Plant leaf disease detection using YOLOv8

Asif Karim
Roll: 210602

Supervisor: Assistant Professor Tarun Debnath

A Project paper submitted to
the Department of Information and Communication Engineering at
Pabna University of Science and Technology
in Partial Fulfillment of the Requirements for Project Course of 3rd year 2nd semester
Bachelor of Science in Engineering
in
Information and Communication Engineering

Pabna University of Science and Technology, Bangladesh

July 2025

I want to cordially thank the supervisor
for giving valuable advice and direction

Acknowledgements

I would like to convey my heartfelt gratitude to Assistant Professor Tarun Debnath sir my project advisor, for his unfaltering support, encouragement, kindness, and guidance. He has given me lots of freedom and sometimes gentle pushes to continue in the right direction during the entire research work. I could not have imagined having a better advisor and mentor like him.

Abstract

Agricultural development plays a vital role in the economic growth of a nation, with the prosperity of many countries closely tied to the productivity of their agricultural sectors. Plant diseases pose a significant threat to crop yields, resulting in millions of dollars in losses annually. Early detection and timely intervention are essential to mitigate these damages and promote sustainable farming practices.

This study presents the development of an Android-based mobile application aimed at the early detection of plant leaf diseases. The application uses image processing and machine learning techniques to analyze leaf images and accurately identify disease symptoms. In addition to diagnosis, the platform enables agricultural experts to access farmers disease reports and geographic data, allowing them to organize field visits more efficiently and provide targeted support. This system enhances communication between farmers and agriculturalists, improving disease management and decision-making in the field. This study used publicly available dataset PlantDoc[7] ,PlantVillage [6] and PlantSeg [9] dataset. The study showed limitation of PlantVillage [6] dataset that causes overfitting.

0.1 Introduction

Agriculture plays a vital role in the economic and social development of many countries, especially those that rely heavily on agriculture for food production, employment, and trade. One of the critical challenges faced by the agricultural sector is the widespread occurrence of plant diseases, which can severely reduce crop yield and quality. These diseases, often caused by fungi, bacteria, or viruses, are responsible for millions of dollars in losses each year. Early detection and timely intervention are essential to prevent disease spread and reduce economic impact.

With advancements in deep learning and mobile computing, intelligent solutions can now be deployed directly in the hands of farmers. In this study, we develop an Android-based mobile application for the early detection of plant leaf diseases. The app leverages a deep learning model based on the YOLOv8n architecture. Specifically, the early convolution layers of YOLOv8n(backbone) are used as a feature extractor, and a custom classification head is built to identify 27 distinct plant disease classes.

To facilitate real-world deployment, the trained model is integrated into a mobile app that not only performs real-time disease detection but also logs farmer-reported cases into a centralized system. Agricultural experts can access these data to monitor disease spread geographically and schedule field visits more efficiently. This organized, technology-driven approach improves disease management, enhances communication between farmers and agriculturalists, and contributes to smarter, more sustainable agriculture.

0.2 Literature Review

There is a significant amount of work done.The study of [5] used publicly available Plant Village dataset[6] and achieved 99% accuracy using CNN.But all the image of the Plant village [6] dataset is captured in same style. So the model suffers from overfitting.The study conducted by [9] shows how the PlantVillage dataset created in-lab is not usable for in-world predictions.The study highlighted that data created for PlantVillage is in lab environment.It showed that the dataset only contains image of a specific resolution which emphasized the fact that the dataset is created in laboratory. It has single leaf with a uniform background in each of the sample of the dataset.

[1] used CNN, YOLOV5, Inception model which gave 97 to 98.75 % accuracy. They used a dataset containing 3150 images of infected tree leaf, those images have class of the disease and annotation of infected area.The used dataset had 5 classes of Soybean crop .It used preprocessing technique like flipping, rotation, scaling, adding noise. The study shows that YOLOV5 can be used to make real time application or monitoring system as it gave high accuracy with a compact model size. But the limitation of the study is that it only trained on Soybean crop. In the study of [2], an effective method to recognize rice disease based on color, shape, size of rice leaves. It binarized the image using Otsu's global threshold to reduce background noises. It helped the model recognize disease based on color, shape, size of rice leaves. The used dataset consist of 4000 healthy and 4000 infected sample. The study used CNN model which achieved 99.7% of accuracy on the dataset. The study of [3] presented an approach to detect tomato leaf disease. It used a publicly available dataset along with the country firm dataset. It used Generative Adversarial Networks (GANs) to generate synthetic dataset to reduce overfitting.They got 99% accuracy both in training and test data. The authors of [4] have focused on the issue and present a DL-based technique for the disease detection and classification in maize crops. Furthermore, the developed technique returns segmented images of affected leaves, allowing them to follow the disease spots on each leaf. A dataset of three maize crop diseases blight, sugarcane mosaic virus, and leaf spot was collected from the University Research Farm Koont, PMAS-AAUR, during different growth stages and under different weather conditions. The data was used to train different models for prediction, such as YOLOv3-tiny, YOLOv4, YOLOv5s, YOLOv7s, and YOLOv8n, with stated prediction accuracy of 69.40%, 97.50%, 88.23%, 93.30%, and 99.04%. The results showed that the YOLOv8n model outperformed the other models in terms of prediction accuracy.

0.3 Proposed Methodology

In this study a deep learning model is developed capable of classifying plant leaf diseases. The methodology involves dataset preparation, preprocessing, feature extraction using a YOLOv8 based architecture, model training, and deployment within an Android application. The detailed steps are as follows:

0.3.1 Dataset Collection

In this study 2 publicly available dataset PlantVillage [6] PlantDoc [7] and PlantSeg**??** dataset is used. PlantVillage [6] dataset available on Kaggle (Plant Village, 2020). This dataset contains over 54304 RGB images of healthy and diseased crop leaves, categorized into 38 different classes covering crops such as tomato, potato, and bell pepper. The dataset is split into 80% for training and 20% for validation, while a separate directory of 247 test images is created for real-world prediction evaluation.Some sample of PlantVillage dataset is shown in Figure 1.It can be clearly seen that all the image is captured in same style.The front surface of the image is captured putting onto a same background. In each of the image there is only single leaf. Those make it difficult for real world use. The PlantDoc [7] is also a publicly available dataset. It contains 2551 RGB image of 27 classes. The images are web scrapped and annotated using 300 human hours.As we can see from the Figure 2 that there are images that are captured from different angles,that have diversity in number of leaves and in other aspects. This diversity of the data set is useful for the real-world use case.So, the PlantDoc dataset is used to train the model. The PlantSeg [9] dataset which created for segmentation contains 11458 RGB images of 115 classes.The image of this dataset also created by web scrapping.The overview of the dataset is show below in Figure3 below. The study of [9] created a plot of image resolution of PlantVillage,PlantDoc,PlantSeg dataset. The plot shows that the PlantVillage dataset contains images of a single resolution. It suggests that all the captured image by a single person in controlled environment. The resolution distribution of PlantSeg has the highest variability. It suggests that the dataset is collected from an in-world environment. Therefore the dataset is more suitable train a model to predict on in-world image. The Plant doc dataset also has a great deal of variability. The resolution distribution is shown in the following Figure4. We added common data from PlantSeg to PlantDoc dataset.The PlantDoc dataset contains 2552 sample.After combining data from PlantSeg dataset, the total number of sample becomes 4630. The data distribution of various class is shown below in Table1,Table2,Table3

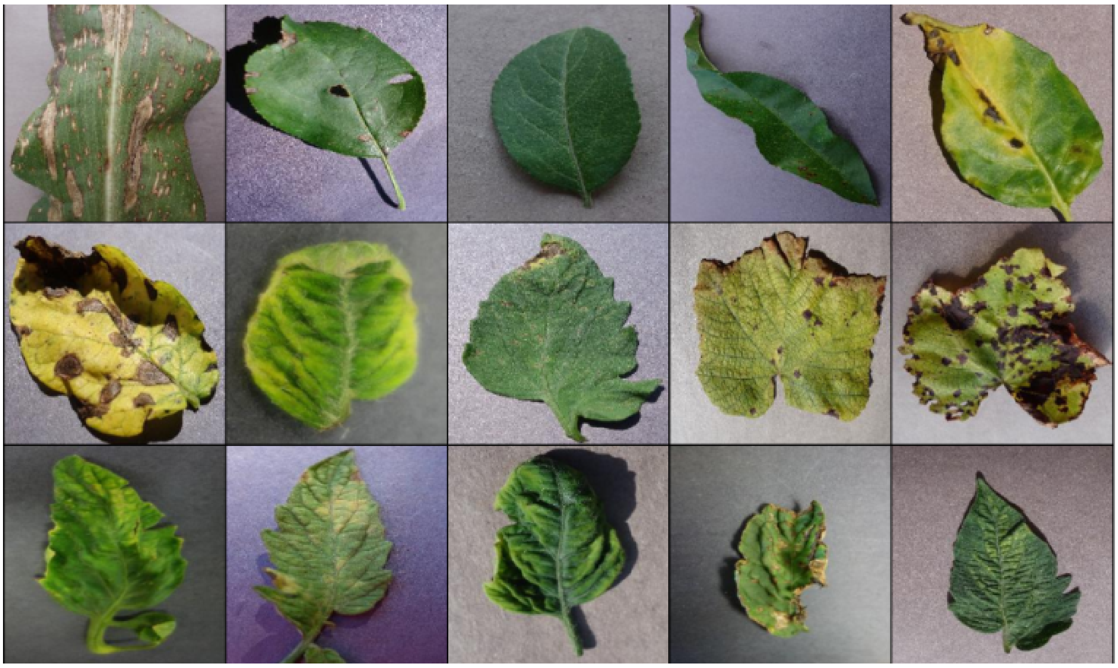

Figure 1: PlantVillage Dataset Sample

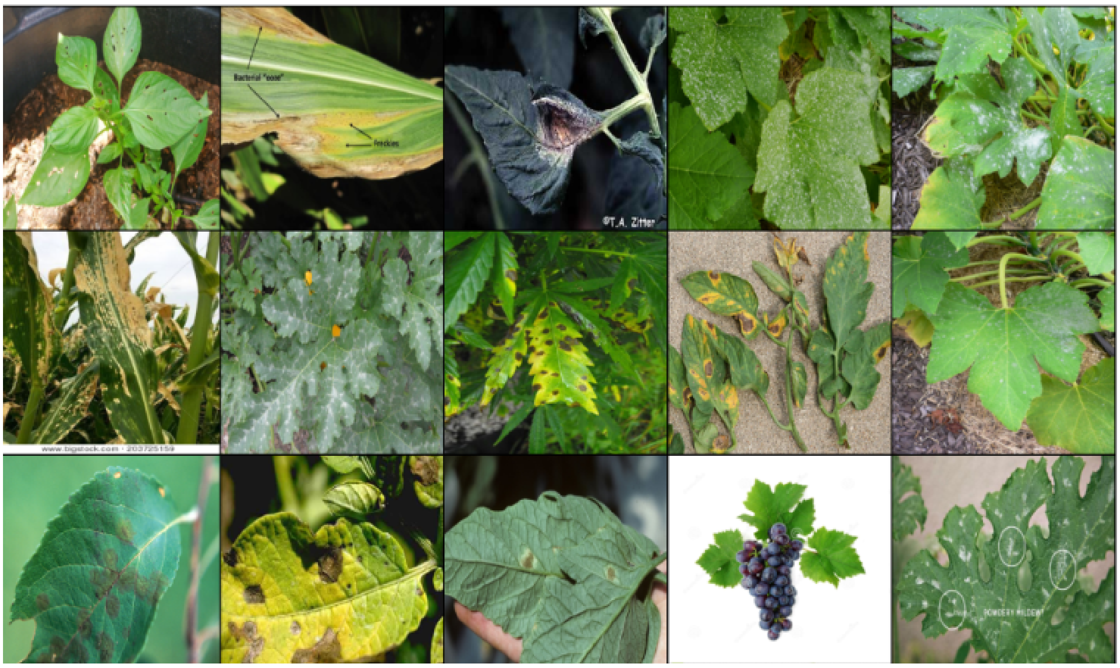

Figure 2: PlantDoc Dataset Sample

Figure 3: PlantSeg Dataset Sample

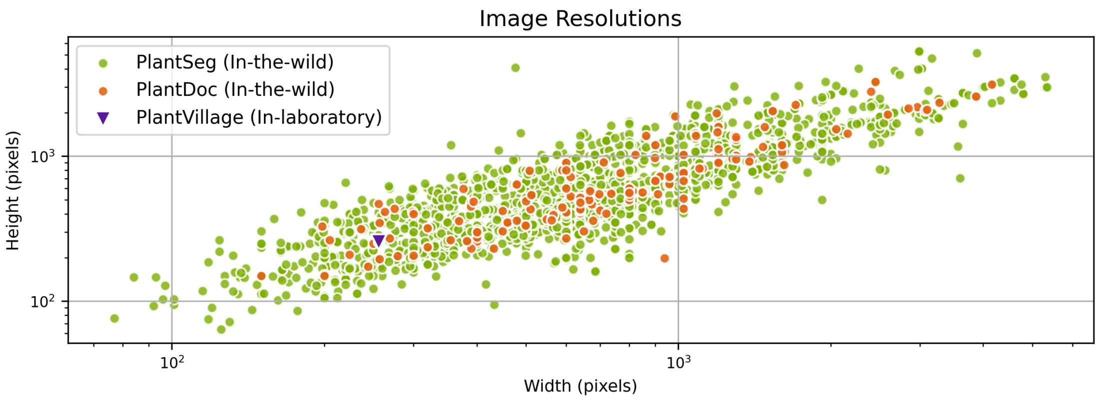

Figure 4: Resolution Distribution of PlantVillage, PlantDoc and PlantSeg Dataset

Table 1: Data distribution per class of training data

Class	**Number of Sample**
Apple Scab leaf	77
Apple leaf	82
Apple rust leaf	219
Bell pepper leaf	53
Bell pepper leaf spot	138
Blueberry leaf	105
Cherry leaf	47
Corn gray leaf spot	171
Corn leaf blight	310
Corn rust leaf	283
Peach leaf	103
Potato leaf early blight	235
Potato leaf late blight	214
Raspberry leaf	109
Soyabean leaf	57
Squash Powdery mildew leaf	305
Strawberry leaf	88
Tomato early blight leaf	261
Tomato septoria leaf spot	267
Tomato leaf	54
Tomato leaf bacterial spot	207
Tomato leaf late blight	264
Tomato leaf mosice virus	107
Tomato leaf yellow virus	162
Tomato mold leaf	241
Grape leaf	57
Grape leaf black rot	178

Table 2: Data distribution per class of validation data

Class	Number of Sample
Apple Scab leaf	8
Apple leaf	7
Apple rust leaf	8
Bell pepper leaf	6
Bell pepper leaf spot	7
Blueberry leaf	9
Cherry leaf	8
Corn gray leaf spot	2
Corn leaf blight	10
Corn rust leaf	8
Peach leaf	7
Potato leaf early blight	6
Potato leaf late blight	6
Raspberry leaf	5
Soyabean leaf	6
Squash Powdery mildew leaf	4
Strawberry leaf	6
Tomato early blight leaf	7
Tomato septoria leaf spot	9
Tomato leaf	6
Tomato leaf bacterial spot	7
Tomato leaf late blight	8
Tomato leaf mosice virus	8
Tomato leaf yellow virus	4
Tomato mold leaf	4
Grape leaf	10
Grape leaf black rot	6

Table 3: Data distribution per class of test data

Class	Number of Sample
Apple Scab leaf	2
Apple leaf	2
Apple rust leaf	2
Bell pepper leaf	2
Bell pepper leaf spot	2
Blueberry leaf	2
Cherry leaf	2
Corn gray leaf spot	2
Corn leaf blight	2
Corn rust leaf	2
Peach leaf	2
Potato leaf early blight	2
Potato leaf late blight	2
Raspberry leaf	2
Soyabean leaf	2
Squash Powdery mildew leaf	2
Strawberry leaf	2
Tomato early blight leaf	2
Tomato septoria leaf spot	2
Tomato leaf	2
Tomato leaf bacterial spot	2
Tomato leaf late blight	2
Tomato leaf mosice virus	2
Tomato leaf yellow virus	2
Tomato mold leaf	2
Grape leaf	2
Grape leaf black rot	2

0.3.2 Data preprocessing

The combined dataset has 4630 samples.This small number of image makes it difficult to use it for Deep Learning model.In this study the image sample are changed -30% to +60% brightness, contrast, saturation, rotated 30% to 60%, flipped vertically and horizontally at random.Thus 3 sample created for each image. Before training, the images are preprocessed to ensure consistency and enhance learning. The image is resized and normalized to standardized input dimensions and pixel range. The images are randomly flipped vertically/horizontally. The images are rotated in range +-30% at random. The brightness, contrast, saturation is changed at +-30%. The data data preprocessing is shown in Figure 5 below:

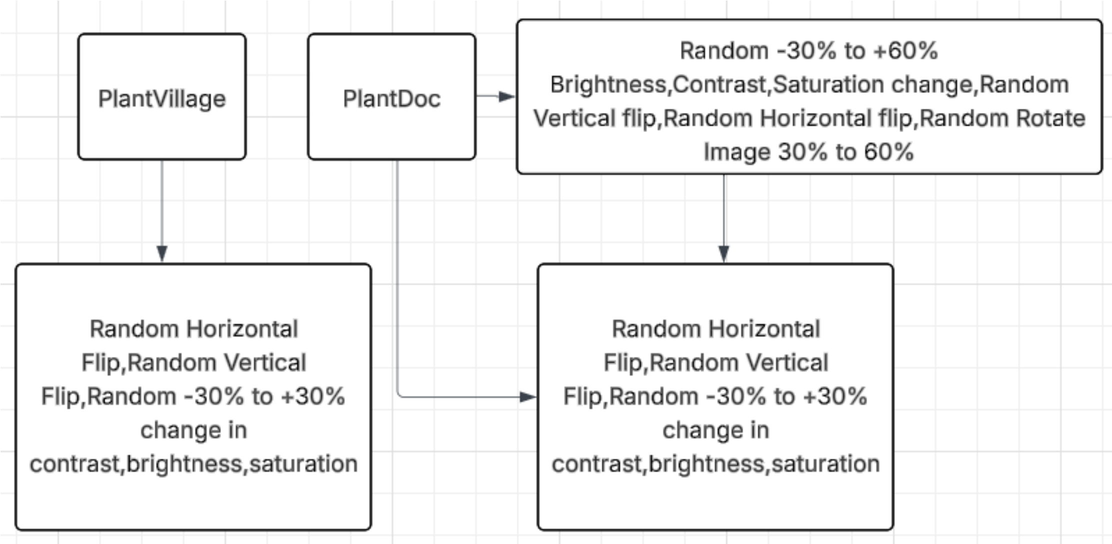

Figure 5: Data Preprocessing

0.3.3 Feature Extraction

In this study the backbone of YOLOv8n**??** is used to extract features from the input image.The first 10 layers of YOLOv8nYaseen is used to extract features.Following Figure 6 shows the object detection process of YOLOv8Yaseen.The first 10 layers of YOLOv8Yaseen returns (1,512,20,20) dimension features map form (1,640,640) dimension input image.

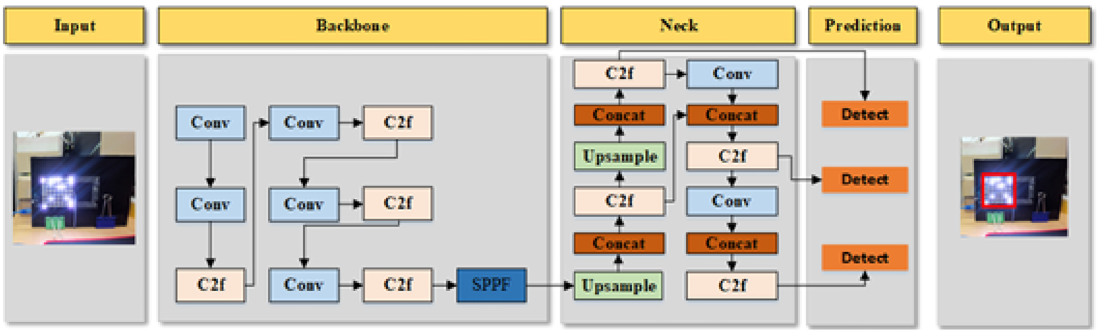

Figure 6: Object Detection Process of YOLO

0.3.4 Classification Head

A custom classification head is appended to the YOLOv8 backbone. The model structure consists of the following components:

Adaptive Max Pooling Layer

The layer is used to reduce the spatial dimensions of the feature maps to 1×1, making the model input-size independent. It compresses the high dimensional data into small dimension reducing the computational resource and training time.It dynamically specifies the mask size to return a fixed sized output.It returns the max of each mask.

Flattening Layer

The output of Adaptive Max Pooling is flatten.

Fully Connected Classifier

The flatten feature is passed to the fully connected classifier.It contains 2 branch of Linear-Relu layer.The first branch contains linear layer of higher dimension and second branch has lower dimension.Output of both the branch is concated and passed to output layer. The model block diagram is shown below in Figure 7

Optimization

Adam is used as optimizer. The learning rate is set to 1e-4 to reduce overfitting. Categorical Cross-Entropy is used as a loss function. The final trained model is then integrated into an Android application, where it is used for real-time detection. The app also stores predictions and sends data to a centralized system where agricultural experts can monitor disease outbreaks and visit farmers in a more organized and informed manner. The model architecture is shown in Figure 8

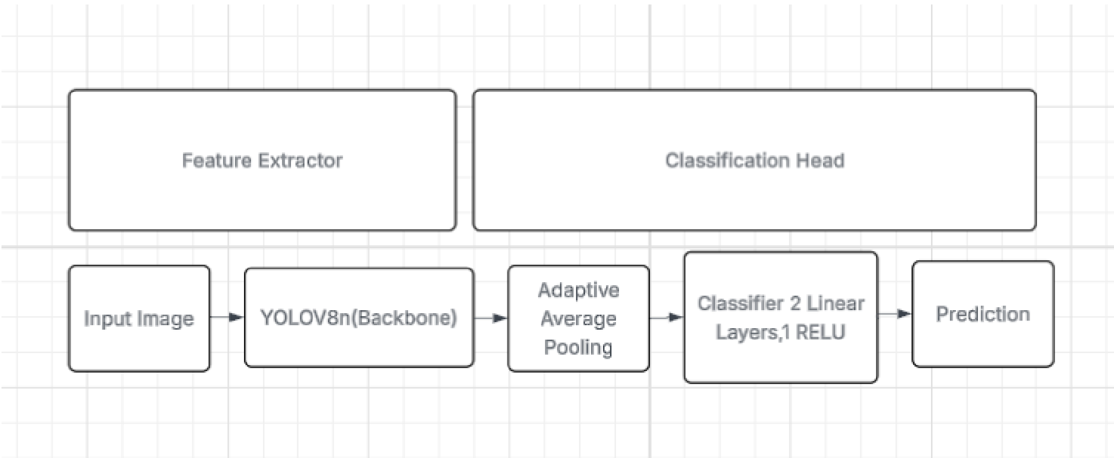

Figure 7: Model Block diagram

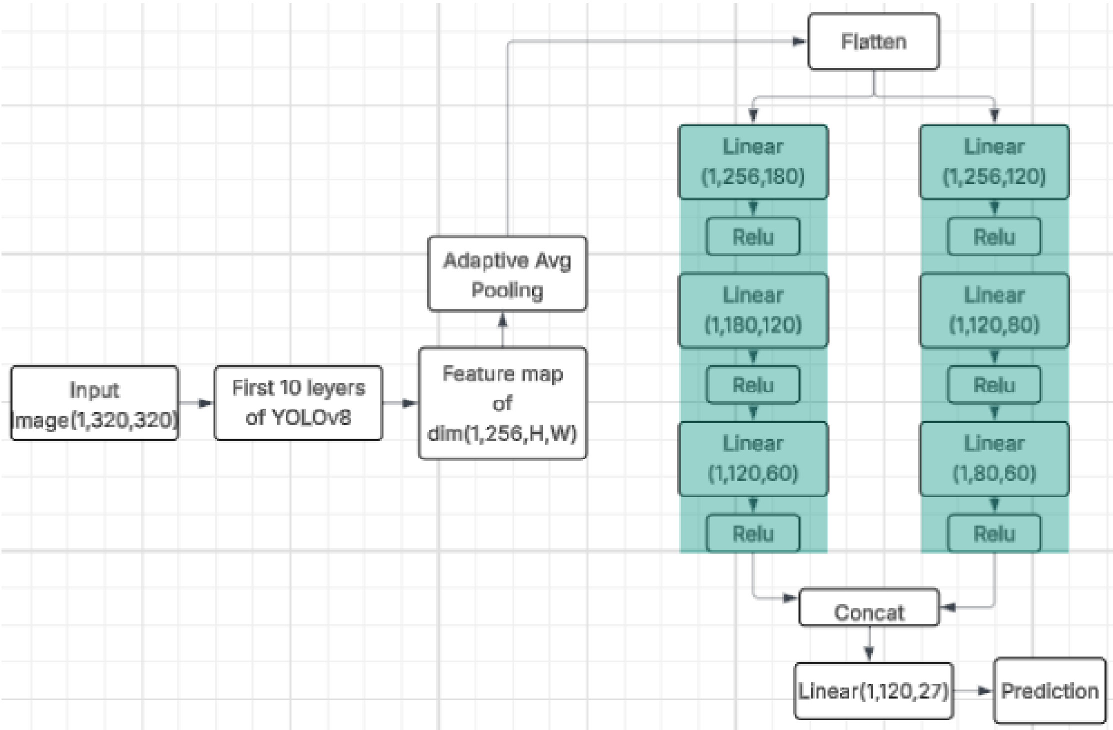

Figure 8: Model Architecture

0.4 Result and Discussion:

The model is trained with PlantVillage dataset.It gave 98.3% accuracy in training and 98.1% accuracy in validation.But when tested with test data from different dataset that contains in-world image its accuracy fell to 12.13%.This is because of overfitting. All the image have almost same background,same style as shown in Figure 1.The dataset contains 54305 images but have a very low variability.Also the image needed to be captured in different angle,different style as shown in PlantDoc dataset of Figure 2.

The training accuracy of various approach using PlantVillage dataset is shown below in Table 4:

Table 4: Training Using PlantVillage dataset

Technique	**Accuracy**
CNN	95.7%
YOLOV8 + Classification head	97.6%
YOLOV8 + Classification head + preprocessing	98.3%

The validation accuracy using PlantVillage dataset is shown below in Table 5 :

Table 5: Validation Using PlantVillage dataset

Technique	**Accuracy**
CNN	93.6%
YOLOV8 + Classification head	97.2%
YOLOV8 + Classification head + preprocessing	98.1%

The accuracy of the model in unkown test data using model trained with PlantVillage dataset is shown below in Table 6:

Table 6: Test of Model trained with PlantVillage dataset

Technique	**Accuracy**
CNN	10.15%
YOLOV8 + Classification head	11.67%
YOLOV8 + Classification head + Preprocessing	12.13%

The training vs validation loss per epoch is shown in for model trained using PlantVillage dataset Figure9.

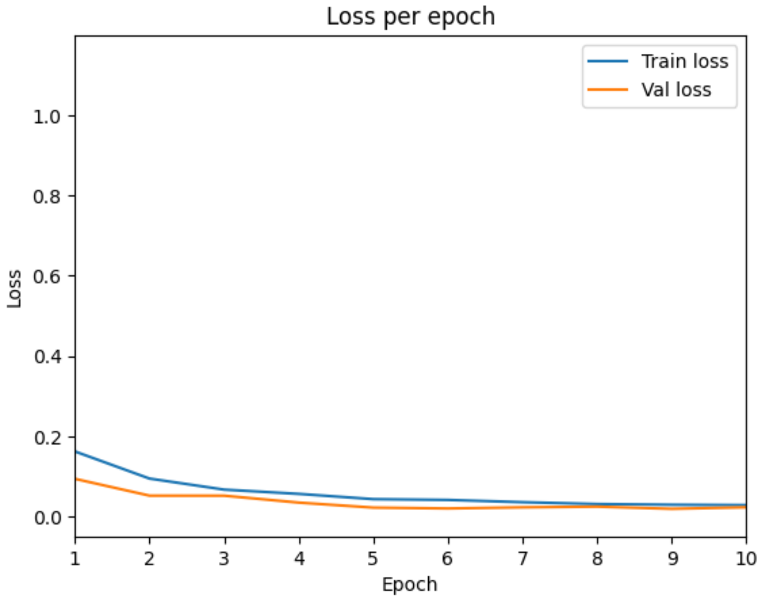

Figure 9: Training vs Validation Loss on PlantVillage Dataset

The training vs validation accuracy per epoch for model trained using PlantVillage dataset is shown in Figure10.

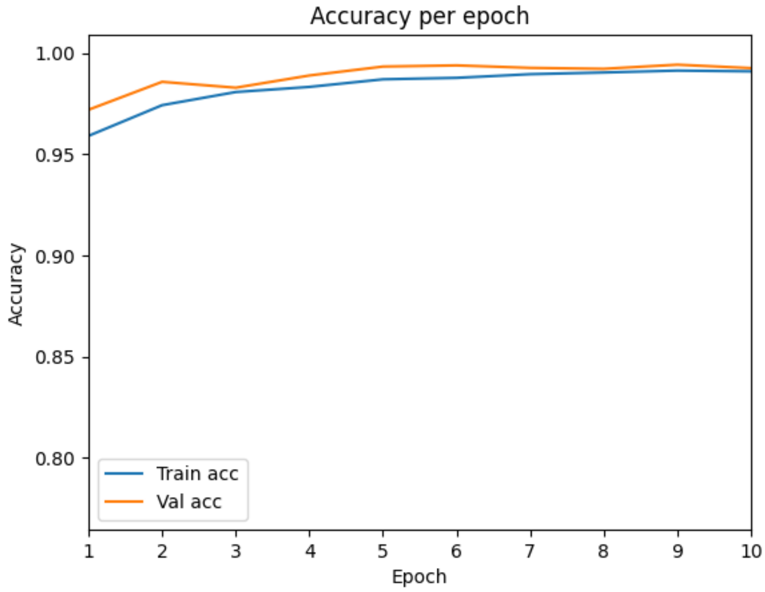

Figure 10: Training vs Validation Accuracy on Plant Village Dataset

The confusion matrix on the validation data for model trained using PlantVillage dataset shown in Figure 11:

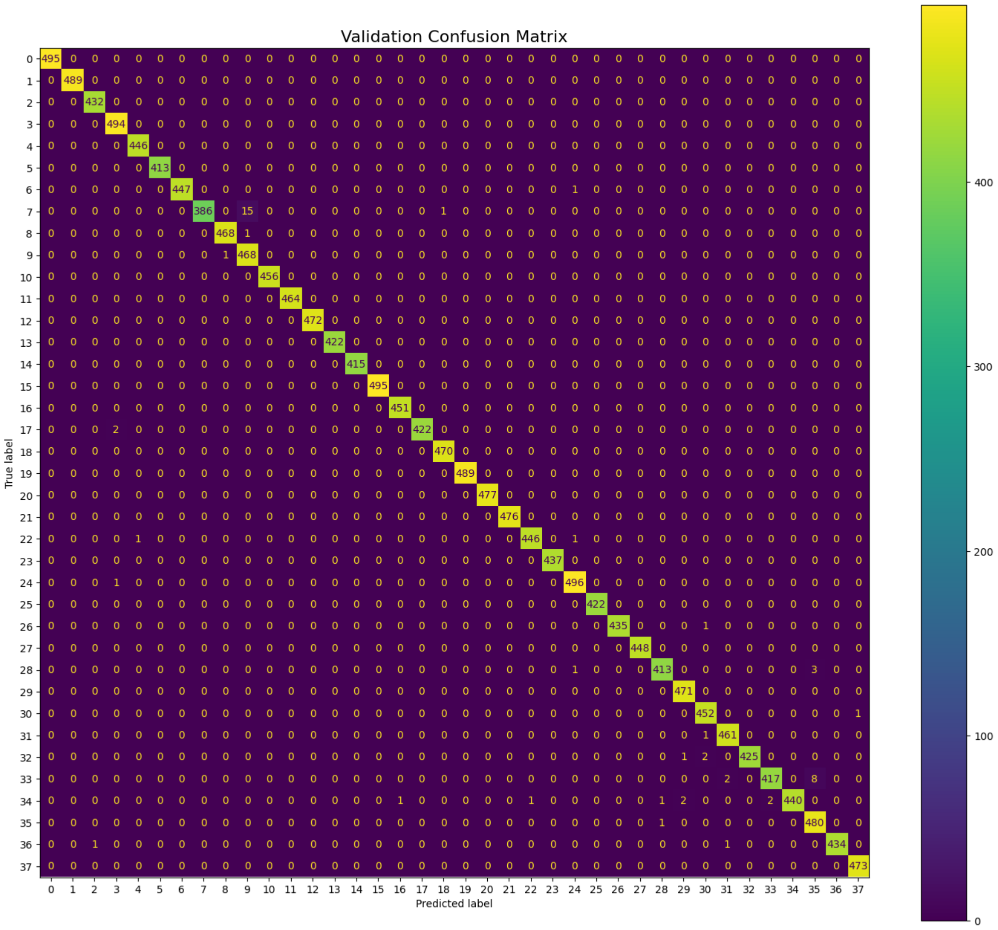

Figure 11: Confusion Matrix on Validation data of PlantVillage dataset

The confusion matrix on the test data for model trained using PlantVillage dataset is shown below in Figure 12:

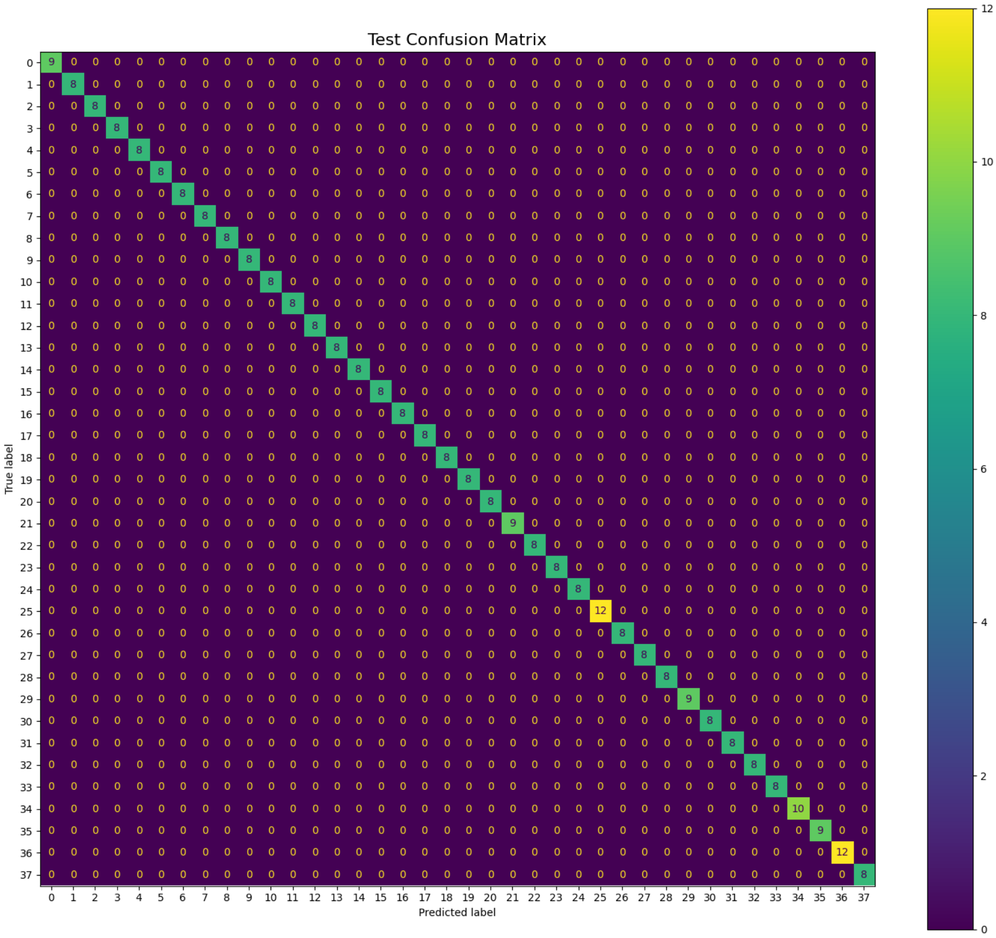

Figure 12: Confusion Matrix on test data of PlantVillage dataset

The training accuracy of various approaches using combination of PlantDoc and PlantSeg dataset is shown below in Table 7:

Table 7: Training Using combination of PlantDoc and PlantSeg dataset

Technique	**Accuracy**
CNN	85.21%
YOLOV8 + Classification head	79.2%
YOLOV8 + Classification head + preprocessing	82.3%

The validation accuracy using combination of PlantDoc and PlantSeg dataset is shown below in Table 8 :

Table 8: Validation Using combination of PlantDoc and PlantSeg dataset

Technique	Accuracy
CNN	59.23%
YOLOV8 + Classification head	52.43%
YOLOV8 + Classification head + preprocessing	54.38%

The accuracy of the model in unknown test data using model trained with combination of PlantDoc and PlantSeg dataset is shown below in Table 9:

Table 9: Test of Model trained with combination of PlantDoc and PlantSeg dataset

Technique	Accuracy
CNN	57.31%
YOLOV8 + Classification head	49.95%
YOLOV8 + Classification head + Preprocessing	53.7%

The training vs validation loss per epoch is shown in for model trained using combination of PlantDoc and PlantSeg dataset Figure 13.

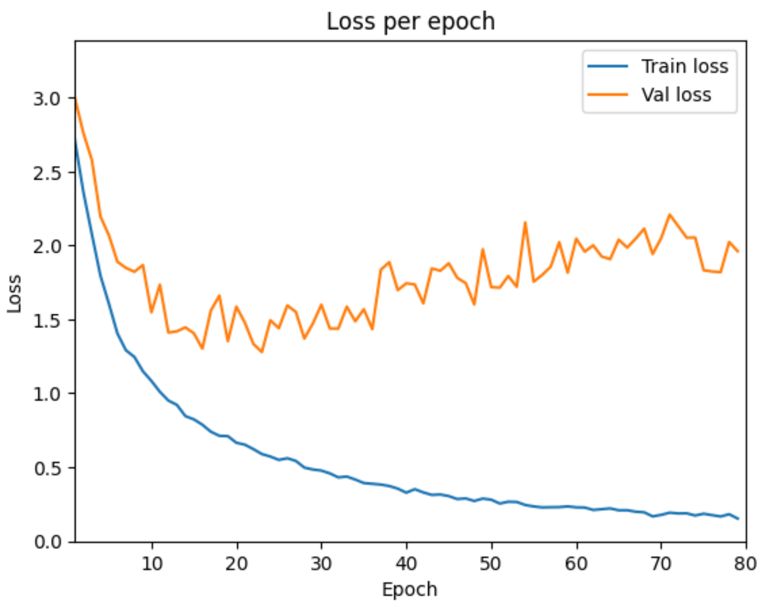

Figure 13: Training vs Validation Loss in Combination of PlantDoc and PlantSeg

The training vs validation accuracy per epoch for model trained using combination of PlantDoc and PlantSeg dataset is shown in Figure 14.

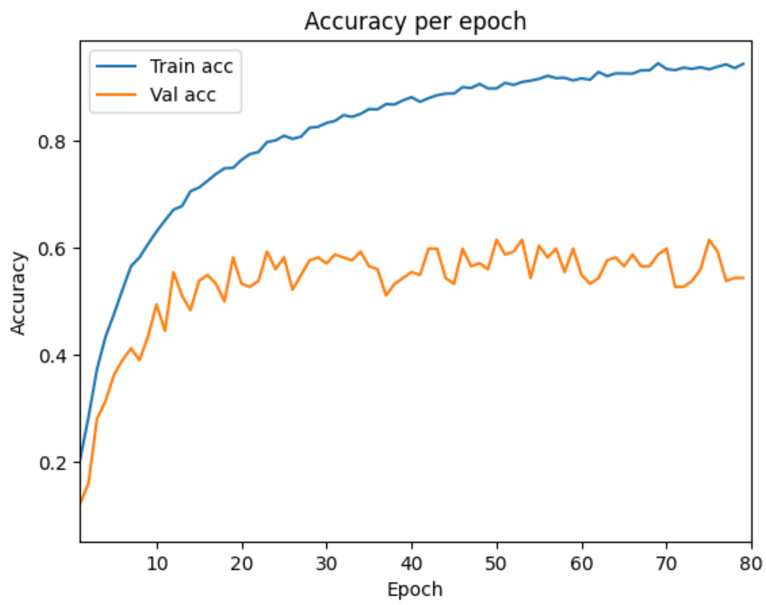

Figure 14: Training vs Validation Accuracy in Combination of PlantDoc and PlantSeg

The confusion matrix on the validation data for model trained using combination of PlantDoc and PlantSeg dataset shown in Figure 15:

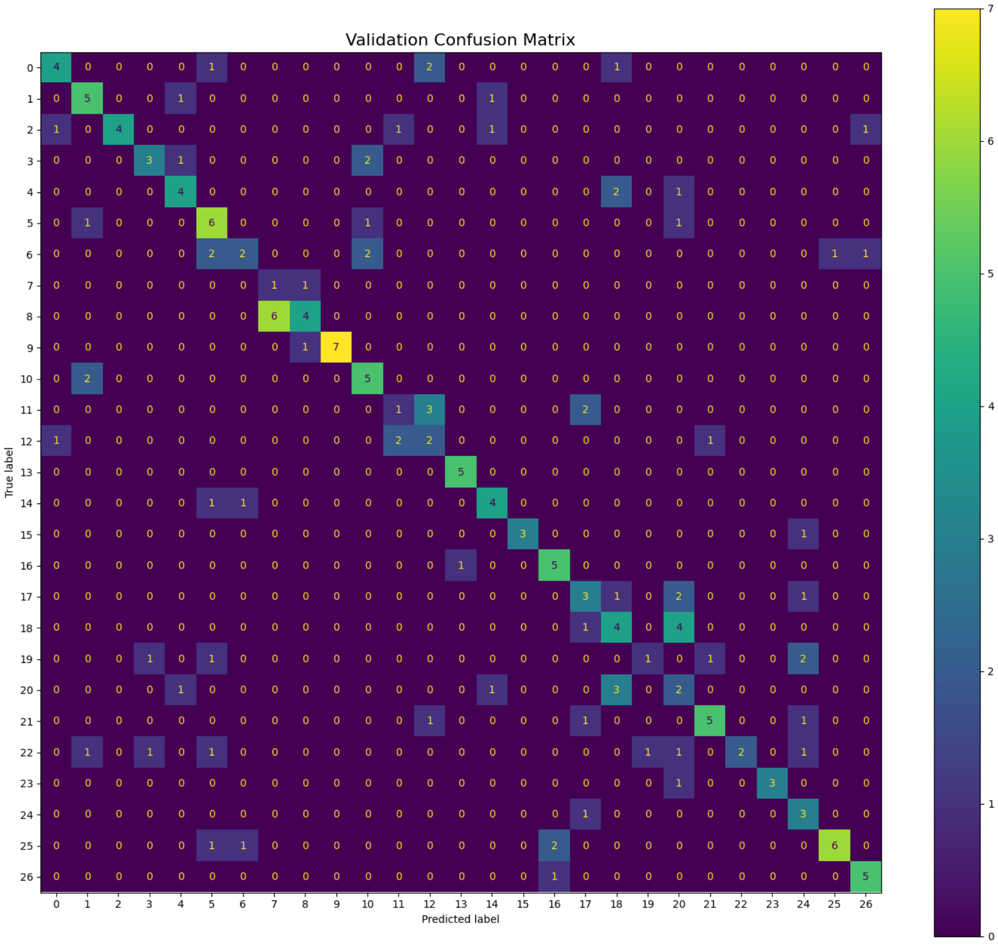

Figure 15: Confusion Matix on Validation data of Combination of PlantDoc and PlantSeg

The confusion matrix on the test data for model trained using combination of PlantDoc and PlantSeg dataset is shown below in Figure 16:

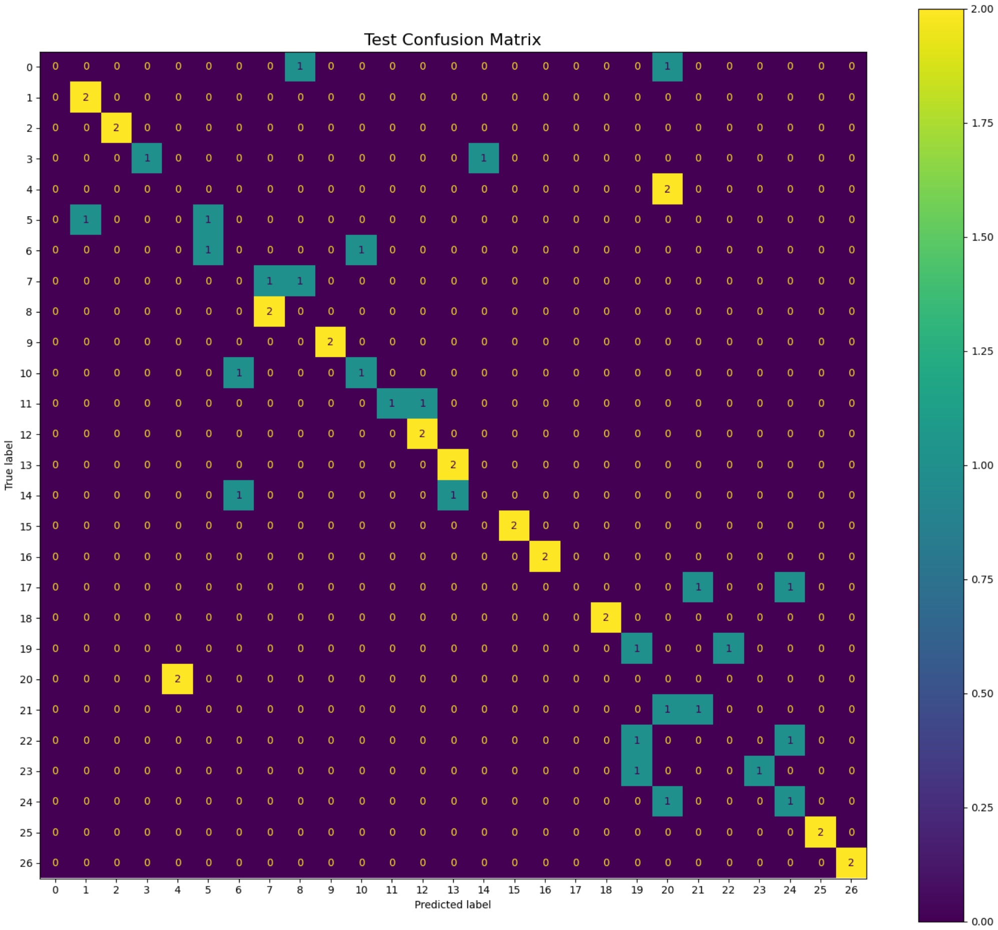

Figure 16: Confusion Matrix on test data of Combination of PlantDoc and PlantSeg

From the result of model trained on PlantVillage dataset it can be seen PlantVillage dataset gave 98.3% accuracy in training .But gave 12.13% accuracy in test.It is for making the dataset in lab environment.The dataset contain only single leaf image with same background. Observing the result of model trained on combination of PlantDoc and PlantSeg dataset we can see the model reached 86.21% accuracy in training. Got 54.38% accuracy in validation and 53.7% accuracy on test.It can be seen the combination of PlantDoc and PlantSeg prformed better than PlantVillage on test data. But the accuracy is not enough for the practical use of the model.There is only 4360 samples to predict 27 classes.It is not sufficient.Atleast 1000 sample needed per class to get good prediction accuracy.

0.5 Android App

An android app is developed so that farmer can easily detect and take preventive measures as early as possible. The app is designed to work both in online and offline mode. Offline mode is introduced so that the user can access the app without net connection.

The offline mode is shown below in Figure 17:

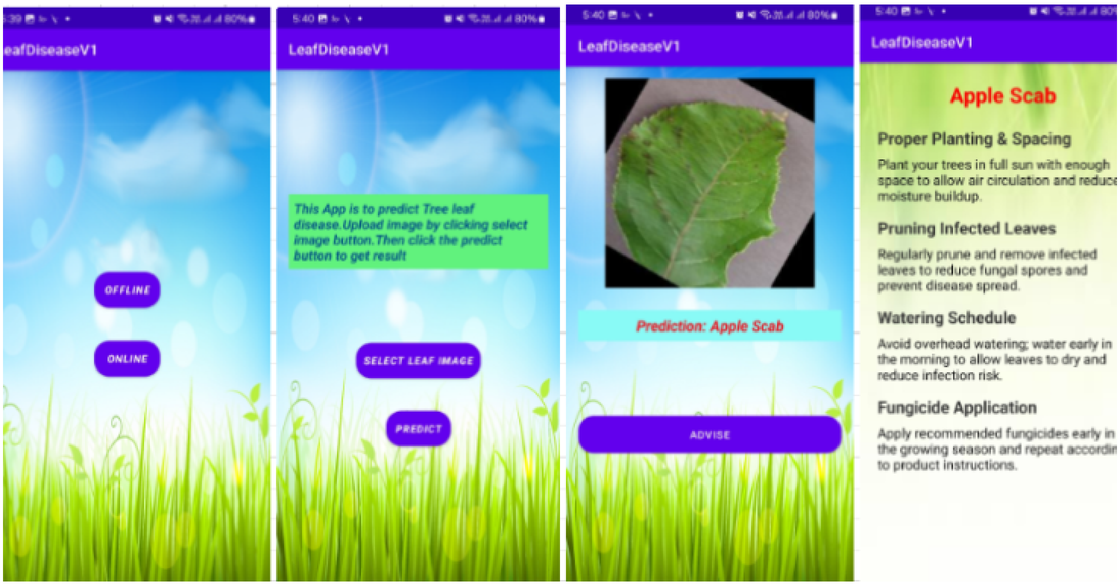

Figure 17: Android app in offline mode

In online mode the user can create their account using email. Once the account is created the predicted disease will be store in their profile. Also it will be sent to the central system. The online mode is shown below in Figure 18 and 19:

In Figure 18 the procedure of user registration is shown. In Figure 19 a disease is predicted. The user can click the Advice button to get advice about the disease. The information also stored in user's profile. Clicking profile button of Figure 18 a user can see all detected disease. Also, a button is shown for each of the detected disease. Clicking the button user will also get advice about the respective disease.

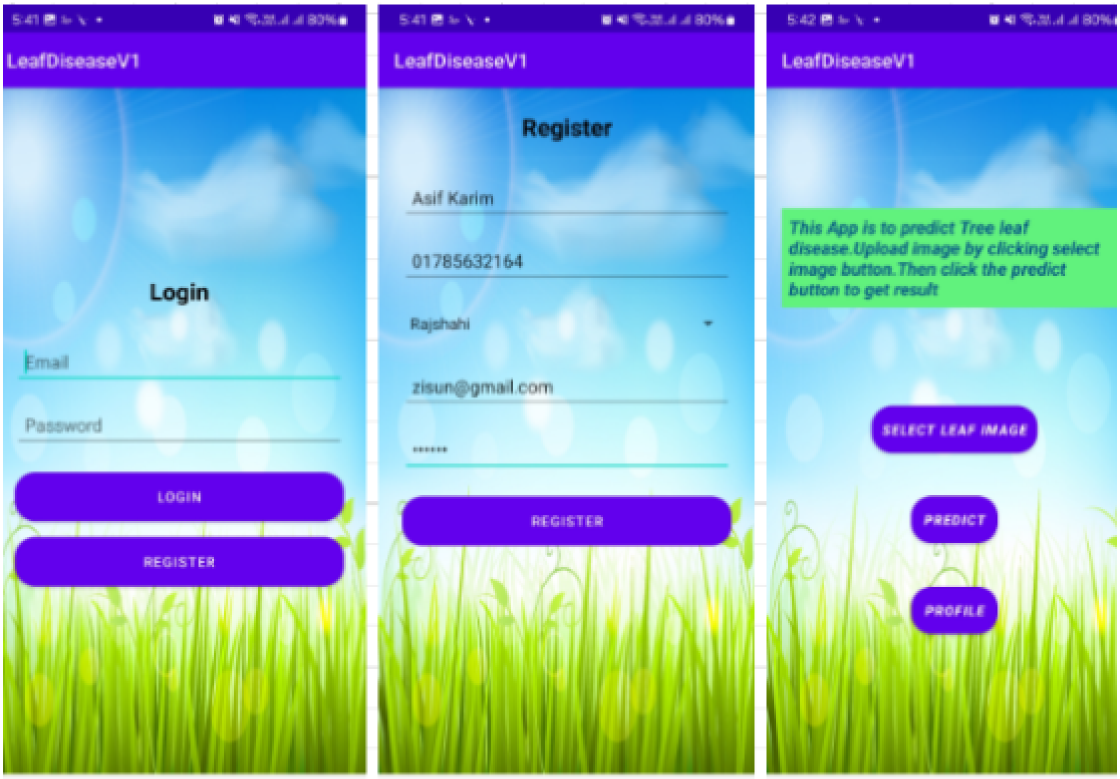

Figure 18: Android app in online mode

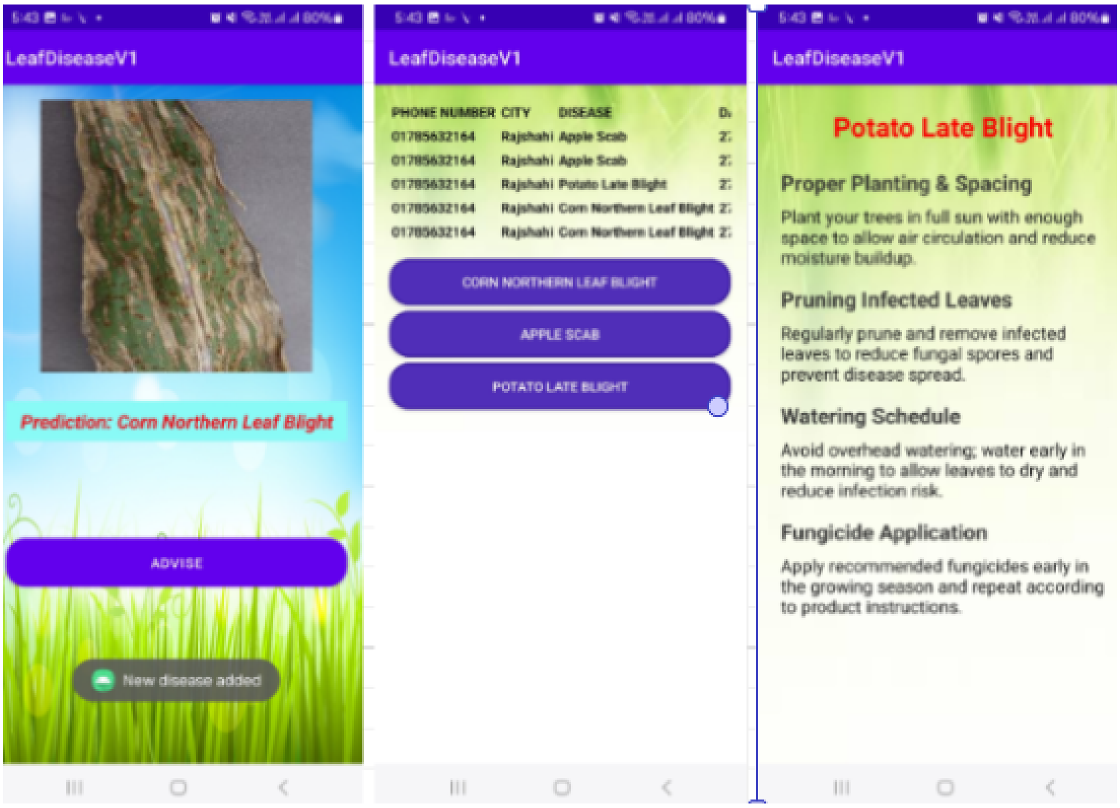

Figure 19: Android app in online mode

Admin can login to the app and see all the information. All the information of past detected disease is shown in the left side image of Figure 20. In the left side image diseases information is divided geographically so that admin easily find out which area is more sensitive and need immediate care.

Figure 20: Admin panel

0.6 Limitations and Future Research

There is a limitation of publicly available dataset.The limitation of PlantVillage dataset is described already.For real time application versatile dataset like PlanetDoc dataset is more suitable.The dataset should contains image captured in natural environment by various device.But its have a very few samples to train a Deep learning model.To train a deep learning model we need adequate dataset.Also publicly available dataset not contains all country plant of Bangladesh.A dataset can be creaded for use in Bangladesh

Bibliography

[1] Divyanshu Tirkey and Kshitiz Kumar Singh and Shrivishal Tripathi , "Performance analysis of AI-based solutions for crop disease identification, detection, and classification "

[2] Upadhyay SK and Kumar A., "A novel approach for rice plant diseases classification with deep convolutional neural network.", Int J Inf Technol, vol.v14, pp. 185–99. , 2022.

[3] Guerrero-Ibañez A and Reyes-Muñoz A., "Monitoring tomato leaf disease through convolutional neural networks.", Electron, 2023.

[4] Khan F and Zafar N and Tahir MN and Aqib M and Waheed H and Haroon Z., "A mobile-based system for maize plant leaf disease detection and classification using deep learning", Front Plant Sci, 2023.

[5] Md. Jalal Uddin Chowdhury and Zumana Islam Mou and Rezwana Afrin and Shafkat Kibria , "Plant Leaf Disease Detection and Classification Using Deep Learning: A Review and A Proposed System on Bangladesh's Perspective"

[6] D.P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics"

[7] Davinder Singh and Naman Jain and Pranjali Jain and Pratik Kayal and Sudhakar Kumawat and Nipun Batra, "PlantDoc: A Dataset for Visual Plant Disease Detection"

[8] Muhammad Yaseen , "WHAT IS YOLOV8: AN IN-DEPTH EXPLORATION OF THE INTERNAL FEATURES OF THE NEXT-GENERATION OBJECT DETECTOR"

[9] Tianqi Wei and Xin Yu and Paul Melloy and Zhi Chen and Scott Chapman and Zi Huang, “PlantSeg: A Large-Scale In-the-wild Dataset for Plant Disease Segmentation”