Deep Learning Based Instance Segmentation of Leaf diseases

By

Shoaib Hossain

ID: 202-35-3112

Supervised by

Mr. Nuruzzaman Faruqui

Assistant Professor

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

A thesis submitted in partial fulfillment of the requirement for the degree of B.Sc. in Software Engineering

Department of Software Engineering

DAFFODIL INTERNATIONAL UNIVERSITY

SPRING-2024

©Daffodil International University

APPROVAL

APPROVAL

This thesis titled on "**Deep Learning-Based Instance Segmentation for Leaf Disease Detection**", submitted by **Shoaib Hossain Khandakar (ID: 202-35-3112)** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Dr. Engr. AKM Masum
Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Chairman

Dr. Marzia Ahmed
Assistant Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 1

Md. Rajib Mia
Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

Internal Examiner 2

Dr. Md. Manowarul Islam
Associate Professor
Dept. of Computer Science and Engineering
Jagannath University

External Examiner

©Daffodil International University

THEIS DECLARATION

THESIS DECLARATION

I hereby declare that, this thesis titled **"Deep Learning Based Instance Segmentation of Leaf Diseases"** is done by me under the supervision of Prof. Mr. Nuruzzaman Faruqui, Assistant Professor, Department Of Software Engineering, Daffodil International University, towards the partial fulfillments of my work. I also state that this project has not been submitted anywhere in the partial fulfillments for any degree of this or any other University. I am also declaring that neither this thesis nor any part therefore has been submitted else here for the award of Bachelor or any degree.

Shoaib Hossain

Submitted by

Shoaib Hossain

ID: 202-35-3112

Department of Software Engineering

Daffodil International University

16-07-21

Certified by,

Mr. Nuruzzaman Faruqui

Assistant Professor

Department of Software Engineering

Daffodil International University

©Daffodil International University

©Daffodil International University

ACKNOWLEDGEMENT

All praise belongs to Almighty Allah. I'm grateful to Allah who has given me the ability to complete this thesis. I am thankful to my mother. Without her support I would not be able to come this far.

I'd like to show my gratitude to my supervisor **Mr. Nuruzzaman Faruqui**, Assistant Professor, Department of Software Engineering, Daffodil International University. He helped throughout the thesis. His knowledge and experience of deep learning helped me achieve my objectives in this thesis.

Finally, I'd like to thank **Dr. Imran Mahmud** Head of the Software Engineering Department at DIU, and all the other professors in the department for their advice, help and kind words.

Shoaib Hossain

©Daffodil International University

TABLE OF CONTENTS

©Daffodil International University

©Daffodil International University

LIST OF FIGURES

©Daffodil International University

ABSTRACT

Crop disease is a significant issue for Bangladesh's economy, but it can be prevented with early detection. This thesis proposes a deep learning-based instance segmentation technique for detecting 10 of the most common crop diseases in Bangladesh. This technique can enable automated crop disease detection on a large scale. The paper introduces a new annotated dataset of around 4600 images for 10 different disease classes. Image annotation is usually the most time-consuming phase for any segmentation task. To reduce this time, the paper proposes a new semi-automated annotation pipeline. It showcases how this pipeline can reduce image annotation time by approximately 85%. After annotating all the images, three different models were trained - two variants of YOLOv8 (YOLOv8, YOLOv8l) and YOLOv9c. These models were trained on three different versions of the dataset: the first version with only manually annotated images, the second version with a combination of manually and semi-automated annotated images, and the third version with an augmented combined dataset. The augmented version did not perform well, but when increasing the dataset using semi-automated annotation, the mAP score increased. YOLOv8l achieved the best mAP score of 0.7417.

Keywords: Instance Segmentation, deep learning, crop disease, semi automated annotation, YOLOv8, YOLOv9

©Daffodil International University

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Bangladesh economy largely depends on agriculture and among popular crops, rice, corn and potato forms the main carrier of the sector. But crop declines caused by diseases result in millions of dollars in economic losses to farmers each year. Their early and accurate detection is important for implementation of effective control measures to limit damage.

Here, the visual inspections by agricultural experts are the basic method of disease detection in traditional ways. It is time-consuming, subjective, error-prone (especially for the detection of diseases at early stages). Furthermore, the extensive diversity in leaf disease symptoms poses an additional difficulty in manual diagnosis.

Artificial intelligence, a subfield in deep learning has become a powerful technique in automatically identifying plant diseases. For CNNs, our methods identify to detect the disease symptoms from images due to their capability to learn intricate patterns from image data. In our case, the deep learning task instance segmentation enabled the exact localization and segmentation of particular disease regions on leaves.

This thesis investigates the application of deep learning for instance segmentation of leaf diseases commonly affecting crops in Bangladesh. For this task I train YOLOv8 and YOLOv9, which are object detection models with segmentation, push to the state-of-the-art. In this paper, a new semi-automated annotation pipeline is proposed to assist with the challenge of manual image annotation, a necessary but tedious operation in deep learning. The pipeline hopes to speed up annotation dramatically while retaining high quality.

The rest of this thesis is organized as follows. Chapter two gives an even closer look at the literature review of deep learning in plant disease detection, compared with instance segmentation techniques. This refers to Chapter 3, which outlines the methodology detailing the deep learning models employed, the semi-automated annotation pipeline implemented as well as the specifications of the image dataset used for training and evaluation. Experimental results of

©Daffodil International University

training and evaluation in chapter 4 is to compare the performances of YOLOv8 and YOLOv9 of instance segmentation of leaf disease. Chapter 5 discusses the fulfilled objective from the thesis. Chapter 6 ends the thesis by discussing the limitations of the proposed approach and offering the possible future directions of the research, recapping the major discoveries and the powerful impact of this work on agriculture practices in Bangladesh and the Semi automated annotation pipeline in terms of reduction of overall annotation time for human annotator.

1.2 Background

Rice, potato and corn are the three most important crops in the Bangladesh, and their leaves are constantly threatened by a variety of diseases. These diseases are highly detrimental to the productivity levels of the crops and also the profit that farmers can make from their farming endeavors. However, traditional approaches to disease detection are slow and subject to error, making it difficult to intervene. This thesis introduces a new method for instance segmentation with deep learning. This novel technique has demonstrated impressive performance in many segmentation challenges, which could solve an efficient disease detection problem autonomously in the crops of Bangladesh. But generating the large datasets required for deep learning is a challenge. In this work, I present a new semi-automated annotation pipeline, built specifically for Bangladeshi crops. The goal of this pipeline is to greatly accelerate the annotation process without reducing the accuracy so that I can efficiently train robust models and ultimately give to farmers a faster and more accurate tool to protect their crops.

1.3 Problem Statement

It is a common knowledge that leaf diseases could pose a major impediment to agricultural productivity particularly in rural communities where there is little availability to expert knowledge and resources. These diseases are mostly diagnosed manually, a process which is both labour-intensive and technically demanding to ensure an accurate diagnosis which usually represents high cost especially for small scale farmers. However, the current literature shows that most of the previously reported researches has adopted traditional segmentation techniques in leaf disease segmentation or completely ignores the segmentation step in the leaf disease segmentation due to

©Daffodil International University

the scarcity of leveraging the powerful deep learning. While these standard techniques may have been satisfactory up to a certain point, they almost always do not offer the precision and applicability needed and were never intended for truly scalable precision agriculture.

1.4 Research Question

Q1. How well does the data from commonly available publicly annotated datasets for leaf diseases cater to the problems associated with Bangladeshi crops?

Q2. Can manual leaf diseases data annotation process be drastically speeded up for Bangladeshi crops?

Q3. Can deep learning based instance segmentation be employed to automate the detection and segmentation of leaf diseases in main Bangladeshi crops such as rice, potato, and corn process efficiently or not?

1.5 Research Gap

- The most of the research in the field is only to one single crop type thus mostly a shortage of data variety
- Data is scarce for the task of leaf disease instance segmentation. Although there are plenty of images of different leaf diseases, there are hardly any datasets which have the corresponding annotated masks.
- Most solutions tends to avoid deep learning-based segmentation techniques.

1.6 Research Objective

This This study focuses on the creation of an accurate deep learning model with an instance segmentation to automate the detection and segmentation of leaf diseases in the most important crops in Bangladesh - rice, potato, and corn.

A unique semi-automated annotation pipeline will be built to solve the biggest problem faced by these Bangladeshi crops - less number of large and well-annotated datasets.

©Daffodil International University

1.7 Motivation of The Study

While In Bangladesh, the huge economic losses due to crop diseases motivates this thesis to conduct a research on automated disease recognition using deep learning. Immediate detection should save the harvest, especially on rice, potato, and corn plants attracting the largest rate of the above-mentioned diseases. Recent advanced in deep learning, particularly in instance segmentation show promise but the real task is creating large, well-annotated datasets, a very labourious and time consuming process. To address this bottleneck, I propose a new semi-automated annotation pipeline that I devise especially for Bangledeshi crops. This combined approach aims to empower farmers with a faster and more accurate tool for safeguarding their crops.

1.8 Research Scope

- Use the technology to identify crop diseases at an early stage and reduce the losses as much as possible.
- Speed up data annotation with a semi-automated annotation pipeline.

Automating crop disease detection erases the need for manually, repetitive, and potentially error-prone traditional practices.This technology offers a more efficient approach to identifying crop diseases, enabling immediate preventive measures. Furthermore, a semi-automatic annotation pipeline is proposed with a purpose to solve the main disadvantage in manual data tagging, too much time used for the whole annotation process can be significantly reduced for annotators.

1.9 Summary

In This research encircle the question of automated identification of disease in crops of Bangladesh (rice, potato, corns) using deep learning with instance segmentation. To cope with the dilemma of data annotation delay, I introduce a new pipeline of semi-automated annotation pipelines to help reduce the overall annotation time needed for a human annotator. The next section of literature review will discuss the previous literature related this thesis.

©Daffodil International University

CHAPTER 2
LETARETURE REVIEW

2.1 Introduction

This section will present a brief discussion about the previous works on disease segmentation for leaves. In total, I have studied about 26 papers related to my work. On this base, I have then filtered out 8 papers that are closest to this thesis. I read most of those papers and have tried to understand the approach to gather data, annotation & models used for this task. So, I analysed all the papers and found out which limitations these research have and I tried to improve on the limitation of the previous paper.

2.2 Previous Literature

Muhammad Shoaib, et al. (2022) did thesis on plant lesion segmentation, subtype identification and survival probability prediction. First, a generative model DCGAN was used to generate new image data. The segmentation and classification of the disease is done by 2 separate deep learning models. To segment the lesion the paper uses a special deep learning model named class agnostic network. After segmentation a 3D CNN model is being used to classify the segmented area of the leaves. Finally, a lasso regression model is being used to predict the longevity of the leaf. The research is computationally expensive. Generating new images using a GAN type model requires lots of computational power. The training dataset also had class imbalance. The amount of healthy leaves were larger than the diseased leaves. And finally using 2 separate models for segmentation and classification increases overall training time and inference time.

Muhammad Shoaib, et al. (2022) uses a Unet deep learning model to segment the leaf area and then passes that particular segmented leaf to Inception net classifier to classify that particular leaf disease. This particular research lacks annotated dataset. The solution is not able to detect leaf diseases, rather it segments the whole leaf and passes it to a classifier. The purpose of the segmentation phase is to detect only the leaf and separate it from the background. This paper also

©Daffodil International University

uses 2 different deep learning models for segmentation and classification which have introduced more complexity. Only a single type of crop (tomato) is being used for the task.

Vijai Singh, et al. (2016) approach leaf disease segmentation without any deep learning methods. The paper utilizes genetic algorithm technique to segment the diseased area from the leaf and uses a machine learning algorithm SVM to classify the disease. This technique is computationally efficient because it is not using any deep learning models. But, there are some limitations such as prior information is needed for segmentation which is not efficient. The technique was used on a limited dataset.

K. Elangovan et al. (2017) introduce K Means clustering and Otsu's threshold for segmenting the diseased area and uses SVM machine learning model to classify the segmented area. This technique like the previous one is also computationally efficient. But the limitation is this method requires 2 different models one for segmentation and one for classification. Also, the dataset size was limited to only one crop (chili).

Parul Sharma, et al. (2020) applied a different method for leaf disease detection. 2 different types of CNN are used. One is F-CNN and another one is S-CNN. F-CNN was trained on a full size diseased images and S-CNN was trained on only segmented part of the diseased images. The author concluded that S-CNN performs better than the F-CNN model by achieving 98.6% accuracy. But, the limitation of this paper is it only focuses on the classification task. The segmentation of images are done manually by cropping the images. So, the model is not able to segment diseased areas from the images. And, only a single type of crop is being used for training both models.

Md Ershadul Haque, et al. (2022) performs leaf disease detection using the YOLO-5 model. The proposed solution created a dataset by manually annotating the diseased area. Then trained the YOLO-5 model on the dataset. Before training the YOLO-5 model the dataset size was increased using Data augmentation. The limitation of this paper is that it is an image detection solution. The proposed methodology is not able to segment diseased areas rather only create a bounding box

©Daffodil International University

around the diseased area. YOLO-5 is an old model. Recent versions of yolo models such as YOLO-8 and YOLO-9 perform better than YOLO-5. But, the proposed methodology only uses YOLO-5. Finally, the model was trained on a single type of crop (rice), there was no crop diversity in the dataset.

Fatimazahra Jakjoud, et al. (2019) solves the classification problem of the diseased leaves. The paper uses deep CNN architecture to classify the diseased leaves. Then the paper compares the result between different optimizers. This paper solves only the classification problem of leaf diseases, not the segmentation problem. Also, the paper doesn't have any model diversity. It uses a single deep CNN and compares the result between different optimizers of the same neural network.

Sharada P. Mohanty, et al. (2016) proposed a solution for classifying leaf diseases based on Alexnet and GoogleNet deep learning architecture. The solution is only limited to classification. There is no segmentation solution in the paper. But, segmented areas can be detected through visualizing the activation of the initial layer of the network. But this method is not efficient at all.

©Daffodil International University

2.3 Research Gap & Objective from Literature Review

Author Name	Methodology	Research Gap	Objective
Muhammad Shoaib, et al. 2022	DCGAN, CANet, 3D CNN	Class Imbalance.	A new dataset containing 10 different crop diseases and a semi automated annotation technique to reduce the manual annotation time.
Muhammad Shoaib, et al. 2022	Unet, Inception net	Limited to only tomato plant disease	
K. Elangovan et al. 2017	Kmeans Clustering, SVM	Limited to only chili plants	
Parul Sharma, et al. 2020	S-CNN, F-CNN	Limited to single crop disease.	
Vijai Singh, et al. 2016	Genetic Algorithm, SVM	No deep learning technique used.	A deep learning based segmentation model to segment the diseased area from the diseased leaf.
Md Ershadul Haque, et al. 2022	YOLOv5	Only limited to segmentation	
Fatimazahra Jakjoud, et al.2019	CNN	Only limited to classification	
Sharada P. Mohanty, et al. 2016	Alexnet, Googlenet	Only limited to classification	

Figure 2.1: Research Gap & Objective from Literature Review

©Daffodil International University

2.4 Summary

In In the above literature review the conclusion from my study is there are numerous methods employed for segmentation of Leaf. However, for the most part, the methods are data constrained. No study shown valid methodology to annotate leaf image data except manual methodology of annotation. And, most of the papers uses multiple network, one of which for segmentation and another one for classification. In my research I tried to tackle these limitations.

©Daffodil International University

CHAPTER 3
METHODOLOGY

3.1 Introduction

The methodology section provides the overall methodology of the thesis. From data collection to data annotation, building a semi automated annotation pipeline and finally training different segmentation models. This section is written in such a way that anyone can reproduce the work.

3.2 Dataset Collection

Data for this study was obtained from two main datasets: PlantVillage (Hughes et al. 2020) dataset and Dhan-Shomadhan (Md Fahad Hossain et al. 2021) dataset. To narrow down the dataset and ensure adequate focus, I selected specific subsets of each dataset. The main criterion was the prevalence of certain identified crop diseases in Bangladesh. The PlantVillage Dataset is used for plant pathologist studies and machine learning research on a variety of crops. A subset of nine classes of leaf images from the PlantVillage Dataset was used in this study.

The dataset contains images of leaves and their conditions of health or diseases. Each of the nine classes is sample images of each type of crop disease or its health status. They are also among the most common diseases among the common crops in Bangladesh and are thus relevant to the study.

The Dhan-Shomadhan dataset is a specialized dataset focusing on rice leaf diseases prevalent in Bangladesh. From this dataset, one specific class of images was extracted for inclusion in our research. This class was chosen due to its significance in the local agricultural context and its potential impact on rice production.

The combined dataset, incorporating ten classes from the Plant Village Dataset and one class from the Dhan-Shomadhan Dataset, provides a robust foundation for analyzing plant diseases across different species and geographical regions.

©Daffodil International University

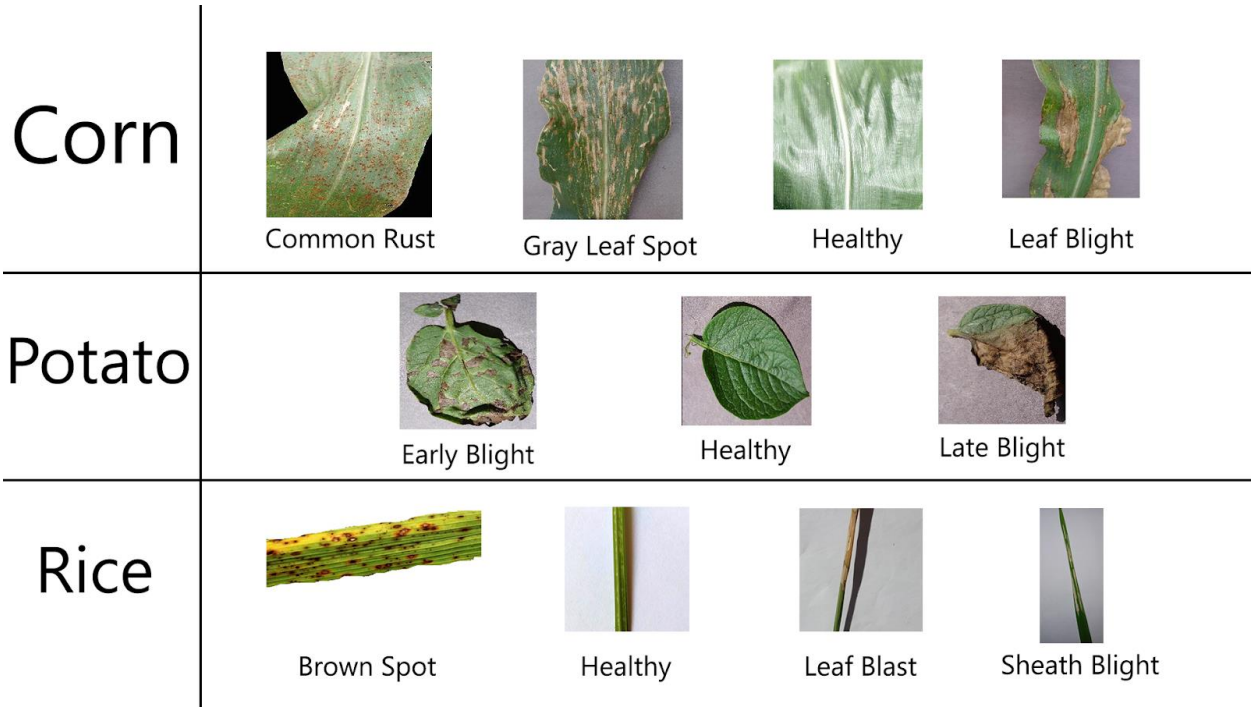

Figure 3.1: Images of Raw Dataset

3.3 Proposed Methodology

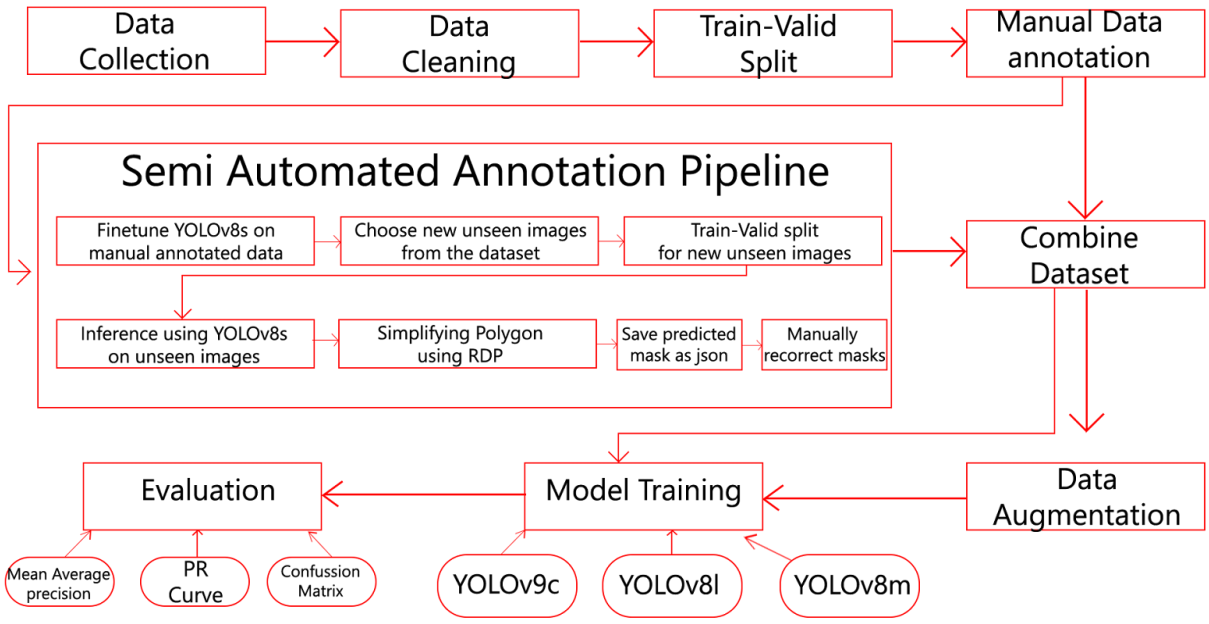

Figure 3.2: Methodology Diagram

©Daffodil International University

Step 1. Data Collection: It is the initial stage of our research where I've gathered data from secondary sources such as PlantVillage and Dhan-somadhan datasets. And, create a new dataset by getting a subset from the datasets and combining them.

Step 2. Data Cleaning: After gathering all the data in the 2nd stage I've cleaned the dataset. First, I've selected a specific amount of images for each class and structured them into a new folder. All the classes were in separate folders. Change the name of images according to class name.

Step 3. Train Valid Split: Split the combined dataset into a training set and a validation set.

Step 4. Manual Data Annotation: Since the research is on instance segmentation hence having only image data with labels isn't enough. In this phase I've annotated the dataset for each class using the VGG Image Annotation tool.

Step 5. Semi Automated Annotation Pipeline: In this phase I've trained the YOLO v8m objective detection model on the manually annotated dataset and made predictions on unseen images. All the predicted masks which have greater than 50% confidence score are considered as valid segmentation. After that, I've manually corrected the predicted masks using the VGG Image Annotation tool. This process helps me to speed up the annotation phase.

Step 6. Combine Dataset: Combine both manual and semi automated annotated datasets.

Step 7. Data Augmentation: Apply different types of data augmentation on the training dataset.

Step 8. Train the Models: Train 3 different models YOLO v8m, YOLO v8l and YOLO v9c using the training dataset.

Step 9. Evaluate Performance: Evaluate each of the model's performance such as mAP@50, mAP@50-95 using validation set.

3.4 Data Cleaning

The For the data cleaning part, I selected some number of images (classes were already collected from Plant Village and Dhan-Shomadhan datasets) from each class. After the images were chosen, they were renamed in a consistent manner for better recognition and organization. After the

©Daffodil International University

renaming process, the images were divided into different folders classwise. Such a clear structure in terms of arranging the images into their class and folders was time- and effort-saving in further analysis and model training to make the data processing pipeline faster.

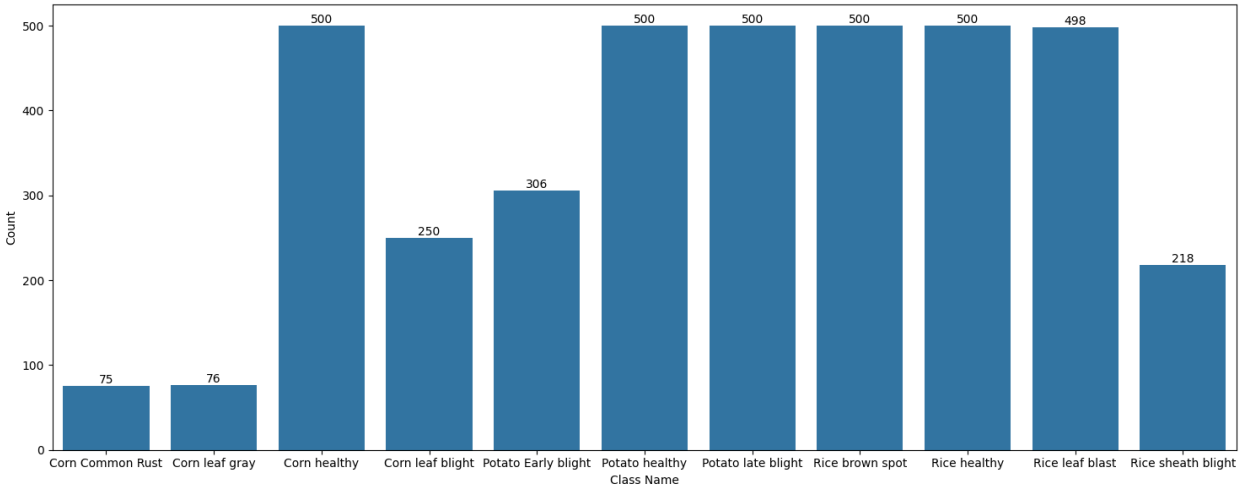

Figure 3.3: Total Images for Each Class

3.5 Train-Validation Split

In this step, I've split the dataset into 2 parts. One is the training set and the other one is the validation set. The reason for doing this step so early is to ensure the validation set has the proper representation of the training set. I've manually picked the images for validation and training sets. The ratio of training and validation is 87:13.

©Daffodil International University

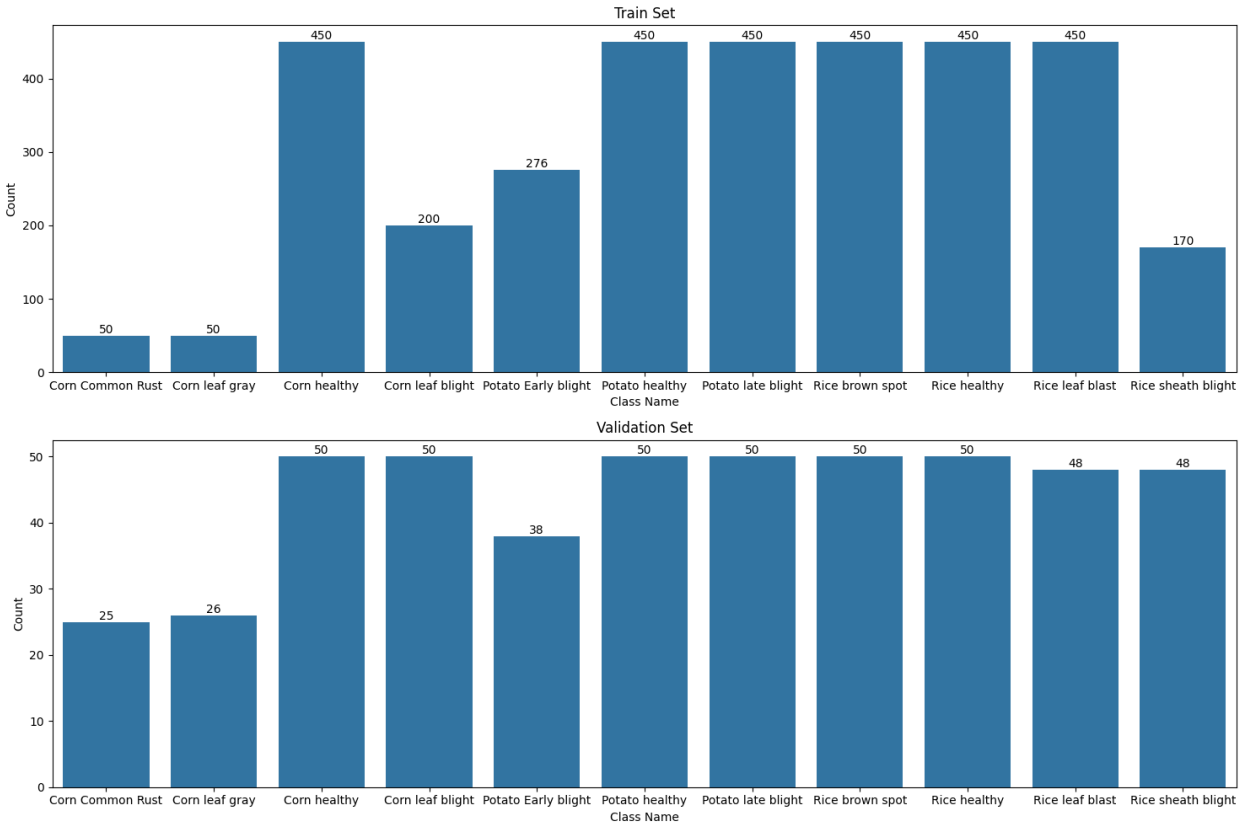

Figure 3.4: Total Images for Each Class in Train-validation set

©Daffodil International University

3.6 Manual Data Annotation

After the data cleaning procedures, the data set was manually annotated to mask the exact diseased region to be segmented in each of the images. For this I used the VGG Image Annotator (VIA), which is a popular and flexible tool for image annotation tasks.

All images were carefully inspected, areas of the disease were masked and were carefully annotated. The manual annotation steps were as follows:

- Loading Images: The cleaned and organized images were loaded into the VGG Image Annotator tool.
- Drawing Masks: For each image, polygons were drawn to outline the diseased areas. These masks accurately represented the regions of interest, ensuring that the disease-affected parts of the leaves were distinctly marked.
- Saving Annotations: The annotated masks were saved alongside the images, creating a detailed and structured dataset that included both the original images and their corresponding disease annotations.

This careful manual annotation ensured high-quality, detailed data that is essential for training accurate and effective machine learning models for disease classification.

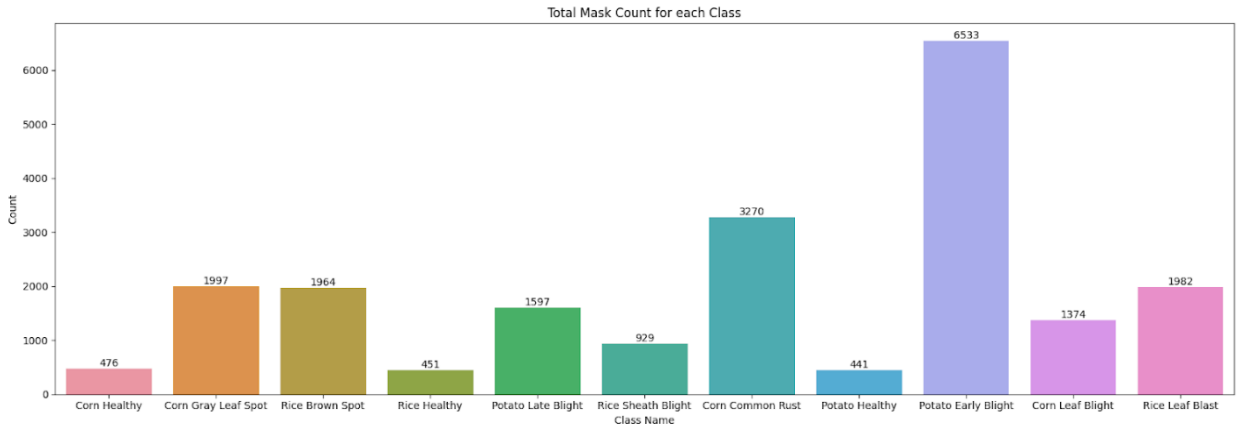

Figure 3.5: Total Mask Count for Each Class

©Daffodil International University

3.7 Semi Automated Annotation pipeline

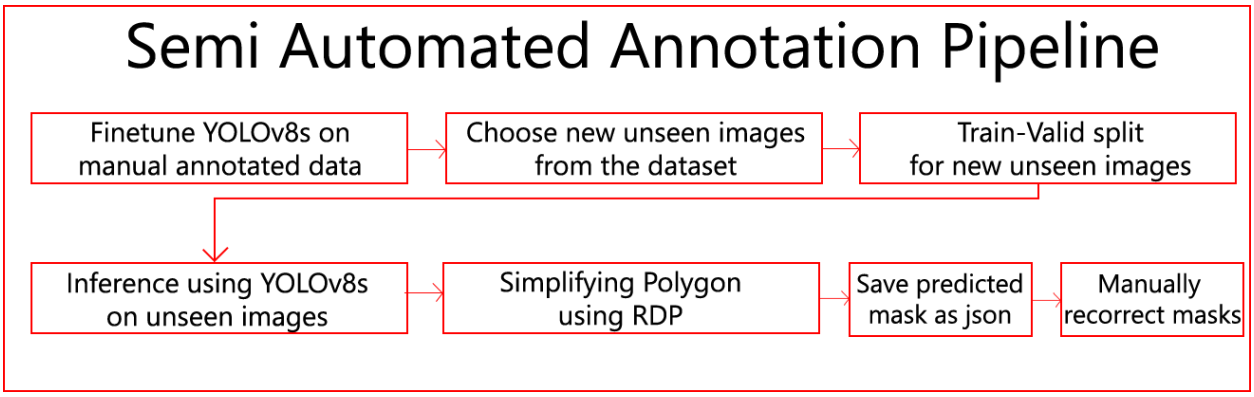

Figure 3.6: Semi Automated Annotation Pipeline

Semi automated annotation pipeline is a process where the model is trained on a subset of annotated data and inferred on unseen data. Then the inferred mask of the unseen data is manually corrected by the human annotator. The purpose of the semi automated annotation pipeline is to increase the total annotated data samples with less effort. Here, human annotator won't have to annotate the images from scratch rather he'd only correct the mistakes of model,

Step 1. Train the model: In the first step of the semi automated pipeline I've trained the YOLOv8m model with the manually annotated dataset.

Step 2. Choose New Samples: After training the model I've chosen new samples from the PlantVillage and the Dhan-Somadhan dataset which will be used for inference in the semi automated pipeline. Since I've changed the name of the original images in the "Data Cleaning" section I've to compare the image hash for the unseen images and already annotated images so that there is no copy of already annotated images in the unseen images.

©Daffodil International University

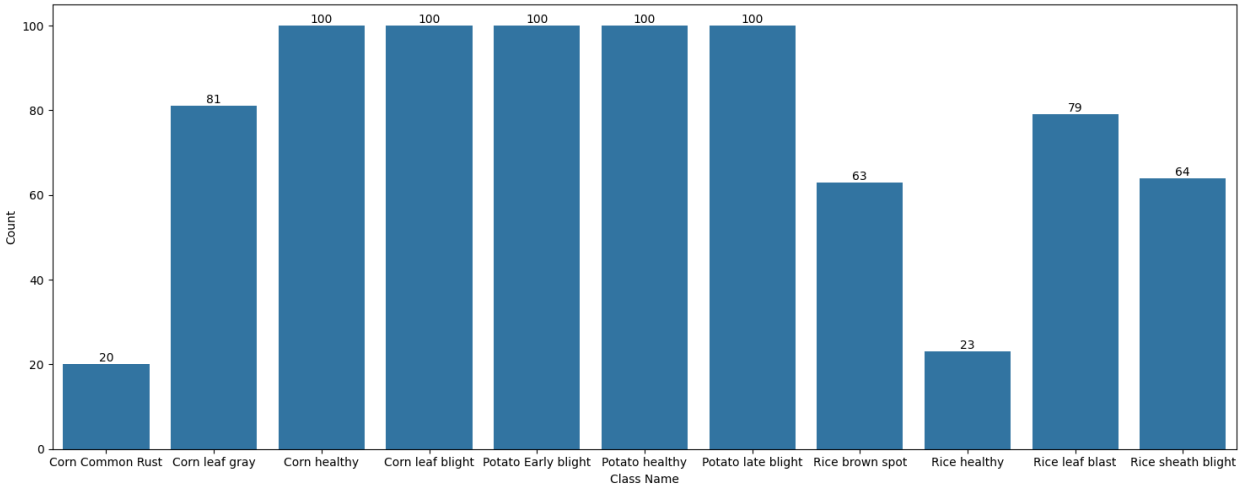

Figure 3.7: Total New Images for Each Class

Step 3. Train-Val split new samples: After choosing the new samples just like before doing the manual annotation I've manually created the train-val set, Here I've also done the same. I've created a train and validation set from the new samples. The reason is to have proper representation in the validation set

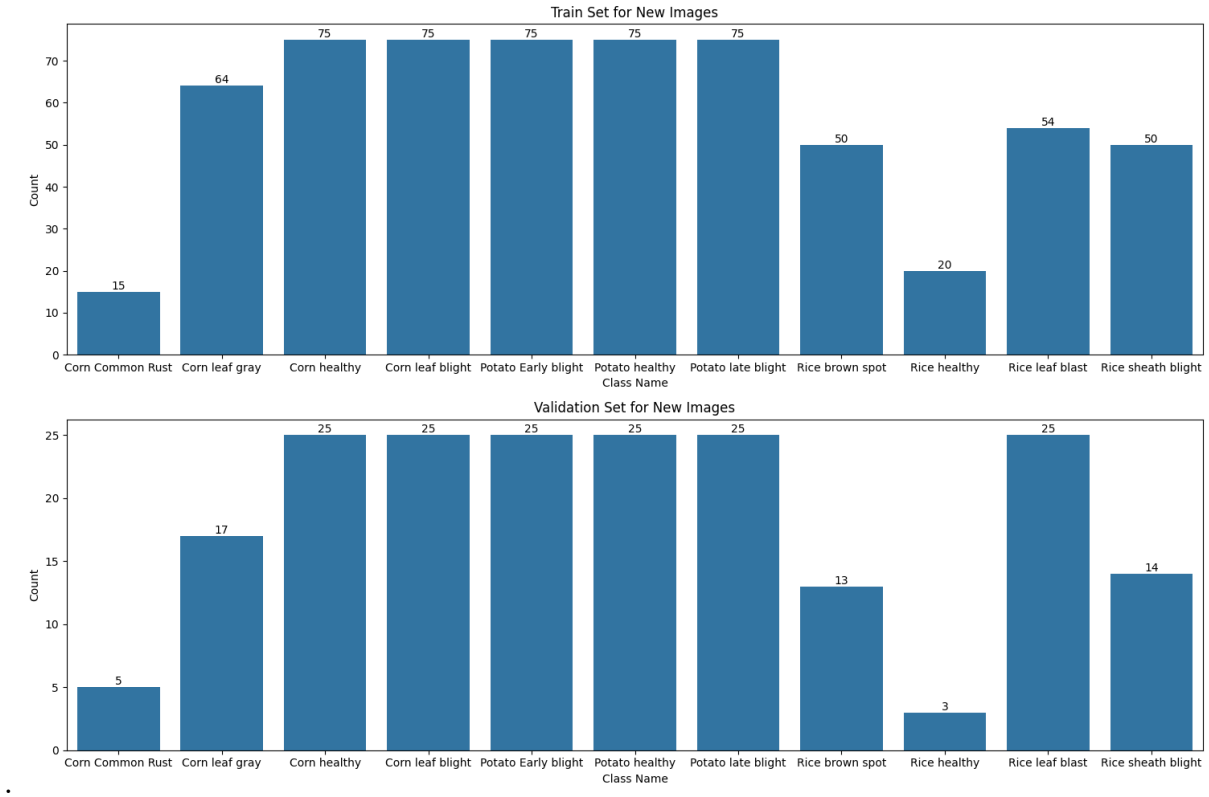

Figure 3.8: Total New Samples for Each Class in Train-validation set

©Daffodil International University

Step 4. Inference on New Samples: I've predicted the new samples using the trained YOLOv8m model for both the new training set and validation set.

Step 5. Simplify Polygon using RDP: The predicted polygon from the model were predicting too many segmentation points. Having too many segmentation points is unnecessary and can be challenging later if we want to update any points. Therefore I've used the "Ramer Douglas Peucker" algorithm to reduce the number of segmentation points and hardly change the original polygon shape of the mask.

Figure 3.9: Reducing Polygon points using RDP

Step 6. Saving Predicted Mask as JSON project: I've saved all the predicted masks (and updated the mask using RDP) inside a JSON file. The json file is structured in such a way that it can be loaded as a saved project from the VGG Image Annotation (VIA) tool and manually update any predicted segment or points inside from the tool. I've saved the training set and validation set to different JSON project files.

©Daffodil International University

Step 7. Manually Correct the Predicted Mask: After opening the json project file inside VGG Image Annotator (VIA) tool I've manually corrected the predicted masks for the new train set and validation set.

3.8 Combine Both Dataset

In the previous section I've created a semi automated annotation pipeline where I've annotated new images with the help of YOLOv8m object detection model. Now, the new train set and validation set is combined with the previous manually annotated dataset. New train set is combined with the previous train set and the new validation set is combined with the previous validation set. After combining the both train and validation set the new ratio for train validation set is 85:15

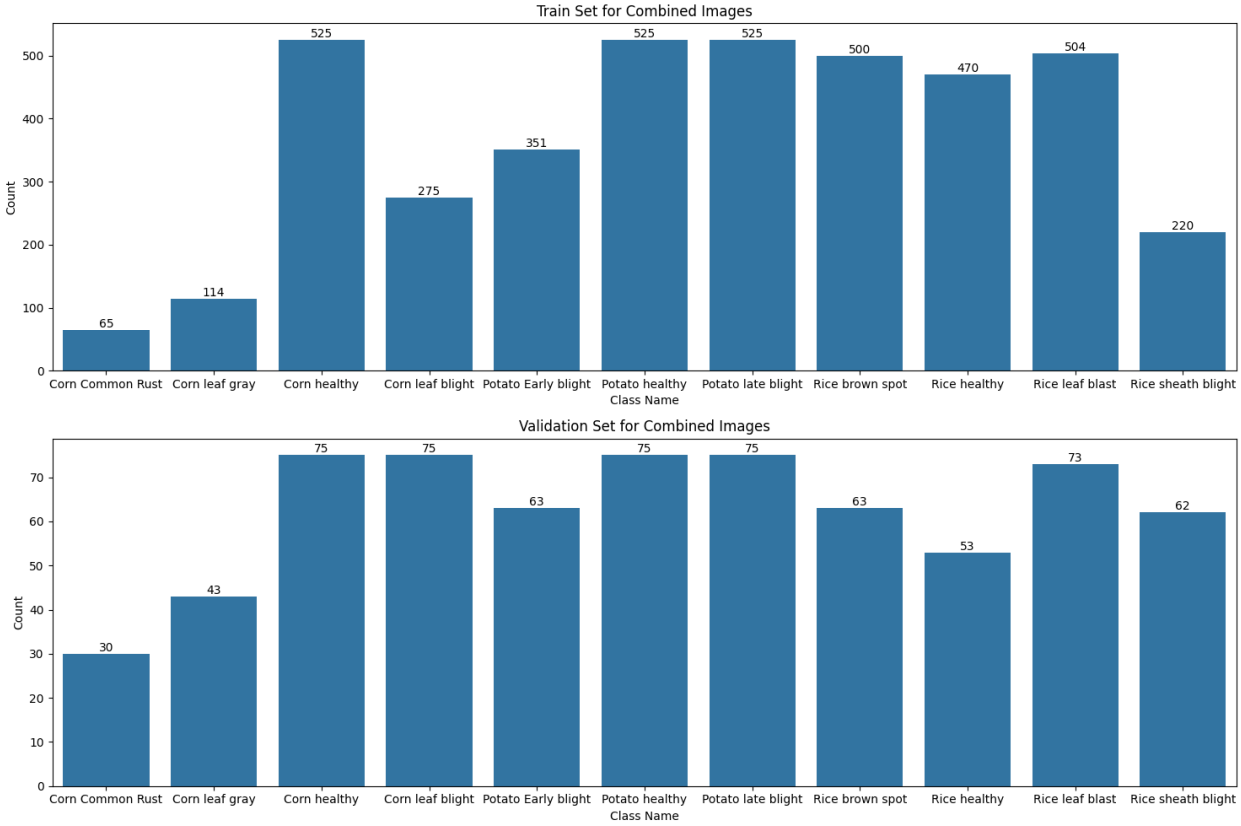

Figure 3.10: Total Combined Samples for Each Class in Train-validation set

©Daffodil International University

3.9 Data Augmentation

In After combining the dataset I've applied data augmentation on the training set only. The augmentation is completed by RoboFlow. Different Types of data augmentation were being used in the training set. Those are: Horizontal and vertical flip, clockwise rotation, anti clockwise rotation, upside down rotation, Saturation between -25% to 25% and brightness between -15% to 15%. After augmentation the training dataset size was increased from 4010 images to 8020 images.

3.10 Model Training

I've created a combined annotated dataset. Where the dataset was annotated manually and using a semi automated annotation pipeline. Now I've chosen 2 State of the art object segmentation deep learning models to train the dataset. These 2 models are YOLOv8 and YOLOv9.

3.10.1 YOLOv8

YOLOv8 is an object detection and segmentation model. It is from the YOLO family. But YOLOv8 is different from the previous YOLO model. YOLOv8 has 3 components and those are backbone, neck and head. Neck component is new to YOLOv8.

Backbone: Similar to YOLOv5, YOLOv8 leverages a convolutional neural network (CNN) as the backbone. This initial stage extracts features from the input image. Notably, YOLOv8 utilizes the CSPDarknet53 model, which is a variant of the Darknet architecture commonly used in YOLO versions.

Neck: Unlike previous YOLO versions that explicitly mentioned a neck component, YOLOv8 integrates feature merging within the backbone itself. This streamlined approach combines features extracted at different resolutions, providing a richer feature set for object detection.

Head: The head is responsible for the final object detections. YOLOv8 employs a novel C2f (Channel Attention Feature) module in the head. This module refines the features by combining high-level information with contextual details, enhancing detection accuracy. Following the C2f module, detection modules predict bounding boxes and class probabilities for the objects present in the image.

©Daffodil International University

Overall, YOLOv8 simplifies the architecture by merging the neck functionality into the backbone and introducing the C2f module for improved contextual awareness in object detection.

There are several variants of YOLOv8 based on the size of the parameters. I've chosen 2 variants of YOLOv8 and those are YOLOv8l and YOLOv8m. YOLOv8m is smaller in size compared to YOLOv8l but has faster inference time. Meanwhile YOLOv8l has better performance.

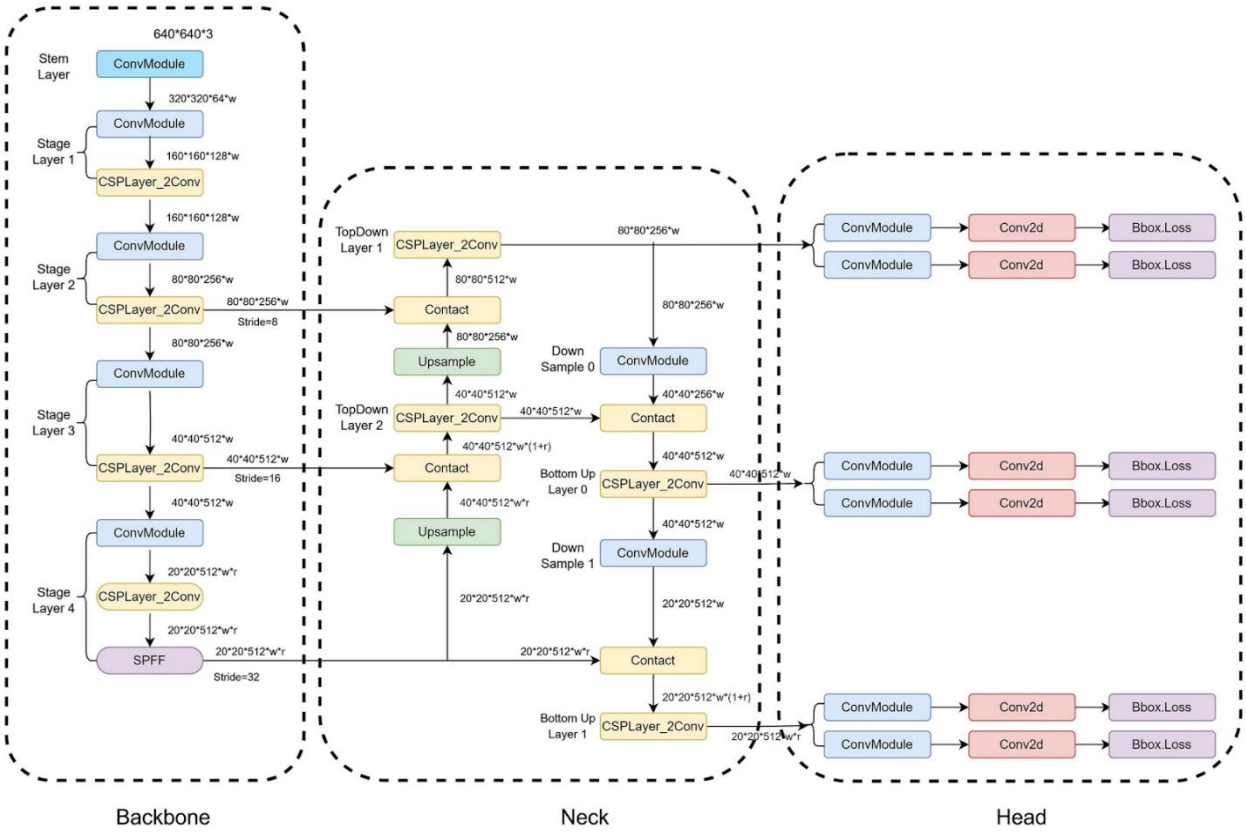

Figure 3.11: YOLOv8 Architecture

Hyper Parameter used for training:

epochs = 50 (Total number of epochs for training)

batch = 16 (Total number of images in a single batch)

imgsz = 640 (Target image size for training)

optimizer = "auto" (Select one of the SGD, Adam etc. based on model configuration)

©Daffodil International University

lr0 = 0.01 (Initial Learning rate)

lrf = 0.01 (Final Learning rate as a fraction of initial learning rate.)

momentum = 0.937 (Momentum factor for SGD or beta1 for Adam optimizers)

Model Segmentation Loss on Val Data while Training:

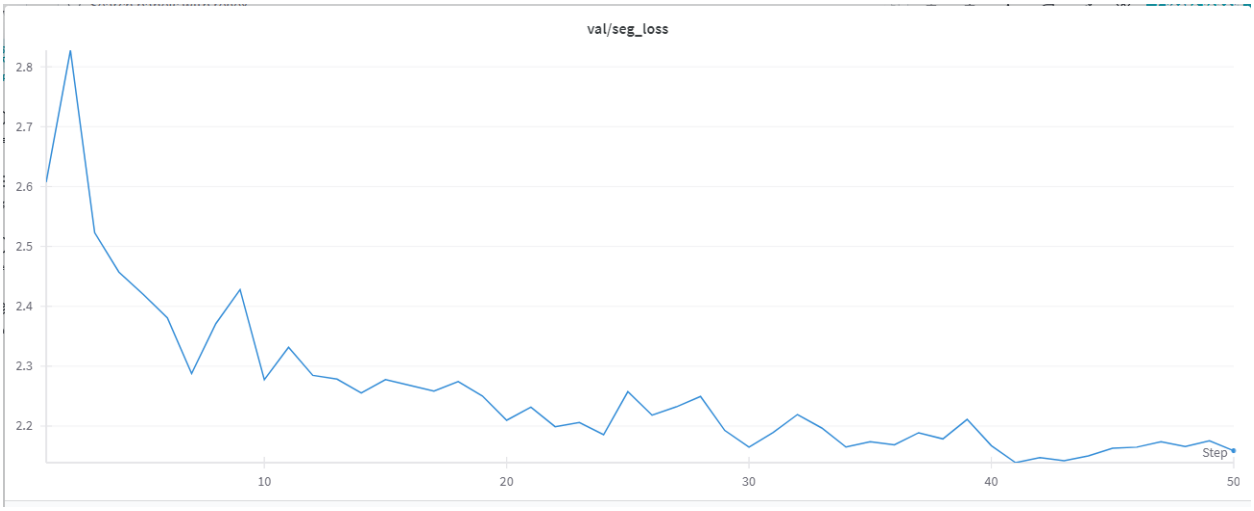

Figure 3.12: Segmentation Loss for YOLOv8m

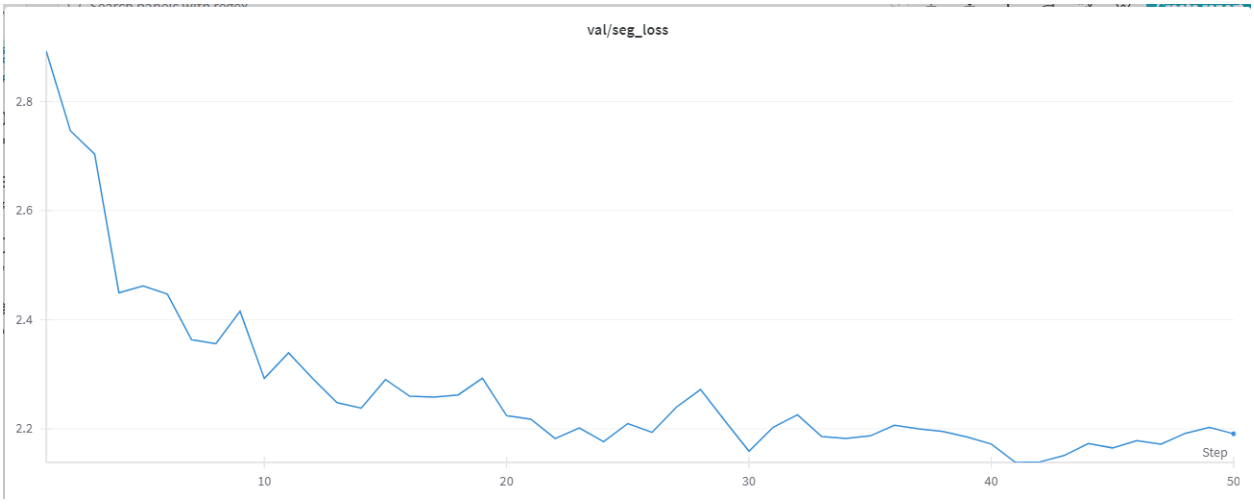

Figure 3.13: Segmentation Loss for YOLOv8l

©Daffodil International University

3.10.1 YOLOv9

YOLOv9 is the latest deep learning model from the YOLO family. Overall architecture of YOLOv9 is similar to YOLOv8 but YOLOv9 introduces some new concepts. It introduces GELAN (Generalized Efficient Layer Aggregation Network) in the backbone. It also introduces a new branch called Auxiliary branch which was not present in YOLOv8.

Backbone: YOLOv9 utilizes a similar concept to YOLOv8, employing a CNN for feature extraction. However, it introduces a new backbone architecture called GELAN (Generalized Efficient Layer Aggregation Network). GELAN focuses on optimizing parameter usage and computational complexity while maintaining accuracy.

Neck: Unlike YOLOv8 which integrated feature merging into the backbone, YOLOv9 likely separates the neck again. This neck component refines and combines features extracted at various resolutions from the backbone, providing a richer dataset for object detection.

Head: The head is responsible for the final detections. YOLOv9 leverages concepts from YOLOv8's head, potentially including the C2f module for feature refinement. The head then predicts bounding boxes and class probabilities for the objects in the image.

Auxiliary Branch: A novel addition in YOLOv9 is the auxiliary branch. This branch improves the training process by providing additional information that bridges the gap between the input data and the final output. This helps mitigate information loss that can occur in deeper layers of the network, leading to more reliable training and potentially improved detection accuracy.

Programmable Gradient Information: Another key concept introduced by YOLOv9 is Programmable Gradient Information (PGI). Traditional deep learning models can suffer from information degradation as data passes through layers. PGI aims to address this by allowing for complete input information to be used when calculating the objective function. This ensures more reliable gradient information is available to update network weights during training, potentially leading to a more accurate model.

Overall, YOLOv9 introduces GELAN for an efficient backbone, potentially reintroduces a dedicated neck for feature fusion, and leverages PGI and an auxiliary branch to enhance training and potentially improve object detection accuracy.

©Daffodil International University

YOLOv9 also has lots of variants. I've chosen YOLOv9c from all the variants. YOLOv9 is the latest model and not all the YOLOv9 variants support segmentation tasks. Therefore I've chosen YOLOv9c.

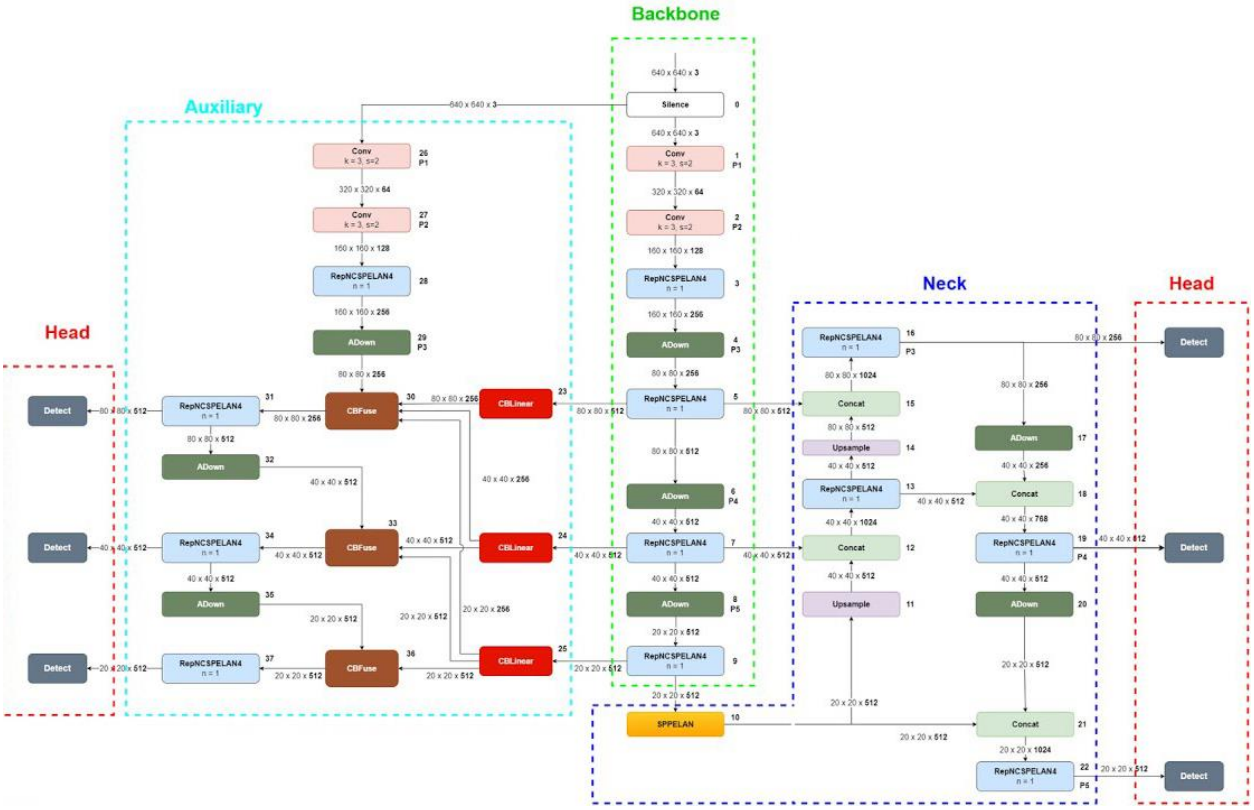

Figure 3.14: YOLOv9 Architecture

Hyper Parameters used for training:

epochs = 50 (Total number of epochs for training)

batch = 16 (Total number of images in a single batch)

imgsz = 640 (Target image size for training)

optimizer = "auto" (Select one of the SGD, Adam etc. based on model configuration)

lr0 = 0.01 (Initial Learning rate)

lrf = 0.01 (Final Learning rate as a fraction of initial learning rate.)

©Daffodil International University

momentum = 0.937 (Momentum factor for SGD or beta1 for Adam optimizers)

Model Segmentation Loss on Val Data while Training:

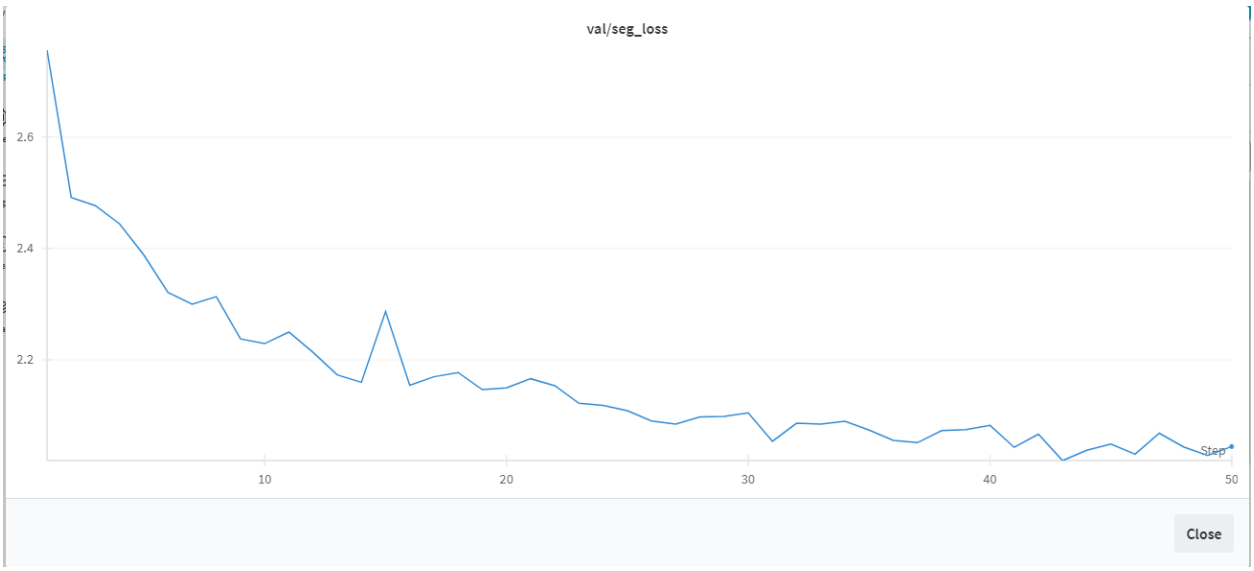

Figure 3.15: Segmentation Loss for YOLO v9c

3.11 Evaluation Method

Evaluation methods for segmentation tasks are different from classification tasks. There are different evaluation methods that are used for evaluating segmentation tasks. The most important evaluation metrics for object segmentation tasks are PR (precision-recall) curve, mAP (mean average precision) etc etc.

3.11.1 Precision & Recall

Here, precision and recall are not directly related for segmentation metrics. But, to calculate different segmentation metrics i.e: mAP or PR curve both precision and recall are needed.

Precision: It measures the ratio of correctly predicted positive detections to the total number of predicted positive detections (including both true positives and false positives).

$$\text{Precision} = TP / (TP+FP)$$

©Daffodil International University

Recall: It represents the proportion of actual positive cases that the model correctly identifies.

$$Recall = TP/ (TP+FN)$$

3.11.2 PR Curve

Precision-Recall Curve: A precision-recall curve is generated for each object class in the dataset. This curve plots the precision values on the Y-axis against the recall values on the X-axis for various detection thresholds.

3.11.3 mAP

Before finding the mAP score, Average Precision needed to be calculated.

AP (average precision): The average precision (AP) for each class is calculated as the area under the precision-recall curve. This single value summarizes the model's performance across all possible detection thresholds for that class.

mAP (mean average precision): The mAP is obtained by averaging the AP values across all object classes in the dataset. This provides a single score that represents the overall performance of the object detection model.

$$mAP = 1 / C * \Sigma(APc)$$

Where, C = total number of object classes in the dataset

APc = Average Precision for class c.

There are 2 different types of mAP score.

mAP@0.5: mAP@0.5 represents the mAP score when the intersection over union value is 0.5

mAP@0.5-0.95: mAP@0.5-0.95 represents the mAP score when the intersection over union value is in between 0.5 to 0.95

©Daffodil International University

3.11.4 F1 Confidence Curve

To find out F1 confidence curve first F1 score has to be calculated.

F1 Score: The F1 score is a harmonic mean of precision and recall, providing a single metric that incorporates both aspects of a model's performance.

$$F1\ Score = 2 * (Precision * Recall)/(Precision + Recall)$$

F1 Confidence Curve: F1 confidence curve plots F1 score for each class in the dataset in the Y-axis and compares its value with different confidence thresholds which are plotted in the X-axis.

3.12 Summary

This section details the dataset creation process. Images of rice, potato, and corn leaf diseases were collected from diverse sources, cleaned, and meticulously split into training and validation sets. Both sets underwent manual annotation, followed by the implementation of a novel semi-automated pipeline to significantly reduce annotation time. The manually and semi-annotated data were then combined to form a comprehensive dataset for training and evaluating various YOLOv8 and YOLOv9 deep learning models. The next result section will include the information of different evaluation metrics for these models on various experiments.

©Daffodil International University

CHAPTER 4

Result

4.1 Introduction

The result section showcases all the experiments that are conducted in the thesis and compares those experiments with different evaluation metrics.

4.2 Experiment Details

I've trained a total of 6 different models.

Experiment 1: In the first experiment I've trained the YOLOv8m model only on the manually annotated dataset. In the training dataset there are 3866 images and in the validation set there are 484 images.

Experiment 2: 2nd experiment I've trained the YOLOv8l model with only on manually annotated dataset.

Experiment 3: In this experiment I've trained the YOLOv8m model with the combined dataset. The dataset contains both manual annotation and semi automated annotation. In the training dataset there are 4010 images and 686 images in the validation dataset.

Experiment 4: This is also similar to experiment 3, but instead of using the YOLOv8m model I've used the YOLOv8l model for this experiment.

Experiment 5: In this experiment I've trained the YOLOv9c model on the combined dataset.

Experiment 6: Lastly, I've trained the YOLOv8m model on the augmentation dataset. The performance of YOLOv8m was poor and that's why I didn't train other YOLO models on the augmentation dataset.

©Daffodil International University

4.3 mAP comparison

In this section I've compared 2 different mAP values (mAP@0.5 and mAP@0.5-0.95) for all the experiments and interpreted the results. mAP values are calculated for both bounding box and segmentation mask.

Experiment Name	mAP Values for Bounding Box	
	mAP@0.5	mAP@0.5-0.95
YOLOv8m with manually Annotated Dataset	0. 7389	0.5099
YOLOv8l with manually Annotated Dataset	0. 7355	0.5097
YOLOv8m with Combined Dataset	0. 754	0.5303
YOLOv8l with Combined Datase	0. 7553	0.5324
YOLOv9c with Combined Datase	0. 7462	0.5315
YOLOv8m with Augmented Dataset	0. 72	0.5079

Figure 4.1: Bounding Box mAP value Comparison

©Daffodil International University

Experiment Name	mAP Values for Segmentation Mask	
	mAP@0.5	mAP@0.5-0.95
YOLOv8m with manually Annotated Dataset	0. 7278	0. 4839
YOLOv8l with manually Annotated Dataset	0. 7238	0. 4802
YOLOv8m with Combined Dataset	0. 7347	0. 5094
YOLOv8l with Combined Datase	0. 7417	0. 5118
YOLOv9c with Combined Datase	0. 7338	0. 5049
YOLOv8m with Augmented Dataset	0. 7177	0. 4946

Figure 4.2: Segmentation mask mAP value Comparison.

From the above tables the mAP values for both Bounding box and segmentation mask are high for the YOLOv8l model which was trained on the combined dataset. Also, the YOLOv8m model trained on combined dataset is right behind og YOLOv8l model. In the table it is clearly shown that the model which was trained on the combined dataset is performing better than the model which was trained only on manual annotated dataset. Finally, we can see the YOLOv8m model which was trained on the augmentation dataset didn't perform well. Its mAP score is even lower than the model which was trained on only manually annotated dataset. So, in this dataset augmentation won't increase the model's performance.

©Daffodil International University

4.4 PR curve comparison

From the above section we can conclude that the model's which are trained on the combined dataset are performing better than the other models. That's why in the upcoming performance metrics I'll be showing performance comparisons only for YOLOv8m, YOLOv8l and YOLOv9c which are trained on a combined dataset. In the segmentation task the PR curve is calculated for both the bounding box and segmentation mask separately. Since, it is segmentation task I'll be focussing on the PR curve of the segmentation mask

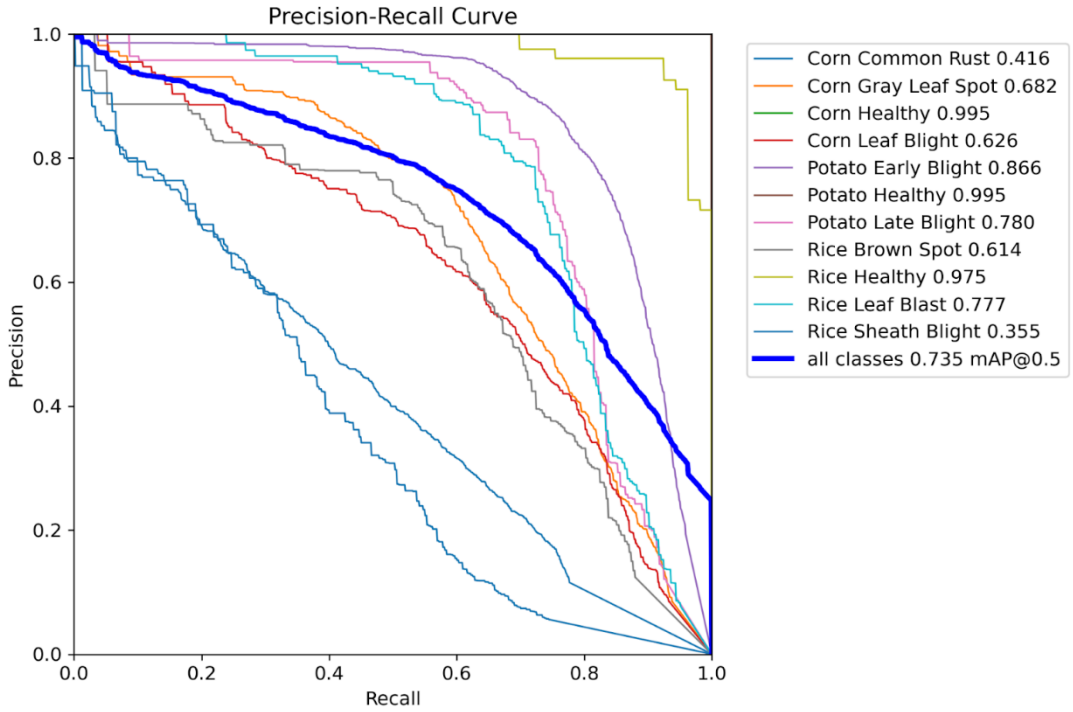

Figure 4.3: YOLOv8m PR curve for Segmentation Mask

©Daffodil International University

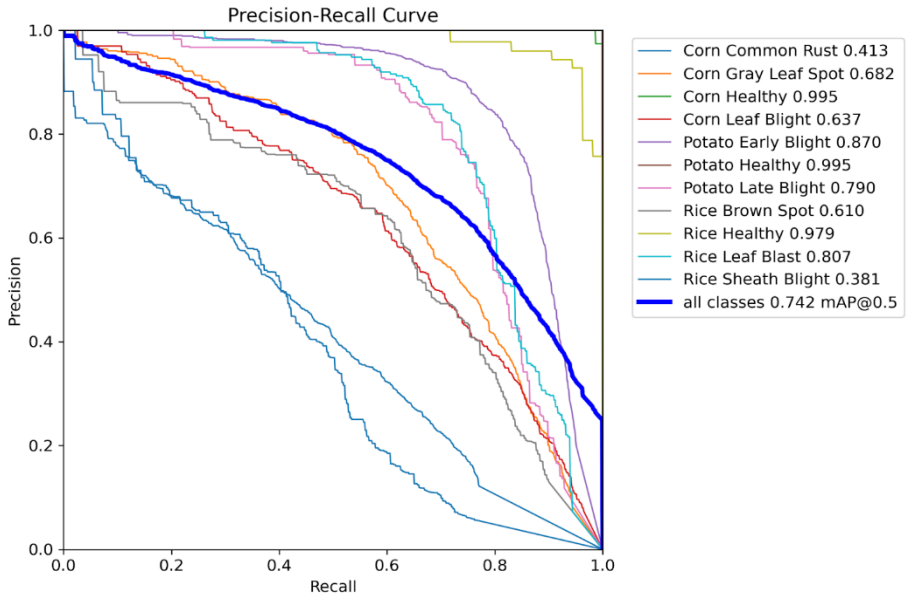

Figure 4.4: YOLOv8l PR curve for Segmentation Mask

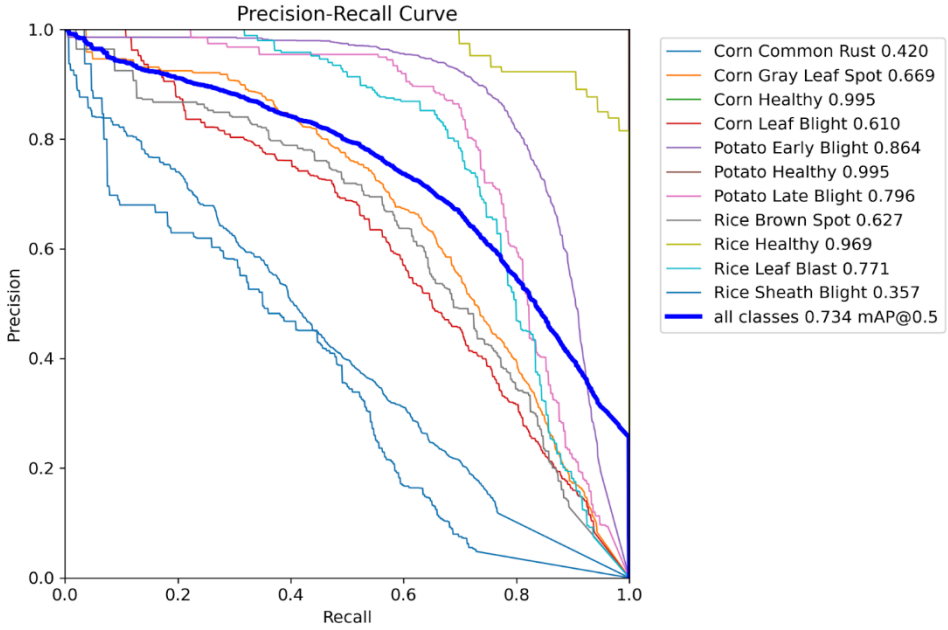

Figure 4.5: YOLOv9c PR curve for Segmentation Mask

©Daffodil International University

From the above PR curve it is visible that YOLOv8l is performing much better than the models. But, the performance of the 2 classes performance are very poor. Those are “Corn Common Rust” and “Rice Sheath Blight”. The exact reason for their poor performance is unknown but the quality of the annotation and the number of samples may have been the root cause of their poor performance.

4.5 F1 Confidence Curve

F1 confidence curve helps to visualize the F1 score for different confidence thresholds.

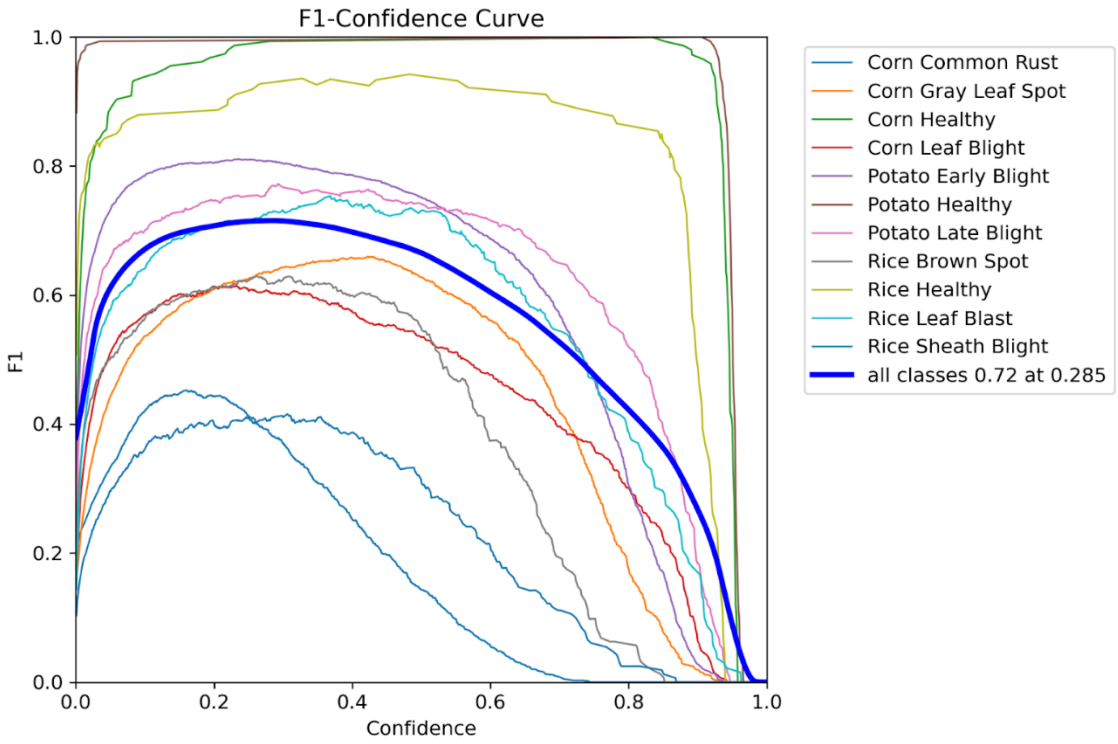

Figure 4.6: YOLOv8m F1-Confidence for Segmentation Mask

©Daffodil International University

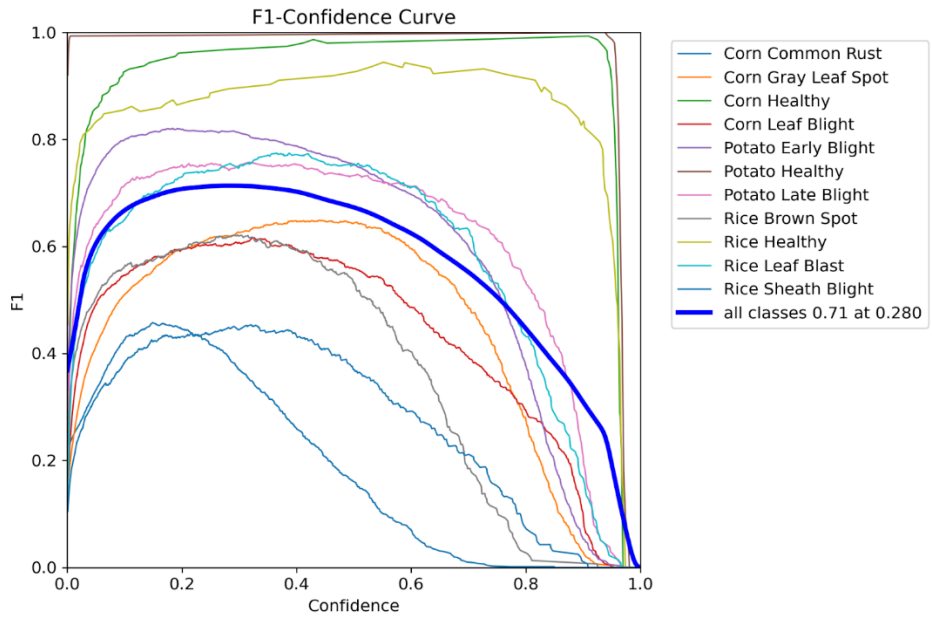

Figure 4.7: YOLOv8l F1-Confidence for Segmentation Mask

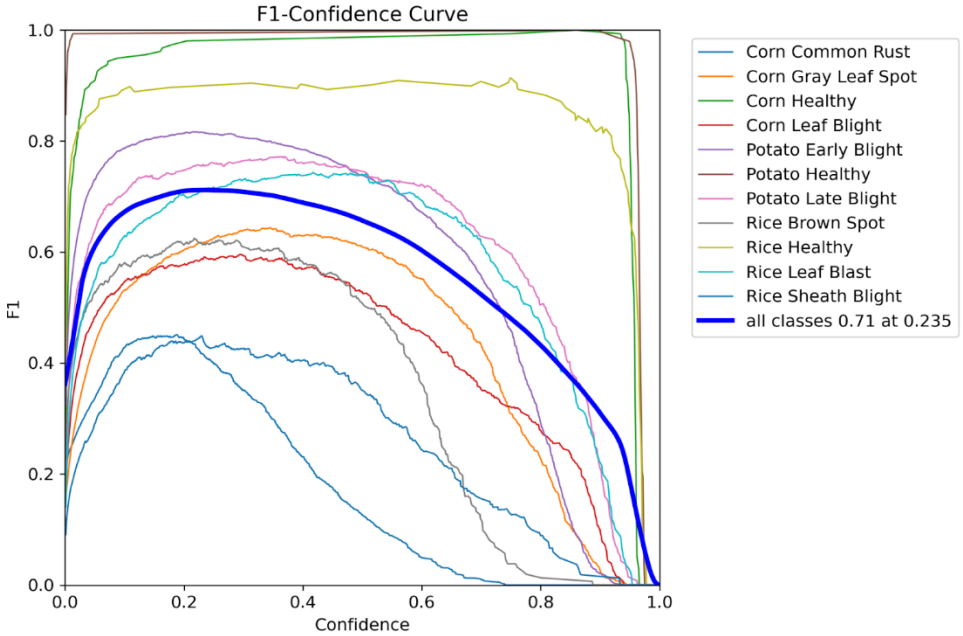

Figure 4.8: YOLOv9c F1-Confidence for Segmentation Mask

From the above graph all the models perform almost identical. But in this metric YOLOv8m performs slightly better than the others.

©Daffodil International University

4.6 Confusion Matrix

In object segmentation, a confusion matrix is a table summarizing how many classes were correctly or incorrectly assigned to each object class. It helps identify missed objects.

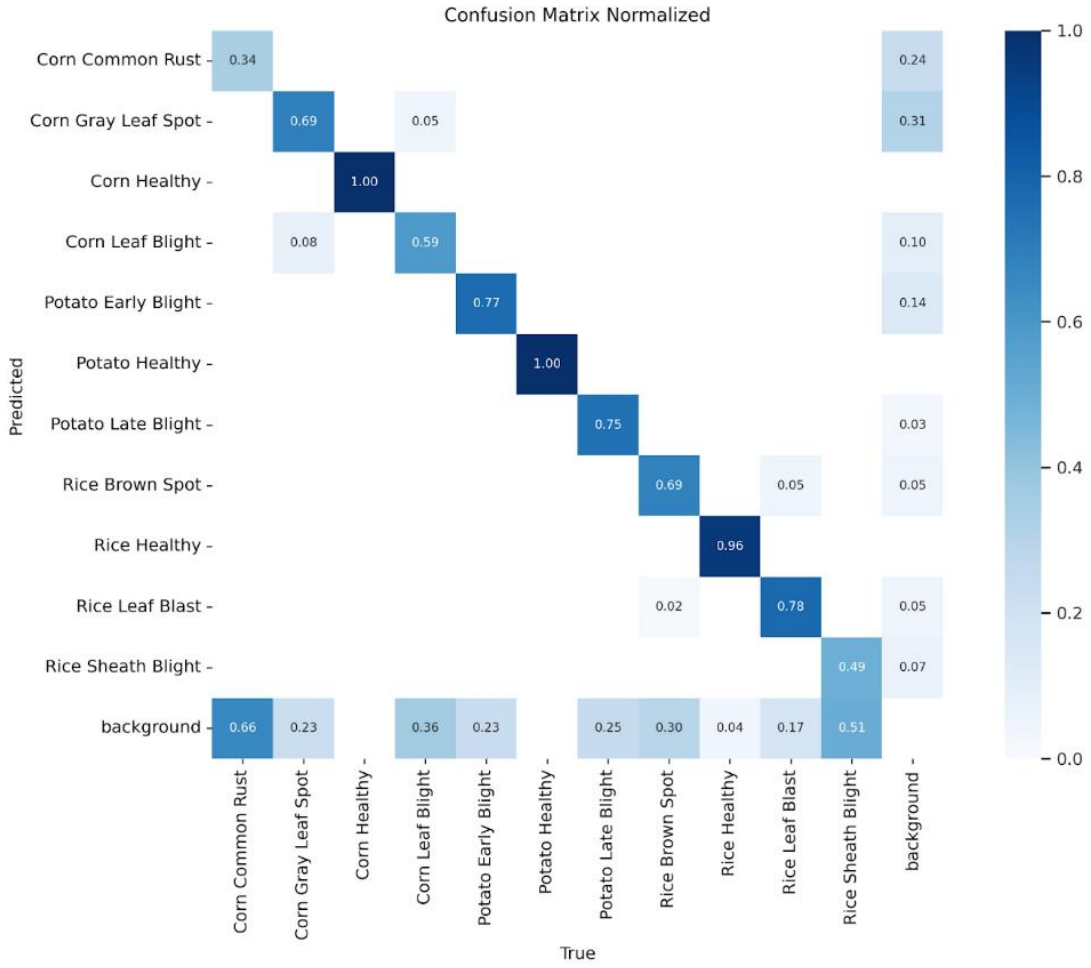

Figure 4.9: Confusion Matrix YOLOv8m

©Daffodil International University

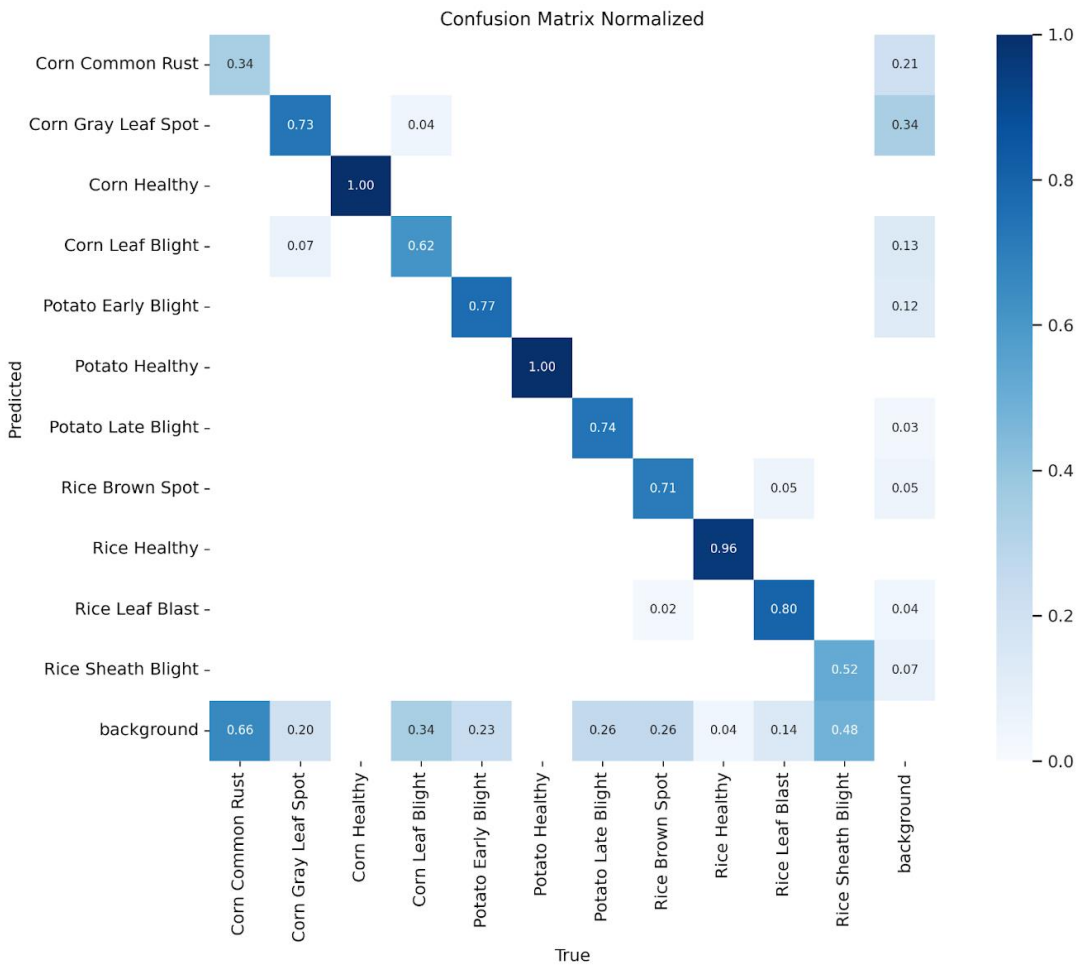

Figure 4.10: Confusion Matrix YOLOv8l

©Daffodil International University

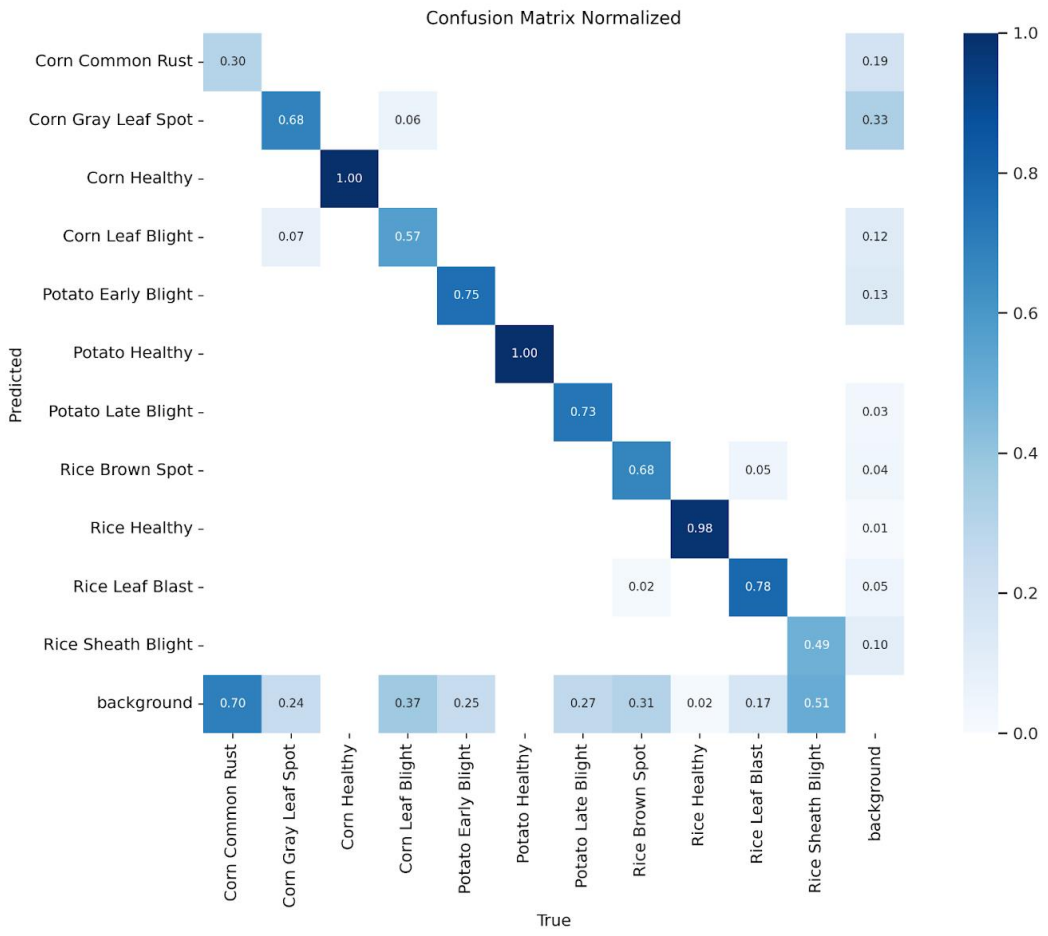

Figure 4.11: Confusion Matrix YOLOv9c

From the above confusion matrix YOLOv8l performs best among others. But, 2 classes that are performing poorly are “Corn Common Rust” and “Rice Sheath Blight”. In the PR curve section I’ve discovered that these 2 classes are performing worse than the other classes. Here also we can see that their normalized confusion matrix is below 50% or close to 50%. Meanwhile almost all the other classes' confusion matrix value is above 70%..

4.7 Summary

This section demonstrates the results of different experiments and compares them to each other with various evaluation metrics. In the next discussion section will include the information about the fulfilled objective of the thesis.

©Daffodil International University

CHAPTER 5

Discussion

5.1 Introduction

Discussion section provides the details of the objective that is fulfilled in this research so far.

5.2 Fulfilled Objective

To address the challenge of crop disease detection in Bangladesh, I have created a novel, Bangladesh-specific annotated dataset for segmentation tasks. This comprehensive dataset encompasses all the key crop diseases impacting Bangladeshi agriculture. To ensure its quality, I meticulously hand-annotated approximately 3900 images.

I have not only completed around 3900 manual annotations but also proposed a unique way of annotating images through semi automated annotation. With the help of a semi-automated annotation pipeline, I was able to annotate around 800 new images. The semi automated annotation pipeline shortened the time to annotate by 85.71% as compared to the manual annotation solution. One of the objectives of this research is to help to increase the annotation speed using a semi automated annotation pipeline. From the above result I can conclude that I've successfully been able to create a semi automated annotation pipeline which is able to reduce the total number of annotation time and increase the annotation speed for a human annotator.

©Daffodil International University

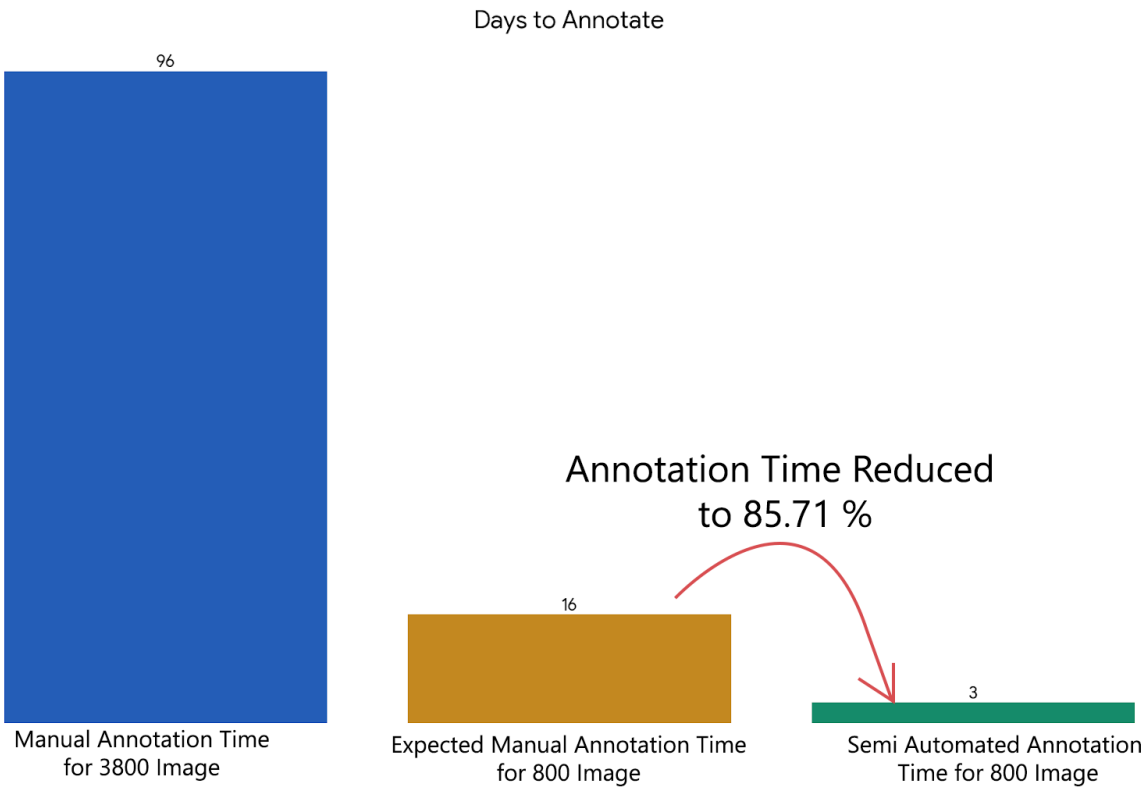

Figure 5.1: Data Annotation time reduced through Semi Automated Annotation

©Daffodil International University

In the end the object segmentation model performance was improved because of increasing the dataset size.

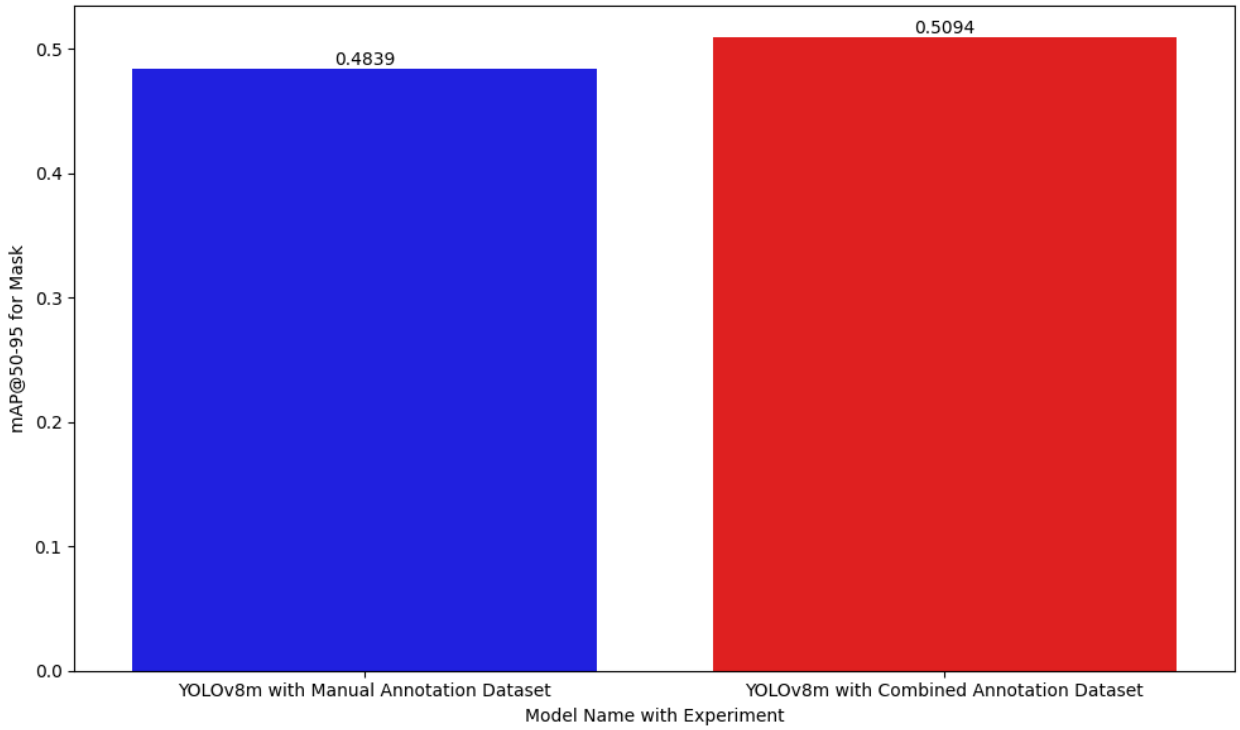

Figure 5.2: mAP@50-95 increased on Combined Dataset for YOLOv8m

5.3 Summary

This research was able to fulfill 2 of the most important objectives: a new annotated dataset and a novel approach to reduce the annotation time using a semi automated annotation pipeline. Finally, the object segmentation model performance was increased because of increasing the size of the dataset. The next conclusion section will include the information about overall research finding, limitations, future work and implications on personal and government level for the research.

©Daffodil International University

CHAPTER 6
Conclusion

6.1 Introduction

Conclusion section will cover the overall research findings from the research, the limitations of the research and future work based on the limitation. It will discuss the implication of the research on personal and governmental level. And, finally finally highlighting my contribution in this thesis.

6.2 Research Finding

From the above chapter we can conclude that the all the finding in this research are:

- A new dataset is being created for segmentation tasks containing around 4600 images over 10 distinct classes.
- A semi automated pipeline has been created to reduce the total annotation time. This semi automated pipeline was able to reduce the total annotation time by 85%
- Finally, the model performance for YOLOv8m, YOLOv8l and YOLOv9c have increased significantly after increasing the dataset size.

6.3 Contribution

While The performance of deep learning models are highly reliant upon abundant, high-quality data. This study aims to solve this problem by building an extensive dataset concentrating only on leaf diseases pertaining to rice, potato, and corn plants which are essential constituents of the agriculture sector of Bangladesh. This dataset has been acquired by Manual annotation from close to 3900 images to include the spectrum of disease symptoms observed on these crops. It can be used as a base dataset for training and testing deep learning based instance segmentation models that will segment disease accurately.

Manual annotation of images for disease segmentation is a notoriously time-consuming and laborious process. Recognizing this bottleneck, this research introduces a novel semi-automated

©Daffodil International University

annotation pipeline specifically designed for Bangladeshi crops. This innovative pipeline leverages various techniques to expedite the annotation process. The findings of this research demonstrate that the pipeline successfully reduces annotation time by an impressive 85%, significantly improving the efficiency of data creation and paving the way for the development of larger and more robust datasets in the future.

6.4 Research Limitation

While conducting the research there are few limitations that were evident. Those limitations were:

- The images that are collected from secondary sources (PlantVillage and Dhan-somadhan) are mainly indoor images. Indoor images don't represent real life scenarios for crop diseases.
- The “Semi-Automated-Annotation-Pipeline” only works with one specific tool and that is VGG Image Annotation.
- The dataset contains annotations for only 10 different crop diseases. Even though other crop disease images exist, those crop disease images are not annotated and not suited for segmentation tasks.

6.5 Future Work

Future will focus on resolving all the limitations that the research has. Those are:

- Collect outdoor images of diseased crops.
- Expand the usage of “Semi-Automated-Annotation-Pipeline” for more image annotation tools not only limiting to VGG Image annotation tool.
- Expand the class number by annotating more images from different classes with the help of “Semi-Automated-Annotation-Pipeline”.

©Daffodil International University

6.6 Implication

On a personal level, the development of this semi-automated annotation tool empowers human annotators and computer vision engineers. The tool tackles the time-consuming and tedious task of manual image annotation, significantly reducing the overall workload. This translates to increased efficiency, allowing researchers and engineers to dedicate more time to analysis, model development, and other critical tasks.

On a government level, this research presents a significant opportunity to empower Bangladeshi agriculture through a segmentation model for crop disease identification. By leveraging the Bangladesh-specific annotated dataset and the segmentation model's capabilities, government agencies can develop robust and user-friendly platforms for farmers. These platforms could integrate the segmentation model, allowing farmers to upload images of their crops. The model would then analyze the image, segmenting the diseased areas, and providing an accurate identification of the specific crop disease. Following this identification, the platform could offer readily available information on treatment solutions, best practices, and potential preventative measures.

6.7 Summary

In this chapter it covers all the research findings such as new dataset, a semi-automated-annotation pipeline and model’s performance. This chapter also covers the limitations of the research such as lack of outdoor image data, semi automated annotation is limited to only a single type of annotation tool and no annotation images for many different classes. In the future work it covers how the limitations can be overcome by collecting outdoor images, support for various annotation tools for semi automated annotation pipeline and annotating new classes so that the number of classes can increase. It also shows my contribution in the research by showcasing a new annotation dataset for the crop disease segmentation task and a novel semi automated annotation pipeline to reduce annotation time. Finally, it covers that for personal implications computer vision engineers or human annotators can reduce annotation time using semi automated annotation and for governmental implications a complete end to end website can help a farmer to identify crop disease and help him to navigate the solution for that particular disease.

©Daffodil International University

CHAPTER 7
REFERECES

1. Elangovan, K., Sivakumar, R., & Dhanasekaran, R. (2020). Plant disease classification using image segmentation and SVM techniques. International Journal of Computational Intelligence and Applications, 19(4), 2040003. https://doi.org/10.1142/S1469026820400034
2. Haque, M. E., Hossain, M. A., & Rahman, M. M. (2021). Rice leaf disease classification & detection using YOLO v5. International Journal of Computer Applications, 183(3), 0975-8887. https://doi.org/10.5120/ijca2021921261
3. Jakjoud, F., Sennoun, S., & Essaaidi, M. (2020). Deep learning application for plant diseases detection. International Journal of Advanced Computer Science and Applications, 11(5), 407-414. https://doi.org/10.14569/IJACSA.2020.0110550
4. Sharma, P., Singh, V., & Kaur, R. (2019). Performance analysis of deep learning CNN models for disease detection in plants using image segmentation. Procedia Computer Science, 167, 987-996. https://doi.org/10.1016/j.procs.2019.10.087
5. Shoaib, M., Abbas, N., & Rahman, S. U. (2021). A deep learning-based model for plant lesion segmentation, subtype identification, and survival probability estimation. Plant Methods, 17(1), 1-12. https://doi.org/10.1186/s13007-021-00789-2
6. Shoaib, M., Khan, M. A., & Rahman, S. U. (2020). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. Computers and Electronics in Agriculture, 170, 105247. https://doi.org/10.1016/j.compag.2020.105247
7. Singh, V., Misra, A. K., & Sharma, P. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. Information Processing in Agriculture, 4(1), 41-49. https://doi.org/10.1016/j.inpa.2016.10.005
8. Hughes, D. P., & Salathé, M. (2015). PlantVillage dataset. Retrieved from https://www.kaggle.com/emmarex/plantdisease

©Daffodil International University

9. Hossain, M. F., Sarker, M. A., & Amin, M. R. (2020). Dhan-Shomadhan: A dataset of rice leaf disease classification for Bangladeshi local rice. Mendeley Data. https://doi.org/10.17632/znsxdctwtt.1
10. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2022). YOLOv9: Learning what you want to learn using programmable gradient information. arXiv preprint arXiv:2209.01579. Retrieved from https://arxiv.org/abs/2209.01579
11. Ultralytics. (n.d.). Ultralytics YOLOv8. Retrieved from https://github.com/ultralytics
12. VGG Image Annotator (VIA). (n.d.). Retrieved from https://www.robots.ox.ac.uk/~vgg/software/via/

©Daffodil International University

202-35-3112

ORIGINALITY REPORT

14%	11%	8%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	Submitted to Higher Education Commission Pakistan Student Paper	1%
3	dspace.daffodilvarsity.edu.bd:8080 Internet Source	1%
4	www.mdpi.com Internet Source	1%
5	www.frontiersin.org Internet Source	1%
6	Submitted to University of Bristol Student Paper	<1%
7	wiredspace.wits.ac.za Internet Source	<1%
8	arxiv.org Internet Source	<1%
9	journal.magisz.org Internet Source	<1%