

**Alliance School of Liberal Arts**

MSc Data Science

Subject:Machine Learning Techniques for Image Processing

Topic: DETECTION OF RICE LEAF DISEASE USING MACHINE LEARNING TECHNIQUE

Submitted To: Mr. Rajesh Sharma R

*Submitted by*

*Meghana Jathan*

*2023MDTS07ALA028*

**DETECTION OF RICE LEAF DISEASE USING MACHINE LEARNING TECHNIQUE**

**Author: Meghana Jathan, Dr Rajesh Sharma R**

**Research Associate**

**Alliance school of Liberal Arts and Humanities**

**Alliance University, Bengaluru, India**

[**smeghanads23@sam.alliance.edu.in**](mailto:smeghanads23@sam.alliance.edu.in)

**Associate Professor**

**Department of CSE, Alliance University**

**Bengaluru, India**

[**rajeshsharma.r@alliance.edu.in**](mailto:rajeshsharma.r@alliance.edu.in)

**ABSTRACT**

Numerous fields have been transformed by artificial intelligence and machine learning. The productivity and quality of rice are seriously threatened by a number of diseases, despite the fact that rice cultivation is crucial to global food security. Innovative approaches to sickness detection and treatment are required to address these challenges. This research focused on YOLOv8, the latest version of this well-liked technology, and its performance in object detection tasks. This study explores the potential of YOLOv8n and YOLOv8s in rice leaf disease detection in order to enhance agricultural sustainability by providing rapid and accurate disease diagnosis.

Data collection, annotation, and model training utilising the YOLO object detection framework comprise the methodology used in this work. Data annotation is done with Cvat.ai. A sizable collection of images of different sick rice leaves is collected and annotated to aid with model training. YOLOv8n and YOLOv8s were then trained to identify several rice leaf diseases using this dataset. Rapid and precise disease diagnosis is made possible by the architecture's inherent efficacy and efficiency.

The results show that YOLOv8n and YOLOv8s are effective in accurately identifying rice leaf diseases. Prompt intervention steps can limit yield losses by ensuring that the model achieves the F1-Confidence curve, Precision-Recall curve, training and validation box and class loss, and confusion-matrix. These findings are significant because they could result in early disease detection and proactive disease management, which would eventually improve agricultural health and output.

The prospective uses of the YOLOv8 model will be examined with the aid of this study. YOLOv8 is a powerful tool for detecting rice leaf disease and is unparalleled in its object detection capabilities. It is a helpful tool for agricultural practices because of its accuracy, effectiveness, and resilience, which enable farmers and other stakeholders to effectively manage the threat of disease. This effort places a strong emphasis on using cutting-edge technology to enhance agricultural sustainability.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No** | **Table Title** | **Page No** |
| 2.1 | Literature Review papers | 7 |
| 3.1 | Data Set | 22 |
| 3.2 | YOLOv8 Models | 29 |
| 4.1 | Model Description | 33 |
| 4.2 | Experimental results for D1 and D3 data set | 41 |
| 4.3 | Experimental results for D2 data set | 41 |
| 4.4 | Comparison evaluation of D1 dataset | 46 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No** | **Figure Title** | **Page No** |
| 2.1 | Types of rice leaf diseases | 12 |
| 2.2 | Structure of YOLO | 14 |
| 3.1 | Workflow of Yolov8 Model | 21 |
| 3.2 | Images from Data set | 23 |
| 3.3 | Data organizing format | 23 |
| 3.4 | Annotation of rice blast disease | 24 |
| 3.5 | Label File | 24 |
| 3.6 | YOLOv8 Architecture Module | 25 |
| 3.7 | YOLOv8 Architecture Workflow | 26 |
| 4.1 | Training and Validation Box Loss | 34 |
| 4.2 | Training and Validation Class Loss | 35 |
| 4.3 | PR Curve | 36 |
| 4.4 | F1-Confidence Curve | 37 |
| 4.5 | Confusion Matrix | 38 |
| 4.6 | Test Result of Bacterial Leaf Blight Diseases | 39 |
| 4.7 | Test Result of Brown Spot Diseases | 39 |
| 4.8 | Test Result of Rice Blast Diseases | 40 |
| 4.9 | Training and Validation Box Loss | 42 |
| 4.10 | Training and Validation Class Loss | 42 |
| 4.11 | PR Curve | 43 |
| 4.12 | F1-Confidence Curve | 43 |
| 4.13 | Confusion Matrix | 44 |
| 4.14 | Test Result on D2 Models | 45 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Contents** | | | | | |
|  | | | | | Page No |
| Abstract | | | |  |  |
| List Of Figures | | | |  |  |
| List Of Tables | | | |  |  |
|  | | | | | |
| **Chapter 1** | | | **INTRODUCTION** | |  |
|  | **1.1** | Introduction | | | 1 |
|  | **1.2** | Motivation | | | 2 |
|  | **1.3** | Objective | | | 3 |
|  | **1.4** | Organization of Report | | | 3 |
|  | | | | | |
| **Chapter 2** | | | **BACKGROUND THEORY** | |  |
|  | **2.1** | Literature Review | | | 4 |
|  | **2.2** | Background Theory | | | 11 |
|  | **2.3** | Evaluation Metrics | | | 16 |
|  | | | | | |
| **Chapter 3** | | | **METHODOLOGY** | |  |
|  | **3.1** | Data Set | | | 22 |
|  | **3.2** | Data Annotation | | | 23 |
|  | **3.3** | YOLO Model Architecture | | | 25 |
|  | **3.4** | Implementation of Model | | | 24 |
|  | | | | | |
| **Chapter 4** | | | **RESULT ANALYSIS** | | 33 |
|  | | | | | |
| **Chapter 5** | | | **CONCLUSION AND FUTURE SCOPE** | |  |
|  | **5.1** | Conclusions and Significance of Results | | | 47 |
|  | **5.2** | Future Scope of Work | | | 48 |
|  | | | | | |
| **Chapter 6** | | | **HEALTH, SAFETY, RISK, AND ENVIRONMENT** | | 49 |
|  | | | | | |
| **REFERENCES** | | | | | 50 |

# CHAPTER 1 INTRODUCTION

The rationale, goals, and methods for developing rice leaf disease detection are established in this chapter, which also introduces a real-time object detection system employing YOLO. The history of the project, its significance, and its intended outcomes are all explored in detail.

## Introduction

The majority of the world's population depends on agriculture, and rice is a staple food that is highly prized in many regions, especially Asia. Since rice is a staple food for a large percentage of the world's population, it is seriously threatened by a number of diseases that can negatively impact productivity and quality. To maintain agricultural sustainability and food security, these risks must be controlled. Experts have previously used manual inspection to identify and diagnose rice leaf diseases.

Despite its effectiveness, this method is labour-intensive, time-consuming, and prone to human error. Growing interest in the development of automated systems that identify and diagnose plant diseases has coincided with technological advancements.

eal-time item identification is becoming a crucial part of many industries and is transforming business processes. Because it is quick and advanced compared to other deep learning frameworks, You Only Look Once (YOLO) is used for object detection tasks. This framework's latest version, YOLOv8, is more feature-rich and performs better. This study's main objective is to use YOLOv8 to identify rice leaf diseases. By using a collection of photos of diseased rice leaves to train the model, the system aims to correctly identify a variety of diseases.

Real-time item identification is becoming a crucial part of many industries and is transforming business processes. Because it is quick and advanced compared to other deep learning frameworks, You Only Look Once (YOLO) is used for object detection tasks. This framework's latest version, YOLOv8, is more feature-rich and performs better. This study's main objective is to use YOLOv8 to identify rice leaf diseases. By using a collection of photos of diseased rice leaves to train the model, the system aims to correctly identify a variety of diseases.

Early detection and timely action are advantages for farmers who employ this automated technique, which will ultimately enhance crop management and increase productivity. It opens the door for intelligent farming solutions that can handle the escalating difficulties in agriculture and guarantee a safe and sustainable future food supply. Additionally, real-time detection systems' scalability makes it easier to implement them in a variety of agricultural environments, which benefits farmers everywhere.

## Motivation

This research is driven by the belief that early and accurate disease detection is crucial to putting management plans into place and preserving food production. Traditional disease detection methods are sometimes labour-intensive, slow, and prone to errors since they rely on the manual inspection of human professionals. We can expedite the detection process and reduce the possibility of misdiagnosis by utilising cutting-edge technology like as YOLO, which provides quick and precise object detection. Using the latest advancements in deep learning and computer vision, this study aims to bridge the gap between traditional farming

methods and modern agricultural practices.

The foundation of earlier research and methodology in the field of rice leaf disease identification has been traditional image processing techniques and machine learning models. Despite being groundbreaking at the time, these techniques had several shortcomings. Early machine learning models, such as Support Vector Machines (SVMs) and k-Nearest Neighbours (k-NN), lack the robustness and adaptability needed to handle the intricate patterns and variations present in photos of damaged rice leaves. Furthermore, these methods' slower processing speeds typically need a lot of preprocessing, which reduces their usefulness in real-time applications.

Its speed and accuracy make the YOLO framework a popular choice for object detection jobs. The YOLOv8 series, in particular YOLOv8n (nano) and YOLOv8s (small), are ideal for agricultural applications due to a variety of unique advantages. YOLOv8n's minimal weight and speed make it ideal for deployment on edge devices and mobile platforms with limited processing power.

Because it maintains a high detection accuracy despite its compact size, it is helpful for field real-time applications. On the other hand, YOLOv8s offer a more powerful yet efficient alternative by striking a balance between speed and precision. Both YOLOv8n and YOLOv8s employ state-of-the-art techniques such mosaic data augmentation, anchor-free detection, and upgraded loss functions due to their superior performance over earlier models. Because of this democratisation of technology, even smallholder farmers with little resources can take advantage of cutting-edge crop protection equipment.

## Objective

The primary objective of the research project are as follows:

* + - To collect rice leaf disease dataset.
    - To annotate the rice leaf images.
    - To apply machine learning techniques to detect various common rice leaf diseases.

## Organization of report

The project will be completed within a defined timeframe, with specific milestones for each stage of development. The project report will be organized into the following:

Chapter 1: Introduction and Project Overview.

Chapter 2: Background Theory; Review of existing research on Machine Learning Techniques.

Chapter 3: Methodology; Detailed description of the environment design and implementation.

Chapter 4: Results and Discussion; Evaluation of machine learning technique and it’s various parameters.

Chapter 5: Conclusion and Future Work; Summary of the project's contributions, limitations, and potential future research directions.

Chapter 6: Health, Safety, Risk, And Environment Aspects; Summary of the health, safety, risk management and also main environmental factors involved during the project execution.

# CHAPTER 2 BACKGROUND THEORY

In this section, theoretical foundations and existing research related to the machine learning algorithm, YOLO, CNN. It explores the present level of the art, relevant background theory, and key findings from previous studies, providing comprehensive context for the development of the rice leaf disease classification. In addition, notable developments in the categorization and identification of rice leaf disease will be reviewed, as well as the shortcomings of current methods. There will also be an exploration of options for enhancing existing models in order to improve the precision and accuracy of disease detection.

## Literature Review

Numerous researchers have worked on the automatic diagnosis of rice disease using a variety of techniques. Machine learning approaches, image processing, and feature extractions. A combination of computer vision and deep learning methods were employed to categorize and identify rice plant illnesses.

In a previous study, Phadikar et.al [1] proposed a method for identifying rice diseases in which images of the diseased rice were classified using a neural network-based Self Organizing Map (SOM). Four distinct image types were employed for testing, and the train images were produced by taking characteristics out of the leaf sections that were infected. In another study, Phadikar et.al [2] suggested an automated method based on morphological changes to classify rice plant illnesses, specifically leaf blast (LB) and leaf brown spot (BS) diseases. A total of 1,000 photos taken in a rice field with digital camera. The outcomes showed that the Bayes and SVM classifiers had 79.5% and 68.1% accuracy, respectively.

Yao et.al [3] study, Rice bacterial leaf blight(BLB), rice sheath blight (SB), and rice blast (RB) were the three rice diseases for which an accurate and automatic technique based on image processing and SVM was created in this research. There were 216 photos in the data set, 72 of which were disease spot images for each condition. The three diseases were categorized using SVM, with the radial basis kernel function being chosen. The outcomes demonstrated that SVM could identify and categorize disease areas with a 97.2% detection rate.

Ramesh et.al [4] proposes detection technique using artificial neural networks (ANNs). This technique is used for the detection of plant diseases. Total 300 images in the data set. The suggested method takes images of leaves that are damaged by RB disease and leaves that are healthy. According to testing, RB and healthy leaves yield accuracy results of 90% and 86%, respectively.

Ahmed et.al [5] this study detects three common diseases affecting rice plants BS, BLB and leaf smut. Following the required pre-processing, a variety of machine learning methods, such as KNN, J48 Decision Tree (DT), Naive Bayes (NB), and Logistic Regression, were used to train the dataset. Using a 10-fold cross-validation process, the decision tree algorithm accuracy of 97% on the test dataset. Azim et.al [6], this study also consists same common diseases as [5]. In this study [6], Extreme Gradient Boosting decision tree ensemble used because of its better performance after a few classification techniques are tried. Hue threshold is used to segregate the disease-affected parts of the photos and removes the images' backgrounds. From the impacted areas, distinguishing characteristics from the domains of colour, shape, and texture are retrieved. On dataset used this model achieves 86.58% accuracy.

Bhartiya et.al [7] in this study, first identify characteristics of rice leaf images of in this study. After that, 400 photos were classified using a quadratic SVM classifier, accuracy of 81.8% was achieved. Shrivastava et.al [8], the authors classify diseases using color features. Four color features, totaling 172 features, were collected from each of the 14 color spaces they analyzed. They made use of a 619-image dataset divided into four classes: sheath blight, rice blast, bacterial leaf blight, and healthy leaves. Tested their strategy using seven different classifiers; RF, DT, NB, KNN, SVM, and discriminant classifier (DC). They state that SVM yields the highest accuracy, with an average of 94.65%.

In the study Rowthu et.al [9], the data is pre-processed using a variety of machine learning methods, such as KNN, NB, and DT algorithm J48. Among these, J48 DT Algorithm produced results with 96% accuracy for BS, BLB, LS, and RB. On the dataset, CNN Classifier is used to improve accuracy. By employing this method, the accuracy rate is raised to 97.58%.

Lu Yang et.al [10], CNNs is trained to recognize 10 rice diseases using a dataset of 500 valid photos of damaged and healthy rice leaves and stems taken at resolutions of 5760×3840 pixels from a rice experimental field. The input image's size is configured to be 224 × 224 × 3. With the 10-fold cross-validation method, CNNs-based model attains 95.48% accuracy.

Hossain et.al [11], presents a unique CNN-based model that lowers the network parameters to identify diseases of rice leaves. utilizing 4199 photos of rice leaf disease from a dataset. Training accuracy of 99.78% and validation accuracy of 97.35% are attained by the suggested model. An independent image collection of rice leaf disease is used to evaluate the effectiveness of the proposed model, and it achieves the greatest accuracy of 97.82% and an area under the curve (AUC) of 0.99. In addition, binary classification tests have been conducted, and the results show that our suggested model is able to recognize RB, BS, BLB, Sheath Blight, and Tungro with identification rates of 97%, 96%, 96%, 93%, and 95%, respectively.

Computer-assisted methods for diagnosing rice leaf disease are becoming increasingly common these days Bari et.al [12] proposes the Faster RCNN method which employs RPN architecture that detects object precisely. At 98.09%, 98.85%, and 99.17% accuracy, respectively, it demonstrated that the recommended deep learning based approach was effective in automatically diagnosing BS, hispa, and RB three different diseases that affect rice leaves.

Reconstructed Disease Aware Convolutional Neural Network (RDA-CNN), used in study of

K. Sathya et.al [13] suggested methodology, combines super resolution layers with classification layers. The dataset has 9857 photos representing six distinct illness classes. The classification accuracy of RDA-CNN with VGG19 architecture is 93.257%. A modified VGG19-based deep learning approach is presented in Latif et.al [14], this paper that can accurately detect and classify six rice leaf diseases, including five disease types (BLB, narrow brown spot(NBS), LS, LB, and healthy leaves). The method's accuracy on the non-normalized augmented dataset is 96.08% on average.

The visual image dataset Paddy Doctor, which can be used to identify paddy diseases, is presented in Petchiammal et al. [15] paper. 16,225 annotated photos of paddy leaves from 13 classes (12 illnesses and normal leaf) are included in this dataset. CNN and four transfer learning models VGG16, MobileNet, Xception, and ResNet34—were employed. During training, all models were trained with a batch size of 32, a learning rate of 0.001, and 100 epochs. With an F1-score of 97.50%, the ResNet34 model was found to have the maximum performance. Xception of 96.57%, VGG16 of 93.20%, and MobileNet of 92.39% accuracy achieved. In contrast, the DCNN model's F1-score of 88.81% was the lowest. According to the results, ResNet34 outperformed the Xception based model with an accuracy of 97.5%, while the latter model came in second with 96.58%.

To detect rice leaf disease, the author Kiratiratanapruk et.al [16] suggest using CNN in conjunction with pre-trained models like Mask RCNN, YOLOv3, RetinaNet, and Faster RCNN. Six groups of rice diseases were studied;BLB, NBS, BS, ragged stunt virus disease, and bacterial leaf streak. A dataset including 6330 photos was used. According to their report, YOLOv3 attained the highest average precision, at 79.19%.

Nihar G et.al [17] proposed Convolutional neural networks (CNNs), make up the YOLO algorithm's backend or core architecture. For this application, Tiny-YOLOv3 and R-CNN (regional CNN) are the primary algorithms utilised. There are 500 images in each of the two classes in the dataset False Smut and Leaf Blast. The two primary algorithms utilised in this application are R-CNN (regional CNN) and YOLO, attained a 98.92% accuracy rate.

Naji et.al [18] in this YOLOv5 detection model to assess whether the rice is healthy or not, while the first part uses CNN to extract and classify diseases from the affected area into nine kinds of diseases. YOLOv5s is used to identify disease using the image. The CNN model categorised the test data 97.28% of the time, while the YOLOv5s achieved a 94.60% accuracy. In addition to 13878 images of typical rice leaves, the collection includes nine distinct types of rare rice diseases; BLB, bacterial leaf streak, RB, hispa, BS, dead heart, downy mildew, bacterial panicle, and tungro.

In this study by Sharma et al. [19], the implementation of Tiny YOLO (T-Yolo) V4 as the base detector is proposed. In an effort to increase the network's accuracy, additional convolutional layers, ghost modules, convolutional block attention module (CBAM), and spatial pyramid pooling (SPP) are added. Seventy percent of the dataset 10819 training samples are used in this paper. Twenty percent of the samples are in the validation. Ten percent of the samples are tested. Mean average precision (mAP) of the upgraded UAV Tiny Yolo Rice (UAV T-yolo- Rice) network is 86%.

The YOLOv5network is used by Kumar et al. [20] in their study to recognise objects of different sizes using Three distinct kinds of output feature maps and eight sample output feature maps. A number of metrics are used to evaluate the effectiveness of the network, including mean average precision 50% (mAP50), average recall, and average precision, which are achieved at 82.8, 75.81, and 69.98, respectively. Data set, 583 total photos, train set, 160 validation images, test set, 107 images. In addition DenseNet-Bi-FAPN with YOLOv5 segmentation strategy can fully segment the leaf areas afflicted by rice disease.

Table 2.1: Literature Review papers

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Author(s)** | **Year** | **Data Set (Images)** | **Method** | **Metrics** | **Remarks** | **Type of Disease** |
| Orillo et.al [21] | 2014 | 134  256×256  Pixel | Back Propagatio n Artificial Neural  Network (BP-ANN) | Precision: 0.83,  Recall: 0.81,  mAP@50: 0.76 | 1. Traditional ANN method. 2. Limited to backpropagati on ANN only. | BLB, BS, RB |
| Narmadha et.al. [22] | 2017 | 134  128×128  Pixel | Image Processing, SVM, ANN | Accuracy:94.70% | 1. Focus on measuring disease symptoms. 2. May lack generalization capability. | RB, BS, NBS |
| Prajapat, et.al. [23] | 2017 | 400 | CNN-  based detection | Precision: 0.85,  Recall: 0.84,  mAP@50: 0.80 | 1. Basic CNN approach. 2. Specific to | BLB, BS, LS |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | 640×480  Pixel | and classificati on, SVM |  | classification tasks. |  |
| Lu et.al. [10] | 2017 | 500  356×256  Pixel | Deep Convolutio nal Neural Networks (CNN) | Precision: 0.86,  Recall: 0.84,  mAP@50: 0.81 | 1. Utilizes   deep CNN.   1. Limited to convolutional neural networks. | RB, BS, BLB |
| Rajmohan et.al. [24] | 2018 | 250  224×224  Pixel | CNN, SVM | Accuracy:87.50% | 1. Integrated deep learning and SVM for disease management. 2. Uses both CNN and   SVM, might be complex. | RB, BS, BLB |
| Gayathri Devi et.al. [25] | 2019 | 134  512×512  Pixel | Image Processing, SVM | Accuracy:94.70% | 1. Focused on diseases in a specific region (Thanjavur, Tamil Nadu). 2. Regional focus may limit applicability. | BS, RB NBS |
| Zhou et.al. [26] | 2019 | 3010  512×512  Pixel | FCM-KM  and Faster R-CNN  Fusion | Precision: 0.87,  Recall: 0.85,  mAP@50: 0.79 | 1. Combines clustering with R-CNN. 2. Complexity in FCM-KM   and R-CNN fusion. | RB, BLB |
| Shrivastava et.al. [27] | 2019 | 619  512×512  Pixel | Transfer Learning with Deep Convolutio nal Neural Networks (DCNN) | Precision: 0.89,  Recall: 0.87,  mAP@50: 0.84 | 1. High   precision and recall rates.   1. Focuses on transfer learning. | RB, BLB  , Sheath Blight |
| Chen et.al. [28] | 2020 | 500 | Deep Transfer Learning | Precision: 0.88,  Recall: 0.87,  mAP@50: 0.83 | 1. Utilizes deep transfer learning. 2. Dependence on deep   transfer learning. | RB,  Sheath blight, and BS |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Li et.al. [29] | 2020 | 1000 | Deep Convolutio nal Neural Network (DCNN)  for video detection | Precision: 0.90,  Recall: 0.88,  mAP@50: 0.82 | 1. Designed for video data. 2. Limited to video detection. | Sheath Blight, Stem Borer, and BS |
| Rahman et.al. [30] | 2020 | 1426 | Convolutio nal Neural Networks (CNN) | Precision: 0.88,  Recall: 0.86,  mAP@50: 0.80 | 1. Good   performance across multiple types.   1. Focuses on CNN only. | FS,  Bacterial Leaf Blight, Brown  Spot |
| Ramesh et.al. [31] | 2020 | 1320 | Optimized Deep Neural Network with Jaya Algorithm | Accuracy:95.70% | 1. Optimization technique enhanced the accuracy. 2. Uses   optimized deep neural network. | RB, BS, BLB |
| Sethy et.al. [32] | 2020 | 5932  640×640  Pixel | Support Vector Machine | Accuracy:98.38% | 1. Used deep learning features for classification. 2. Relies on SVM for classification. | BLB,  BS, and Tungro |
| Kiratiratanap ru, et.al [16] | 2020 | 6330  640×640  Pixel | YOLOv3 | Average precision:79.19%. | 1. Provides a comprehensive approach using deep learning for rice disease detection. 2. The method might be limited by the quality of the images and dataset used. | BLB, NBS,  BS, and Bacterial leaf streak |
| Nihar G et.al [17] | 2020 | 500  640×640  Pixel | Tiny- YOLOv3 | Accuracy:98.92% | 1. Utilizes YOLO algorithm, which is  known for real-time object detection. 2.  The model might struggle | False Smut and LB |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | with high variability in rice disease symptoms or overlapping issues in the field images. |  |
| Jiang Z et al. [33] | 2021 | 80 | Multi-task Deep Transfer Learning | Precision: 0.87,  Recall: 0.85,  mAP@50: 0.78 | 1. Multi-task learning approach. 2. Multi-task   learning may be complex | BLB,  BS, and LS |
| Bari et.al [12] | 2021 | 2400 | Faster R- CNN | Precision: Rice blast: 98.09%,  Brown spot: 98.85%, Hispa:  99.17%, | 1.Applied faster R-CNN for real-time diagnosis.  2. Real-time approach could be challenging. | RB, BS,  and Hispa |
| Krishnamoor thy et.al [34] | 2021 | 5200 | CNN,  InceptionR esNetV2 | Accuracy:95.67% | 1. Used transfer learningfor disease prediction. 2. Uses   transfer learning for prediction.. | LB, BLB,  and BS |
| Kathiresan et.al [35] | 2021 | 12496 | Deep Learning Techniques | Accuracy:98.79% | Limited to transfer learning techniques. | Bacterial Leaf Streak, and Rice False Smut |
| Julianto et.al [36] | 2021 | 5932 | CNN,  ResNet50, DenseNet2 01,  MobileNet, EfficientN etB3 | EfficientNetB3 Accuracy:90.14% | 1. Evaluated CNN architecture performance. 2. Evaluation of CNN   architecture only. | LS, BLB,  and BBS |
| Jena et.al [37] | 2021 | 120 | Supervised Machine Learning, SVM | Accuracy:86% | 1. Utilized supervised learning for classification. 2. Supervised learning only. | BS,  Hispa, and LB |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Gogoi et.al. [38] | 2023 | 5000 | 3-Stage CNN  Architectur e with  Transfer Learning | Precision: 0.91,  Recall: 0.89,  mAP@50: 0.85 | 1. Effective three-stage architecture. 2. Complex 3- stage CNN architecture. | Multiple rice diseases including RB and BLB |
| Naji et.al [18] | 2023 | 13878  640×640  Pixel | YOLOv5 | Accuracy:94.60% | 1. Integrates YOLO and CNN for rice disease detection, which could enhance accuracy. 2. The publication lacks detailed performance metrics and comparison with other  methods. | BLB,  Bacterial leaf streak, RB,  Hispa, BS, and Tungro |
| Kumar et al. [20] | 2023 | 850  640×640  Pixel | YOLOv5  with DenseNet- Bi-FAPN | Precision: 75.81%,  Recall: 82.8%, mAP@50: 69.98% | 1. Employs a sophisticated model combining bidirectional feature attention with YOLO v5 for improved detection. 2. The complexity of the model might lead to increased computational requirements and longer  processing times. | BLB, BS  and RB |

Note: Bacterial Leaf Blight (BLB), Brown Spot (BS), Rice Blast (RB), Narrow Brown Spot (NBS), Leaf Scald (LS), False Smut (FS).

## Background Theory:

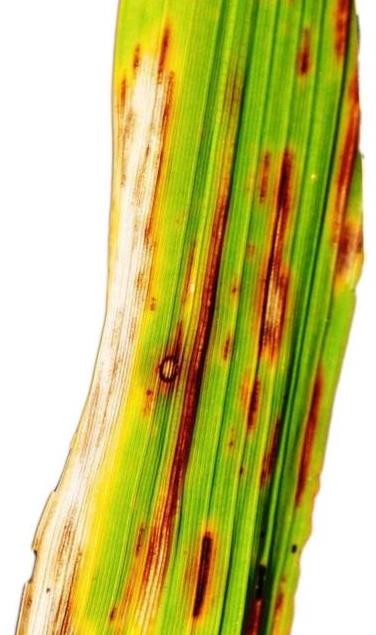
In this section, fundamental concept and theories relevant to rice leaf diseases detection using machine learning technique are explored and YOLOv8 model with a focus on different types rice leaf disease.

## Different types of rice plant disease

Different types of rice plant disease such as bacterial leaf blight (BLB), brown spot (BS), rice blast (RB), leaf scald (LS), narrow brown spot (NBS), and healthy leaf (H) will be taken for detection.



A: Bacterial Leaf Blight B: Rice Blast C: Brown Spot



D: Leaf Scald E: Narrow Brown Spot F: Healthy Figure 2.1 Types of rice leaf diseases

Various types of rice leaf diseases shown in the figure 2.1. Figure A is bacterial leaf blight (BLB) affects leaves of the plant. The symptoms are characterised by elongated lesions that are several inches long on the tip of the leaf. Because of microorganisms, the lesion colour changes from yellow to white [23]. Caused by fungs Xanthomonas oryzae. Rice blast (RB) image shown in figure B is caused by caused by fungus Magnaporthe oryzae. It can affect any part of the rice plant, including the collar, node, neck, leaves, parts of the panicle, and sometimes the leaf sheath. The presence of blast spores is required for the development of blast illness [39].

The brown spot (BS) image in figure C impairs the plant's leaves. The disease's symptoms are shaped like rounds or ovals. The lesion is dark brown to reddish brown in hue [23]. These lesions start off small, round, and can be anywhere from dark brown to purplish-brown in colour [39]. Figure D is leaf scald (LS) image, rice fungus Microdochium oryzae is the source of leaf scald. Elongated, dark brown to reddish-brown lesions with a noticeable yellow halo are how it usually appears, and these lesions are frequently found around the margins of leaves. Large necrotic regions that resemble scalds can result from the aggregation of these lesions.

Narrow brown spot (NBS) image shown in figure E, the fungus Cercospora janseana causes thin brown spots, which appear as linear, brown lesions on leaves that may have a yellowish border. The amount of leaf area accessible for photosynthesis is significantly reduced as a result of these lesions, which are usually little but can occasionally be numerous. In extreme circumstances, the spots may combine, resulting in significant harm and affecting the general well-being and yield of the rice plant. Figure F is healthy rice leaf image, it doesn’t have any kind of diseases.

## YOLOv8 overview

Rice leaf disease detection is a crucial task in precision agriculture, aimed at identifying and mitigating the impact of various diseases on rice crops. One of the most recent iterations of the YOLO family of object identification models, YOLOv8 (You Only Look Once, version 8) is useful for this kind of work because of its accuracy, efficiency, and compact size in real-time object detection. YOLOv8 itself contains different versions of architecture including YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. The lightest model is YOLOv8n, and the heaviest model is YOLOv8x [39].

It's vital to remember that Ultralytics, the developer of YOLOv8, is also the one behind YOLOv5. Because YOLOv8n and YOLOv8s meet our requirements for precision, interference speed, and lightness, they are being examined for the rice plant disease detection problem in this research.

There has been a lot of interest in the One-Stage YOLOv8 algorithm. With only one network needed to predict the categorisation probability and bounding box point, it is extremely fast.

The enhanced performance of YOLOv8n and YOLOv8s architecture is attributed to the introduction of multiple critical components.

An advanced backbone network that effectively extracts significant characteristics from input photos. An anchor-less object proposal-based anchor-free detection head that improves bounding box predictions. A more precise loss function that maximises the model's capacity for item detection and classification.

The YOLOv8n model compared to larger models, it has fewer convolutional layers and parameters, making it extremely quick and efficient. It is most suited for deployment on edge devices and smartphones, which have constrained computational capacity. Despite its compact size, YOLOv8n maintains a respectable level of accuracy, it is usefull for real-time applications in the field.

Another model YOLOv8s provides a balance between efficiency and precision. It offers improved feature extraction and detection capabilities over YOLOv8n because it has more layers and parameters. It is also suitable for applications where both speed and precision are important.

Accuracy and speed are carefully balanced in the design of the YOLOv8n and YOLOv8s versions. Through use of cutting-edge architectural elements including anchor-free detection, FPN, PAN, and CSPNet, these models provide cutting-edge performance in object identification tasks, which makes them ideal for real-time applications in agriculture and other fields. These advanced techniques enable the models to handle the complexity and variability of rice leaf images effectively, providing robust and reliable disease detection. Basic structure of YOLOv8 is shown in the figure 2.2, it shows backbone head and detect part of the YOLOv8 structure.

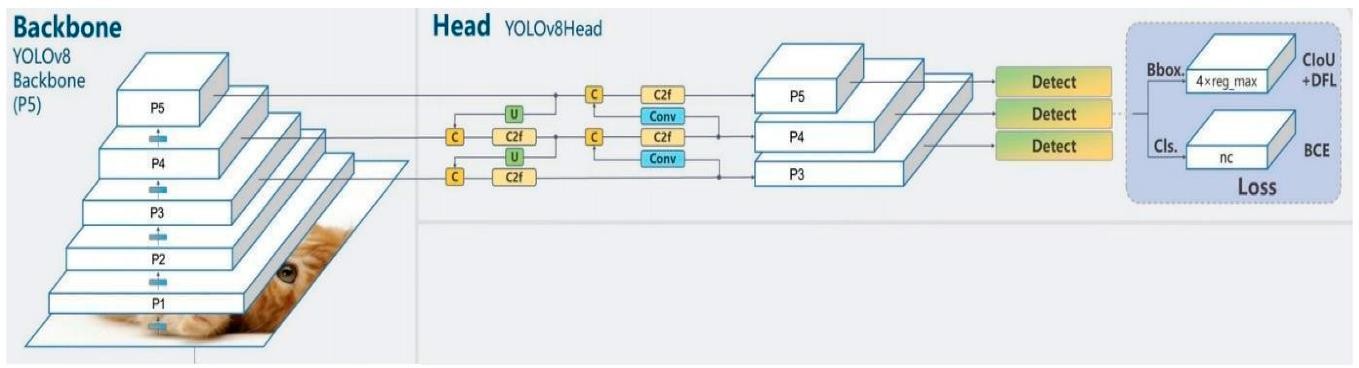


Figure 2.2: Basic Structure of YOLOv8

## Backbone of yolov8

The YOLOv8 architecture's initial step, known as the Backbone, is responsible for removing features from the input picture. Using a sequence of convolutional layers, it processes the image to create feature maps that capture various abstraction levels, ranging from low-level edges to high-level textures and shapes. The Cross Stage Partial (CSP) network is updated and used as the foundation for YOLOv8 models. It is a modified version of CSPDarknet53 architecture, which has convolutional layers, improves information transfer between the network's multiple tiers by using a method known as cross-stage partial connections. The efficiency of CSPDarknet53 networks and its capacity to lower computational costs without sacrificing performance are well known.

## Spatial pyramid pooling fast (SPPF)

The neck portion of YOLOv8 is made up of Spatial Pyramid Pooling Fast (SPPF), an enhanced form of SPP. The SPPF is allows the model to effectively gather and encode information from objects of various sizes, regardless of the object's spatial positioning. The backbone feature maps will be subjected to max-pooling with different kernel sizes for each channel, resulting in three feature maps of different sizes in SPPF. This feature map will then be produced as a fixed-length vector after that.

## Head of YOLOv8

The Head, is a top-level hierarchical structure of the neural network model and the last stage of YOLOv8 architecture, is in charge of producing the last set of predictions. After processing the multi-scale feature maps from the Neck, it yields bounding box coordinates, class probabilities, and objectness scores for every object that is discovered. This approach improves the model's overall accuracy while allowing each branch to focus on its specific task.

In the head part of YOLOv8 architecture it includes three important components such as detection layers, up sample layers and route layers. Input feature maps are transformed into detection bounding boxes by the detection layers. The bounding box coordinates surrounding the observed items are predicted by the bounding box regression component. It defines each bounding box using the width, height, and centre point.

To improve the feature map's resolution, up sample layers are employed. To accomplish up- sampling and transform low-resolution feature maps into high-resolution ones, these layers usually employ deconvolution processes. The main purpose of the up-sampling layer is to increase the model's sensitivity to small objects. Also, different level feature maps are connected via the route layer.

## Advance Architecture: Anchor-Free Detection

Unlike other YOLO models, which employ pre-established anchor boxes, YOLOv8 models take an anchor-free stance. As a result, the computational complexity of the model is decreased. The accuracy of the anchor-free detection method is improved, particularly for small items, as it directly predicts the centre of objects. Anchor-free detection speeds up the time-consuming Non-Maximum Suppression (NMS) step, which filters possible detections following inference, by lowering the number of box predictions. When many anchor boxes are not needed for every grid cell, a significant reduction in memory and processing complexity is needed. Additionally, as the model is less sensitive to the choice of anchor box-specific hyperparameters, it is easier to train and fine-tune.

Overall, YOLOv8 performs better when anchor-free detection is integrated, making it a reliable and adaptable solution for a range of object detection workloads. It emphasises how YOLO models are always evolving, pushing the envelope in real-time object recognition and demonstrating their relevance in a variety of real-world situations.

## Evaluation Metrics

This research on rice leaf disease identification using YOLOv8 models depends heavily on metrics evaluation. To evaluate the effectiveness of the created models and guarantee their applicability in actual agricultural environments, precise and trustworthy measurements are crucial. The model's performance is being measured in this project using a number of important assessment measures, such as precision, recall, average precision (AP), and mean average precision (mAP). Key performance indicators for assessing the completeness and accuracy of the model's detections are precision and recall.

These measurements are essential for comprehending how well the YOLOv8 models identify different rice leaf illnesses. They facilitate the identification of both the model's strong points and places in need of development. In order to help farmers manage crop health and increase agricultural productivity, the project intends to establish a robust and dependable system for identification of rice leaf diseases by a thorough evaluation of these parameters.

## Precision

One of the key criteria used to assess the effectiveness of object identification models, such as YOLOv8, is precision. It gauges how well the model forecasts the good outcomes. To be more precise, precision can be defined as the ratio of true positive detections to the total of false positive and true positive detections, as show in Eq.21. In the context of rice leaf disease

detection, successfully identified infected leaves are considered true positives, whereas false positives are healthy leaves that are mistakenly labelled as diseased.

𝑃𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 =

𝑇𝑟𝑢𝑒 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒

𝑇𝑟𝑢𝑒 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒 + 𝑇𝑟𝑢𝑒 𝑁𝑒𝑔𝑎𝑡𝑖𝑣𝑒

𝐸𝑞. 2.1

To what extent the model can prevent false alarms is revealed by precision. Good precision means the model can accurately identify healthy and damaged leaves, which lowers the chance of needless treatments and their related expenses. This holds significant importance in real- world agricultural applications, since false positives can result in resource waste and higher operating costs.

In the event that the model incorrectly labels healthy leaves as diseased, for example, farmers may apply treatments needlessly, adding to the expense and maybe risking crop damage. In order to minimise false positives and hence lower the costs and number of needless treatments, precision is essential.

## Recall

The model's recall determines how well it identified every relevant instance of a certain class. It is also known as true positive rate or sensitivity at times. Remember that the definition of recall for rice leaf disease detection is the proportion of true positives to all true positives and false negatives detected; diseased leaves that the model missed are considered false negatives, equation to calculate recall is shown in the Eq. 2.2.

𝑅𝑒𝑐𝑎𝑙𝑙 =

𝑇𝑟𝑢𝑒 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒

𝑇𝑟𝑢𝑒 𝑃𝑜𝑠𝑖𝑡𝑖𝑣𝑒 + 𝐹𝑎𝑙𝑠𝑒 𝑁𝑒𝑔𝑎𝑡𝑖𝑣𝑒𝑠

𝐸𝑞. 2.2

An understanding of recall helps to determine how well the model can identify damaged leaves overall. Efficient illness management and prompt action depend on the model to identify the majority of diseased cases, which is evident from its high recall. When a false negative (missing a diseased instance) can have serious repercussions, recall becomes even more crucial. For instance, strong recall makes sure the model doesn't overlook any infected leaves in a high- stakes agricultural context where early disease outbreak detection is crucial, allowing for prompt and efficient response.

## Mean Average Precision (mAP)

Mean Average Precision (mAP) is used to evaluate the effectiveness of an object identification

across many classes. Each class's precision-recall curve is displayed, and mAP is computed by

calculating the area under the curve (AUC). It is the average precision (AP) for every class. After that, the AP values for every class are averaged, and the mAP is determined using the formula in Eq. 2.3.

∑𝑁

𝑃(𝑘)𝑅(𝑘)

𝑚𝐴𝑃 = 𝐾=1

𝐶

𝐸𝑞. 2.3

Where P(k) is the precision, R(k) is the recall, N is the number of IOU thresholds, and k is the IoU threshold. The number of target rice leaf classifications is C.

mAP50, when we talk about mAP50, we are talking about the mean average precision that is determined at an Intersection over Union (IoU) threshold of 0.50. The overlap between the ground truth bounding box and the anticipated bounding box is measured by IoU. When the anticipated and actual bounding boxes overlap by at least 50%, the IoU equals 0.50.

A single value, provided by mAP50, represents the model's general effectiveness in locating and identifying items across all classes. It provides a comprehensive understanding of the model's capabilities by striking a balance between recall and precision. When it comes to rice leaf disease identification, a high mAP50 means that the model can identify diseases in rice leaves, enabling disease monitoring and treatment.

mAP50-95, In comparison to mAP@50, mAP@50-95 is a more tough evaluation criteria. The average precision (AP), which ranges from 0.50 to 0.95 in steps of 0.05, is averaged over a number of IoU criteria. Providing a more in-depth knowledge of the model's localisation capabilities, this statistic assesses the model's performance over a range of IoU thresholds. An elevated mAP@50-95 suggests that the model functions well across a range of IoU thresholds. The capacity of the model to accurately localise diseases at varying degrees of overlap between the predicted and real bounding boxes may be assessed with mAP@50-95 in particular.

## F1- Score

F1-score is an effective metric that provides a model's accuracy by combining recall and precision as shown in Eq.2.4. Because it strikes a balance between recall and precision, it is specially helpful when the dataset exhibits an unbalanced class distribution. Because recall and precision are equally important.

𝐹1 − 𝑆𝑐𝑜𝑟𝑒 = 2 ×

𝑝𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 × 𝑟𝑒𝑐𝑎𝑙𝑙

𝑝𝑟𝑒𝑐𝑖𝑠𝑖𝑜𝑛 + 𝑟𝑒𝑐𝑎𝑙𝑙

𝐸𝑞. 2.4

The F1-score is essential for assessing the model’s efficacy in precisely identifying a variety of diseases in rice leaf disease detection. Reliability in identifying diseased leaves is ensured by a high F1-score, which shows that the model has both high precision and good recall. If the model exhibits high precision but low recall, it indicates that many diseased leaves are missing even though the diseased leaves are accurately detected. In contrast, a model with low precision but high recall will identify the majority of infected leaves while incorrectly classifying a large number of healthy leaves as positives.

## Box Loss

The method used to calculate the difference between the predicted and ground truth bounding boxes is called box loss. For object detection to be effective, objects must be localised accurately. The goal of CIoU or Complete IoU , a loss function designed to improve bounding box regression, is to overcome the drawbacks of previous iterations of IoU loss functions. To do this, YOLOv8 commonly uses the Generalised Intersection over Union (GioU) loss or CioU loss in combination with other losses.

𝐿 − 𝐶𝐼𝑜𝑢 = 1 − 𝐼𝑜𝑈 +

𝑝2(𝑏, 𝑏𝑔𝑡)

𝐶2 +𝖺 𝖯 𝐸𝑞. 2.5

Above equation Eq.2.5 is of CIoU loss, where the width-to-height ratio disparity is calculated using α and v, whereas p represents the Euclidean distance between the center points of b and b(gt).

## Classification Loss (CLS Loss)

The Classification Loss measure is used to assess the accuracy of the estimated class probabilities for each object. YOLOv8 often use binary cross-entropy loss or cross-entropy loss for multi-class classification. Cross-entropy loss quantifies the discrepancy between the true distribution and the expected probability distribution.

𝐶𝑟𝑜𝑠𝑠 𝐸𝑛𝑡𝑟𝑜𝑝𝑦(𝑝, 𝑦) = − ∑ 𝑦𝑖 log(𝑝𝑖) 𝐸𝑞. 2.6

𝑖

In the above equation Eq.2.6, where p of I is the predicted probability for class I, and y of I is the ground truth label.

## Distribution Focus Loss (DFL Loss)

Distribution Focal Loss (DFL) is used to alleviate the problem of class imbalance in object detection procedures. It ensures that the model gives these scenarios more thought during

training by focussing on the most difficult examples. There are several aspects that can affect the target item inside each image, such as blur, occlusion, shadow, or an unclear boundary. Ground-truth labels are therefore not always trustworthy. DFL was created to force the network to concentrate on quickly learning the probabilities associated with values in the surrounding continuous regions of the goal bounding boxes.

𝐷𝐹𝐿(𝑆𝑖, 𝑆𝑖+1) = −((𝑦𝑖+1 − 𝑦) log(𝑆𝑖) + (𝑦 − 𝑦𝑖) log(𝑆𝑖+1)) 𝐸𝑞. 2.7

In the above equation Eq.2.7, where yi+1 and yi−1 represent the two values that are closest to y, or the ground-truth bounding box, respectively. The probabilities for yi+1 and yi−1 can be expressed as Si and Si+1.

# CHAPTER 3 METHODOLOGY

The YOLO approach is an essential part of the methodology used in rice leaf disease categorisation in order to obtain precise and efficient disease diagnosis. This section provides a thorough overview of the approach used, outlining the main steps in the process. The first step in developing a model was collecting and preparing the data. Each of these pictures is labelled after going through the annotation process.

Choosing an appropriate model architecture is the next stage in effectively using YOLOv8 for disease detection. The performance of the trained YOLO model is then evaluated using a variety of measures, including accuracy, the F1-, Precision-, Recall-, and Confusion-matrices. These criteria measure how well the model distinguishes between healthy and unhealthy rice leaves. The application of a systematic approach that results in accurate and efficient disease identification can lead to increased agricultural output and sustainability.

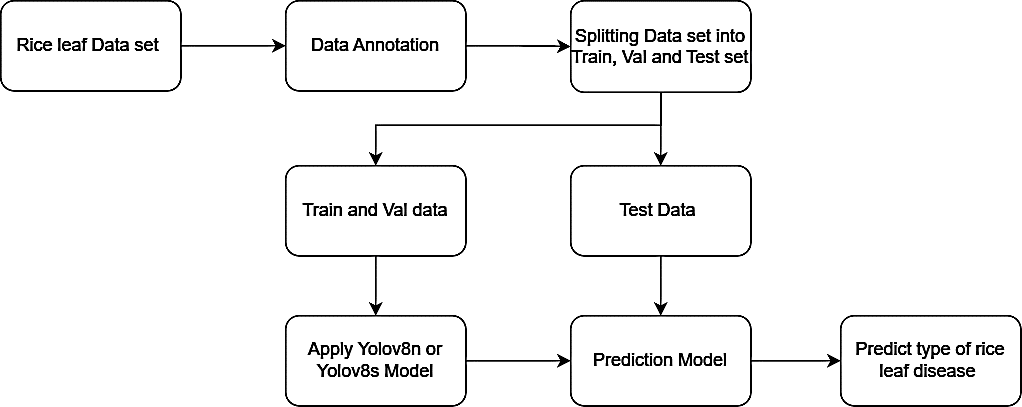


Figure 3.1: Workflow of Yolov8 Model

Figure 3.1 illustrates the YOLOv8 Model's process. Following the collection of the rice leaf data set, the data is annotated. Each data set is then converted into a train, val, and test set. The train and val set is used to train and generate the model used in this study, either YOLOv8n or YOLOv8s. The test data set is then used to evaluate the prediction model that was produced during model training. The anticipated output of rice leaf disease is then shown in a bounding box.

## Data Set

In this investigation, three distinct data sets are being used.Images of both healthy and various diseased rice leaf varieties have been gathered from a variety of sources, such as digital repositories and field surveys, in order to create a diverse dataset.

Table 3.1 Data Set

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Data Set 1 | Data Set 2 | Data Set 3 |
| Brown Spot | 236 | 631 | 776 |
| Bacterial Blight | 370 | 551 | 720 |
| Leaf Scald | - | 564 | - |
| Rice Blast | 246 | 663 | 768 |
| Narrow Brown Spot | - | 619 | - |
| Healthy | - | 705 | 833 |
| Total | 850 | 3733 | 3097 |

Table 3.1 lists the three data sets utilised in this investigation along with the different disease classes. Data set 1 [40] preparation, which includes 850 photos of rice leaf disease in totalTraining, validation, and testing use a total of 583, 160, and 107 images, respectively. It includes three distinct diseases: brown spot, bacterial blight, and rice blast. Each image has a 300 x 300 pixel dimension, an RGB background, and very few have a white background.

PNG is the image file format. Every picture has a label.

Data set data set 2 [41] preparation; there are 3733 photographs of rice leaf disease in this data collection. 247, 489, and 2997 photos are used for testing, validation, and training, respectively. Five distinct diseases—brown spot, bacterial blight, leaf scald, rice blast, and narrow spot—are included in this data set's six classes. It has healthy leaves as well. Each image has a 640 x 640 pixel dimension, and the majority of them have a white background. PNG is the image file format. Preprocessing has already been completed, including identifying the photos and automatically orienting the pixel data.

Data set 3 [42] includes 2754 photos of rice leaf disease in total. 2754, 172, and 171 photos are used for training, validation, and testing, respectively. It includes a healthy view of rice leaves as well as three distinct diseases known as rice blast, bacterial blight, and brown spot. Each image is 640 x 640 pixels in size, with a mostly white and sparsely coloured RGB background. PNG is the image file format.



A. Rice Blast B. Bacterial Leaf Blight C. Brown Spot Figure 3.2: Images from Data set

Data Set [40-42] have different image backgrounds as show in the figure 3.2. First figure A is a bacterial leaf blight disease image from data set 1, it has a random colour background. Another figure B is bacterial leaf blight from the data set 2, which has white background. Last figure C is brown spot disease image from the data set 3, which has different background.

In this project, all six data set must be arranged using the Ultralytics YOLO format, data is organized in your training and validation images and labels as, correct folder structure. Should have train/ and val/ top-level directories, and within each, an images/ and labels/ subdirectory, shown in figure 3.3.

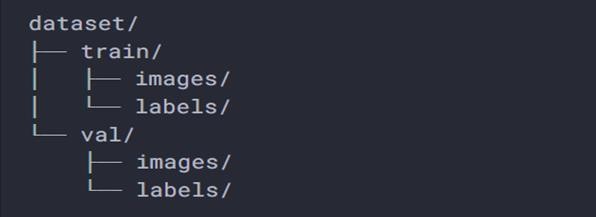


Figure 3.3 Data organizing format

## Data Annotation

In this investigation, three distinct data sets are being used. To produce a diverse dataset, photos of both healthy and diseased rice leaves are collected from a range of sources, including field surveys and digital repositories.

Each image is labelled, the process of labelling the images is called as annotation. So, by categorizing or marking the diseased part with relevant labels, it enables the model to

understand and learn from it. An essential first step in creating artificial intelligence models that are precise and successful is data annotation. Data annotation is done on the data set 1, other two data set were already annotated.

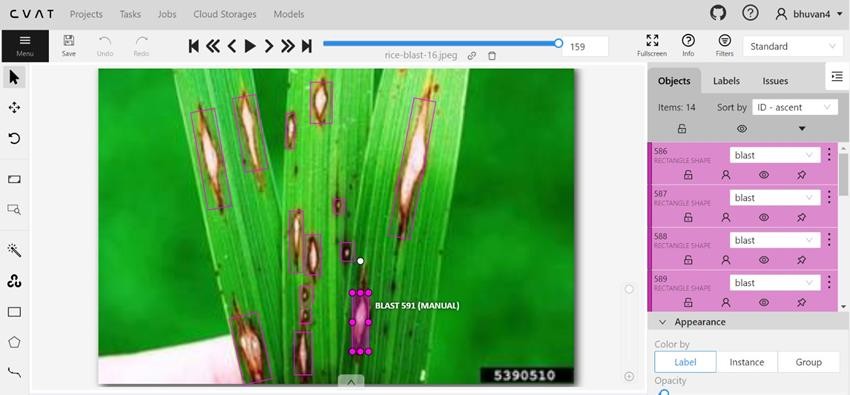


Figure 3.4: Annotation of rice blast disease

The web program cvat.ai, a computer vision annotation tool, is used to annotate the rice blast illness in picture 3.4. The process is completed by labelling the type of disease and drawing a bounding box around every affected area. After every image has been annotated, a label file is produced for each one.

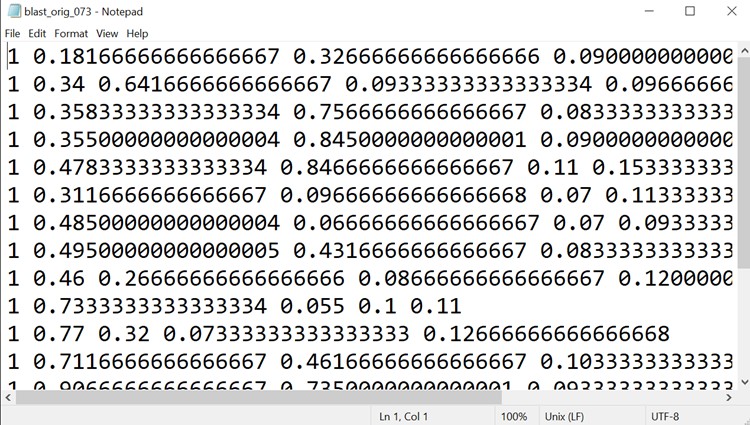


Figure 3.5: Label File

In the figure 3.5, it is a label file of rice blast disease. This label file is in text format and it consists, co-ordinates of the bounding box which is labelled in cvat.ai.

## YOLO Model Architecture

Figure 3.7 displays the YOLOv8 object detection framework architecture, whereas Figure 3.6 displays the components. YOLOv8 is designed for accurate and efficient object recognition with the use of several key modules, such as ConvModule, C2f, SPPF, DFL, Sequential, Bottleneck, Concat, Upsample, and the Detect head. These components work together to process input pictures, extract features, and forecast the locations and classes of objects.

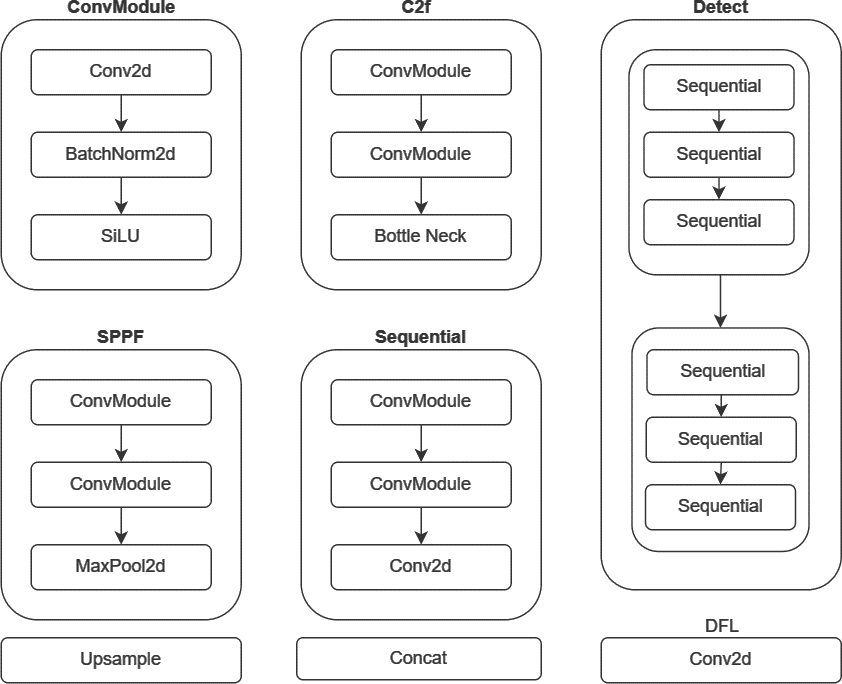


Figure 3.6: YOLOv8 Architecture Modules

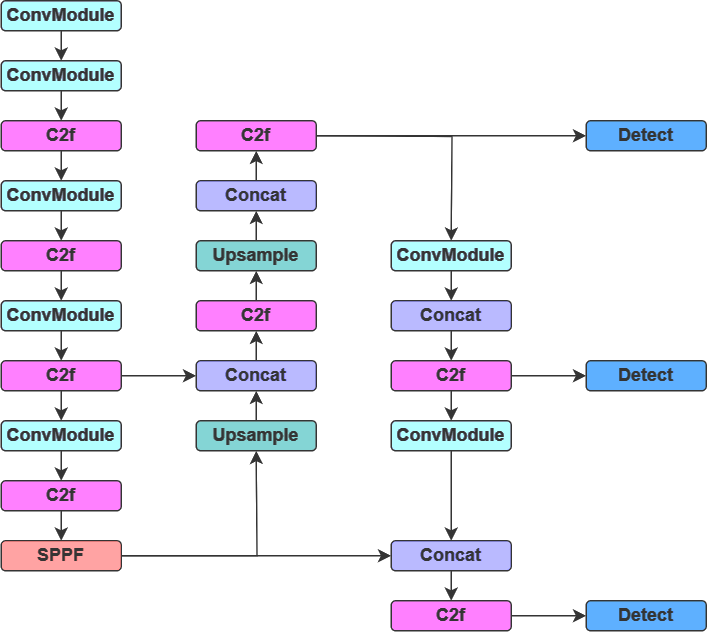


Figure 3.7: YOLOv8 architecture work flow

## ConvModule:

Conv2d, BatchNorm2d, and SiLU are included. The Conv2d module performs the 2D convolution function, which applies a number of filters, sometimes known as kernels, to an input image in order to produce a feature map. A tiny matrix of changeable parameters makes up each filter in the convolutional layer, which is applied to the input image. Its architecture consists of filters, strides, padding, and activation and batch normalisation algorithms. The convolution technique multiplies each filter element by the corresponding element of the image and adds up the results to create a single value in the feature map. Early on in the YOLOv8 process, Conv2d layers are responsible for extracting basic features from the input image. Conv2d modules in intermediate layers capture more complex patterns, such as fragments of objects or textures specific to specific classes. Prior to the detection head predicting the bounding boxes and class probabilities, the Conv2d modules aid in enhancing the features in the last phases of YOLOv8.

BatchNorm2d is used to normalise the output of a convolutional layer across a mini-batch of data. This involves scaling and adjusting the activations to ensure that each layer's output has a consistent distribution during training. Usually, BatchNorm2d is applied prior to the activation function and following the convolution operation. The SiLU (Sigmoid Linear Unit) activation function, also known as the Swish activation function, is a relatively recent addition to the family of activation functions used in deep learning. The way YOLOv8 implements it ensures that the model can capture intricate patterns, maintain effective gradient flow, and perform well across a variety of detection tasks, making it an essential component of contemporary deep learning architectures.

## C2f

The "Cross-Stage Partial with Bottleneck," or C2f, module is a specialized convolutional block that combines the advantages of bottleneck structures and Cross-Stage Partial (CSP) connections. CSP connections, which divide the feature map into two halves and then merge them after processing, are well known for their capacity to improve gradient flow and lower computing costs. In contrast, the bottleneck structure lowers the number of computations and parameters, increasing network efficiency without sacrificing performance.

Several parts make up the C2f module in YOLOv8, and they all function together to extract features at various levels of abstraction. ConvModule and Bottle neck. The C2f module's bottleneck structure is a more compact and effective convolutional block that decreases the input feature map's dimensionality, conducts the required processing, and then expands it back to its original dimensions. This architecture is critical to preserving the network's efficiency and preserving key functionalities.

## SPPF

The SPPF (Spatial Pyramid Pooling Fast) module is an essential component of the YOLOv8 architecture, designed to enhance the model's ability to capture contextual data at different sizes. In order to handle photos of varying sizes without having a fixed input size, spatial pyramid pooling was initially developed. This concept is expanded upon in the SPPF module. To accurately recognise objects of different sizes in a picture, the network must be able to record attributes at several scales, which is made possible by the spatial pyramid technique for object recognition.

SPPF module mainly contains ConvModule andMaxPool2d layer. The SPPF module's core, MaxPool2d Layers, performs several max-pooling operations at various kernel sizes. These max-pooling layers preserve the most noticeable characteristics while shrinking the feature map's spatial dimensions. Pooling is carried out at several scales in the SPPF module.

## Concat and Upscaling

Feature maps can have their spatial resolution increased through the technique of upscaling. Higher resolution feature maps help the model detect smaller objects and record finer features, which is especially helpful in object detection tasks.

In the Neck of the YOLOv8 architecture, where feature maps from many network stages are integrated, the upscaling module is very crucial. Feature maps are frequently upscaled in the Neck to get ready for fusion with higher-resolution data after being down sampled in the Backbone to collect high-level features. This aids in the model's preservation of contextual information, which is necessary for processing the image, as well as spatial detail, which is crucial for identifying small things.

In YOLOv8 networks, feature maps from several layers or phases of the network are combined using the Concat module. This is especially helpful in systems such as YOLO, where the detection of objects of different sizes depends on multi-scale feature aggregation.

One important innovation in YOLOv8 that helps the network perform well on a variety of object detection tasks is the mix of upscaling and concatenation. With the help of these modules, the model may continue to provide accurate detections even in difficult situations, integrate multi-scale data, and preserve high-resolution characteristics.

## Detection Head

The network's last stage is called the Detection Head. It generates the final predictions using the feature maps that have been analysed in the earlier levels. It is in charge of converting the Backbone and Neck's detailed, multi-scale feature maps into accurate, useful forecasts. The YOLOv8 Detection Head module is composed of sequential layers, or simply stacks of processes applied one after the other. These processes involve convolutional layers, activation functions, and often some kind of post-processing to enhance the results.The convolutional layers in the Detection Head further refine the features that have been extracted by the Backbone and Neck. By using convolutional filters on the input feature maps, these layers generate new feature maps that gather important data required for object detection.

The Detection Head estimates the bounding box of each object it finds in the input image. These predictions include the coordinates of the bounding box, the class probabilities, and the objectness score, which indicates the degree of confidence that an object is inside the box.

The inclusion of three detection modules in the YOLOv8 architecture is a strategic choice that enhances the model's ability to recognise objects at different scales. With the use of this multi- scale technique, YOLOv8 is able to precisely detect small, medium, and big objects while maintaining high detection accuracy over a broad range of object sizes and minimising false positives. YOLOv8 strikes a speed-accuracy balance with its use of a pyramidal feature hierarchy and Non-Maximum Suppression step optimisation, which qualifies it for a range of real-time object identification applications.

Table 3.2: YOLOv8 Models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| YOLO Model | Size (Pixel) | Speed(ms) CPU ONNX | Speed(ms) TesnsorRT | Parameters (M) | Flops |
| YOLOv8n | 640 | 80.4 | 0.99 | 3.2 | 8.7 |
| YOLOv8s | 640 | 128.4 | 1.20 | 11.2 | 28.6 |

In the table 3.2 represents YOLOv8 Models, two models YOLOv8n which refers to the nano version of YOLOv8 and YOLOv8s refers to small version of YOLOv8. Input image size of both YOLOv8n and YOLOv8s is of 640 pixels. Speed CPU ONNX provides the model's inference speed in milliseconds when utilizing the CPU with ONXX runtime. Inference speed of the model in milliseconds of 100 TensorRT. Number of parameters is represented in millionsFloating Point Operations Per Second, or FLOPS, is a measure of the model's computational complexity.

In conclusion of both models, YOLOv8n is smaller which have lesser parameters and FLOPS but have faster inference speed on both CPU ONXX and TensorRT. Where YOLOv8s is larger model and is more complex with more parameters and higher FLOPS, gives better results but at the cost of slower inference speeds.

## Implementation of Model

Implementing rice leaf disease detection using YOLOv8 and involves a structured process that includes setting up the environment, data setup, training the model, and performing inference and evaluation.

This project two models YOLOv8n and YOLOv8s is used for detection of rice leaf disease. These models is implemented on each of the three data set. Hence, a total of six different model is being implemented in this study.

## Environment Setup

Installing all of the libraries needed for training and model construction completes the setup. The model is constructed using the necessary Ultralytics libraries for YOLOv8 in Google Colab.Google Colab is mostly used since it provides free access to GPUs and TPUs. TesnorFlow and other deep learning libraries are pre-installed in Colab. The T4 GPU is the runtime type used in this project. The T4 has more than enough power to manage large datasets and intricate models thanks to its 16 GB of GDDR6 RAM. For many compute-intensive jobs, the T4 GPU is a strong and effective option. Since it contains information on the number of classes, the structure of the dataset, and the names of the classes, the data.yaml file is an essential component of configuring the YOLOv8 model. The model will be able to accurately perform training and evaluation and interpret data if this file is set up properly. The data.yaml file is created inthe root directory of the dataset. This is carried out when coding in Google Colab.

## Train Model

The engine/trainer configuration is a comprehensive setup for training a YOLOv8 model on a specific dataset using the YOLOv8s.pt or YOLOv8n model architecture. This configuration controls how the model learns from the data by setting up a number of settings that optimise the training process. The YAML file contains a list of parameters that influence how the model learns from the given dataset. These settings are carefully chosen to optimise the model's training process.

This is a detailed explanation:

Pretrained=True indicates that the model's backbone is has already been pretrained, meaning weights from previous tasks have been initialized in it. In transfer learning contexts, this pretraining usually improves the model's capacity to converge more quickly and perform better on the particular dataset being utilized.

A specific dataset that is given in a.yaml file is referred to by the data parameter. This YAML file should contain the class names as well as the paths to the training and validation datasets. The model is configured to train for 50 epochs, meaning the dataset will be iterated over 50 times during training, in order to lower the loss function and improve the model's performance.

In light of the requirement for the model to update its weights based on a small but sufficiently representative portion of the data, the batch size of 16 was selected in an equitable mannerThe

640x640 pixel resolution is used to make sure the model can effectively process the training images within the GPU's memory constraints. This offers a compromise between maintaining detailed characteristics in the images.

This training configuration makes extensive use of data augmentation techniques. By piecing together four distinct images, mosaic augmentation (mosaic=1.0) is used to create fresh training samples. This increases the model's capacity to generalise to new data by exposing it to a variety of settings and object sizes. In the same way, adding occlusions to the training images using random erasing (erasing=0.4) strengthens the model's resistance to situations in which objects may be partially obscured. The auto augmentation technique (auto augment= randaugment) applies a number of random picture alterations, such as colour changes, scaling, rotation, and more, to the training data to add another layer of unpredictability.

The augmentations specified Blur, MedianBlur, ToGray, and CLAHE are designed to enhance the model's robustness by introducing variability in the training data, With a probability of 0.01 for both the Blur and Median Blur augmentations, there is a 1% chance that every given image will experience these blurring effects throughout training. Blur uses an average of pixel values inside a given window (blur limit set between 3 and 7 pixels) to soften the image and reduce its sharpness. This mimics situations in which pictures might appear marginally out of focus. The ToGray transformation, also applied with a probability of 0.01, converts images to grayscale. Using a clip limit between 1 and 4.0, the CLAHE is applied with a probability of

0.01 to avoid over-amplification of noise while enhancing the visibility of image details. By applying local histogram equalisation within each of the small sections created by the tile grid size of (8, 8), the image's local contrast and edge definition are enhanced.

Because they penalise errors, the loss functions are essential to the training process because they help the model make better predictions. To represent their relative importance in the object detection task, the distribution focal loss (dfl=1.5), classification loss (cls=0.5), and bounding box loss (box=7.5) are all weighted separately. Non-Maximum Suppression (NMS), an important post-processing step, is impacted by the iou=0.7 option, which establishes the Intersection over Union (IoU) threshold for deciding whether overlapping bounding boxes should be merged. This enhances the final set of bounding boxes by removing duplicates and ensuring that the model produces the most accurate set of detections.

The robustness and flexibility of the model training procedure allow it to handle a wide range of training conditions. Because caching is turned off (cache=False), training loads the dataset in real time rather than storing it in memory in its entirety. This helps control RAM usage but may cause the training process to lag a little bit. By ensuring that data loading is managed effectively over several CPU cores, Workers=8 expedites the training process in its entirety.

Ultimately, the whole training process including logs and model checkpoints is stored to the directory that has been designated (save directory=path of file). This configuration makes sure that all outputs are kept intact in case additional research is done or the trained model is used in real-world scenarios. With its ability to handle different augmentations and optimisations, along with its use of advanced features like overlap masking (overlap mask=True), which is especially helpful in situations where objects overlap, this training configuration is both powerful and comprehensive, making it appropriate for a variety of object detection tasks.

Optimizer configuration has been set to 'optimizer=auto', which instructs the system to automatically select the most suitable optimizer and corresponding hyperparameters, such as the learning rate (lr0) and momentum. The optimiser selected is AdamW, a variation of the Adam optimiser that prevents the model from overfitting and helps in regularization by including weight decay directly into the update process. The selected learning rate (lr=0.001) is an important hyperparameter that controls the step size at which the model's weights are updated. By smoothing out these updates with a momentum value of 0.9, the model is able to converge more successfully by accounting for the direction of prior gradients.

Weight and bias parameters are used by the optimiser with varying decay values, 57 weight parameters are set to decay at 0.0 (no decay), 64 weight parameters at 0.0005, and 63 bias parameters at 0.0 as well. This meticulous differentiation aids in optimising the regularisation and efficacy of the model.

# CHAPTER 4 RESULT ANALYSIS

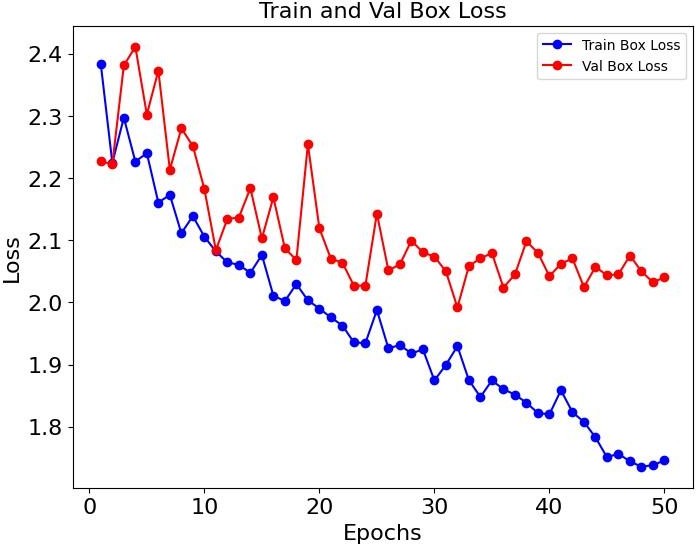
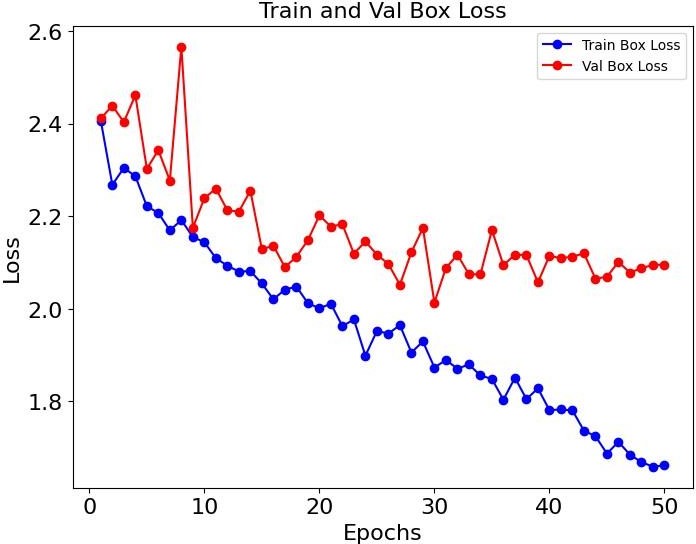
This chapter displays the results of training and evaluating the YOLOv8n and YOLOv8s models. Two models are used to analyse performance on the training and validation data sets in order to identify the diseases on each of the data sets used in this study. Recall, mAP50, F1 score, precision, and other metrics are evaluated. plots the graphs of the precision recall curve, box loss, class loss, F1-score, and confusion matrix. The analysis of the YOLOv8n and YOLOv8s models for the identification of rice leaf diseases is demonstrated.   
  
Using Python 3.10.12 and Google Colab, the suggested YOLOv8n technique is programmed using a Tesla T4, 15102MiB, and configured with two CPUs and 12.7 GB of RAM.

In this project, a total of six model of 3 different data set is being built and trained. In the table

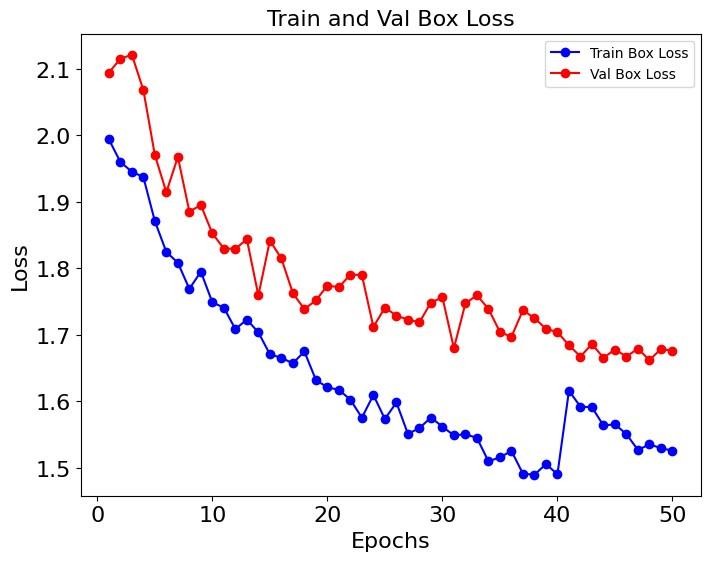
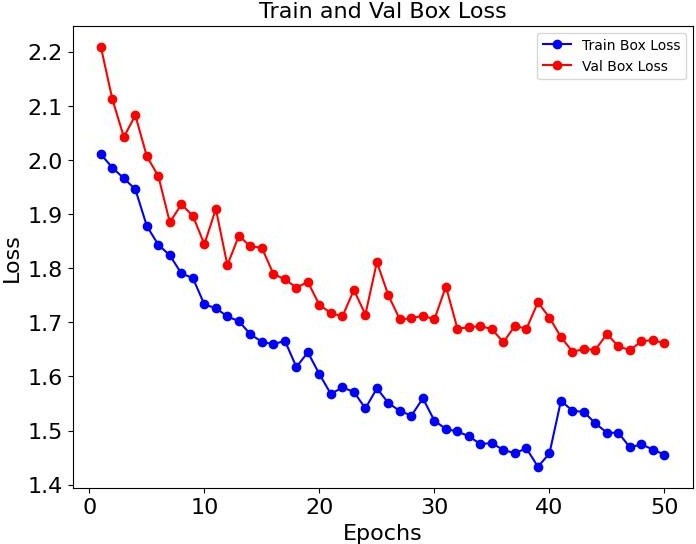
4.1 model description of six different model is being given. In each data set two different model YOLOv8n and YOLOv8s is being represented as Yn and Ys respectively. Data set 1, 2, and 3 is being represented as D1, D2, and D3 respectively.

Table 4.1 Model Description

|  |  |
| --- | --- |
| Model | Description |
| Yn\_D1 | Yolov8n Model of Data Set 1 |
| Ys\_D1 | Yolov8s Model of Data Set 1 |
| Yn\_D2 | Yolov8n Model of Data Set 2 |
| Ys\_D2 | Yolov8s Model of Data Set 2 |
| Yn\_D3 | Yolov8n Model of Data Set 3 |
| Ys\_D3 | Yolov8s Model of Data Set 3 |

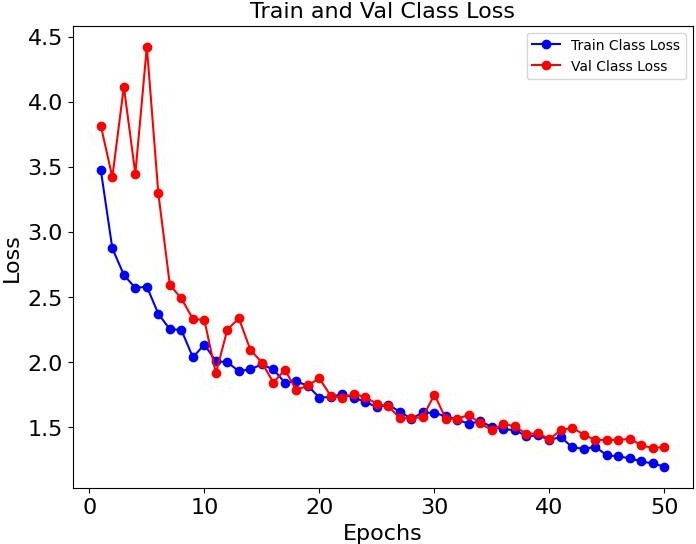
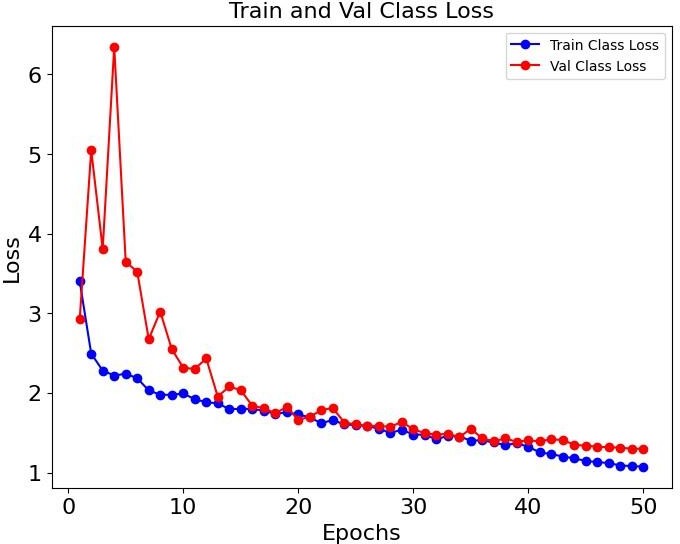
A B



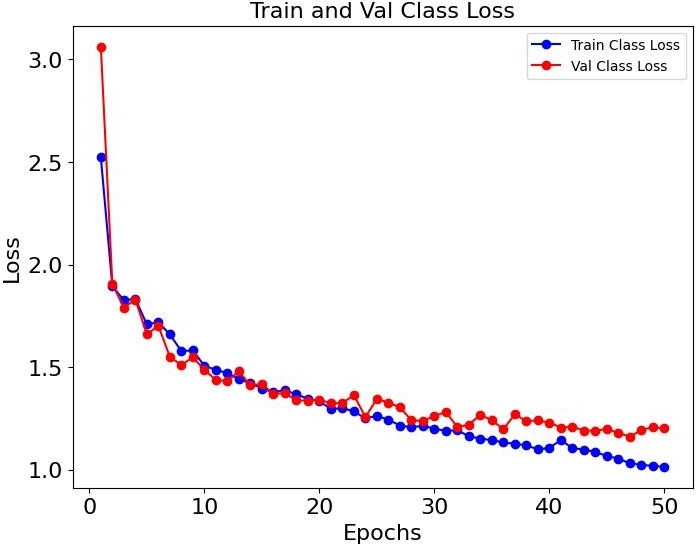
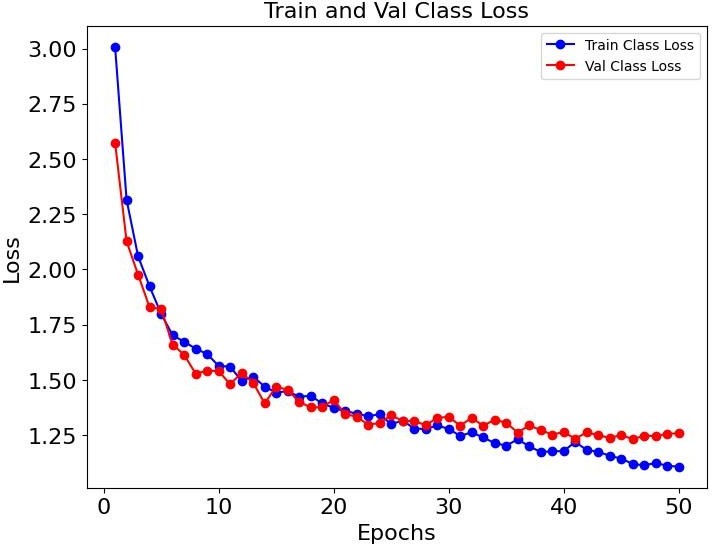
C D

Figure 4.1: Training and Validation Box Loss

Figure 4.1 compares the four models and shows how they perform differently in terms of validation box loss and training box loss. Both the training and validation losses are consistently reduced in the Yn\_D1 model (A) and the Yn\_D3 model (C), suggesting stable learning and improved generalisation. On the other hand, the Yn\_D1 model's validation loss line somewhat higher than the training loss, indicating minor overfitting. The validation loss of the Ys\_D1 model (B) varies and occasionally spikes, suggesting instability and possible problems with the model's generalisation. In the meantime, the Ys\_D3 model (D) shows the best convergence, showing a well-balanced model with reduced overfitting and strong performance, as both training and validation losses decrease smoothly and closely aligning.

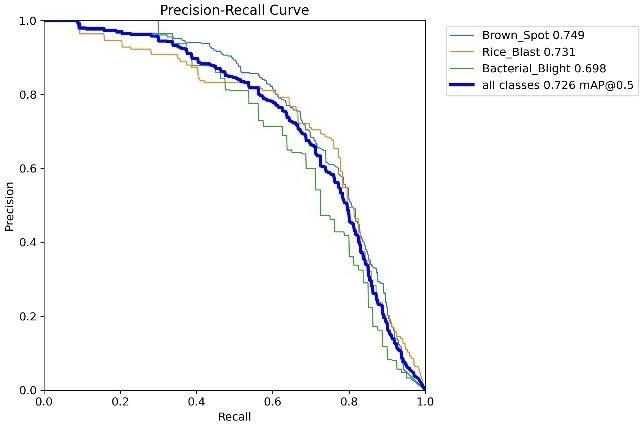
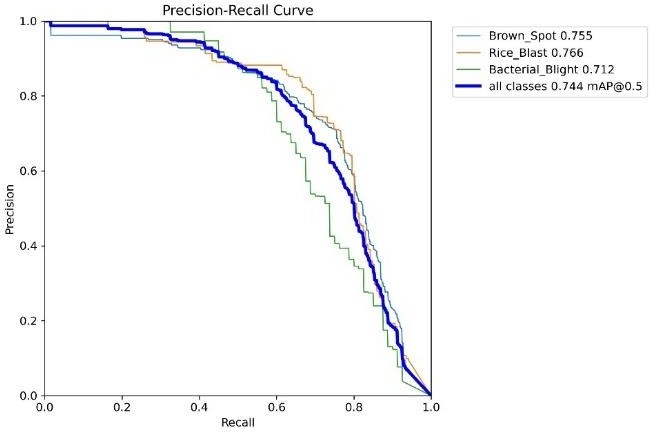
A B



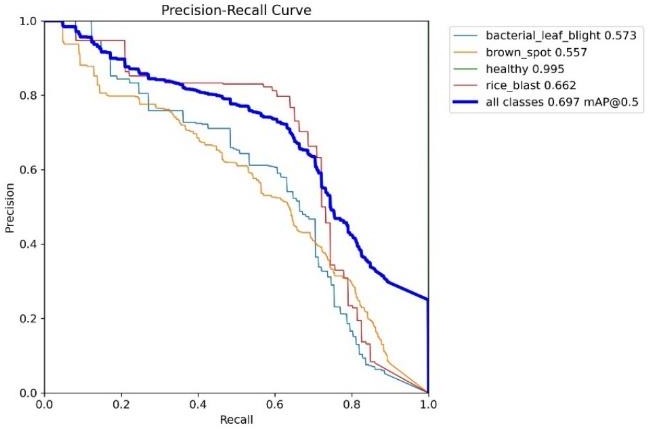
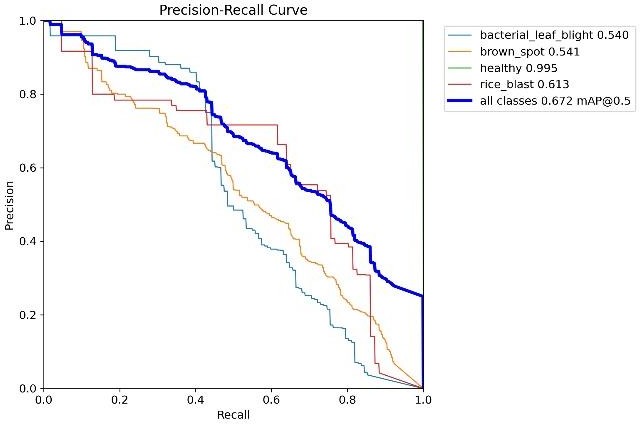
C D

Figure 4.2: Training and Validation Class Loss

In Figure 4.2, which presents the training and validation class loss for models Yn\_D1 (A), Ys\_D1 (B), Yn\_D3 (C), and Ys\_D3 (D), several observations can be made. With very slight variations in validation loss, the Yn\_D1 model (A) exhibits a consistent decrease in training and validation losses, suggesting a modest level of stability. Nonetheless, the large loss at first indicates that there was some initial trouble with the model before it stabilised. Although both losses gradually converge to lower values, the Ys\_D1 model (B) starts with noticeably greater losses, especially in the validation set, which may suggest initial overfitting. Strong generalisation is suggested by the Yn\_D3 model (C), which exhibits steady and continuous loss reduction with close alignment between training and validation. In the same way, early in training, the Ys\_D3 model (D) exhibits well-aligned losses with a sharp decline in both metrics, suggesting efficient learning.

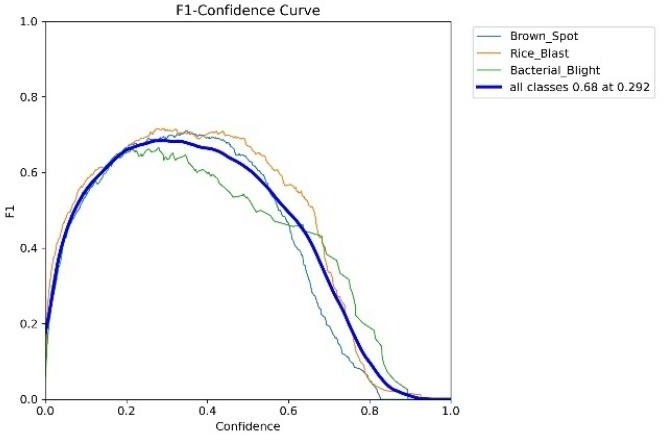
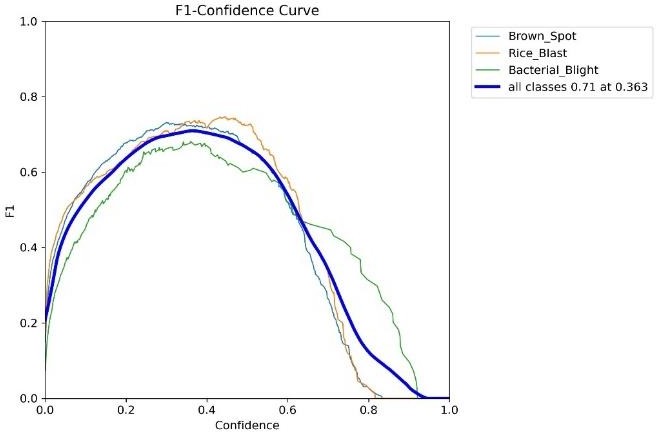
A B



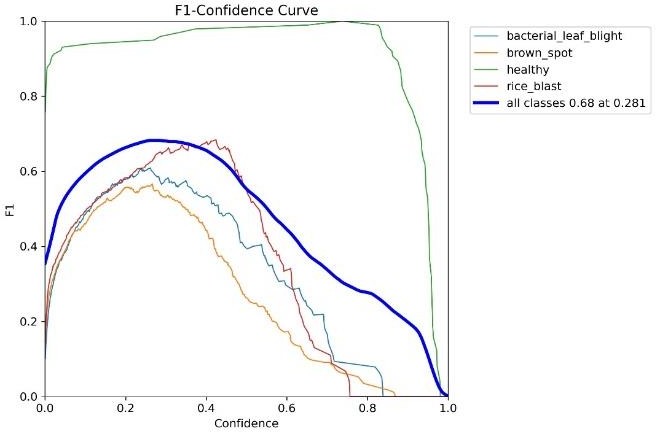
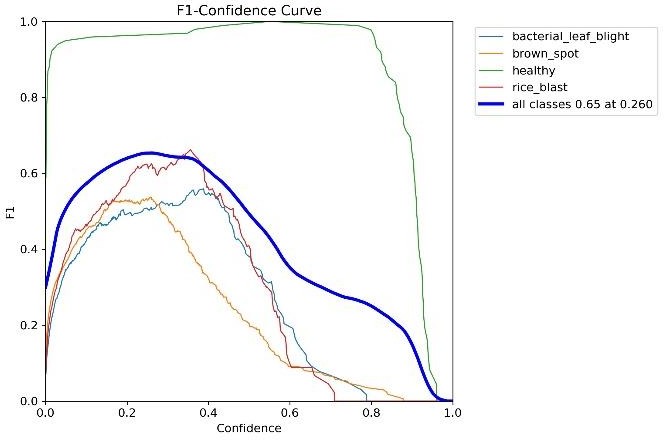
C D

Figure 4.3: PR Curve

The Precision-Recall (PR) curves for the four models Yn\_D1 (A), Ys\_D1 (B), Yn\_D3 (C), and Ys\_D3 (D) are contrasted in Figure 4.3. With mAP@50 values of 0.726 and 0.744, respectively, the Yn\_D1 and Ys\_D1 models perform better, showing greater precision and recall balance across the all classes. By comparison, the Yn\_D3 and Ys\_D3 models perform lower, with respective mAP values of 0.672 and 0.697. Particularly, the disease classes, such as bacterial leaf blight, brown spot, and rice blast, exhibit change, whereas the healthy class consistently achieves high mAP in all models. Overall, the Ys\_D1 model (B) has the best performance, indicating that it can discriminate between various plant health situations, the Ys\_D1 model appears to offer the most reliable classification performance.

A B

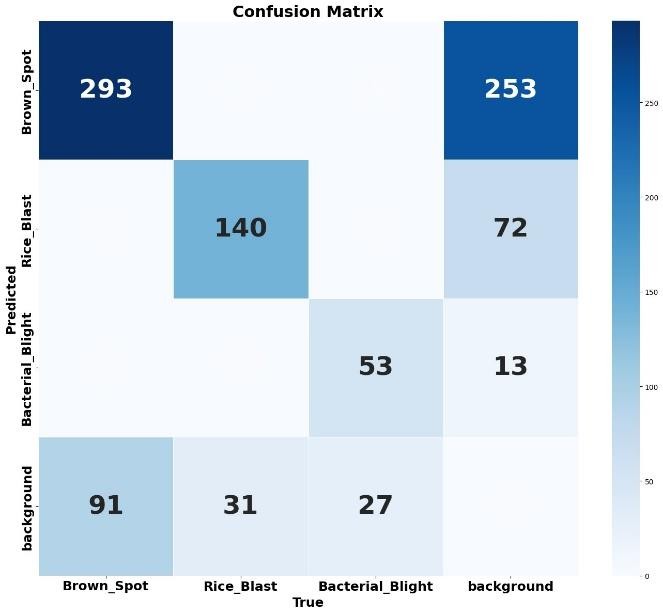
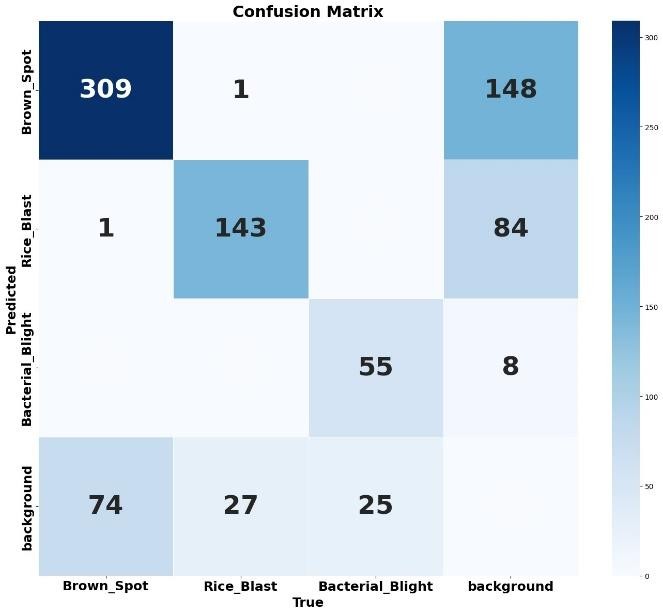


C D

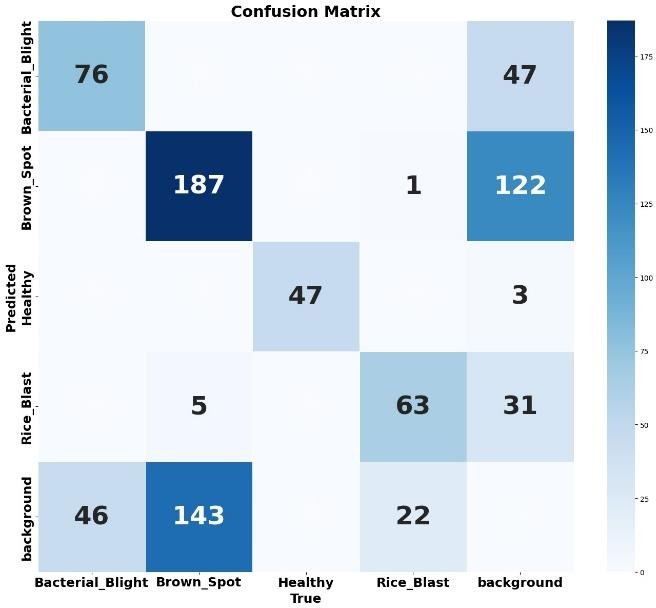
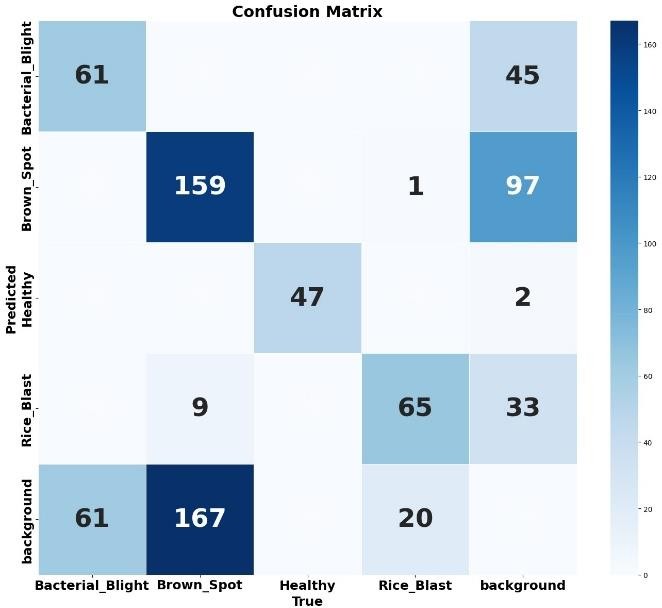
Figure 4.4: F1-Confidence Curve

Figure 4.4 displays the F1-Confidence curves for the four models Yn\_D1 (A), Ys\_D1 (B), Yn\_D3 (C), and Ys\_D3 (D). The Ys\_D1 model (B) has the highest overall F1 score, with an F1 score of 0.717 and a confidence threshold of 0.363, indicating that it achieves the optimal balance between recall and precision across all classes. In contrast, the Yn\_D1 model (A) receives a somewhat lower F1 score of 0.68 with a confidence level of 0.292. For the Yn\_D3 (C) and Ys\_D3 (D) models, lower peak F1 values of 0.65 and 0.68, respectively, suggest less reliable performance.

Interestingly, the healthy class consistently receives high F1 scores in all models, but the sick classes show greater fluctuation, particularly in the Ys\_D3 model. The most reliable model for this classification task is the Ys\_D1 model (B), which outperforms the others in terms of balancing precision and recall across a range of confidence levels.

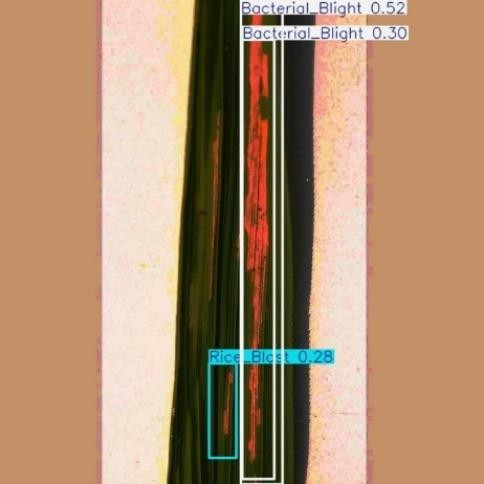
A B



C D

Figure 4.5: Confusion matrix

The confusion matrices in Figure 4.5 illustrate the performance of four different YOLOv8 models on the classification task. Each matrix represents how well the models identify and distinguish between multiple classes, including “Brown Spot”, “Rice Blast”, “Bacterial Blight”, “Healthy” and background elements. The accuracy of the models Yn\_D1 (A) and Ys\_D1 (B) varies, Yn\_D1 does better in some classes but has challenges with "Bacterial Blight." However, Ys\_D1 seems better balanced even though there is still considerable misclassification, especially in the area between "Brown Spot" and background. Similar trends are seen in Yn\_D3 (C) and Ys\_D3 (D), with Ys\_D3 typically obtaining greater precision in all classes, particularly in the distinction between the "Rice Blast" and "Healthy" categories. In general, the Ys\_D3 model (D) appears to provide the best results, classifying data more accurately across all categories.

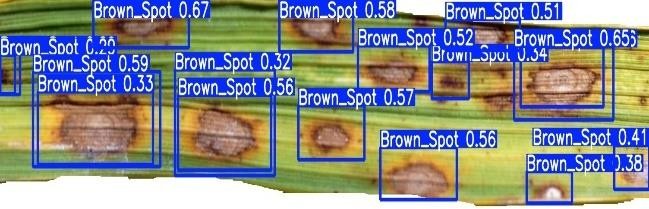
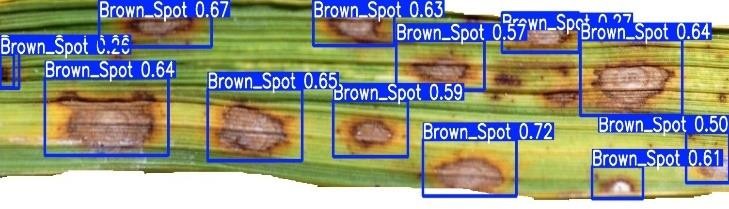
Yn\_D1 Ys\_D1



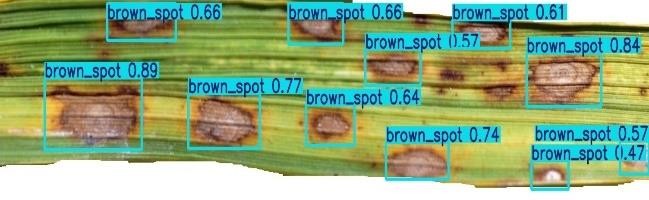
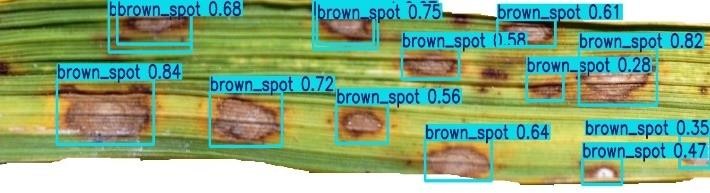
Yn\_D3 Ys\_D3

Figure 4.6: Test Result of Bacterial Leaf Blight Diseases

Bacterial leaf blight image is taken from Data set 3 for testing, result of the test image is shown in figure 4.6. Ys\_D3 have good result on this image across 4 models. Ys\_D3 have detected 7 bacterial leaf blight disease with better confidence.



Yn\_D1 Ys\_D1



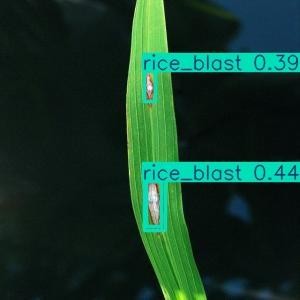
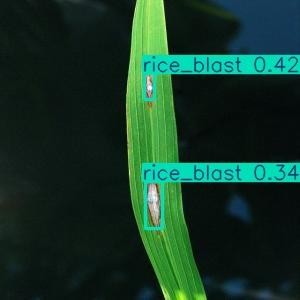
Yn\_D3 Ys\_D3

Figure 4.7: Test Results of Brown Spot Diseases

Brown Spot diseases image is taken from data set 2 for testing, result for the test is show in figure 4.7. Ys\_D1 could detect 17 brown spot diseases in the image with confidence. Even Ys\_D3 have good confidence and could detect 11 brown spot diseases.



Yn\_D1 Ys\_D1



Yn\_D1 Ys\_D1

Figure 4.8: Test Results of Rice Blast Diseases

Rice Blast diseases image is taken from data set 1 for testing, result for the test is show in figure

4.8. Yn\_D1 and Ys\_D1 have performed equally well and predicted 3 rice blast disease with good confidence level. It can be seen in the figure that both models of D3 data set have lesser confidence then D1 models.

Table 4.2 Experimental results for D1 and D3 data set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Epochs | Precision | Recall | mAP50 | F1-Score |
| Yn\_D1 | 50 | 70.2 | 65.8 | 72.5 | 68 |
| **Ys\_D1** | **50** | **76.2** | **66.9** | **74.4** | **71** |
| Yn\_D3 | 50 | 66.8 | 66.2 | 67.2 | 65 |
| Ys\_D3 | 50 | 69.9 | 67.8 | 69.7 | 68 |

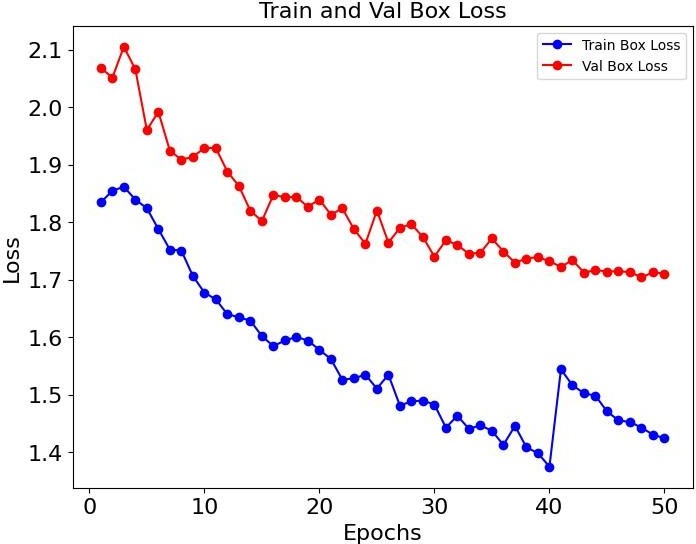
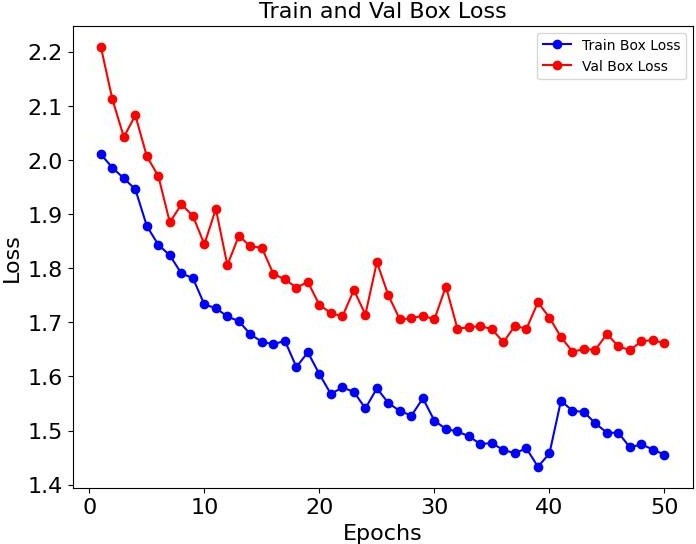
The conducted experiment results on the four models are show in Table 4.2 for two different data set. Number of training epochs of each model is 50. Comparing Yn\_D1 and Ys\_D1 model, Ys\_D1 model have better performance metrics. Ys\_D3 performs slightly better than Yn D3 in terms of recall, mAP50, mAP50-95, and F1-ScoreYs models generally show better precision and balanced metrics compared to Yn models, indicating that it is more refined or better configured versions of the same base model.

Models Yn\_D1, Ys\_D1, Yn\_D3 and Ys\_D3 were trained and evaluated on three types of diseases BLB, RB and BS. To train and evaluate on more types of diseases we have used D2 data set which consists of six types of diseases BLB, RB BS and leaf scald.

Table 4.3 Experimental results for D2 data set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Epochs | Precision | Recall | mAP50 | F1-Score |
| Yn D2 | 50 | 48.9 | 49.8 | 44.7 | 49.3 |
| Ys D2 | 50 | 49.3 | 51.8 | 46.1 | 50.5 |

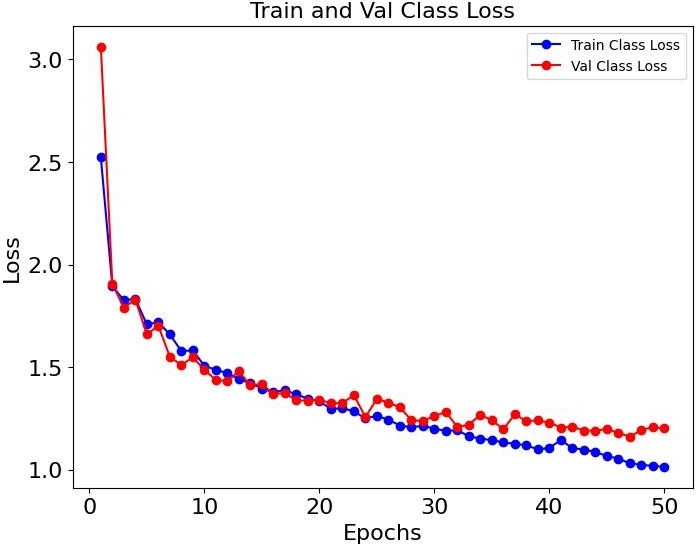
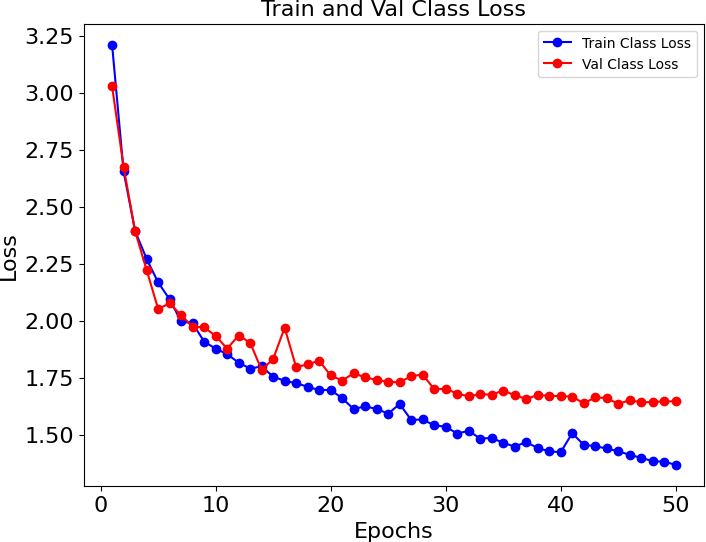
Experimental results for D2 data set show in the table 4.2. D2 data set was trained for 5 different disease and one healthy leaf. The D2 dataset's experimental findings indicates Ys\_D2 model performs better than the Yn\_D2 model in most metrics, such as recall, mAP@50, mAP@50- 95, F1-score, and overall accuracy. Even though the precision of both models is the same, Ys\_D2 performs better at precisely identifying and localising illnesses compared to Yn D2 model.

Yn\_D2 Ys\_D2

Figure 4.9: Training and Validation Box Loss

Both the Yn\_D2 and Ys\_D2 models' training and validation box loss graphs shown in figure 4.9, which exhibit a broad trend of loss decrease across 50 epochs, suggesting that both models are learning efficiently. Stable learning is suggested by the Yn\_D2 model's smoother and more consistent loss decline, especially after the 10th epoch. However, the validation loss of the Ys\_D2 model exhibits a small fluctuation, especially around the 20th epoch, suggesting overfitting or instability in the learning process. Despite this, both models show convergence, with the Ys\_D2 model achieving a slightly lower final validation loss

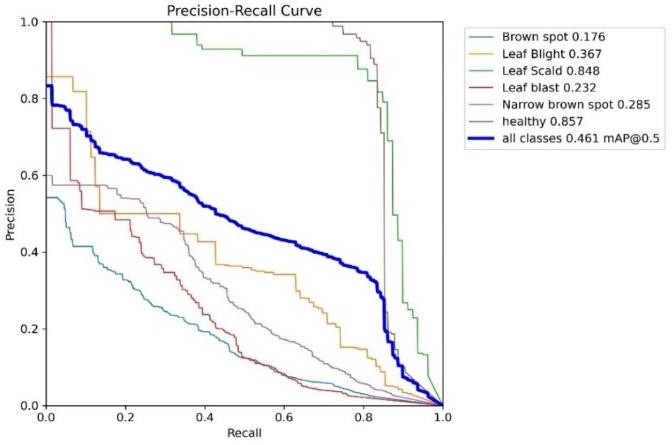
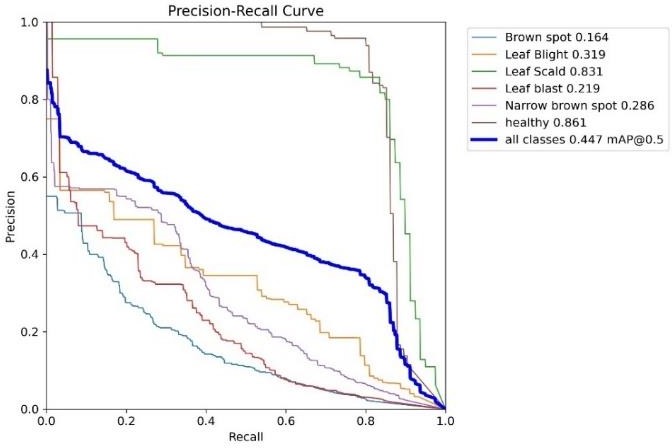


Yn\_D2 Ys\_D2

Figure 4.10: Training and Validation Class Loss

Training and Validation Class loss over 50 epochs is shown in the figure 4.10. The Yn\_D2 model demonstrates a steady decline in training and validation loss, suggesting consistent learning and a successful decrease in classification error. In contrast, the Ys\_D2 model exhibits an earlier initial decline in loss along with more noticeable fluctuations, particularly in

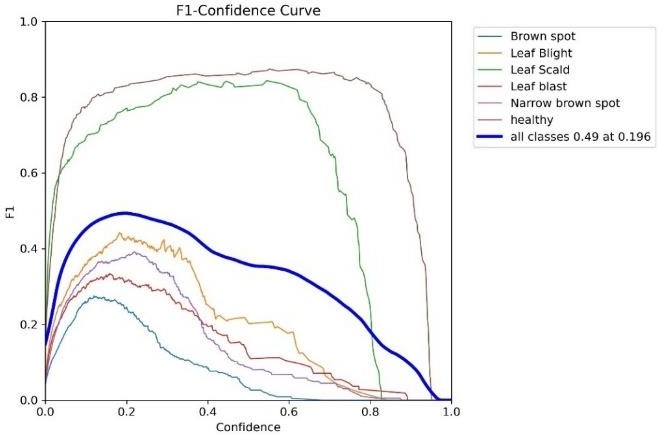
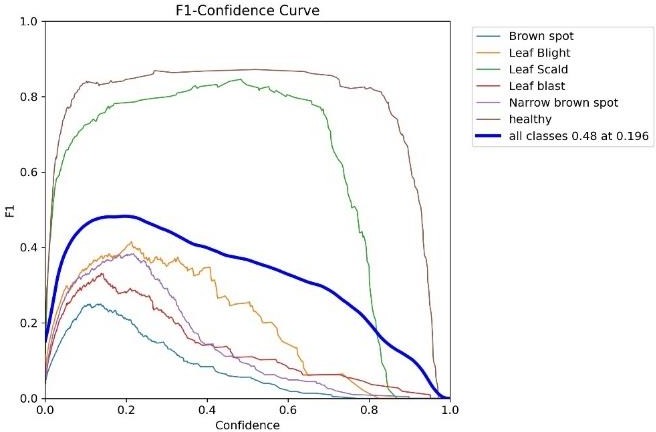
validation loss. This implies that while the Ys\_D2 model may be better capable of learning, overfitting is also more likely to occur.



Yn\_D2 Ys\_D2

Figure 4.11: PR Curve

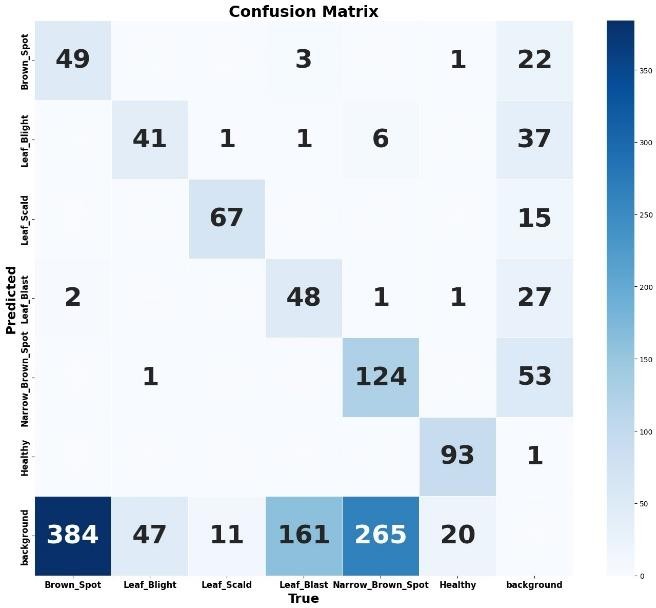
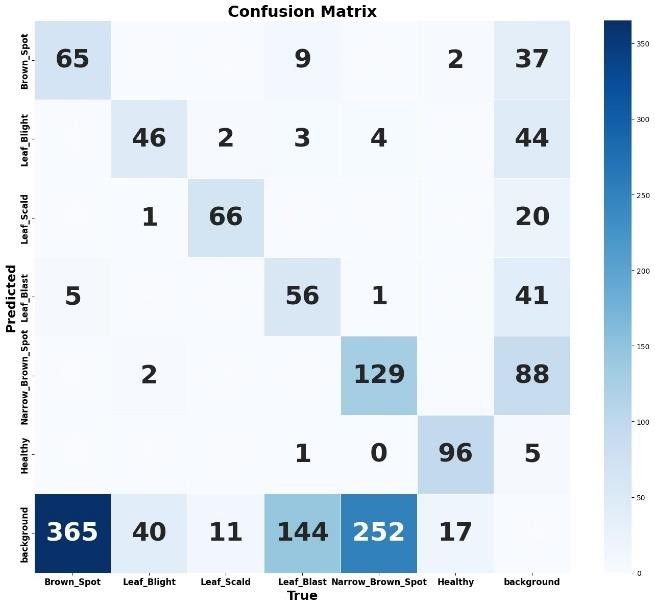
The Precision-Recall (PR) curves for the Yn\_D2 and Ys\_D2 models illustrate their performance across different disease classes in figure 4.11. The Yn\_D2 model shows relatively lower precision and recall values across most classes, With mAP@50 values of 0.447 and 0.461 respectively, the Yn\_D2 and Ys\_D2 models as indicated by the steep drop-off in the curves, particularly for challenging classes like Brown Spot and Leaf Blight. In contrast, the Ys D2 model exhibits better performance, with higher precision and recall values, especially for the Healthy class, which maintains a near-perfect curve.



Yn\_D2 Ys\_D2

Figure 4.12: F1-Confidence Curve

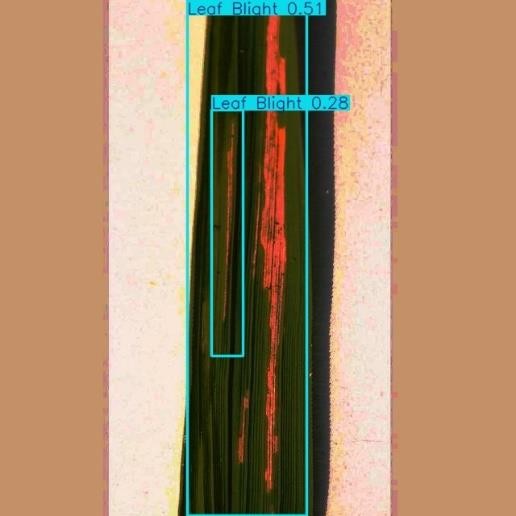
The Yn\_D2 and Ys\_D2 models' F1-Confidence curves shown in figure 4.12, indicates how well the models perform for each class at various confidence criteria. The Yn\_D2 model's overall F1 score peaks at 0.48, while the Ys\_D2 model reaches a slightly higher peak at 0.49, indicating a marginally better balance between precision and recall.

Yn\_D2 Ys\_D2

Figure 4.13: Confusion Matrix

The Yn\_D2 and Ys\_D2 models' confusion matrices show varying trends in classification performance for various rice leaf diseases in figure 4.13. While the Yn\_D2 model performs well in identifying categories such as Leaf Scald and Healthy, it has trouble classifying data from Leaf Blight and Narrow Brown Spot, as seen by increased misclassification rates. With regard to Brown Spot and Leaf Blight, on the other hand, the Ys\_D2 model exhibits superior generalisation and increased accuracy.

A B

C D

E F

4.14: Test Result of D2 Models

Test results of D2 models is shown in the figure 4.14. Bacterial leaf blight image is taken from Data set 3 for testing, result of the test image is shown in figure A and figure B. Brown Spot diseases image is taken from data set 2 for testing, result for the test is show in figure C and figure D. Rice Blast diseases image is taken from data set 1 for testing, result for the test is show in figure E and figure F. Figure A, C and E is test result of Yn\_D2 model, figure B, D and F is test result of model Ys\_D2. It can be observed that Ys\_D2 model have better performance and detecting accuracy than Yn\_D2 model.

YOLOv8n model is used in paper [9], evaluated on the rice leaf data set which consisted of 3175 images and achieved precision of 90%, recall of 84.4% and mAP@50 of 89.9%. Similarly in study [42] have performed YOLOv8s model on mixed-traffic driving environment dataset and achieved precision value 78%, recall value 76%.

Table 4.4. Comparison evaluation of D1 dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Models | Epochs | Precision | Recall | mAP@50 |
| YOLOv5-Bi FAPN | 100 | 75.8 | 82.8 | 69.9 |
| Proposed method (YOLOv8s) | 50 | 76.2 | 66.9 | 74.4 |
| Proposed method (YOLOv8n) | 50 | 70.2 | 65.8 | 72.5 |

As shown in the table 4.4, Ys D1 model YOLOv8s have better average precision of 76.2% and mAP@50 of 74.4%, then the paper [20] which obtained average precision of 75.8% and mAP@50 of 69.98% by performing YOLOv5 model with Bi-FAPN is used for extracting the features proposed on data set 1. Yn\_D1 model is also achieved better mAP@50 of 72.5% than model proposed in study[20], it was trained with 100 epochs whereas our model Yn D1 and Ys D1 was trained for only 50 epochs. In proposed model Ys D1 have slight increment of approximate 1% in the precision and nearly 5% increases in the mean Average Precision(mAP@50) is observed. It is also observed that YOLOv8s model have better overall performance than YOLOv8n model.

The data set D3 for Yn model has achieved precision 66.8%, recall 66.2% and mAP@50 67.2%. For Ys model it has achieved better precision, recall and mAP@50 of 69.9%, 67.8% and 69.7% receptively. Also for the data set D2 where more types of diseases were used to trained, for Yn model it has achieved precision 48.9%, recall 49.8% and mAP@50 44.7%. For Ys model it has achieved better precision, recall and mAP@50 of 51.8%, 46.1% and 50.5% receptively.

As observed from all the models Ys model of all three data set have performed better than the Yn model. Overall performance of Ys\_D1 model was best among all. The best model Ys\_D1 has achieved better results and is trained with 50 Epochs.

# CHAPTER 5

**CONCLUSION AND FUTURE SCOPE OF WORK**

This chapter summarizes the key findings of the project, reflects on the significance of the results obtained, and outlines potential avenues for future research and development.

## Conclusions and Significance of Results

Three distinct data sets were employed in this study for evaluation and training. Three disease categories were included in the D1 dataset: rice blast, brown spot, and bacterial leaf blight. Three disease types—bacterial leaf blight, brown spot, and rice blast—as well as healthy leaves were included in the D2 dataset. In addition to healthy leaves, the D3 dataset included five different disease types: bacterial leaf blight, brown spot, rice blast, narrow brown spot, and leaf scald. Two different models, YOLOv8n and YOLOv8s, were applied to each data set. Six models in all—Yn\_D1, Ys\_D1, Yn\_D2, Ys\_D2, Yn\_D3, and Ys\_D3—were used in this investigation.

Data annotation was done on the D1 dataset, 850 images were annotated in this process. Dataset D2 and D3 were annotated before, no annotation process was done on D2 and D3 dataset. On each data set two various models YOLOv8n and YOLOv8s were implemented. A total of 6 models Yn\_D1, Ys\_D1, Yn\_D2, Ys\_D2, Yn\_D3 and Ys\_D3 were implements in this study. These models were trained on train and val dataset and evaluation metrics were calculated. Train and val box loss and class loss graphs, Precision Recall curve(PR curve), F1-Confidence curve and confusion matrix for all the models are shown.

The best model Ys\_D1 has achieved better results and is trained with 50 Epochs. Ys\_D1 model YOLOv8s have better average precision of 76.2 and mAP@50 of 74.4, then the paper [5] which obtained average precision of 75.81 and mAP@50 of 69.98 by performing YOLOv5 model with Bi-FAPN is used for extracting the features proposed on data set 1. Yn\_D1 model is also achieved better mAP@50 of 72.5 than model proposed in study[20], it was trained with 100 epochs whereas our model Yn\_D1 and Ys\_D1 was trained for only 50 epochs. In proposed model Ys\_D1 have slight increment of approximate 1% in the precision and nearly 5% increases in the mean Average Precision(mAP@50) is observed.

The Yn model's data set D3 has obtained 66.8% precision, 66.2% recall, and 67.2% mAP@50. Better recall, accuracy, and mAP@50 of 69.9%, 67.8%, and 69.7% receptively have been attained for the Ys model. Additionally, for the data set D2, which included a greater variety of disorders in its training, the Yn model scored 48.9% precision, 49.8% recall, and 44.7% mAP@50. Better recall, accuracy, and mAP@50 of 51.8%, 46.1%, and 50.5% responsively have been attained for the Ys model. It is also observed that YOLOv8s model have better overall performance than YOLOv8n model.

YOLOv8n and YOLOv8s showcases remarkable precision and efficiency in identifying and classifying rice leaf diseases. Its real-time detection capabilities empower farmers with timely insights into disease outbreaks, enabling swift decision-making and targeted disease management strategies for enhanced crop protection and productivity.

## Future Scope of Work

1. Enhancement of Model Accuracy: Predicting disease more accurately, improvising recall rate of Yn\_D1 and Ys\_D2 models, so that overall accuracy can be increased. Improvement in precision and recall rate of Yn\_D3 and Ys\_D3 models which may lead to better models. Also Yn\_D3 and Ys\_D3 models needs a lot of improvement due to low F1 score.
2. Expansion of Dataset Diversity: An increasingly diverse dataset containing photos of various disease symptoms that afflict rice plants will be useful in training the algorithm to identify symptoms on different parts of the plant. This could help stop the sickness from spreading and lower yield loss by enabling the model to identify the disease more quickly and precisely.
3. Deployment of Edge Computing Solutions: Using edge computing solutions to infer YOLO models in real-time on farm equipment or portable devices can help with timely diagnostic and treatment decisions. By using YOLO models that are optimised for edge devices and lightweight, farmers with minimal resources might be better prepared to identify diseases.

In conclusion, the future scope of work for rice leaf disease identification utilising the YOLO technique has a lot of space for creativity. Building dependable and scalable solutions for managing illnesses in rice farming can be helped by addressing problems with model accuracy, diversity of datasets, sensor integration, edge computing, decision support systems, transfer learning, and field validation.

# CHAPTER 6

**HEALTH, SAFETY, RISK, AND ENVIRONMENT**

Important considerations of risk, safety, health, and environmental impact are taken into account while using YOLO V8 to detect rice leaf illnesses. YOLO V8 contributes to crop health maintenance and safer food production by facilitating early and accurate disease identification, hence decreasing the need for hazardous pesticides. In addition to preventing soil and water contamination, this reduces health concerns for agricultural workers. Because YOLO V8 offers prompt action, its high precision and recall effectively reduce the danger of crop loss and economic harm. Environmentally, less environmental deterioration results from the decreased need for substantial pesticide use, which encourages sustainable farming techniques. By supporting a safer, more sustainable agricultural ecosystem, YOLO V8 thereby improves the efficacy and efficiency of disease management.

# REFERENCES

* 1. Phadikar, Santanu, and Jaya Sil. "Rice disease identification using pattern recognition techniques." In *2008 11th International Conference on Computer and Information Technology*, pp. 420-423. IEEE, 2008.
  2. Phadikar, Santanu, and Jaya Sil. "Rice disease identification using pattern recognition techniques." In *2008 11th International Conference on Computer and Information Technology*, pp. 420-423. IEEE, 2008.
  3. Yao, Qing, Zexin Guan, Yingfeng Zhou, Jian Tang, Yang Hu, and Baojun Yang. "Application of support vector machine for detecting rice diseases using shape and color texture features." In *2009 international conference on engineering computation*, pp. 79-83. IEEE, 2009.
  4. Ramesh, S., and D. Vydeki. "Rice blast disease detection and classification using machine learning algorithm." In *2018 2nd International Conference on Micro- Electronics and Telecommunication Engineering (ICMETE)*, pp. 255-259. IEEE, 2018.
  5. Ahmed, Kawcher, Tasmia Rahman Shahidi, Syed Md Irfanul Alam, and Sifat Momen. "Rice leaf disease detection using machine learning techniques." In *2019 International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1-5. IEEE, 2019.
  6. Azim, Muhammad Anwarul, Mohammad Khairul Islam, Md Marufur Rahman, and Farah Jahan. "An effective feature extraction method for rice leaf disease classification." *Telkomnika (Telecommunication Computing Electronics and Control)* 19, no. 2 (2021): 463-470.
  7. Bhartiya, Varun Pramod, Rekh Ram Janghel, and Yogesh Kumar Rathore. "Rice leaf disease prediction using machine learning." In *2022 Second International Conference on Power, Control and Computing Technologies (ICPC2T)*, pp. 1-5. IEEE, 2022.
  8. Shrivastava, Vimal K., and Monoj K. Pradhan. "Rice plant disease classification using color features: a machine learning paradigm." *Journal of Plant Pathology* 103, no. 1 (2021): 17-26.
  9. Rowthu, Lakshmana Rao, M. Naveen Kumar, and Ch Narayana Rao. "Early identification of rice plant diseases using machine learning algorithms." *J Inf Comput Sci* (2020): 368-372.
  10. Lu, Yang, Shujuan Yi, Nianyin Zeng, Yurong Liu, and Yong Zhang. "Identification of rice diseases using deep convolutional neural networks." *Neurocomputing* 267, 2017: 378-384.
  11. Hossain, Syed Md Minhaz, Md Monjur Morhsed Tanjil, Mohammed Abser Bin Ali, Mohammad Zihadul Islam, Md Saiful Islam, Sabrina Mobassirin, Iqbal H. Sarker, and SM Riazul Islam. "Rice leaf diseases recognition using convolutional neural networks." In *Advanced Data Mining and Applications: 16th International Conference, ADMA 2020, Foshan, China, November 12–14, 2020, Proceedings 16*, pp. 299-314. Springer International Publishing, 2020.
  12. Bari, Bifta Sama, Md Nahidul Islam, Mamunur Rashid, Md Jahid Hasan, Mohd Azraai Mohd Razman, Rabiu Muazu Musa, Ahmad Fakhri Ab Nasir, and Anwar PP Abdul Majeed. "A real-time approach of diagnosing rice leaf disease using deep learning- based faster R-CNN framework." *PeerJ Computer Science* 7, 2021: e432.
  13. Sathya, K., and M. Rajalakshmi. "RDA-CNN: Enhanced Super Resolution Method for Rice Plant Disease Classification." *Computer Systems Science & Engineering* 42, no. 1, 2022.
  14. Latif, Ghazanfar, Sherif E. Abdelhamid, Roxane Elias Mallouhy, Jaafar Alghazo, and Zafar Abbas Kazimi. "Deep learning utilization in agriculture: Detection of rice plant diseases using an improved CNN model." *Plants* 11, no. 17, 2022: 2230.
  15. Petchiammal, Briskline Kiruba, Murugan, and Pandarasamy Arjunan. "Paddy doctor: A visual image dataset for automated paddy disease classification and benchmarking." In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, pp. 203-207. 2023.
  16. Kiratiratanapruk, Kantip, Pitchayagan Temniranrat, Apichon Kitvimonrat, Wasin Sinthupinyo, and Sujin Patarapuwadol. "Using deep learning techniques to detect rice diseases from images of rice fields." In *International conference on industrial, engineering and other applications of applied intelligent systems*, pp. 225-237. Cham: Springer International Publishing, 2020.
  17. Nihar, G., V. Raghavendra, V. Suresh, and M. Sandhya. "Rice crop disease detection using yolo algorithm." In *National Conference On Advances in Electronics Signal Processing and Communications (AESPC-2020)*, vol. 6, no. 3. 2020.
  18. Naji, DS Alwan MH. “Rice diseases detection and classification using you only look once and convolutional neural network”, International Journal on “Technical and Physical Problems of Engineering”, IJTPE, Iss. 57, Vol. 15, No. 4, 2023.
  19. Arun Kumar Sangaiah, Fan-Nong Yu, Yi-Bing Lin, Wan-Chi Shen, Akashdeep Sharma. “An Enhanced Tiny Yolo Networks for Rice Leaves Diseases Detection in Paddy Agronomy.” IEEE Transactions on Network Science and Engineering. PP. 1-16, 2024.
  20. Kumar, V. Senthil, M. Jaganathan, A. Viswanathan, M. Umamaheswari, and J. J. E. R.

C. Vignesh. "Rice leaf disease detection based on bidirectional feature attention pyramid network with YOLO v5 model." *Environmental Research Communications* 5, no. 6, 2023: 065014.

* 1. Orillo, John William, Jennifer Dela Cruz, Leobelle Agapito, Paul Jensen Satimbre, and Ira Valenzuela. "Identification of diseases in rice plant (oryza sativa) using back propagation Artificial Neural Network." In *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pp. 1-6. IEEE, 2014.
  2. Narmadha, R. P., and G. Arulvadivu. "Detection and measurement of paddy leaf disease symptoms using image processing." In *2017 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-4. IEEE, 2017.
  3. Prajapati, Harshadkumar B., Jitesh P. Shah, and Vipul K. Dabhi. "Detection and classification of rice plant diseases." *Intelligent Decision Technologies* 11, no. 3, 2017: 357-373.
  4. Rajmohan, R., M. Pajany, R. Rajesh, D. Raghu Raman, and U. Prabu. "Smart paddy crop disease identification and management using deep convolution neural network and SVM classifier." *International journal of pure and applied mathematics* 118, no. 15, 2018: 255-264.
  5. Gayathri Devi, T., and P. J. C. C. Neelamegam. "Image processing based rice plant leaves diseases in Thanjavur, Tamilnadu." *Cluster Computing* 22, no. Suppl 6, 2019: 13415-13428.
  6. Zhou, Guoxiong, Wenzhuo Zhang, Aibin Chen, Mingfang He, and Xueshuo Ma. "Rapid detection of rice disease based on FCM-KM and faster R-CNN fusion." *IEEE access* 7, 2019: 143190-143206.
  7. Shrivastava, Vimal K., Monoj K. Pradhan, Sonajharia Minz, and Mahesh P. Thakur. "Rice plant disease classification using transfer learning of deep convolution neural network." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 42, 2019: 631-635.
  8. Chen, Junde, Defu Zhang, Yaser A. Nanehkaran, and Dele Li. "Detection of rice plant diseases based on deep transfer learning." *Journal of the Science of Food and Agriculture* 100, no. 7, 2020: 3246-3256.
  9. Li, Dengshan, Rujing Wang, Chengjun Xie, Liu Liu, Jie Zhang, Rui Li, Fangyuan Wang, Man Zhou, and Wancai Liu. "A recognition method for rice plant diseases and pests video detection based on deep convolutional neural network." *Sensors* 20, no. 3, 2020: 578.
  10. Rahman, Chowdhury R., Preetom S. Arko, Mohammed E. Ali, Mohammad A. Iqbal Khan, Sajid H. Apon, Farzana Nowrin, and Abu Wasif. "Identification and recognition of rice diseases and pests using convolutional neural networks." *Biosystems Engineering* 194, 2020: 112-120.
  11. Ramesh, S., and D. Vydeki. "Recognition and classification of paddy leaf diseases using Optimized Deep Neural network with Jaya algorithm." *Information processing in agriculture* 7, no. 2, 2020: 249-260.
  12. Sethy, Prabira Kumar, Nalini Kanta Barpanda, Amiya Kumar Rath, and Santi Kumari Behera. "Deep feature based rice leaf disease identification using support vector machine." *Computers and Electronics in Agriculture* 175 (2020): 105527.Jiang Z, Dong Z, Jiang W, Yang Y. Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. Computers and Electronics in Agriculture. 2021 Jul 1;186:106184.
  13. Krishnamoorthy, N., LV Narasimha Prasad, CS Pavan Kumar, Bharat Subedi, Haftom Baraki Abraha, and V. E. Sathishkumar. "Rice leaf diseases prediction using deep neural networks with transfer learning." *Environmental Research* 198, 2021: 111275.
  14. Kathiresan, Gugan, M. Anirudh, M. Nagharjun, and R. Karthik. "Disease detection in rice leaves using transfer learning techniques." In *Journal of Physics: Conference Series*, vol. 1911, no. 1, p. 012004. IOP Publishing, 2021.
  15. Julianto, Afis, and Andi Sunyoto. "A performance evaluation of convolutional neural network architecture for classification of rice leaf disease." *IAES International Journal of Artificial Intelligence* 10, no. 4, 2021: 1069.
  16. Jena, Kalyan Kumar, Sourav Kumar Bhoi, Debasis Mohapatra, Chittaranjan Mallick, and Prachi Swain. "Rice disease classification using supervised machine learning approach." In *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, pp. 328-333. IEEE, 2021.
  17. Gogoi, Munmi, Vikash Kumar, Shahin Ara Begum, Neelesh Sharma, and Surya Kant. "Classification and detection of rice diseases using a 3-stage CNN architecture with transfer learning approach." *Agriculture* 13, no. 8, 2023: 1505.
  18. Trinh, Dong Cong, Anh Tuan Mac, Khanh Giap Dang, Huong Thanh Nguyen, Hoc Thai Nguyen, and Thanh Dang Bui. "Alpha-EIOU-YOLOv8: an improved algorithm for rice leaf disease detection." *AgriEngineering* 6, no. 1, 2024: 302-317.
  19. https://doi.org/https://[www.kaggle.com/datasets/nischallal/rice-disease-dataset.](http://www.kaggle.com/datasets/nischallal/rice-disease-dataset)
  20. Roboflow Universe, [https://universe.roboflow.com/project-khcjh/rice-leaf-disease-](https://universe.roboflow.com/project-khcjh/rice-leaf-disease-detection-obj) [detection-obj.](https://universe.roboflow.com/project-khcjh/rice-leaf-disease-detection-obj)
  21. Roboflow Universe, [https://universe.roboflow.com/rice-disease-detection-olj7i/rice-](https://universe.roboflow.com/rice-disease-detection-olj7i/rice-disease-detection-qiyjo) [disease-detection-qiyjo](https://universe.roboflow.com/rice-disease-detection-olj7i/rice-disease-detection-qiyjo).
  22. Afdhal, Afdhal, Khairun Saddami, Sugiarto Sugiarto, Zahrul Fuadi, and Nasaruddin Nasaruddin. "Real-time object detection performance of yolov8 models for self-driving cars in a mixed traffic environment." In *2023 2nd International conference on computer system, information technology, and electrical engineering (COSITE)*, pp. 260-265. IEEE, 2023.