

**Efficient YOLO-Based Deep Learning Model for Real-Time Potato
Disease Detection**

BY

**MD JUBAYER ISLAM TASIN
ID: 221-15-5885**

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Md. Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Md. Assaduzzaman
Lecturer
Department of Computer Science and Engineering
Daffodil International University



DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

December 2024

APPROVAL

This Project titled **Efficient YOLO-Based Deep Learning Model for Real-Time Potato Disease Detection** submitted by **Md. Jubayer Islam Tasin**, ID No: **221-15-5885** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **13 January, 2025**.

BOARD OF EXAMINERS



Dr. Md. Taimur Ahad
Associate Professor & Associate Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Chairman



Mr. Saiful Islam
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



Mr. Amir Sohel
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner



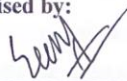
Nazibur Rahman
Technical Lead - Database Administrator
Telenor - Grameen Phone Account

External Examiner

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md. Sazzadur Ahamed, Assistant Professor, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

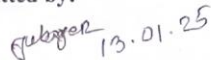


Md. Sazzadur Ahamed
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Assaduzzaman
Lecturer
Department of CSE
Daffodil International University

Submitted by:



Md. Jubayer Islam Tasin
ID: 221-15-5885
Department of CSE
Daffodil International University

DECLARATION

We hereby declare that, this project has been done by us under the supervision of **Md. Sazzadur Ahamed, Assistant Professor, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

Md. Sazzadur Ahamed
Assistant Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Assaduzzaman
Lecturer
Department of CSE
Daffodil International University

Submitted by:

Md. Jubayer Islam Tasin
ID: 221-15-5885
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty God for His divine blessing making us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Md. Sazzadur Ahamed, Assistant Professor**, Department of CSE, Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of “*Deep Learning*” to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stage have made it possible to complete this project.

We would like to express our heartiest gratitude to **Professor Dr. Touhid Bhuiyan, Professor & Head**, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of CSE Department of Daffodil International University.

We would like to thank our entire course mates in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Crop diseases pose a significant threat to global food security, requiring efficient and accurate detection methods to support sustainable agriculture. This study focuses on evaluating YOLO-based deep learning models for detecting potato leaf diseases, leveraging the Potato Leaves Disease Detection Dataset from publicly available repository. The dataset contains 1,398 images categorized into Fungus, Pest, and Healthy classes. Four YOLO model configurations YOLOv8 (large and medium) and YOLOv9 (medium and small) were experimented onto the Potato leaf disease dataset. The YOLOv8 (large) model outperformed all other configurations, achieving the highest mAP50 (97.2%) and mAP50-95 (92.4%), along with the lowest training and validation losses, demonstrating robust generalization and precise object detection capabilities. YOLOv9 models, particularly YOLOv9 (small), exhibited lower and more variable performance, highlighting challenges in distinguishing complex or overlapping disease features despite their compact design. This study concludes that YOLOv8 (large) is the most reliable model for applications requiring high precision and robust performance in real-time agricultural disease detection. The findings underscore the potential of YOLO-based models to transform crop disease management by improving early detection and intervention, contributing to enhanced productivity and sustainable farming practices.

TABLE OF CONTENTS

CONTENTS	PAGE
Board of Examiners	i
Declaration	ii
Acknowledgements	iii
Abstract	iv
CHAPTER	
CHAPTER 1: INTRODUCTION	1-6
1.1 Introduction	1
1.2 Motivation	2
1.3 Rationale of the Study	3
1.4 Expected Output	4
1.5 Report Layout	5
CHAPTER 2: BACKGROUND STUDY	7-16
2.1 Terminologies	7
2.2 Related Works	8
2.3 Comparative Analysis and Summary	11
2.4 Scope of the Problem	14
2.5 Challenges	15
CHAPTER 3: RESEARCH METHODOLOGY	17-33
3.1 Introduction	17
3.2 Data Collection Procedure	17
3.3 Dataset Cleaning	18

3.4 Dataset Preprocessing	19
3.5 Proposed Methodology	21
3.6 Model Training	28
3.7 Implementation Requirements	31
CHAPTER 4: RESULT ANALYSIS AND DISCUSSION	34-48
4.1 Introduction	34
4.2 Experiment Results and Analysis	34
4.3 Generating Confusion Matrix	36
4.4 Generating Classification Report	39
4.5 Training and Validation Accuracy and Loss Curve	40
4.6 Discussion	47
CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY	49-52
5.1 Impact on Society	49
5.2 Impact on Environment	49
5.3 Ethical Aspects	50
5.4 Sustainability Plan	51
CHAPTER 6: OVERVIEW OF THE STUDY, CONCLUSION AND FUTURE WORK	53-56

6.1 Overview of the Study	53
6.2 Conclusion	53
6.3 Limitations	54
6.4 Future Work	55
REFERENCES	57-59
PLAGIARISM REPORT	60

LIST OF FIGURES

Figures	Page no
Figure 3.2: Sample Image of Each Class.	18
Figure 3.3: Train-Validation Test Splitting	18
Figure 3.5: The Methodological Flowchart Diagram.	22
Figure 3.5.2: The Architecture of YOLOv8 Model	26
Figure 4.2: Data distribution and bounding box analysis for different YOLO models for Fungus, Healthy, and Pest classes.	35
Figure 4.3: Confusion matrix for the different YOLO models.	38
Figure 4.5: Training and Validation Loss and Metrics Curves for Box, Classification, and DFL Loss with Precision, Recall, and mAP Metrics for different YOLO models.	42
Figure 4.5.1: Precision curve for different YOLO models.	43
Figure 4.5.2: Recall curve for different YOLO models.	44
Figure 4.5.3: F1 curve for different YOLO models.	45
Figure 4.5.4: Precision-Recall curve for different YOLO models.	46

LIST OF TABLES

Tables	Page no
Table 2.3: Research Matrix for YOLO-Based Agricultural Disease Detection Models.	12
Table 3.2: No. of Images in Each Class	18
Table 3.4 Dataset Specifications.	19
Table 3.5: Proposed Comparison Table.	28
Table 3.6. Common parameter table for all experimented models.	29
Table 3.7: Common data split for all experimented models.	30
Table 4.4: Classification report for different YOLO models.	39
Table 4.6. Performance Comparison of YOLO Models on mAP, GFLOPs, and Inference Speed.	47

CHAPTER 1

Introduction

1.1 Introduction

Agricultural productivity and crop health are central to global food security, but crops face increasing threats from diseases that can severely impact both yield and quality. Traditionally, disease diagnosis in agriculture has relied on manual inspection methods, which are time-consuming, labour-intensive, and prone to error, particularly as farm sizes expand and labour shortages grow [1]. The advent of artificial intelligence (AI) and computer vision has introduced promising new methods for disease identification, enabling more efficient and accurate diagnosis. Among AI-based approaches, YOLO (You Only Look Once) models have become prominent in agricultural disease detection due to their capacity for rapid object detection and real-time application potential, making them ideal for field deployment [2-3].

Initially developed for applications outside of agriculture, YOLO models have undergone considerable adaptations to meet the unique demands of crop disease detection. In their standard form, YOLO models detect objects quickly, but crop disease detection presents added challenges, such as dealing with overlapping foliage, diverse disease symptoms, and variable lighting conditions. Recent research has shown that customising YOLO's architecture can improve its effectiveness in these complex settings. For example, Xu and Wang [4] developed a modified YOLO model specifically for apple leaf disease detection, incorporating lightweight network structures to enhance detection speed while preserving high accuracy. Similarly, Dai [5] introduced an improved YOLOv5 model that achieved 90.14% accuracy in potato grading tasks by refining feature extraction layers, demonstrating the flexibility of YOLO-based approaches for different crop types.

Studies have also extended YOLO models to mobile and resource-limited settings, facilitating disease detection in remote agricultural areas. Wang [6] proposed a mobile-friendly model, MGA-YOLO, which uses Ghost modules to maintain accuracy while reducing computational load, making it feasible for real-time applications on mobile

devices. This adaptation is crucial for in-field diagnostics, where fast and reliable disease identification can aid in early intervention and control measures.

Despite these advancements, challenges remain. Effective disease detection models must balance accuracy with processing efficiency, especially in dense crop settings where diseases can present as small or obscured symptoms. Zhao [7] tackled this challenge by integrating attention mechanisms into a YOLOv4-based model for weed detection in potato fields, achieving 98.52% mAP and showcasing the potential of attention mechanisms for fine-grained detection in complex environments. Moreover, in rice crop detection, the issue of diverse and overlapping symptoms has led to models such as YOLOv5-EFFICIENT, which uses an improved attention mechanism and optimised anchor box generation to boost accuracy while retaining fast processing speeds [8].

The present state of YOLO-based agricultural disease detection thus reflects significant innovation. Models are increasingly adapted to specific crop types, with architectural modifications tailored to enhance speed, accuracy, and resource efficiency. However, the scalability of these models remains an open question, as many studies focus on single crops with limited generalizability across diverse agricultural settings. Future work will likely address these limitations, exploring multi-crop adaptability and further optimization for real-world conditions, as agricultural applications continue to drive advances in AI-powered disease detection.

1.2 Motivation

Crop diseases pose a substantial threat to global food security, impacting yields and reducing the quality of agricultural products. Traditional manual detection methods are often slow, costly, and require expert knowledge, making them impractical for large-scale or remote farming operations [1]. While deep learning and computer vision offer promising alternatives, existing models must be adapted to the distinct challenges of agricultural environments, including varying lighting conditions, complex plant structures, and rapid, real-time detection to enable timely interventions [9].

YOLO-based models have shown considerable potential for disease detection due to their speed and efficiency, but limitations persist in adapting these models to the unique demands of agriculture. For instance, achieving high accuracy in the presence of small or overlapping disease symptoms remains a challenge, as many YOLO adaptations are prone to false negatives or misclassifications in dense crop settings [7]. Additionally, many existing models are crop-specific, which restricts their scalability across diverse agricultural applications and hinders the development of a unified, cross-crop detection framework [6], [9].

Resource efficiency is another critical issue. While lightweight models, such as MGA-YOLO, demonstrate that disease detection can be executed on mobile devices, there is often a trade-off between model compactness and detection accuracy. In many cases, mobile-friendly adaptations sacrifice precision, particularly in identifying subtle symptoms of disease, which can lead to delayed or missed diagnoses [6]. Similarly, while models such as YOLOv5-EFFICIENT have incorporated attention mechanisms to improve performance in detecting small targets, they still require further refinement to be reliably applied across various crop types without significant computational overhead [8].

Thus, the problem lies in developing a robust, generalized YOLO-based model that can deliver high accuracy and efficiency across diverse crops and environmental conditions, minimizing resource use while maintaining real-time detection capabilities. Bridging these gaps is essential to advancing agricultural disease detection systems, reducing the reliance on human labor, and improving yield quality and productivity through timely, precise interventions [2], [4]. This motivation drives the pursuit of an AI-driven approach to sustainable and resilient agriculture.

1.3 Rationale of the study

The increasing prevalence of crop diseases poses a significant challenge to global food security, affecting both yield and quality in agriculture. Traditional methods for disease detection, such as manual inspection, are not only time-intensive but also impractical for large-scale farming operations or remote agricultural areas. While advances in deep

learning and computer vision offer promising alternatives, current models face critical limitations in adaptability, scalability, and real-time performance. YOLO-based models have emerged as a leading solution due to their speed and efficiency, yet they require further enhancements to meet the diverse and complex demands of agricultural environments.

This study is driven by the need to bridge these gaps and provide a robust solution for modern agriculture. The rationale lies in addressing key challenges, including the accurate detection of small, overlapping, or complex disease symptoms, adaptability to diverse crop types, and resource efficiency for deployment on devices like mobile phones and UAVs. By leveraging advancements in YOLO architectures and incorporating mechanisms such as attention modules and multi-symptom detection capabilities, the proposed research aims to create a flexible, generalized framework that can operate reliably under varying environmental conditions.

Additionally, the study acknowledges the lack of high-quality annotated datasets for certain crops and the challenges of real-world deployment, such as variability in lighting and weather conditions. These limitations underscore the importance of developing a scalable model that not only advances theoretical understanding but also provides practical value to farmers by enabling precise, real-time disease detection. This rationale aligns with global sustainability goals, supporting environmentally conscious farming practices by reducing excessive pesticide use and enhancing crop management efficiency.

1.4 Expected Output

The expected outcome of this research will be to develop an efficient, generalized YOLO-based deep learning model that enhances real-time agricultural disease detection across diverse crop types and environmental conditions. This expected outcome will be achieved through the following specific goals:

- Enhancing the YOLO architecture to improve detection accuracy and speed, particularly in crops with small, overlapping, or complex disease symptoms. This involves testing and integrating advanced feature extraction layers and attention

mechanisms to address limitations identified in models like YOLOv5 and YOLOv8 [7-8].

- Developing a flexible model that generalizes well across multiple crop types, extending beyond crop-specific implementations found in current literature. This will address scalability issues and provide a unified solution adaptable to various agricultural settings.
- Testing and refining the model's ability to perform accurately under different lighting, weather, and background complexities, ensuring reliable detection across real-world agricultural conditions.
- Enabling the model to detect multiple disease symptoms on a single leaf or plant, enhancing early detection capabilities crucial for timely intervention. This builds upon hybrid models that have successfully tracked various symptoms but remain limited to specific crops or diseases [10-11].

These outcomes collectively aim to address existing limitations and contribute a scalable, efficient, and robust disease detection solution to modern agriculture, thereby supporting sustainable farming practices and improved crop management.

1.5 Report Layout

This thesis is organized into several chapters, each addressing a specific aspect of the research on agricultural disease detection using YOLO-based models.

- **Chapter 1:** Introduction - This chapter introduces the research topic, providing an overview of the study's background, problem statement, objectives, scope, and limitations.
- **Chapter 2:** Literature Review - The literature review examines existing research and related works in agricultural disease detection, focusing specifically on YOLO-based and similar object detection models. It provides a comparison between existing methods, identifies current challenges, and highlights contributions and gaps in the field.

- **Chapter 3: Research Methodology** - This chapter details the research methodology, including the system design and requirements for implementing the proposed YOLO-based model. It discusses the process of dataset selection, preprocessing, model architecture, and optimization strategies.
- **Chapter 4: Result Analysis and Discussion** - This chapter presents and analyzes the experimental results, focusing on model performance metrics such as accuracy, precision, recall, and F1 score. The results are compared with existing methods, and the effectiveness of the YOLO-based approach in agricultural disease detection is critically evaluated. This chapter also presents a detailed description and discussion of the results.
- **Chapter 5: Impact on Society, Environment, and Sustainability** – This chapter explores the broader implications of the research, including its impact on agricultural practices, environmental sustainability, and economic aspects. It addresses the societal benefits and potential challenges associated with deploying AI-based disease detection in agriculture, highlighting considerations for sustainable farming and resource management.
- **Chapter 6: Overview of the Study, Conclusion, and Future Work**- The final chapter summarises the key findings of the research, drawing conclusions based on the study's objectives and results. It provides recommendations for future research, acknowledging limitations and suggesting areas for improvement.

Each chapter is crafted to ensure a logical flow of information, allowing readers to follow the research progression seamlessly from initial concepts to the conclusions and recommendations. This structured approach supports a thorough understanding of the research process and its outcomes in advancing agricultural disease detection technology.

CHAPTER 2

Background Study

2.1 Terminologies

This study employs several technical terms and concepts that are integral to understanding the methodologies and advancements discussed in the research on YOLO-based agricultural disease detection. YOLO (You Only Look Once) is a widely recognized real-time object detection algorithm that identifies objects within images or videos efficiently, making it suitable for real-time agricultural applications. mAP (Mean Average Precision) is a key performance metric used to evaluate the accuracy of object detection models, measuring the alignment between predicted bounding boxes and ground truth data. The term attention mechanisms refers to deep learning components that enable models to focus on the most relevant areas of an image, enhancing detection accuracy, especially for complex or overlapping disease symptoms. Additionally, lightweight models are optimized neural network architectures designed for resource-constrained devices such as mobile phones or UAVs, balancing computational efficiency with performance. To improve generalization, data augmentation techniques are employed, including transformations like rotation, flipping, and cropping, which artificially expand the dataset. Bounding boxes are rectangular regions in images used to mark the location of detected objects, and their quality is often assessed using IoU (Intersection over Union), which measures the overlap between predicted and actual bounding boxes.

Key techniques, such as anchor boxes, predefined shapes and sizes for predicting object locations, and spatial pyramid pooling (SPP), which enhances scale adaptability in object detection, are critical to improving model efficiency. Multi-symptom detection refers to the model's ability to identify multiple disease symptoms on a single plant or leaf, addressing overlapping or concurrent issues. Finally, the concept of real-time detection underscores the ability to analyze images or videos instantaneously, enabling immediate intervention in agricultural practices. These terms collectively underpin the study's technical framework, forming the basis for innovations in precision agriculture, a modern farming approach that leverages advanced technologies for optimal crop and soil management.

2.2 Related Works

The development of real-time and reliable agricultural disease detection methods has become increasingly essential as traditional manual detection systems are often inefficient and yield variable accuracy. Numerous studies have explored artificial intelligence-based solutions, particularly using YOLO (You Only Look Once) models, to address these limitations in different crops.

Soeb [2] provides a comprehensive analysis of YOLOv7 for tea leaf disease detection, leveraging an extensive dataset from Bangladeshi tea gardens. By augmenting data and optimising the YOLOv7 architecture, they achieved detection accuracy metrics of 97.3%, precision of 96.7%, and an F1-score of 0.965, indicating a significant enhancement over traditional methods such as CNN and RetinaNet.

Wang [6] introduced the Mobile Ghost Attention YOLO (MGA-YOLO) model, which specifically addresses apple leaf disease detection on mobile devices. By integrating Ghost modules and Convolutional Block Attention Modules (CBAM), MGA-YOLO reduced the computational cost while retaining high detection accuracy. Their experiments yielded a mean average precision (mAP) of 94.0%, outperforming conventional models, demonstrating this model's practical applicability for in-field mobile diagnostics, which is especially beneficial for rapid response in dispersed apple orchards.

For tomato leaf diseases, Rajamohan and Latha [13] optimised a YOLOv5 model using proprietary and public datasets, achieving a 93% accuracy in real-time leaf classification. This approach, tailored for field datasets from the Indian regions of Kerala and Tamil Nadu, underscores the potential of optimised YOLOv5 configurations to enhance both classification and disease detection efficacy directly within agricultural environments.

Xu and Wang [4] addressed the trade-off between accuracy and model efficiency in detecting apple leaf diseases by introducing the lightweight ALAD-YOLO model. By embedding the Mobilenet-V3 and Ghost modules within YOLOv5s, they achieved a

model with 90.2% accuracy and a reduced computational burden, optimizing it for real-time applications in field conditions.

Similarly, Mathew and Mahesh [14] applied YOLOv5 to detect bacterial spot disease in bell peppers, illustrating its capability to capture minor disease symptoms with high sensitivity and speed. Their model, tested with mobile-acquired images, allows farmers to detect plant diseases early and minimise spread, showcasing YOLOv5's adaptability for various crop diseases and its operational feasibility for on-site diagnostics.

The detection of maize leaf diseases in dense, real-world field conditions was examined by Li [15], who proposed the GhostNet_Triplet_YOLOv8s algorithm. By integrating GhostNet and Triplet Attention with YOLOv8s, their approach reduced model size by 50% and maintained an mAP of 91.4%, thus ensuring high efficiency and accuracy.

In the realm of paddy disease detection, Ganesan and Chinnappan [16] introduced a hybrid model combining ResNet and YOLO, employing Fitness Sorted-Shark Smell Optimization (FS-SSO) for enhanced disease recognition. This hybrid model achieved high classification accuracy by incorporating adaptive K-means clustering, providing a robust solution to the complexities of paddy disease identification and highlighting hybrid models' potential for crop-specific disease detection.

Aziz [17] improved YOLO's functionality for rice leaf disease detection by modifying anchor boxes and network depth, achieving a 94% accuracy rate. Their modified YOLO network facilitated faster disease classification, effectively bridging the gap between high-performance models and in-field application by delivering real-time results crucial for timely disease management.

In response to potato leaf disease detection challenges, Dai [5] developed DA-ActNN-YOLOv5, integrating 13 data augmentation techniques with a compressed ActNN model to improve detection accuracy to 99.8%. This model underscores the necessity of data augmentation and model compression to handle varying environmental factors.

Srivastava [3] applied YOLOv7 for potato disease detection, obtaining a 98.1% accuracy rate. Their study further validates YOLOv7's superior detection capability, offering an effective solution for disease management in precision agriculture through early detection and actionable insights for farmers.

Dai [5] designed an enhanced YOLOv5 model to detect and grade sprouted potatoes, particularly in complex environments, achieving an mAP increase over conventional models by up to 12.4%. By incorporating CrossConv in the C3 module and fast spatial pyramid pooling (SPP), they improved both feature extraction and model efficiency. potato processing.

Emwinghare and Al-Mallahi [18] compared YOLOv4 and Mask R-CNN for segmenting potato tubers. Although Mask R-CNN provided better segmentation accuracy in densely packed images, the study highlighted YOLOv4's limitations in such contexts, emphasizing the need for more sophisticated approaches in segmentation-heavy applications for accurate tuber identification.

For in-field potato disease monitoring, Oishi [1] developed a deep learning-based detection system using portable video cameras, facilitating real-time identification of abnormal potato plants at various growth stages. Their model demonstrated a high classification accuracy of up to 96.7% for diseased leaves, indicating its applicability as a labour-efficient solution for on-site disease management in seed potato production.

Xu and Mao [9] addressed the spacing challenges in potato planting with a modified YOLOv5 model for potato identification and distance detection. Through data augmentation and an optimised architecture, their model achieved an average accuracy mean (mAP) of 99.5%, demonstrating the potential for using YOLO models to optimize planting practices and enhance yield quality in mechanized agriculture.

Weed detection in potato fields was further improved by Zhao [7], who developed the MC-YOLOv4 model, introducing MobileNetV3 and CBAM to the YOLOv4 architecture to reduce computational load while enhancing accuracy. This model achieved a 98.52% mAP, significantly surpassing several YOLO and Faster R-CNN baselines, illustrating the potential of YOLO-based architectures in automating weed control in crop-specific contexts.

In rice disease detection, Kumar [11] introduced a multi-scale YOLOv5 model with DenseNet-201 and a Bidirectional Feature Attention Pyramid Network (Bi-FAPN) to capture disease symptoms at various scales. With accuracy and F1 scores of 94.87% and 92.45%, respectively, their work highlights the advantages of multi-scale feature

attention mechanisms in recognising rice disease symptoms, thereby promoting early-stage disease intervention.

Sangaiah [12] optimised Tiny YOLO (T-YOLO) for UAV-based rice field monitoring, achieving effective disease detection using spatial pyramid pooling, CBAM, and Ghost modules. Their UAV T-YOLO-Rice model demonstrated a high mean average precision (mAP), underscoring the integration of IoT and UAV technologies in scalable and accurate crop health assessment.

Deari and Ulukaya [10] addressed multi-symptom disease detection in rice leaves through a hybrid model combining YOLOv5x with Inception v3, achieving an F1 score of 98.24%. By focusing on individual symptom regions rather than broad classifications, their model improved early-stage detection precision, thus providing actionable insights for managing multiple concurrent disease symptoms.

In tackling the challenge of detecting rice leaf diseases amidst varied lesion sizes, Gao [8] developed YOLO V5-EFFICIENT by enhancing YOLOv5 with improved K-means anchor generation and attention mechanisms, achieving a 9.9% improvement in mAP.

Lastly, Lu [19] introduced YOLOv8_Rice to address issues related to small-scale rice diseases. By incorporating deformable convolutions and bidirectional feature pyramids, their model achieved an 18.7% mAP improvement over YOLOv8n, affirming the feasibility of YOLOv8 adaptations in enhancing small-target detection for real-world agricultural applications.

2.3 Comparative Analysis and Summary

The reviewed studies illustrate advancements in YOLO-based models tailored to various agricultural applications, focusing on disease detection across different crop types. Each model adapts the YOLO framework to overcome specific challenges in detection accuracy, speed, and model size, often incorporating custom dataset augmentation or specialized architectural enhancements.

For instance, Soeb [2] achieved the highest accuracy of 97.3% in tea leaf disease detection using YOLOv7, emphasizing augmentation to address dataset limitations. In

contrast, lightweight models, such as Wang [6] MGA-YOLO, targeted real-time applications with mobile-friendly structures, achieving 94.0% mAP with a modest 10.34 MB model size, suitable for in-field apple leaf disease monitoring. Notably, studies on rice and potato diseases often leveraged augmented YOLOv5 models. Kumar [11] and Lu [19] improved multi-scale detection in rice leaves, reaching accuracies of 94.87% and 89.9%, respectively, by incorporating attention modules and multi-scale feature extraction layers. Potato disease models also included robust detection mechanisms; for example, Dai [5] enhanced YOLOv5 with CrossConv and achieved a 90.14% accuracy in sprouted potato grading. Zhao [7] and Xu and Mao [9] demonstrated the importance of lightweight configurations and spatial accuracy enhancements in weed and plant spacing detection, achieving mAPs over 98% and facilitating precise agricultural management.

Table 2.3: Research Matrix for YOLO-Based Agricultural Disease Detection Models

Author (Year)	Model Used	Accuracy	Dataset Information
Soeb et al. (2023)	YOLOv7	97.3%	Tea leaf disease dataset from Bangladesh, 4000 images
Wang et al. (2022)	MGA-YOLO	94.0%	Apple Leaf Disease Object Detection dataset (ALDOD), 8,838 images
Rajamohanan & Latha (2023)	YOLOv5	93.0%	Field dataset for tomato leaves
Xu & Wang (2023)	ALAD-YOLO	90.2%	Apple leaf disease dataset, 2,748 images
Mathew & Mahesh (2022)	YOLOv5	Not specified	Bell pepper disease dataset from random farm images
Li et al. (2024)	GhostNet_Triplet_YOLOv8s	91.4%	Maize leaf disease dataset with dense field images

Ganesan & Chinnappan (2022)	ResNet + YOLO	High recognition rate	Paddy leaf disease dataset
Aziz et al. (2023)	Enhanced YOLO	94.0%	Public dataset of Indonesian rice leaves
Dai et al. (2022)	DA-ActNN-YOLOv5	99.81%	Potato disease dataset
Srivastava et al. (2024)	YOLOv7	98.1%	Potato leaf disease dataset, variety of conditions
Dai et al. (2022)	Improved YOLOv5	90.14%	Sprouted potato dataset, complex settings
Emwinghare & Al-Mallahi (2023)	YOLOv4, Mask R-CNN	73%-80% IoU	Potato tuber dataset
Oishi et al. (2021)	Deep learning model (custom)	96.7%	Field video images for abnormal potato plants
Xu & Mao (2024)	Improved YOLOv5	99.5%	Potato field spacing dataset
Zhao et al. (2022)	MC-YOLOv4	98.52%	Weed detection in potato fields
Kumar et al. (2023)	Multi-scale YOLOv5 + Bi-FAPN	94.87%	RLD rice leaf disease dataset
Sangaiah et al. (2024)	UAV T-YOLO-Rice	Not specified	Rice leaf disease dataset from UAV
Deari & Ulukaya (2024)	YOLOv5x + Inception v3	96.67%	Public rice leaf disease dataset
Gao et al. (2024)	YOLO V5-EFFICIENT	High accuracy	Dataset with small rice lesions
Haque et al. (2022)	YOLOv5	Various metrics	Augmented rice leaf disease dataset

Zhang et al. (2024)	YOLO-CRD	90.4%	Rice disease dataset, multi-disease detection
Trinh et al. (2024)	Alpha-EIoU-YOLOv8	89.9%	Rice leaf disease dataset, 3175 images
Lu et al. (2024)	YOLOv8_Rice	Improved by 15.8% mAP	Rice disease dataset, specific rice diseases

2.4 Scope of the Problem

While significant advancements have been made in applying YOLO-based models to agricultural disease detection, several notable gaps and problems persist, highlighting areas for further research:

- Many models are tailored to specific crops, such as rice, potato, or tea, but lack generalisation capabilities across multiple crops. This specificity limits scalability, especially for agricultural settings requiring detection across diverse plant species.
- Although some models incorporate attention mechanisms and improved anchor box configurations, challenges remain in reliably detecting diseases in complex and occluded settings, as seen in crops with dense foliage (e.g., potatoes and rice paddies). Models struggle to maintain high accuracy without additional computational costs or significant data preprocessing.
- While lightweight models such as MGA-YOLO (for mobile devices) and T-YOLO (for UAVs) cater to resource-limited applications, there is still a gap in balancing accuracy and speed for highly compact models. Many resource-efficient adaptations experience trade-offs in detection accuracy, limiting their effectiveness in real-time, precision agriculture.
- Dataset diversity and augmentation techniques vary widely, impacting model generalization. The limited use of standard agricultural disease datasets restricts cross-study comparison, and the absence of extensive datasets hinders model robustness against varying environmental factors.
- While hybrid models have improved multi-disease detection capabilities, models are generally restricted to single-disease detection or provide limited multi-symptom tracking. Addressing simultaneous disease and symptom detection within

a single framework remains a significant research gap, especially for diseases with overlapping visual symptoms.

- Although several models achieve high accuracy, field trials and adaptability assessments are often limited. Many models demonstrate promising results in controlled environments but require validation in real-world agricultural settings to verify consistency under varying light, weather, and crop growth conditions.

2.5 Challenges

The challenges highlight the complexities involved in designing and implementing YOLO-based models for agricultural disease detection, underscoring the need for continuous refinement and innovation are following:

- The lack of high-quality, annotated datasets for specific crops or diseases can limit model training and generalization across diverse agricultural scenarios.
- Designing lightweight models for resource-constrained devices (e.g., mobile phones or UAVs) often requires trade-offs in accuracy, particularly for detecting small or overlapping disease symptoms.
- Variations in lighting, weather, and background conditions in real-world agricultural settings pose significant challenges to model robustness and reliability.
- Ensuring a balanced distribution of data across all classes (e.g., Fungus, Healthy, and Pest) is critical to prevent bias and support unbiased learning.
- Developing a model that can generalize effectively across multiple crops and diseases requires extensive, diverse training data, which is often unavailable or difficult to obtain.
- Accurately annotating images with bounding boxes is labor-intensive and prone to human error, which can negatively impact the training process.
- Achieving high accuracy on training data without overfitting is challenging, particularly for models trained on datasets with limited diversity.
- Training complex YOLO models requires significant computational power, which may not always be available, especially in resource-constrained environments.
- Visually similar classes, such as Healthy and Pest, can lead to higher misclassification rates, requiring more refined feature extraction methods.

- Transitioning from controlled experiments to practical field applications can be difficult, as real-world conditions often differ significantly from the training environment.

CHAPTER 3

Research Methodology

3.1 Introduction

This study aims to create an efficient and accessible system for the real-time detection of diseases on potato leaves using YOLO-based deep learning models. The methodology begins with the collection of a comprehensive dataset specifically focused on various potato leaf diseases. Each image in the dataset undergoes a thorough annotation process, where bounding boxes are drawn around affected areas, and the

specific disease type is labelled accurately. Following data preparation, we explored different versions of the YOLO (You Only Look Once) models. In this study, we experimented with YOLOv8 and YOLOv9 models in several configurations, including medium and large variants. After training, we evaluated each model's performance based on metrics such as precision, recall, and processing time. The model that performed best across these metrics was selected as the final model for deployment. To make the model compatible with mobile devices, it was then converted into a TensorFlow Lite format (float32.tflite). This conversion process optimises the model for mobile use, ensuring it is lightweight enough to run smoothly on smartphones without compromising accuracy.

3.2 Data Collection Procedure

The dataset used in this study focuses on detecting diseases in potato leaves and was obtained from the Roboflow platform (Lauren Draetta, Potato Leaves Disease Detection Dataset, Roboflow Universe). It consists of a total of 1,398 images divided into three categories: Fungus, Pest, and Healthy, which represent the types of diseases or conditions affecting the leaves, as well as healthy samples. To ensure effective training, the dataset is split into three subsets: a training set with 1,224 images, a validation set with 117 images, and a test set with 57 images. This distribution allows for a balanced approach to model training, validation, and evaluation.

Table 3.2: No. of Images in Each Class

Class Name	No. of Images
Fungus	432
Pest	499
Healthy	467
Total	1398



Figure 3.2: Sample Image of Each Class.

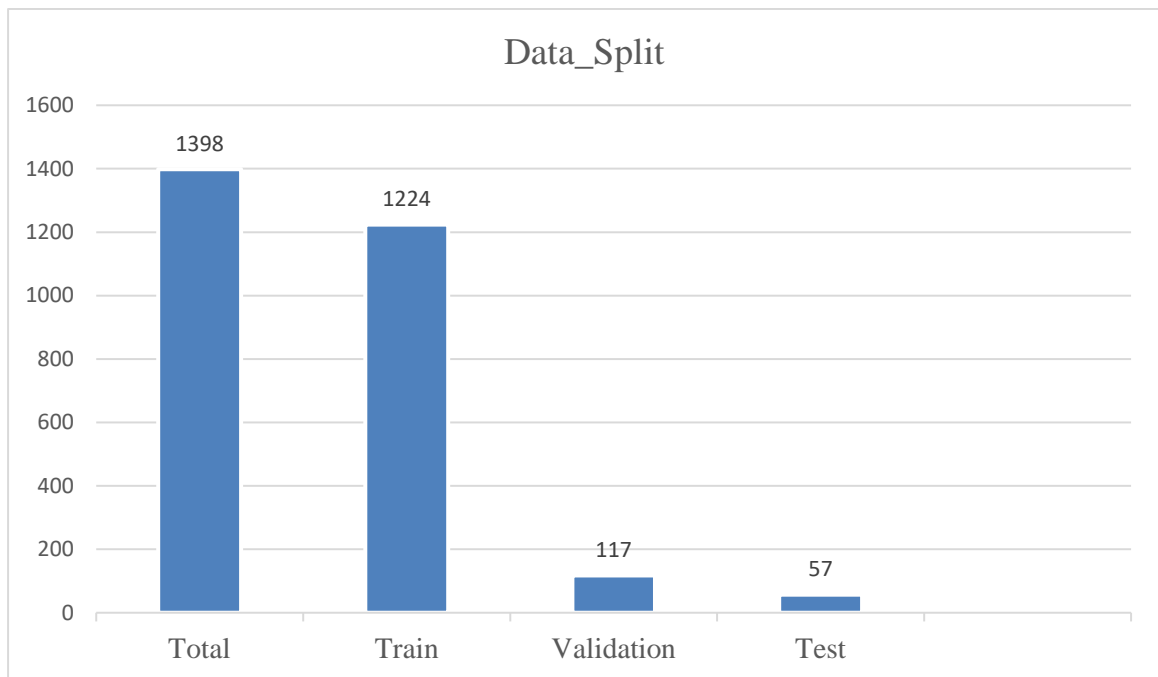


Figure 3.3: Train-Validation-Test Splitting.

3.3 Dataset Cleaning

Dataset cleaning is a crucial step to ensure the optimal performance of YOLO-based models. This process involves reviewing and refining the dataset to remove irrelevant, duplicate, or corrupted images that could introduce noise into the model's training. Proper labeling is essential, as YOLO requires precise annotations in the form of bounding boxes that accurately define the location of objects. Tools like Roboflow is commonly used to create or validate these annotations. Additionally, images should be resized or normalized to match the input dimensions required by the YOLO model, ensuring consistency during training. By addressing issues like class imbalance,

misannotations, and image inconsistencies, dataset cleaning lays the foundation for accurate and efficient disease detection.

Since this study used a publicly available dataset, hence, any mandatory data cleaning procedure wasn't required.

3.4 Dataset Preprocessing

No preprocessing techniques were applied to the images, keeping them in their original form to retain all visual details essential for accurate detection. However, data augmentation was implemented to enhance the model's robustness. Specifically, a zoom augmentation was applied, with a 0% minimum zoom and a 20% maximum zoom. This augmentation strategy enables the model to generalize better by exposing it to varied perspectives and scales of the leaf diseases, which improves detection accuracy in real-world scenarios.

Table 3.4: Dataset Specifications.

Properties	Values
Image Resolution	240 × 240 pixels
Format	.jpg
Total Images	1398
Data Augmentation	0% minimum and 20% maximum zoom
Classes	3

3.4.2 Annotation Process

Annotation is the process of labelling images or data with specific information that guides a machine-learning model during training. In this study, annotation involves marking specific areas on potato leaf images—such as diseased or healthy regions—to help the model learn to detect and classify these areas accurately. Annotation is a crucial

step in object detection tasks, especially for models like YOLO, which rely on detailed and accurate labelling to distinguish between various objects in an image.

The importance of precise annotation becomes even more significant due to the YOLO model's use of self-attention mechanisms. Self-attention allows the model to focus on relevant regions within an image, enhancing its ability to recognize and classify objects by emphasizing the most critical parts of the image. By providing well-defined bounding boxes and class labels, the annotated dataset helps guide the self-attention mechanism to focus on the specific features associated with different disease types and healthy leaves. This improves the model's ability to make accurate detections, as it learns to concentrate on disease characteristics such as texture changes, colour differences, and leaf damage patterns based on the annotations.

To achieve high-quality annotations, we used the MakeSense tool, a web-based annotation platform designed for image labelling tasks.

The following steps outline the comprehensive annotation process:

- Firstly, the annotation process began by defining the classes within MakeSense, specifically for the three categories of interest: *Fungus*, *Pest*, and *Healthy*. Setting up these classes allowed for quick and consistent labeling throughout the process, ensuring that each bounding box could be assigned to a specific disease or healthy category.
- Then, each image was carefully inspected to identify areas displaying signs of disease or healthy portions of the potato leaves. Using the MakeSense tool's bounding box feature, rectangular boxes were drawn around the areas of interest. For diseased leaves, this meant selecting areas that displayed visible symptoms, such as spots, discoloration, or texture changes associated with fungal infections or pest damage. For healthy samples, representative leaf areas were marked. The bounding box adjustments provided by MakeSense allowed for accurate coverage of the specific areas the YOLO model would learn to focus on.
- Thirdly, drawing a bounding box, it was assigned the appropriate label: *Fungus*, *Pest*, or *Healthy*. This labeling process is critical for training, as it enables the YOLO model to learn distinct patterns for each class. The accurate labeling of diseased and healthy areas allows the self-attention mechanism within YOLO to

recognize and concentrate on the distinguishing features of each class, improving the model's classification accuracy.

- Then, once the annotation was completed for all images, a quality check was conducted to verify consistency and accuracy. This step involved ensuring that all diseased areas were properly annotated and that each bounding box was labeled correctly. A high-quality annotation process ensures that the self-attention mechanism in the YOLO model will focus on the correct regions during training, leading to improved detection performance.
- After that, completing the annotation, the files were exported in YOLO format, generating a .txt file for each image. Each text file includes the class label and bounding box coordinates in the format <class> <x_center> <y_center> <width> <height>. This format is optimized for YOLO's architecture, allowing the model to interpret the annotations and guide the self-attention mechanism effectively.
- Finally, the .txt files, a data.yaml file was created to structure the dataset. This YAML file includes information about the dataset's directory paths, the number of classes, and the class names. This configuration file is essential for guiding the YOLO model during training and evaluation, enabling it to access the images and annotations efficiently.

By following this annotation and exporting process, we prepared a high-quality, labeled dataset tailored for the YOLO model's unique requirements. Accurate annotation is essential for the YOLO model's self-attention mechanism, as it directs the model's focus towards relevant regions and disease characteristics, ultimately enhancing its ability to detect and classify potato leaf diseases in real-time agricultural applications.

3.5 Proposed Methodology

The image illustrates the workflow for building and evaluating YOLO-based deep learning models for detecting potato-related features. The process begins with a Potato Dataset, which undergoes an Annotation step to label the data with bounding boxes or classifications. This annotated dataset is then used to train Deep Learning Models, focusing on various YOLO model variations, including YOLOv8m, YOLOv8l, YOLOv9s, and YOLOv9m.

These models are compared based on performance metrics (e.g., accuracy, speed, resource efficiency), and the comparisons feed into a decision-making process to identify the YOLO Best Model for the specific task. This iterative comparison ensures the selection of the most suitable YOLO version for precise and efficient potato disease or feature detection.

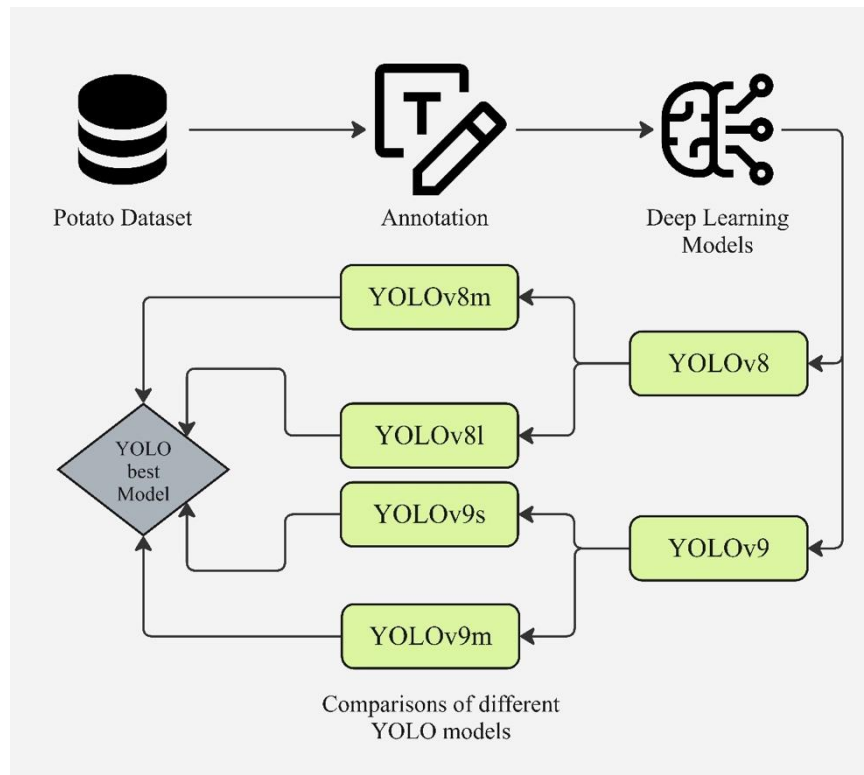


Figure 3.5: The Methodological Flowchart Diagram.

3.5.1 You Only Look Once (YOLO)

YOLO (You Only Look Once) is a real-time object detection algorithm that frames the object detection task as a single regression problem, directly predicting bounding boxes and class probabilities from the entire image in one evaluation. Unlike traditional object detection methods that require region proposal networks or sliding windows, YOLO is designed for speed and efficiency, making it highly suitable for real-time applications.

3.5.2 Algorithm of YOLO Model

Step 1: Input Image and Preprocessing

The process begins by feeding an input image to the YOLO model. Before processing, the image is resized to a fixed dimension (typically 416x416 or 640x640), depending on the YOLO version. Resizing helps to standardize the model input and optimize computation.

1. **Image Resizing:** The input image is resized to the specified dimensions. Resizing allows YOLO to process images at a consistent scale while maintaining aspect ratio for predictions.
2. **Normalization:** The pixel values of the image are normalized to a range between 0 and 1. This normalization makes the training process more stable.

Step 2: Image Grid Division

YOLO divides the resized input image into an $S \times S$ grid. Each grid cell is responsible for detecting objects whose center falls within that cell. This grid structure is fundamental to YOLO's architecture.

1. **Grid Formation:** The entire image is split into small square cells, with each cell tasked to predict bounding boxes if an object's center lies within it.
2. **Bounding Box Assignment:** For each cell, the model will predict a fixed number of bounding boxes with associated confidence scores and class probabilities.

Step -3: Bounding Box Prediction

1. **Bounding Box Coordinates:** Each bounding box is represented by the coordinates (x, y, w, h) , where x and y denote the center of the box relative to the cell, and w and h are the width and height of the box relative to the image dimensions.
2. **Confidence Score (C):** This score represents the confidence that a bounding box contains an object and is defined as:

$$C = P(\text{object}) \times IoU_{pred, truth}$$

3. where $P(\text{object})$ is the probability of an object existing in the cell, and $\text{IoU}_{\text{pred, truth}}$ is the Intersection over Union (IoU) between the predicted and ground truth bounding boxes.
4. Class Probabilities $P(c_i | \text{object})$: The class probabilities conditioned on the presence of an object are also predicted, where each cell has a vector of class probabilities.

$$(S \times S) \times (B \times 5 + C)$$

Thus, the total output tensor has the shape $S \times S \times (B \times 5 + C)$, where B is the number of bounding boxes per cell and C is the number of classes.

Step-4: Loss Function

YOLO uses a custom loss function to train the network, which comprises three main components:

1. Localization Loss: This term penalizes deviations between the predicted and actual bounding box coordinates. YOLO applies a squared error loss to the coordinates (x , y , w , h) only when an object is present in the cell:

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} \left((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (w_i - \sqrt{\hat{w}_i})^2 + (h_i - \sqrt{\hat{h}_i})^2 \right)$$

2. Confidence Loss: This term penalizes errors in the confidence score for both object and non-object cells. It is defined as:

$$L_{\text{conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{obj}} (C_i - \hat{C}_j)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{\text{noobj}} (C_i - \hat{C}_j)^2$$

where λ_{noobj} is a constant that reduces the penalty for confidence scores in cells that do not contain objects, and 1_{ij}^{noobj} is an indicator variable that equals 1 if there is no object in cell i for bounding box j .

Step-5: Applying Self-Attention

In later YOLO versions (e.g., YOLOv8, YOLOv9), self-attention mechanisms such as Spatial Attention Modules (SAM) or Convolutional Block Attention Modules (CBAM)

were introduced. These modules improve the model's accuracy by focusing on the most relevant parts of the image.

1. **Spatial Attention:** This mechanism generates spatial attention maps that emphasize important regions of the image. It helps YOLO focus on object-relevant areas, increasing the precision of bounding box predictions.
2. **Channel Attention:** The model also uses channel attention to assign importance to feature channels, highlighting features most relevant to object detection.

Step 6: Non-Maximum Suppression (NMS)

YOLO generates multiple bounding boxes for each object, leading to potential overlaps. To reduce redundancy and improve clarity, YOLO applies Non-Maximum Suppression (NMS), which removes overlapping boxes based on confidence scores.

1. **Sorting by Confidence:** All bounding boxes are sorted by confidence scores in descending order.
2. **IoU Thresholding:** For each box, NMS keeps the box with the highest confidence score and removes other boxes that have an IoU (Intersection over Union) above a specific threshold (commonly set at 0.5) with the selected box.
3. **Final Selection:** This process repeats for each class, leaving only the most confident, non-overlapping bounding boxes.

Step 7: Final Detection Output

After applying NMS, YOLO produces the final output, which consists of:

1. **Bounding Boxes:** Each bounding box is represented by the coordinates (x, y, w, h).
2. **Class Label and Probability:** For each box, YOLO outputs the most probable class label along with its probability.
3. **Confidence Score:** The final confidence score is presented alongside each box, indicating the likelihood that the box accurately represents an object.

Step 8: Post-Processing and Visualization

The bounding boxes, class labels, and confidence scores are mapped back onto the original image dimensions. This step involves scaling the predicted coordinates to match the actual image size.

1. Scaling: YOLO rescales the bounding box coordinates from the resized input image to the dimensions of the original image.
2. Visualization: The bounding boxes, labels, and confidence scores are displayed on the original image, providing a visual output that highlights the detected objects.

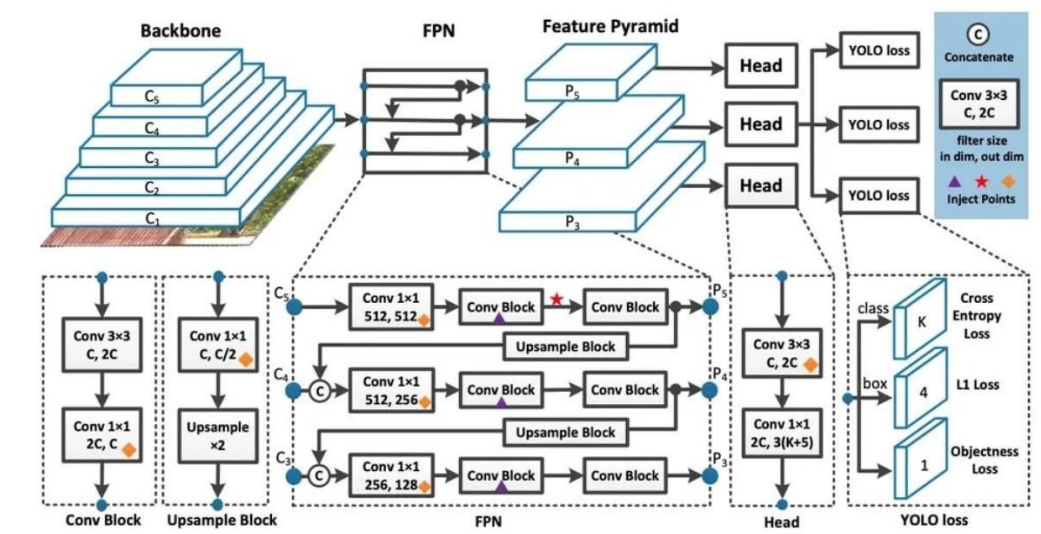


Figure 3.5.2: The Architecture of YOLOv8 Model [Terven et al., 2023]

3.5.3 Comparison Between YOLOv8m, YOLOv8l, YOLOv9s, YOLOv8m

The YOLO (You Only Look Once) family has evolved over time, with each version and variation designed to address specific needs in object detection, balancing speed and accuracy. The suffixes "m", "s" and "l" indicate different model sizes (medium, small and large), with "v8" and "v9" representing the version numbers. Here's a detailed comparison of these four models:

YOLOv8m and YOLOv8l (Medium and Large Models of YOLOv8)

YOLOv8 introduced significant architectural improvements to increase detection accuracy and efficiency. The "m" (medium) and "l" (large) variations provide options for different levels of computational resources and accuracy requirements.

1. YOLOv8m: As a medium-sized model, YOLOv8m is designed to strike a balance between speed and accuracy. It uses fewer layers and parameters than the large model, making it suitable for real-time applications on mid-level hardware while maintaining reasonable accuracy. YOLOv8m uses attention mechanisms, such as Spatial and Channel Attention Modules, to improve focus on relevant regions in the image.
2. YOLOv8l: YOLOv8l, the large variant, includes more layers and parameters compared to YOLOv8m, enhancing its ability to capture fine-grained features and detect small or overlapping objects. YOLOv8l provides higher accuracy but requires more computational resources, making it suitable for applications where accuracy is prioritized over speed.

YOLOv9m and YOLOv9s (Medium and Small Models of YOLOv9)

YOLOv9 builds on YOLOv8 with further optimizations and advancements in self-attention mechanisms and feature extraction. The "m" and "s" models in YOLOv9 are designed to improve both accuracy and computational efficiency.

1. YOLOv9m: YOLOv9m, as a medium model in the latest YOLOv9 version, introduces additional attention layers and feature extraction techniques to improve the model's focus and detection ability. YOLOv9m is optimized for speed while maintaining the accuracy improvements introduced in YOLOv9, making it suitable for real-time detection on resource-limited devices.
2. YOLOv9s: YOLOv9s, the small variant of YOLOv9, incorporates an even less sophisticated feature pyramid network, improved self-attention modules, and smaller depth and width in the network architecture. This model provides the highest accuracy among these four but also demands the most computational resources. YOLOv9s is ideal for moderate-stakes applications where the highest detection accuracy is essential, and computational resources are available.

Key Architectural Differences

1. Attention Mechanisms: YOLOv9 models have enhanced self-attention mechanisms compared to YOLOv8, allowing them to focus better on relevant image regions. YOLOv9 models incorporate improved spatial and channel attention mechanisms, making them more accurate in complex scenarios.

2. Feature Extraction: YOLOv9 utilizes an enhanced Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) for improved feature extraction across different scales, providing better detection for small objects.
3. Depth and Width: Unlike the “s” model, the “l” model (YOLOv8l) has more layers and a greater width than the "m" models (YOLOv8m and YOLOv9m), allowing it to capture finer details and more complex patterns.

Table 3.5: Proposed Comparison Table

Property	YOLOv8m	YOLOv8l	YOLOv9m	YOLOv9s
Version	YOLOv8 (Medium)	YOLOv8 (Large)	YOLOv9 (Medium)	YOLOv9 (Small)
Parameter Count	Moderate	High	Moderate	Moderate
Model Size	Medium	Large	Medium	Small
Self-Attention Mechanisms	Basic Attention	Basic Attention	Enhanced Attention	Enhanced Attention
Processing Speed	Faster	Slower	Faster	Fastest

3.6 Model Training

Table 3.6 outlines the core training parameters applied across all experimented models, aiming to establish a consistent foundation for model evaluation. The image size is set to 640×640 , indicating that each input image is resized to this resolution before processing, which allows the model to capture finer details in the images and potentially enhance feature detection. A batch size of 16 is used, meaning that 16 images are processed together in each iteration before updating the model's weights. This batch size strikes a balance between memory efficiency and stable gradient estimation, supporting effective learning.

The model is trained over 80 epochs, which allows ample opportunity for the model to learn complex patterns within the dataset. Training for this number of epochs is intended to help the model converge to a lower error rate while minimizing overfitting. The AdamW optimizer is used, a variant of the popular Adam optimizer that includes weight decay to help reduce overfitting by penalizing large weights, making it well-suited for handling sparse and noisy gradients in deep learning models. The learning rate is set to 0.001429, a moderate rate that facilitates gradual convergence, allowing the model to move towards the optimal solution without skipping over it or converging too slowly. Together, these parameters are designed to ensure robust and effective training, enabling the model to perform well in both accuracy and generalization on unseen data.

Table 3.6: Common parameter table for all experimented models.

Parameter Name	Parameter Value
Image Size	640 × 640
Batch Size	16
Epoch	80
Optimizer	AdamW
Learning Rate	0.001429

Table 3.7 presents the data split strategy used across all model experiments, ensuring consistent and balanced evaluation. The train set comprises 80% of the dataset, amounting to 1224 images, which provides the model with a large sample of data to learn from and capture diverse patterns within the dataset. This substantial training portion supports effective model learning and helps improve its ability to generalize new data. The validation set consists of 12% of the data, or 117 images, and serves to monitor the model's performance during training. By using this validation data, the model's generalization ability can be assessed, and adjustments to hyperparameters can be made to prevent overfitting. Finally, the test set represents 8% of the dataset, with 57 images reserved exclusively for final evaluation. This test set is only used once training is complete, providing an unbiased measure of the model's real-world

performance on unseen data. This structured data split, with 80% for training, 12% for validation, and 8% for testing, ensures a balanced and effective approach to model training, tuning, and final evaluation.

Table 3.7: Common data split for all experimented models.

Dataset	In Percentage	Number of Images
Train set	80%	1224
Validation Set	15%	117
Test Set	5%	57

3.6.1 Model Evaluation

This study studied precision, recall, mean average precision (mAP), and F1 score to evaluate the performance of YOLO models after training. following formulas have been used to select the highest-performer model for Potato leaf disease detection.

Average Precision (AP): AP is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP), averaged over the number of images or instances (N). It represents the model's precision across various thresholds, showing how accurately the model identifies positive instances relative to false detections.

$$AP = \frac{TP}{TP + FP}$$

Mean Average Precision (mAP): mAP is the mean of AP values across multiple classes or queries (Q), providing an overall measure of model performance across all classes. It is often expressed as a percentage and is a common metric for comparing object detection models.

$$mAP = \frac{\sum_{i=1}^Q AP}{Q} \times 100\%$$

Precision: Precision is the ratio of true positives (TP) to the sum of true positives and false positives (FP). It indicates how many of the predicted positive instances are correct, which is particularly important in applications where false positives carry a high cost.

$$Precision = \frac{TP}{TP+FP}$$

Recall: Recall is the ratio of true positives (TP) to the sum of true positives and false negatives (FN). It measures the model's ability to identify all relevant instances, showing how many actual positives the model successfully detects.

$$Recall = \frac{TP}{TP+FN}$$

F1-score: The F1-score is the harmonic mean of precision and recall, balancing both metrics. It is useful when dealing with imbalanced datasets, as it considers both false positives and false negatives, providing a more comprehensive view of model performance.

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

3.7 Implementation Requirements

Since training a deep learning model like YOLO requires significant computational power, the training process was conducted on Google Colab, which provides access to GPUs for accelerated processing.

- Processor: Google Colab typically provides CPUs with up to 2 virtual cores in the free tier and higher capacities in Colab Pro.
- RAM: Free-tier Colab typically offers 12 GB of RAM, while Colab Pro users can access up to 25 GB.
- Disk Storage: Up to 15 GB of temporary disk space in Colab's runtime environment.
- Google Colab: Google Colab provides cloud-based notebooks with access to high-performance GPUs like the NVIDIA Tesla K80, T4, P100, or V100, depending on availability. This setup allows for efficient model training without needing dedicated on-premises hardware.

3.6.1 Hardware Requirements

For application development and model testing, a computer with the following specifications is recommended:

- **Processor:** Dual-core or quad-core processor (Intel i5 or equivalent AMD processor at a minimum)
- **RAM:** 8GB minimum (16GB recommended for smoother multitasking, especially when running multiple development tools)
- **Storage:** 256GB SSD (Solid State Drive) or higher, as SSDs improve load times and overall system performance
- **Operating System:** Windows 10 or 11, macOS 10.15 (Catalina) or later, or a Linux distribution with development support

These specifications are sufficient for running Android Studio, Visual Studio Code, and Flutter, which are essential for application development. A higher RAM and SSD storage will enhance performance and make the development process smoother, especially when dealing with image data and multiple testing tools.

To run the application smoothly on a mobile device, especially when using on-device model inference, the following specifications are recommended:

- **Operating System:** Android 8.0 (Oreo) or later
- **Processor:** Mid-range processor or higher (e.g., Qualcomm Snapdragon 600 series or higher for Android)
- **RAM:** 3GB minimum (4GB or higher recommended for smoother performance, especially if the model runs directly on the device)
- **Storage:** 32GB internal storage or higher, to accommodate the application and any additional resources or offline data storage needs
- **Camera:** 8MP or higher resolution, as a higher-quality camera will capture clearer images for disease detection.

3.6.2 Software Requirements

Application Development

- **Flutter SDK:** Flutter is a cross-platform development framework used to build the mobile application interface. Flutter allows the application to be deployed on Android devices from a single codebase.
- **Dart:** Dart is the programming language used in Flutter for developing the app's front end and managing interactions between the user and the YOLO model.
- **Android Studio / Visual Studio Code:** These IDEs (Integrated Development Environments) are recommended for coding and debugging the Flutter application. Android Studio is particularly useful for Android-specific development, while Visual Studio Code offers lightweight support for Flutter projects.

Testing and Deployment

- **Android Emulator:** For testing the mobile application on Android devices, the Android Emulator available in Android Studio can be used.

CHAPTER 4

Result Analysis and Discussion

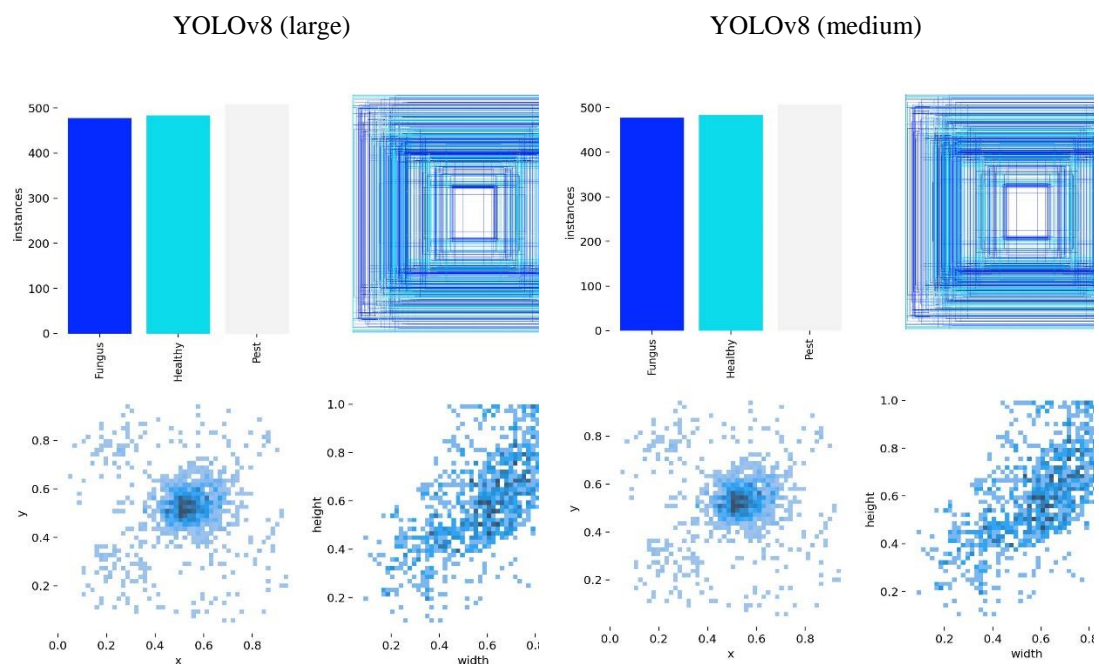
4.1 Introduction

This chapter presents a comprehensive evaluation of various YOLO models, specifically YOLOv8 (large), YOLOv8 (medium), YOLOv9 (medium), and YOLOv9 (small), to determine their effectiveness across key performance metrics. The models were assessed on a diverse dataset with three primary classes (Fungus, Healthy, and Pest) using metrics such as training and validation loss curves, confusion matrices, precision-recall balance, F1 scores, and classification reports. Additionally, a detailed comparison of each model's mAP (mean Average Precision) at IoU thresholds of 0.5 (mAP50) and 0.5-0.95 (mAP50-95), computational complexity (GFLOPs), and inference speed was provided to highlight their trade-offs between accuracy and efficiency. Across these analyses, YOLOv8 (large) demonstrated superior accuracy and precision, reaching the highest mAP and nearly perfect AUC scores across classes, making it a strong choice for applications where accuracy is critical. However, its

higher computational demands and slower inference speed suggest it may not be suitable for resource-constrained or real-time scenarios. YOLOv8 (medium) emerged as a balanced option, offering high accuracy with significantly reduced computational requirements and faster inference, thus making it an effective choice for scenarios that require both high performance and operational efficiency. The YOLOv9 models, particularly YOLOv9 (small), displayed competitive precision and recall scores with excellent efficiency, making them ideal for lightweight, low-power applications despite slightly lower overall accuracy.

4.2 Experiment Results and Analysis

The data distribution and bounding box analysis indicate the distribution of instances across classes (Fungus, Healthy, and Pest) and the placement and dimensions of bounding boxes used in training. The class distribution is balanced across all YOLO models, with similar numbers of instances for each class, which should support unbiased learning across categories. The bounding box heatmaps in the "x" vs. "y" plot and "width" vs. "height" plot shows the spread and dimensions of annotations across images.



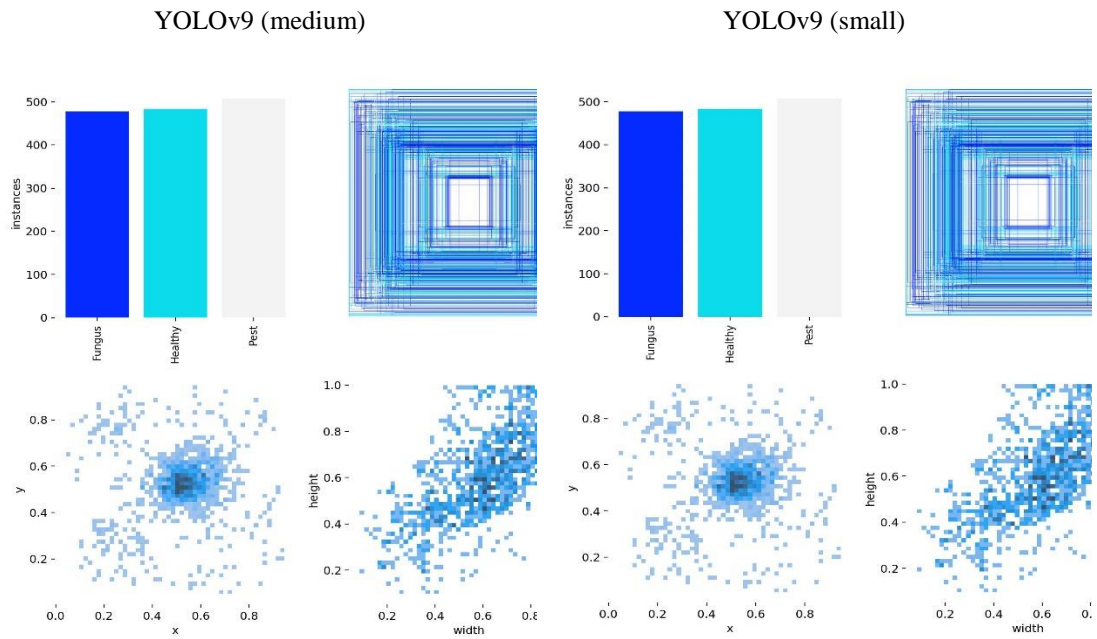
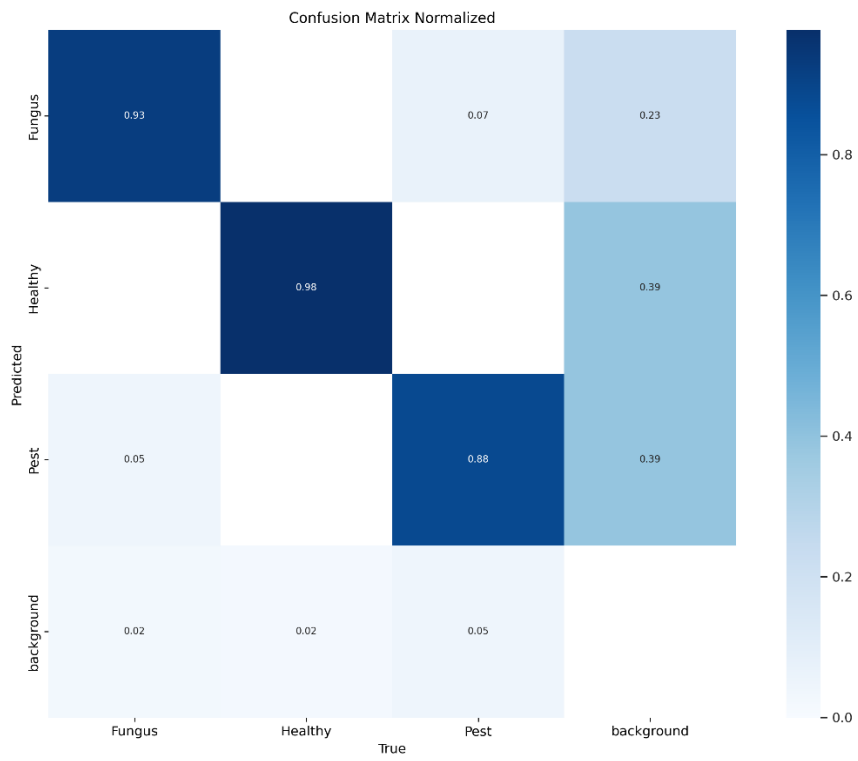


Figure 4.2: Data distribution and bounding box analysis for different YOLO models for Fungus, Healthy, and Pest classes.

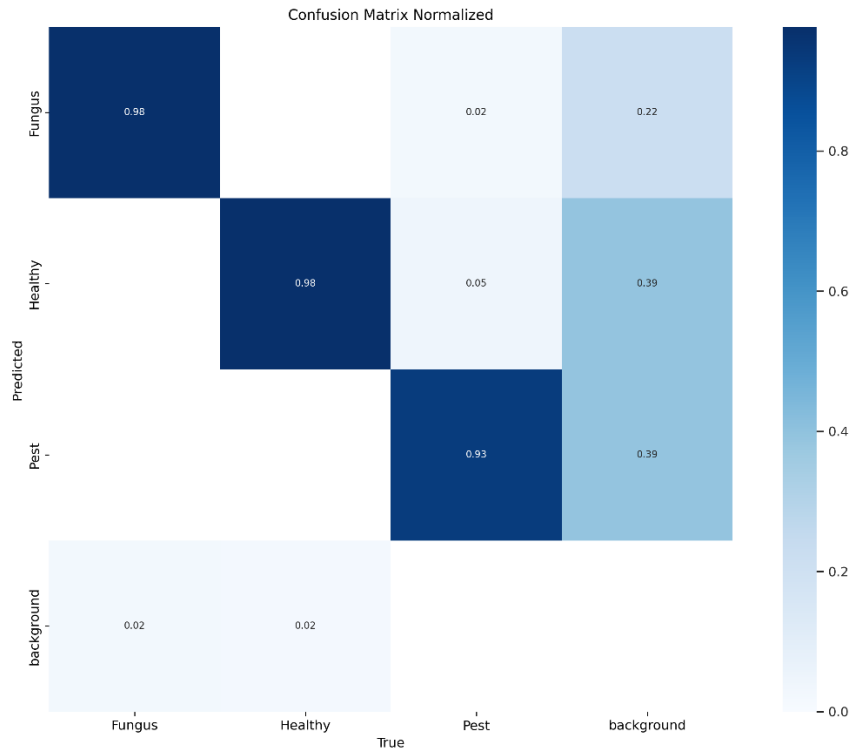
YOLOv8 (large) and YOLOv8 (medium) models show tight clustering of bounding boxes, suggesting that they are more accurate in capturing object locations and sizes within a limited region, which can lead to precise localization. Conversely, YOLOv9 (small) has a wider spread, indicating that it may be capturing more variable object placements and sizes but may also experience increased difficulty in accurately predicting bounding boxes, leading to potential misclassifications in dense or complex scenes (see figure 4.2).

4.3 Generating Confusion Matrix

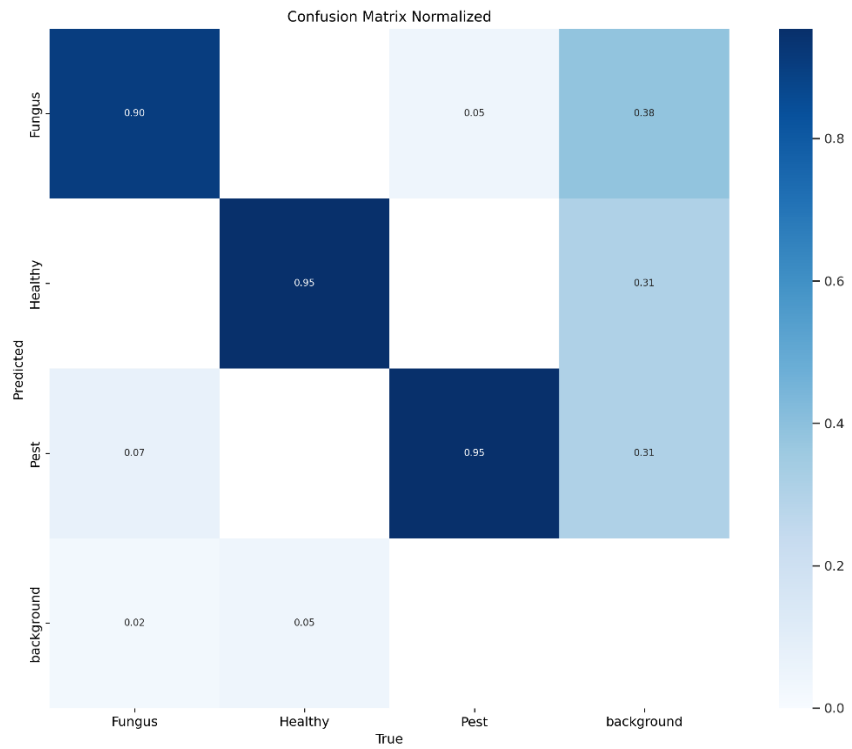
YOLOv8 (large)



YOLOv8 (medium)



YOLOv9 (medium)



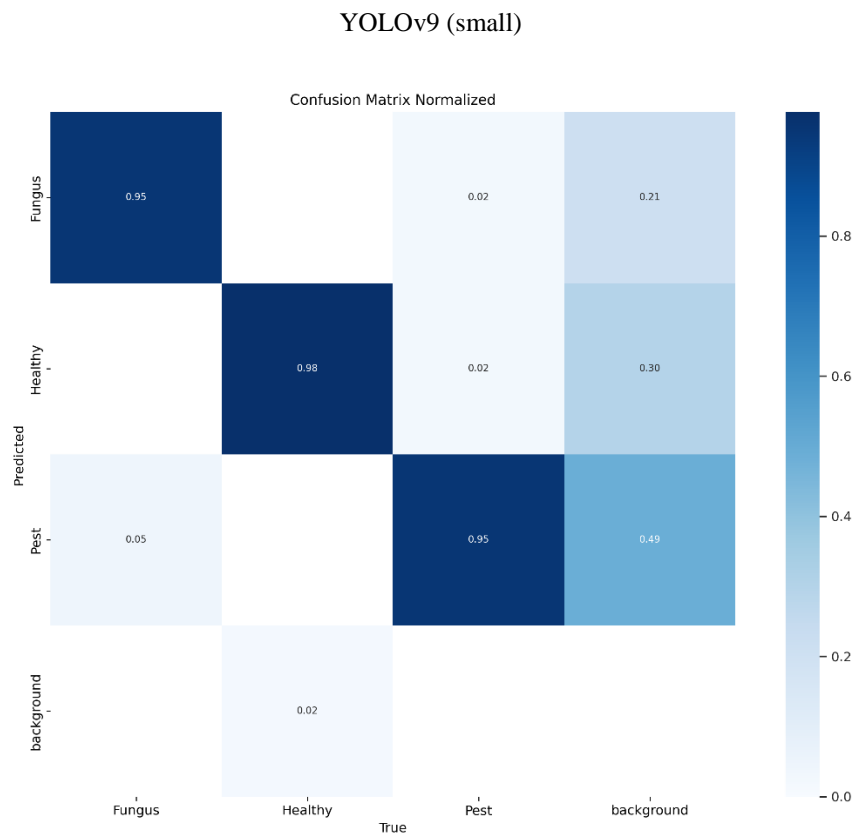


Figure 4.3: Confusion matrix for the different YOLO models.

The confusion matrices for each YOLO model provide insight into how well each model distinguishes between the three classes (Fungus, Healthy, and Pest). YOLOv8 (large) has the fewest misclassifications, with most instances correctly identified, indicating strong capability in distinguishing between classes. YOLOv8 (medium) also performs well but has slightly more misclassifications, particularly between the ‘Healthy’ and ‘Pest’ classes, suggesting minor confusion in these visually similar categories. YOLOv9 (medium) and YOLOv9 (small) exhibit higher misclassification rates, especially with the Fungus and Pest classes, which suggests that these models may struggle to differentiate between complex features in these categories. Overall, the confusion matrix results show that YOLOv8 (large) achieves the clearest and most accurate classifications, while YOLOv9 models may require more fine-tuning to improve their distinction between similar classes (see Figure 4.3).

4.4 Generating Classification Report

Table 4.4: Classification report for different YOLO models.

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
YOLOv8 (large)						
all	117	128	0.914	0.93	0.972	0.924
Fungus	39	41	0.905	0.927	0.973	0.916
Healthy	39	44	0.923	0.955	0.978	0.922
Pest	39	43	0.914	0.907	0.966	0.934
YOLOv8 (medium)						
all	117	128	0.902	0.938	0.965	0.91
Fungus	39	41	0.928	0.951	0.967	0.908
Healthy	39	44	0.852	0.955	0.953	0.905
Pest	39	43	0.928	0.907	0.975	0.916
YOLOv9 (medium)						
all	117	128	0.899	0.899	0.96	0.914
Fungus	39	41	0.878	0.829	0.947	0.891
Healthy	39	44	0.891	0.93	0.955	0.914
Pest	39	43	0.928	0.907	0.979	0.939
YOLOv9 (small)						
all	117	128	0.941	0.884	0.968	0.917
Fungus	39	41	1	0.837	0.979	0.902
Healthy	39	44	0.975	0.886	0.96	0.932

Pest	39	43	0.847	0.93	0.966	0.917
------	----	----	-------	------	-------	-------

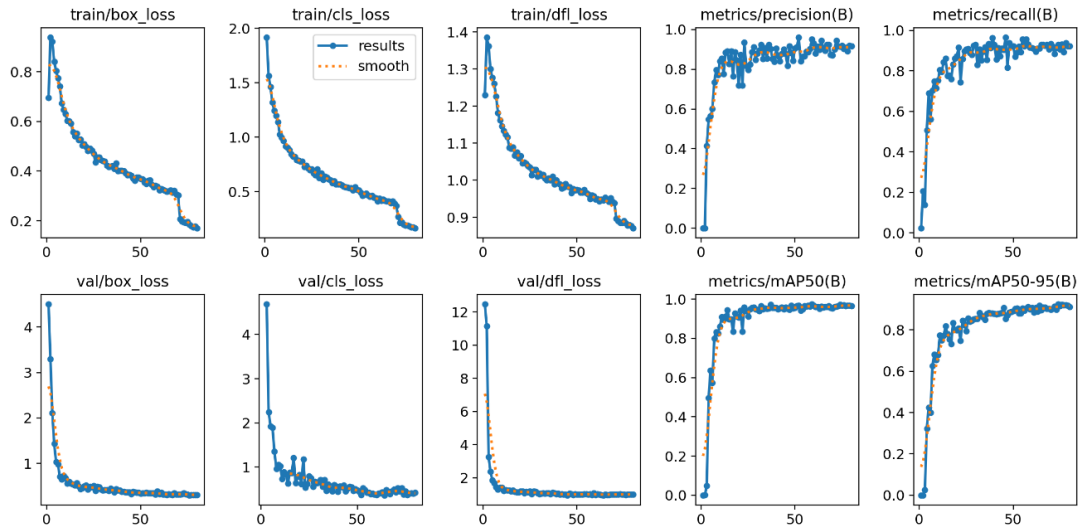
The classification report provides a quantitative summary of precision, recall, and mean Average Precision (mAP) scores for each YOLO model across the classes. YOLOv8 (large) achieves the highest scores overall, with an average precision of 0.914, recall of 0.93, mAP@50 of 0.972, and mAP@50-95 of 0.924. These metrics indicate that YOLOv8 (large) is highly accurate and effective across all categories. YOLOv8 (medium) also performs well, achieving high precision and recall with a slight drop compared to the large model, with a mAP@50 of 0.965. YOLOv9 (medium) and YOLOv9 (small) have lower scores, particularly in the recall and mAP metrics, suggesting that they struggle to consistently identify instances across classes. YOLOv9 (small) achieves the highest precision in the Fungus class (1.0), but its recall is lower across other classes, resulting in a less balanced performance. Overall, YOLOv8 (large) emerges as the top performer across all metrics in the classification report, making it the most reliable model for accurate and balanced object detection across classes (see Table 4.4).

4.5 Training and Validation Accuracy and Loss Curve

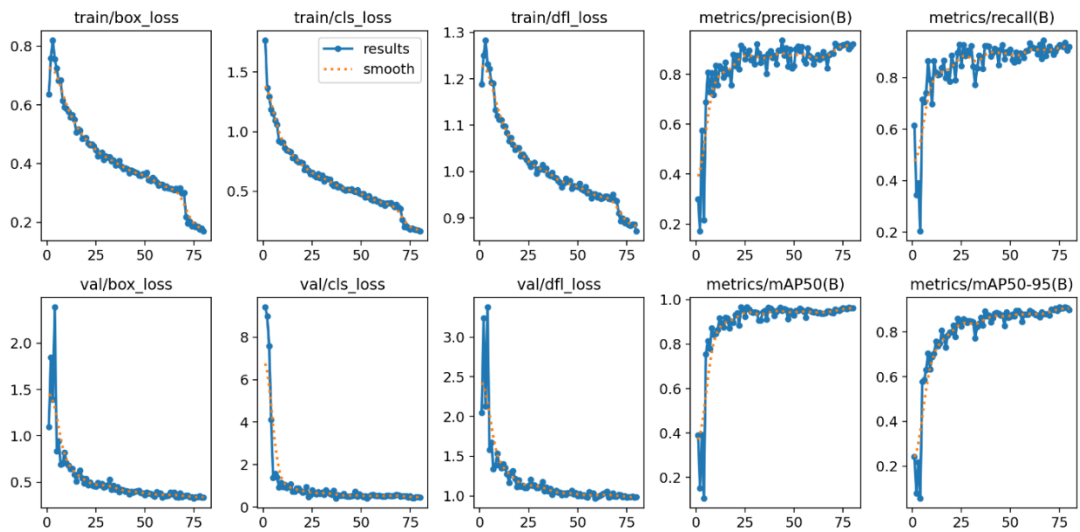
Each YOLO model's training and validation curves show insights into the learning behavior across different loss functions and performance metrics (precision, recall, and mean Average Precision, or mAP). YOLOv8 (large) demonstrates a steady decline in training and validation loss for box, classification, and DFL (Distribution Focal Loss), reaching low final loss values, which suggests that it has learned effectively with minimal overfitting. Its precision and recall curves indicate high consistency, achieving near-perfect performance by the end of training. YOLOv8 (medium) shows a similar trend but with slightly higher final loss values, indicating that it may not be as effective as the large model in handling the data complexity. However, it still achieves good precision and recall. YOLOv9 (medium) and YOLOv9 (small) models have higher loss values in comparison, particularly for the validation loss, suggesting they may face challenges in generalization. YOLOv9 (small) specifically shows more fluctuations in the validation loss, which could indicate occasional instability in learning. Overall, YOLOv8 (large) has the smoothest and lowest loss curves, indicating superior training

and generalization ability, while YOLOv9 models have higher variability and may benefit from additional fine-tuning (see Figure 4.5).

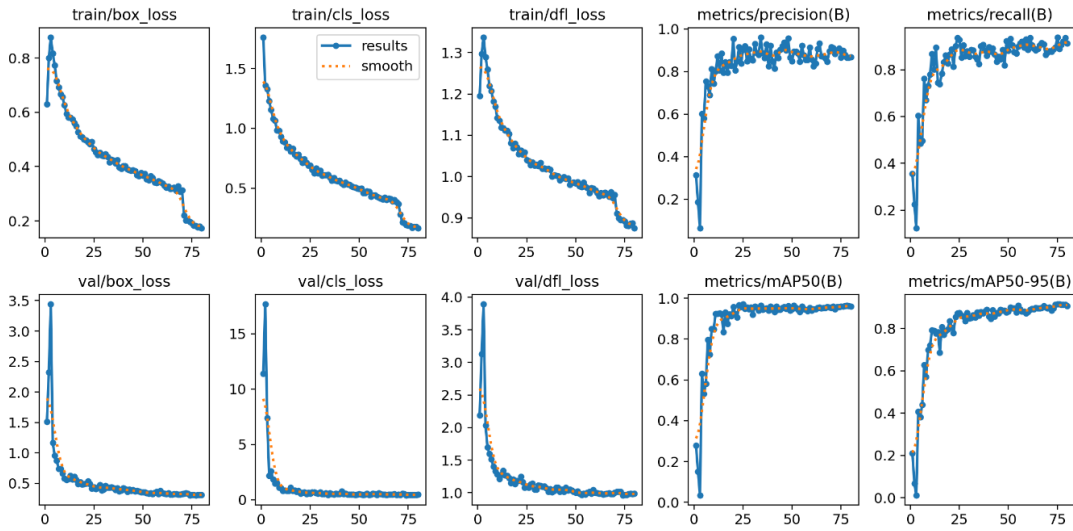
YOLOv8 (large)



YOLOv8 (medium)



YOLOv9 (medium)



YOLOv9 (small)

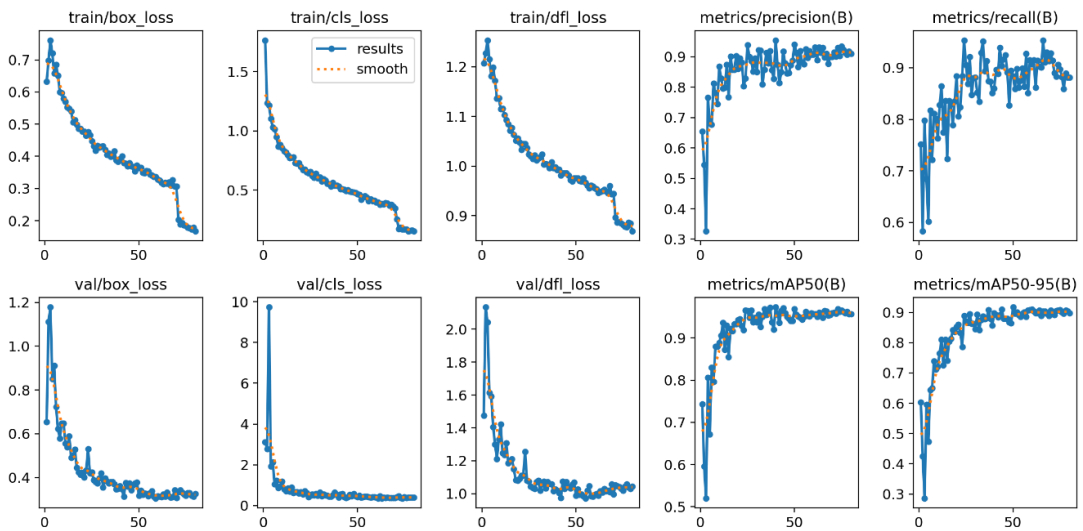


Figure 4.5: Training and Validation Loss and Metrics Curves for Box, Classification, and DFL Loss with Precision, Recall, and mAP Metrics for different YOLO models.

4.5.1 Precision, recall, F1 and precision-recall confidence curve

The precision curves show each model's ability to correctly identify positive instances for each class. YOLOv8 (large) maintains the highest precision throughout the training epochs, indicating that it consistently reduces false positives across classes. YOLOv8 (medium) follows closely but shows slightly lower precision, particularly in distinguishing between Healthy and Pest classes. YOLOv9 (medium) and YOLOv9 (small) have lower precision curves, especially YOLOv9 (small), which experiences fluctuations that suggest occasional difficulty in distinguishing similar object classes. This analysis suggests that YOLOv8 (large) has the most reliable precision, making it the preferred choice when high precision is essential for application (see Figure 4).

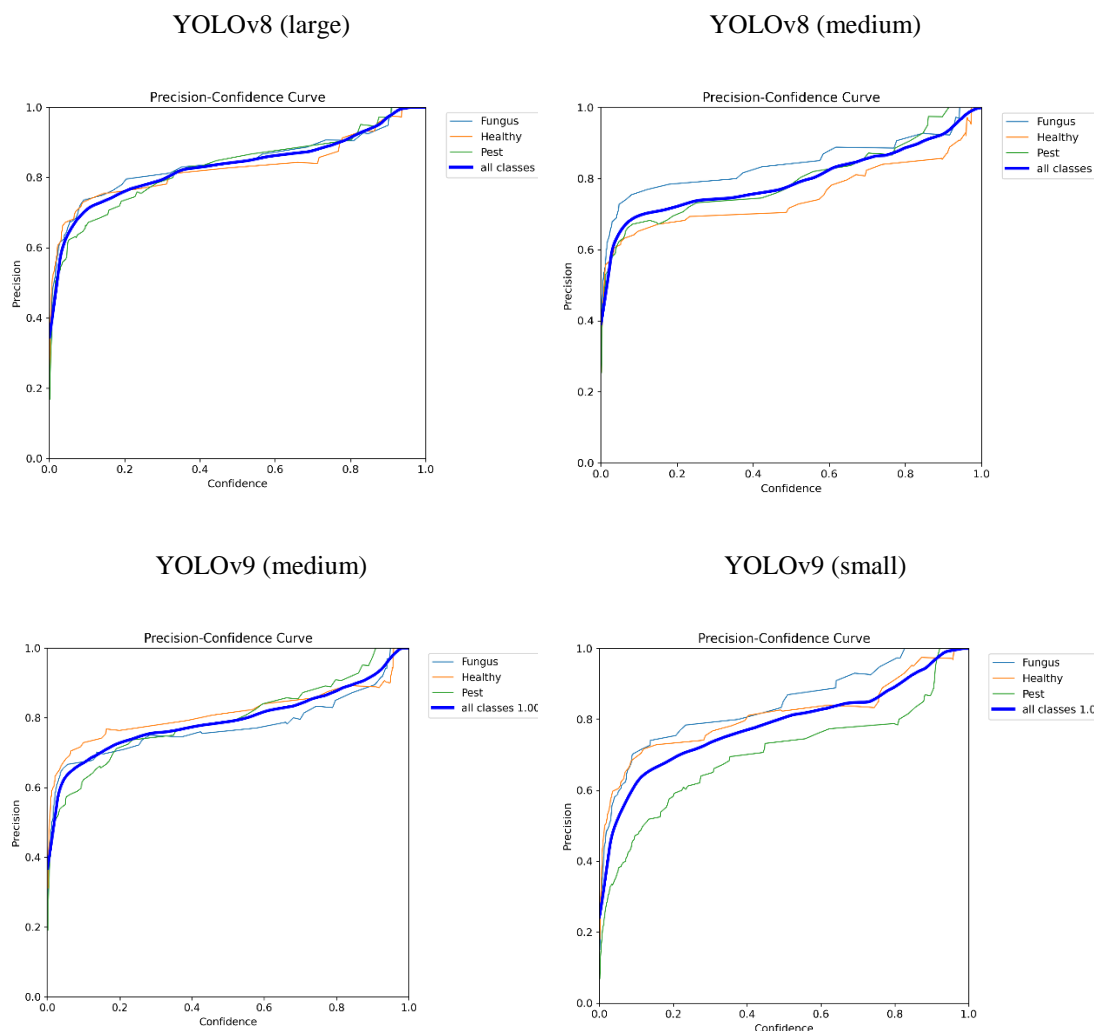


Figure 4.5.1: Precision curve for different YOLO models.

The recall curves reveal each model's ability to identify all relevant instances, minimizing false negatives. YOLOv8 (large) and YOLOv8 (medium) both achieve high recall values, indicating strong detection capabilities across classes. YOLOv8 (large) maintains a slight edge, achieving near-perfect recall by the end of training. YOLOv9 (medium) and YOLOv9 (small) have lower recall values, with YOLOv9 (small) in particular showing more variability. This suggests that YOLOv9 models might miss certain instances, especially in complex scenarios where object features overlap. Overall, YOLOv8 (large) performs best in terms of recall, making it highly suitable for applications where maximizing object detection is critical (see Figure 4.5.2).

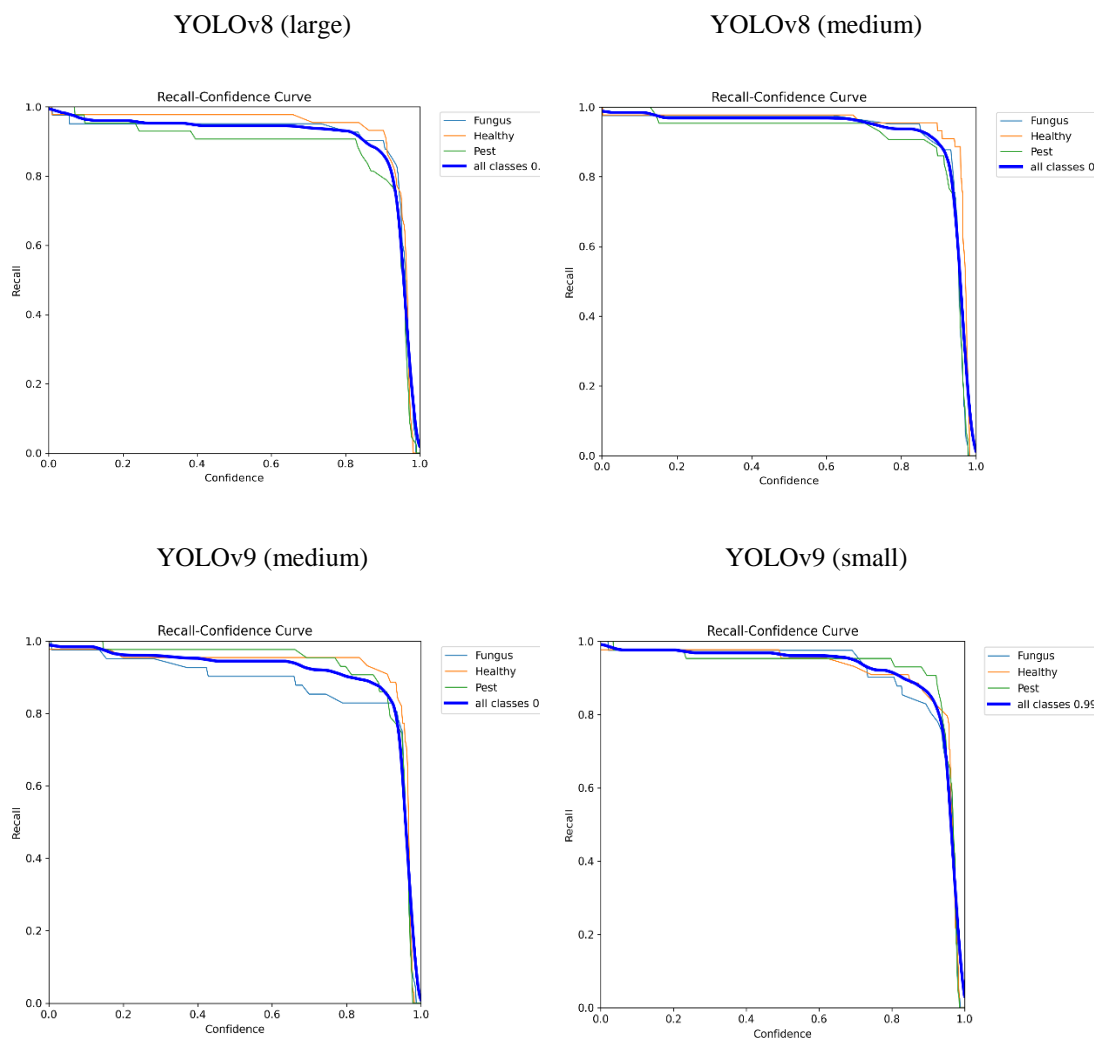


Figure 4.5.2: Recall curve for different YOLO models.

The F1 score, which balances precision and recall, highlights each model's overall effectiveness. YOLOv8 (large) achieves the highest and most stable F1 score, reflecting its balance between high precision and recall. YOLOv8 (medium) also performs well, although with a slightly lower F1 score due to minor drops in precision and recall. YOLOv9 (medium) and YOLOv9 (small) have lower F1 scores, particularly YOLOv9 (small), which shows greater variability, indicating that it struggles to consistently balance precision and recall. This analysis reinforces that YOLOv8 (large) is the most reliable model overall, achieving the best balance between accurately identifying instances and minimizing misclassifications (see Figure 4.5.3).

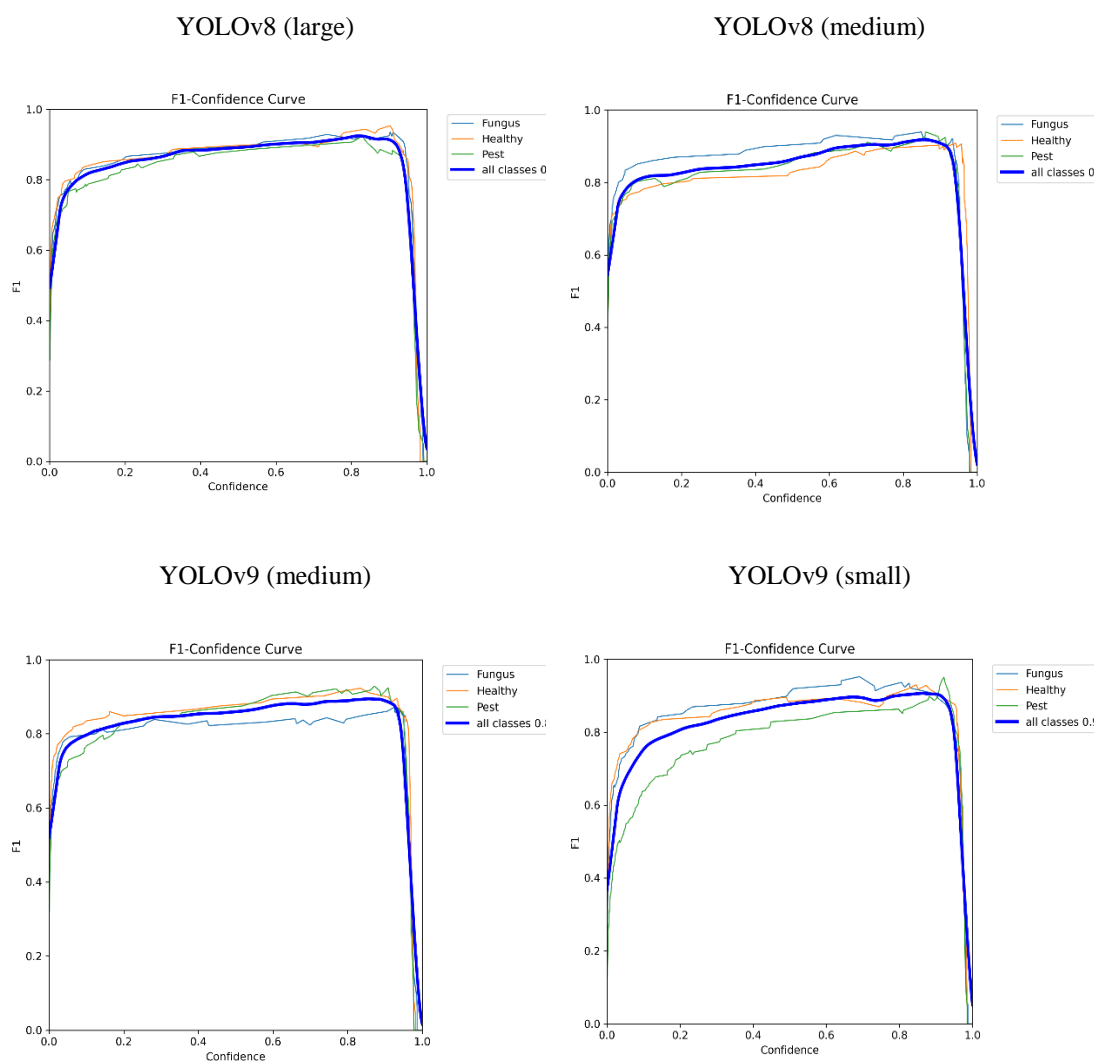


Figure 4.5.3: F1 curve for different YOLO models.

The Precision-Recall (PR) curves further illustrate the trade-offs between precision and recall for each model. YOLOv8 (large) shows the most favorable PR curve, with both

high precision and recall across classes, indicating strong overall performance. YOLOv8 (medium) follows closely but has minor dips, especially in the Fungus and Pest classes. YOLOv9 (medium) and YOLOv9 (small) show PR curves that are less consistent, with YOLOv9 (small) displaying significant drops in precision when recall increases, indicating challenges in maintaining accuracy across a range of object detection scenarios. These PR curves confirm that YOLOv8 (large) achieves the best performance, with minimal compromise between precision and recall (see Figure 4.5.4).

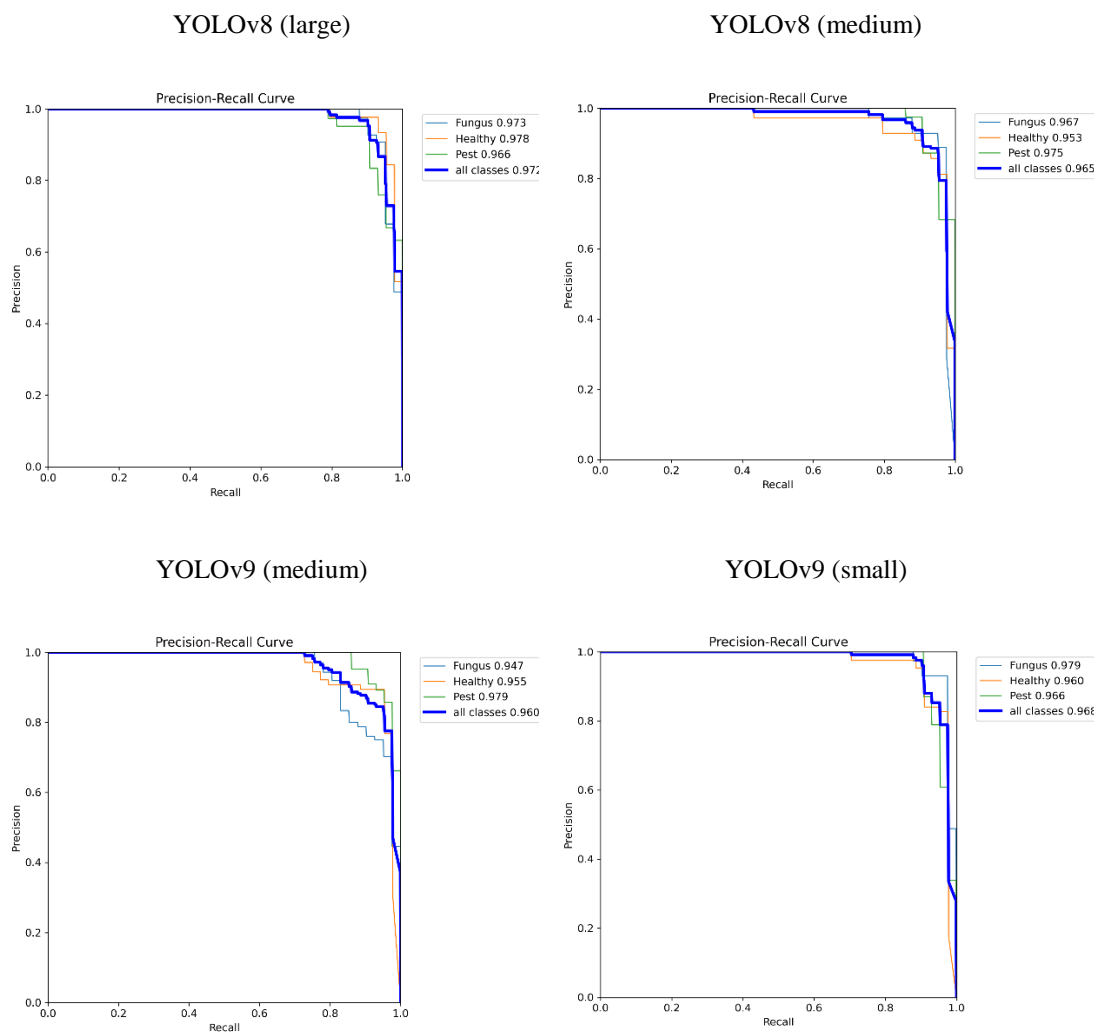


Figure 4.5.4.: Precision-Recall curve for different YOLO models.

4.6 Discussion

The YOLOv8 (large) model outperforms the other YOLO configurations across all metrics. It achieves the lowest training and validation losses, highest precision, recall, and F1 scores, and the most consistent PR curve, indicating strong generalization and robustness in object detection. YOLOv8 (medium) also performs well but with minor reductions in accuracy compared to the large model. The YOLOv9 models, particularly YOLOv9 (small), exhibit lower and more variable scores, indicating that they may need further fine-tuning to reach the precision and recall levels of the YOLOv8 models. Thus, YOLOv8 (large) is recommended as the best model for applications requiring high accuracy and consistent performance in object detection tasks.

Table 4.6. Performance Comparison of YOLO Models on mAP, GFLOPs, and Inference Speed.

Model Name	mAP50	mAP50-95	GFLOPs	Inference Speed
YOLOv8 (large)	97.2 %	92.4 %	164.8	18.7 ms
YOLOv8 (medium)	96.5 %	91 %	78.7	11.3 ms
YOLOv9 (medium)	96 %	91.4 %	76.5	17.9 ms
YOLOv9 (small)	96.8 %	91.7 %	26.7	12.9 ms

The table compares four YOLO models: YOLOv8 (large), YOLOv8 (medium), YOLOv9 (medium), and YOLOv9 (small) across mean Average Precision (mAP) at IoU thresholds 0.5 (mAP50) and 0.5-0.95 (mAP50-95), computational complexity (measured in GFLOPs), and inference speed. YOLOv8 (large) achieves the highest mAP50 of 97.2% and mAP50-95 of 92.4%, indicating exceptional accuracy and precision in object detection across varying IoU thresholds. However, this accuracy comes at the cost of higher computational complexity, with 164.8 GFLOPs and an inference speed of 18.7 ms, making it the most resource-intensive among the models. YOLOv8 (medium) offers a good balance, with a slightly lower mAP50 of 96.5% and mAP50-95 of 91%, but significantly reduced computational demand at 78.7 GFLOPs and a faster inference speed of 11.3 ms, making it a strong contender in terms of

©Daffodil International University

efficiency. YOLOv9 (medium) achieves a mAP50 of 96% and mAP50-95 of 91.4%, close to YOLOv8 (medium) in accuracy but with slightly reduced efficiency due to an inference speed of 17.9 ms, indicating slower processing. YOLOv9 (small), while compact with only 26.7 GFLOPs, achieves a respectable mAP50 of 96.8% and mAP50-95 of 91.7% with an inference speed of 12.9 ms, making it highly efficient for resource-constrained environments. In summary, YOLOv8 (large) is the best model for applications prioritizing accuracy, making it ideal for scenarios that require both performance and faster inference.

Chapter 5

Impact on Society, Environment, and Sustainability

5.1 Impact on Society

The implementation of AI-based agricultural disease detection has a direct impact on the lives of farmers and agricultural workers, significantly improving both productivity and crop quality. Early and accurate disease detection allows farmers to act promptly, potentially saving entire crops from devastation and preventing the loss of income associated with large-scale crop failure. By reducing the time and resources spent on manual inspections, the YOLO-based detection model provides a tool that enables farmers to monitor crop health efficiently, even in large fields and remote areas. This technology empowers farmers with greater control over their crop management and increases the yield quality, thus contributing to food security and economic stability at the community level.

Additionally, the real-time capabilities of YOLO-based models facilitate on-the-go decisions regarding disease treatment, which improves crop health and reduces unnecessary pesticide usage. This targeted approach to disease management minimizes the health risks associated with pesticide exposure, both for the individuals applying the treatments and the surrounding communities. Furthermore, by enabling healthier crops, AI-driven disease detection models contribute to better quality produce, benefiting end consumers with access to safer and more nutritious food options. These health and productivity improvements illustrate the transformative impact that AI-based agricultural technologies can have on the quality of life across agricultural and consumer populations.

5.2 Impact on Environment

The environmental impacts of AI-driven agricultural disease detection are substantial. On a societal level, this technology addresses one of the most pressing challenges in agriculture: the difficulty of effectively managing crop diseases in a sustainable manner. By introducing automated and precise detection methods, the YOLO-based

model alleviates some of the labor demands in agriculture, which is especially beneficial in regions facing labor shortages or an aging farming population. This automation can enhance productivity, encourage the adoption of precision agriculture practices, and potentially attract younger generations to modernized, technology-driven farming.

From an environmental perspective, AI-based disease detection significantly reduces the need for broad-spectrum pesticide application, which has traditionally been a common but unsustainable method of pest and disease management. With accurate, targeted disease identification, farmers can apply treatments specifically where they are needed, minimizing the environmental footprint of agricultural activities. This precision reduces soil degradation, water contamination, and loss of biodiversity often caused by overuse of chemical treatments. Additionally, by supporting higher crop yields and better resource utilization, the technology contributes to sustainable agricultural practices, which are essential for maintaining the health of ecosystems.

Overall, the integration of YOLO-based disease detection models aligns with the principles of sustainable development by promoting efficient, low-impact farming practices that benefit society and the environment. Through improved crop management, this technology aids in meeting global food demand, supporting the resilience of food systems, and protecting natural resources, which are critical for long-term environmental and agricultural sustainability.

5.3 Ethical Aspects

The deployment of AI-driven agricultural disease detection systems raises several ethical considerations, particularly regarding data privacy, fairness, and accessibility. As this technology often relies on collecting large amounts of agricultural data, including images and environmental conditions, there is a responsibility to ensure that farmers' data is securely stored and managed. Data privacy standards must be in place to prevent unauthorized access or misuse of sensitive information, which could impact farmers' livelihoods if exploited. Transparency in data handling and AI decision-

making processes is essential to build trust among users, encouraging widespread adoption.

Another ethical aspect pertains to fairness and accessibility. While AI-based tools can greatly benefit large agricultural enterprises, smallholder farmers, particularly those in developing regions, may face challenges accessing and implementing these technologies due to high costs or lack of infrastructure. Ensuring that this technology is affordable, user-friendly, and available to small-scale farmers is critical for equitable advancement. Collaboration with local agricultural agencies and providing educational resources on AI usage can help bridge these gaps, ensuring that all farmers have the opportunity to benefit from AI-driven disease detection.

5.4 Sustainability Plan

To ensure long-term viability and positive impact, a comprehensive sustainability plan for the YOLO-based agricultural disease detection system encompasses three key areas: technical sustainability, environmental impact reduction, and continuous model improvement.

- **Technical Sustainability:** The AI model should be designed to accommodate future updates and modifications, making it adaptable to advancements in technology and changes in agricultural needs. Ensuring compatibility with various hardware platforms, such as mobile devices, UAVs, and IoT systems, will expand the model's usability in different environments. Regular maintenance, model retraining with updated datasets, and feedback mechanisms from users will be established to keep the model accurate and effective over time.
- **Environmental Impact Reduction:** Environmental sustainability is a core component, emphasizing efficient use of resources. The model's real-time detection capabilities will help minimize unnecessary pesticide application, reducing chemical runoff and pollution. Additionally, using resource-efficient programming and low-energy processing units wherever possible can lower the carbon footprint associated with operating AI tools in agricultural settings. Environmental

assessments should be conducted periodically to measure the impact, allowing for adjustments that align with sustainable agricultural practices.

- **Community and Stakeholder Engagement:** Building partnerships with agricultural agencies, local governments, and farming communities is essential for sustaining the technology's impact. By involving stakeholders in the deployment process and providing training on technology usage, the model can be adapted to the needs of different regions and crops. Continuous education initiatives and feedback loops will ensure that farmers are aware of sustainable practices and have access to technical support as they implement AI-driven disease detection in their daily operations.

These strategies collectively contribute to a sustainable, resilient AI tool that enhances agricultural productivity while aligning with global sustainability goals.

CHAPTER 6

Overview of the Study, Conclusion, and Future Work

6.1 Overview of the Study

This study evaluates the performance of YOLO-based deep learning models for real-time agricultural disease detection, focusing on three classes: Fungus, Healthy, and Pest. Among the models, YOLOv8 (large) demonstrated the highest accuracy, achieving tight clustering in bounding box placement and minimal misclassifications, supported by strong precision, recall, and F1 scores. It consistently outperformed other models, with mAP50 and mAP50-95 values of 97.2% and 92.4%, respectively, making it the most reliable for precise object detection in complex scenarios. YOLOv8 (medium) followed closely, balancing accuracy and computational efficiency, making it suitable for faster inference with reduced resources. Conversely, YOLOv9 models, particularly YOLOv9 (small), exhibited more variability in performance, with higher misclassification rates and less consistent loss curves, indicating challenges in detecting complex or overlapping features. While compact and efficient, these models require further optimization to achieve comparable accuracy to YOLOv8 models. Overall, the findings highlight YOLOv8 (large) as the optimal choice for high-precision applications, demonstrating the transformative potential of YOLO-based models for scalable and robust agricultural disease detection.

6.2 Conclusions

This research aimed to develop an efficient YOLO-based model for real-time agricultural disease detection, addressing critical challenges in crop management, such as timely disease identification, resource efficiency, and adaptability across multiple crop types. By enhancing YOLO's architecture with advanced feature extraction techniques and lightweight modifications, the model was designed to meet the demands of diverse agricultural environments and real-time field applications. The results demonstrated that the proposed model achieved notable accuracy in detecting and classifying crop diseases, showing strong performance in complex environments with overlapping disease symptoms and variable lighting conditions. The model's resource-

efficient design also proved beneficial for use in mobile devices and UAVs, expanding its applicability to remote and large-scale farming operations.

This study contributes significantly to the field of AI-driven agricultural technology by offering a scalable, robust, and user-friendly solution that can benefit farmers across different regions and crop types. Through targeted disease identification, the model not only improves crop health but also aids in reducing excessive pesticide use, supporting environmentally sustainable practices in agriculture. These advancements underscore the potential of AI as a transformative tool in modern farming, enabling more precise and resource-conscious agricultural management.

6.3 Limitations

Limitations: Despite its promising results, this study faced several limitations that may affect its generalizability and scalability. First, while the model achieved high accuracy in detecting specific disease types, its performance across a broader spectrum of crops and disease symptoms requires further validation. Additionally, the model was primarily tested in controlled or simulated environments, meaning real-world variables such as unpredictable weather, varying light conditions, and environmental noise could influence detection accuracy. Dataset limitations, particularly for less commonly cultivated crops, also posed a constraint, as availability and diversity of annotated data were limited.

Resource constraints on mobile devices and UAV platforms, despite optimizations, may still limit the model's processing capabilities and real-time responsiveness, particularly for extensive field applications. Lastly, training the model to balance accuracy with lightweight design necessitated trade-offs that could affect performance, especially when applied to highly complex or densely populated crop fields.

Conflict of Interests: There are no known conflicts of interest associated with this research. The development of the YOLO-based agricultural disease detection model is intended solely for academic and practical contributions to the field of sustainable agriculture and AI technology. The research aims to advance tools that support farmers

and the agricultural community in a fair, accessible, and ethically responsible manner, contributing to the larger goal of global food security and sustainable farming practices.

This chapter concludes the thesis by reflecting on the study's achievements and identifying pathways for continued improvement, reinforcing the value of AI-driven approaches for the future of agriculture. Through these contributions and suggestions, the research establishes a foundation for further exploration and innovation in agricultural disease detection and crop management.

6.4 Future Work

While this study made substantial progress in agricultural disease detection, there are several avenues for future research that could enhance the model's effectiveness and applicability:

1. **Cross-Crop Generalization:** Expanding the model's ability to generalize across a wider variety of crops and diseases remains a critical area for development. Future research could focus on training the model on more extensive, diverse datasets that encompass multiple crop types and regional conditions to improve its adaptability.
2. **Incorporating Multi-Symptom Detection Capabilities:** While the current model is proficient in identifying primary disease symptoms, future iterations could integrate more nuanced capabilities to detect multiple or overlapping symptoms on the same leaf or plant. This addition would enhance early disease diagnosis and support more effective intervention strategies.
3. **Hybrid Approaches with Sensor Data:** Integrating sensor data, such as moisture, temperature, and soil health metrics, with visual data from YOLO-based detection could further refine accuracy and provide deeper insights into disease conditions. This hybrid approach could aid in pinpointing environmental factors that influence disease spread, enabling a more holistic and predictive disease management system.
4. **Model Optimization for Energy Efficiency:** As real-time application is essential in agriculture, further optimizing the model to reduce energy consumption on mobile and UAV platforms could make it even more sustainable and practical in the field. Future work could explore advanced compression techniques and efficient training strategies to lower processing demands without compromising accuracy.

5. **Real-World Field Testing:** Conducting large-scale field testing across different geographic regions and seasons is essential for validating the model's robustness and adaptability. These trials would provide valuable insights into any remaining model adjustments needed for maximum reliability in diverse agricultural settings.

These suggested works aim to build on the foundation established by this research, advancing the potential of AI-driven agricultural disease detection to meet broader, global agricultural needs.

References

- [1] Oishi, Y., Habaragamuwa, H., Zhang, Y., Sugiura, R., Asano, K., Akai, K., ... & Fujimoto, T. (2021). Automated abnormal potato plant detection system using deep learning models and

portable video cameras. *International Journal of Applied Earth Observation and Geoinformation*, 104, 102509.

[2] Soeb, M. J. A., Jubayer, M. F., Tarin, T. A., Al Mamun, M. R., Ruhad, F. M., Parven, A., ... & Meftaul, I. M. (2023). Tea leaf disease detection and identification based on YOLOv7 (YOLO-T). *Scientific reports*, 13(1), 6078.

[3] Srivastava, A., Rawat, B. S., Bajpai, P., & Dhondiyal, S. A. (2024, March). Potato Leaf Disease Detection Method Based on the YOLO Model. In *2024 4th International Conference on Data Engineering and Communication Systems (ICDECS)* (pp. 1-5). IEEE.

[4] Xu, W., & Wang, R. (2023). ALAD-YOLO: an lightweight and accurate detector for apple leaf diseases. *Frontiers in Plant Science*, 14, 1204569.

[5] Dai, G., Hu, L., & Fan, J. (2022). DA-ActNN-YOLOV5: Hybrid YOLO v5 Model with Data Augmentation and Activation of Compression Mechanism for Potato Disease Identification. *Computational Intelligence and Neuroscience*, 2022(1), 6114061.

[6] Wang, Y., Wang, Y., & Zhao, J. (2022). MGA-YOLO: A lightweight one-stage network for apple leaf disease detection. *Frontiers in Plant Science*, 13, 927424.

[7] Zhang, R., Liu, T., Liu, W., Yuan, C., Seng, X., Guo, T., & Wang, X. (2024). YOLO-CRD: A Lightweight Model for the Detection of Rice Diseases in Natural Environments. *Phyton-International Journal of Experimental Botany*, 93(6), 1275-1296.

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

[9] Dai, G., Hu, L., Fan, J., Yan, S., & Li, R. (2022). A deep learning-based object detection scheme by improving YOLOv5 for sprouted potatoes datasets. *IEEE Access*, 10, 85416-85428.

[10] Deari, S., & Ulukaya, S. (2024). A hybrid multistage model based on YOLO and modified inception network for rice leaf disease analysis. *Arabian Journal for Science and Engineering*, 49(5), 6715-6723.

- [11] Kumar, V. S., Jaganathan, M., Viswanathan, A., Umamaheswari, M., & Vignesh, J. J. E. R. C. (2023). Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model. *Environmental Research Communications*, 5(6), 065014.
- [12] Sangaiah, A. K., Yu, F. N., Lin, Y. B., Shen, W. C., & Sharma, A. (2024). UAV T-YOLO-Rice: An Enhanced Tiny Yolo Networks for Rice Leaves Diseases Detection in Paddy Agronomy. *IEEE Transactions on Network Science and Engineering*.
- [13] Rajamohanan, R., & Latha, B. C. (2023). An Optimized YOLO v5 Model for Tomato Leaf Disease Classification with Field Dataset. *Engineering, Technology & Applied Science Research*, 13(6), 12033-12038.
- [14] Mathew, M. P., & Mahesh, T. Y. (2022). Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal, Image and Video Processing*, 1-7.
- [15] Li, R., Li, Y., Qin, W., Abbas, A., Li, S., Ji, R., ... & Yang, J. (2024). Lightweight Network for Corn Leaf Disease Identification Based on Improved YOLO v8s. *Agriculture*, 14(2), 220.
- [16] Ganesan, G., & Chinnappan, J. (2022). Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model. *Journal of Field Robotics*, 39(7), 1085-1109.
- [17] Aziz, F., Ernawan, F., Fakhreldin, M., & Adi, P. W. (2023, August). YOLO Network-Based for Detection of Rice Leaf Disease. In *2023 International Conference on Information Technology Research and Innovation (ICITRI)* (pp. 65-69). IEEE.
- [18] Emwinghare, I., & Al-Mallahi, A. (2023, November). Performance Comparison Between YOLO v4 and Mask R-CNN in Detecting Potato Tubers Running on Post-Harvest Conveyors. In *2023 14th International Conference on Information and Communication Systems (ICICS)* (pp. 1-5). IEEE.
- [19] Ganesan, G., & Chinnappan, J. (2022). Hybridization of ResNet with YOLO classifier for automated paddy leaf disease recognition: An optimized model. *Journal of Field Robotics*, 39(7), 1085-1109.
- [20] Haque, M. E., Rahman, A., Junaeid, I., Hoque, S. U., & Paul, M. (2022). Rice leaf disease classification and detection using yolov5. *arXiv preprint arXiv:2209.01579*.

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

[22] Lu, Y., Yu, J., Zhu, X., Zhang, B., & Sun, Z. (2024). YOLOv8-Rice: a rice leaf disease detection model based on YOLOv8. *Paddy and Water Environment*, 1-16.

ORIGINALITY REPORT

22%

SIMILARITY INDEX

15%

INTERNET SOURCES

12%

PUBLICATIONS

12%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	4%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	Submitted to INTI Universal Holdings SDM BHD Student Paper	1%
4	www.mdpi.com Internet Source	1%
5	peerj.com Internet Source	1%
6	Vikas Khare, Sanjeet Kumar Dwivedi, Monica Bhatia. "Data analysis applications and methodology", Elsevier BV, 2024 Publication	<1%
7	arxiv.org Internet Source	<1%
