A

MAJOR PROJECT-II REPORT

on

A Deep Learning Approach For Rice Leaf Disease Detection

Submitted by:

Anish Borkar (210220)

D Veera Harsha Vardhan Reddy (210258)

Vandamasu Sai Sumanth (210257)

Nichenametla Karthik Raja (210371)

Godavarthi Sai Nikhil (210214)

under mentorship of

Dr. Manisha Saini (Assistant Professor)

Dr. Himanshu Upreti (Assistant Professor)

Department of Computer Science Engineering

School of Engineering and Technology

BML MUNJAL UNIVERSITY, GURUGRAM (INDIA)

May 2024

CANDIDATE'S DECLARATION

I hereby certify that the work on the project entitled, **"A DEEP LEARNING APPROACH FOR RICE LEAF DISEASE DETECTION",** in partial fulfillment of requirements for the award of Degree of **Bachelor of Technology** in School of Engineering and Technology at BML Munjal University, having University Roll Nos. **210220, 210258, 210257, 210371, 210214** is an authentic record of our work carried out during a period from **Jan 2024 to May 2024** under the supervision of **Dr. Manisha Saini** and **Dr. Himanshu Upreti**.

By-

Anish Borkar

D Veera Harsha Vardhan Reddy

Vandamasu Sai Sumanth

Nichenametla Karthik Raja

Godavarthi Sai Nikhil

SUPERVISOR'S DECLARATION

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Faculty Supervisor Name: Dr. Manisha Saini
Signature:

Faculty Supervisor Name: Dr. Himanshu Upreti
Signature:

ACKNOWLEDGEMENT

I am highly grateful to **Dr. Manisha Saini and Dr. Himanshu Upreti,** Assistant Professors at BML Munjal University, Gurugram, for providing supervision to carry out the Major Project III from February-May 2024.

Dr. Manisha Saini and Dr. Himanshu Upreti has provided great help in carrying out my work and is acknowledged with reverential thanks. Without wise counsel and able guidance, it would have been impossible to complete the project.

I extend my sincere thanks to **Dr. Manisha Saini and Dr. Himanshu Upreti** for their continuous encouragement throughout the project. I am also thankful to the entire team at BML Munjal University for their contributions. Additionally, I am grateful to my friends who generously dedicated their time and assistance, contributing significantly to the successful completion of our group project.

Anish Borkar
D Veera Harsha Vardhan Reddy
Vandamasu Sai Sumanth
N Karthik Raja
Godavarthi Sai Nikhil

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Full Form
Machine Learning	ML
Deep Learning	DL
Vision Transformer	ViT
You Only Look Once	YOLO
Support Vector Machine	SVM
Least Squares SVM	LS-SVM
K-Nearest Neighbors	KNN
Convolutional Neural Networks	CNN
User Interface	UI
Feed-Forward Network	FFN
Feature Pyramid Network	FPN
Pyramid Attention Network	PAN
Mean Squared Error	MSE
Mean Average Precision	mAP
Intersection over Union	IoU

Table of Contents

ABSTRACT

In the agricultural sector, the timely detection and management of crop diseases are crucial for ensuring food security and maximizing crop yields. This project presents "A Deep Learning Approach for Rice Leaf Disease Detection," aimed at addressing the challenge of identifying common diseases affecting rice plants. The focus diseases include bacterial leaf blight, brown spot, leaf blast, leaf scald, and narrow brown spot.

The project depends on DL techniques for both back-end disease detection and front-end user interaction. Using a live camera stream or from provided photographs, a full mobile application has been built with Flutter that allows users to identify diseases in rice leaves. When a disease is detected, the application gives information regarding it and from the type of disease the user can take appropriate courses of action suggested in the application as we planned to design in such a way the application contains all the information regarding it.

The report covers the methodology which we planned for disease detection, including the structure of the dataset on how we passed it to use for model training, the system architecture of the DL model, and the integration of the model with the mobile application. It also lists the components and materials needed to put the plan into practice.

1. INTRODUCTION

1.1 OVERVIEW

In many parts of the world, rice is a staple food that is full of vital nutrients. Billions of people around the world eat it. However, many kinds of diseases can impact rice fields, sometimes seriously impairing food security and significantly lowering productivity. The immediate and accurate identification of those diseases is essential for putting control measures in place and reducing harm to the crops.

That's where technology comes into the picture. DL and mobile application development work together to offer a powerful solution for rice leaf disease diagnosis. DL, a subfield of ML, allows algorithms to find complicated patterns in vast volumes of data. In this scenario, DL models can be trained using enormous datasets of images of rice leaves, both damaged and healthy. The models can then be integrated into mobile apps built with the popular Flutter framework for accessible app development.

The Rise of Mobile-Based Rice Leaf Disease Detection:

The Development of Rice Leaf Disease Detection using a Mobile application combining DL and Flutter offers numerous benefits. Field farmers can use smartphone applications to swiftly identify rice leaf diseases from the farms. DL models may diagnose diseases quickly and correctly, leading to better crop management. Furthermore, farmers with varied levels of technological experience may find it easy to use mobile applications with basic UI.

This plan might greatly impact the rice-growing sector by doing the following:

a) Improving food security through early disease detection and improved crop management.
b) Giving farmers easy access to diagnostic tools.
c) Supporting educated decision-making regarding disease prevention strategies.
d) Promoting sustainable agricultural practices by lowering reliance on use of chemicals.

The sections that follow go into more detail about this process and also look at developing a mobile application that detects rice leaf disease using DL and Flutter.

1.2 EXISTING SYSTEM

Traditionally, disease detection in crops, including rice, has heavily relied on manual inspection by agricultural experts. This approach is time-consuming, labor-intensive, and often prone to errors due to human subjectivity. While some automated systems for disease detection have been developed in recent years, they often lack accuracy, scalability, or accessibility for end-users.

Existing automated systems for rice disease detection typically involve image processing techniques combined with ML algorithms. However, these approaches may struggle with accurately distinguishing between different types of diseases or require extensive computational resources for real-time processing. Furthermore, the UI of these systems are often complex and not tailored to the needs of farmers or agricultural workers in the field.

1.3 USER REQUIREMENT ANALYSIS

Understanding the demands and preferences of end customers is critical when developing an effective and user-friendly solution. Farmers, agricultural extension agents, and other crop management stakeholders are the primary users of rice leaf disease detection. Surveys and research were conducted to identify the specific needs, issues, and preferences of target users and assemble their requirements.

This analysis identified important user needs, which include:

a) **Accuracy:** In order to take appropriate treatment activities, users want the system to consistently identify and classify distinct types of rice leaf infections.
b) **Accessibility:** The system must be available across several platforms and usable by users with varying levels of technical proficiency.
c) **Real-time Detection:** In order to decrease crop damage and promote fast response, users must be able to detect conditions in real-time.
d) **Easy-to-Use Interface:** The UI should be easy to use and straightforward, with clear instructions for taking and processing leaf photos.
e) **Offline Functionality:** Given the frequent lack of connectivity in rural agricultural areas, the system should include offline capabilities to ensure continuous use.

1.4 FEASIBILITY STUDY

Technical Feasibility

a) **Data Availability:** A wide range of photographs of rice leaf disease are readily available in public sources. Furthermore, we may be able to collaborate with nearby research centers or agriculture departments to create a more specialized dataset addressing local rice illnesses.

b) **Computational Resources:** Training DL models requires a significant amount of computational power. Nonetheless, university-based DL workstations and cloud-based

platforms such as Google Colab can provide the necessary resources for model development and training.

c) **Hardware Compatibility:** Currently available cameras and smartphones can capture high-resolution photos suitable for DL processing. Furthermore, using the appropriate libraries, the system may be configured to work with a variety of hardware combinations.

d) **Algorithm Suitability:** DL algorithms in Computer Vision have proven exceptional performance in image recognition tasks, making them an excellent choice for diagnosing rice leaf disease.

Market Feasibility

1. **Problem Significance:** Rice, an important crop, is badly endangered by a number of diseases. Early and precise sickness detection is critical to reducing production loss and allowing for appropriate management.

2. **Target Market:** This strategy benefits researchers, agricultural extension professionals, and farmers alike. It can help farmers detect infections early on, allowing them to apply tailored pesticides and take other essential procedures. Extension professionals can utilize it to conduct field evaluations and offer farmers further guidance. It can be used by researchers to track diseases and develop new control strategies.

3. **Competitive Landscape:** Although there may be commercial solutions available, there is an opportunity to produce an approachable, affordable, and potentially open-source solution that can be used by a larger group of people.

Operational Feasibility

a. **Project Team:** The five members of our team are well-versed in DL concepts and have finished the course. We can use the faculty's help throughout the assignment to increase our level of understanding and completion.

b. **Timeline:** If you have a clear project strategy, assign work efficiently, and have regular team contact, you can complete the project by the end of the semester.

2. LITERATURE REVIEW

Rice Leaf Disease Detection Using Machine Learning and Deep Learning

Recent study has looked into the use of DL and ML algorithms for automating rice leaf disease identification. These methods provide promising advantages such as objectivity, precision, and speed. This study looks at a number of research papers, and journals which explored different approaches to ML and DL for diagnosing rice leaf disease.

Machine Learning Techniques for Rice Leaf Disease Detection

Several studies have looked into the use of ML approaches for classifying rice leaf diseases. Typically, these research involve collecting a dataset of images of rice leaves that have been classified into several disease groups (for example, bacterial leaf blight, brown spot, and leaf smut). Following that, the images are pre-processed for consistency and quality. In order to discriminate between healthy and diseased leaves, relevant features are found using feature extraction techniques. Shapes, textures, and color characteristics are commonly used elements.

a) Support Vector Machine (SVM) & Least Squares SVM (LS-SVM):

A supervised ML technique called SVM [16,7] has been applied to tasks for both classification and regression. Typically employed in classification problems, numerous studies have produced excellent outcomes using various strategies. SVM and LS-SVM performance for rice leaf disease classification were examined in one study [16]. Although SVM is a strong technique for a variety of classification applications, its computing cost may be high. In order to overcome this complexity, LS-SVM substitutes error variables for some of the SVM's components. The results of the study showed that LS-SVM has the potential to be a computationally efficient disease classification method by outperforming SVM in terms of accuracy for detecting four types of rice leaf diseases.

b) K-Nearest Neighbors (KNN), Logistic Regression, Decision Tree, and Naive Bayes:

Numerous research investigated the efficacy of various ML classifiers, including Naive Bayes, KNN, Decision Trees, and Logistic Regression, for diagnosing rice leaf disease [2,15]. One of them [2] employed color attributes derived from previously processed pictures. After evaluating each classifier's performance, they decided which model was most effective for the given dataset and disease categories.

Deep Learning Techniques for Rice Leaf Disease Detection

DL, a subfield of ML, has grown in popularity as an image recognition and classification tool. CNNs are a popular DL architecture that performs extremely well in image interpretation. CNNs may learn complicated properties without explicitly extracting them from visual input.

a) Convolutional Neural Networks (CNNs) with Transfer Learning:

CNNs have been utilized in numerous studies to classify rice leaf diseases. One method employs transfer learning, in which a CNN model that has previously been trained on a large picture dataset is updated for the specific aim of identifying rice disease [18,7,10,15]. Compared to training a CNN from scratch, this strategy reduces training time and allows the model to benefit from past knowledge. In contrast to the traditional VGG model, the study's novel CNN model achieves high accuracy while using less memory by merging a multi-scale convolution module with a pre-trained VGG model.

Recognizing that some disease classes might be under-represented in the data, the authors implemented a balanced class weight distribution technique. This helped the model pay closer attention to these classes during the training process. Finally, to optimize the performance of the model's fully connected layers, they utilized a method inspired by the **"bandit approach"**. In another paper [7] a hybrid CNN (Inception-ResNet) - SVM model for detecting and treating damaged rice leaves has been developed. In this designed model, the images are collected and gathered by capturing rice leaf using a camera at the agricultural field. These images are refined to improve image quality and visibility for reliable estimation, and then segregated using Grab-Cut algorithm to eliminate undesired sections of image. Features of the segmented images are extracted and classified using hybrid CNN (Inception-Resnet V2)-SVM algorithm.

In a paper [14] the researchers built a comprehensive dataset of rice leaf images encompassing healthy and diseased leaves at various stages (mild/severe blight, tungro, blast, brown spot) and to enrich the dataset and address potential imbalances, they employed data augmentation techniques. Next, they proposed custom CNN models specifically tailored for rice leaf disease classification. The performance of these models was evaluated against established transfer learning approaches like VGG16, Xception, ResNet50, DenseNet121, Inception ResnetV2, and Inception V3. This comparison aimed to identify the most effective model for rice leaf disease classification. Finally, to enhance interpretability and understanding of the classification process, the researchers visualized the feature extraction performed by each layer of the most successful model (custom VGG16). This graph explains how the algorithm distinguishes between healthy and sick leaves.

In the researcher’s one study [4] proposed a different approach for diagnosing tomato leaf diseases using a modified AlexNet architecture-based CNN. They utilized a dataset of over 18,000 training images and 4,500 testing images categorized into ten different tomato leaf diseases. The CNN architecture consisted of six layers: three convolutional layers with ReLU activation and max pooling, and three fully-connected layers with ReLU activation for feature extraction and classification. The final layer used a Softmax activation function for multi-class classification. This approach achieved high accuracy demonstrating the effectiveness of the modified AlexNet CNN for tomato leaf disease detection.

b) Vision Transformers (ViT) for Classification:

One of the unique approach is implemented by the researchers [13]. In their paper they compared three approaches: (A) CNNs with channel and spatial attention for improved feature extraction, (B) transfer learning with Inception V3 for leveraging pre-trained knowledge, and (C) ViT models (ViT-1 and ViT-2) with different hyperparameter configurations. They used Grad-CAM for visualizing disease locations in CNNs and a DINO technique for ViT attention visualization to understand which image regions the models focus on for classification. This multi-pronged approach aimed to achieve robust and accurate disease classification, surpassing existing methods, which they achieved with good enough to call it a success.

c) U-Net for Semantic Segmentation:

A different approach utilizes U-Net, a DL architecture specifically designed for semantic segmentation tasks [11,9]. Semantic segmentation aims to assign a specific class label to each pixel in an image. In the context of rice leaf disease detection, U-Net can be used to segment diseased regions within rice leaf images. This study [11] employed U-Net to segment rice leaf blast from images captured in rice fields. They optimized the U-Net model's hyperparameters using different search methods (Hyperband, Random Search, Bayesian optimization) to achieve the most accurate segmentation of rice blast. In other study[9], they compared two models, U-Net and Modified U-Net, for identifying the specific diseased regions within the leaf images. It achieved superior performance with higher accuracy outperforming the standard U-Net model.

d) YOLO Network:

YOLO is a real-time object detection algorithm which is a single-stage object detector that uses a CNN to predict the bounding boxes and class probabilities of objects in input images.

YOLOv3 [6] is a popular improvement version of YOLO. In YOLOv3, the bounding box prediction methods were improved and the convolutional neural network backbone’s size was enlarged. YOLOv3 uses a binary cross-entropy loss function with logistic activation for the object's class determination which was used as one of its important models in one study to detect plant disease. In the object location bounding box localize part, they used logistic regression and sum squared error to predict the objectness score and to find regression loss.

Another study proposed an improved YOLOv5 network for rice leaf disease classification [1]. This YOLOv5 network was trained on a publicly available dataset of Indonesian rice leaves with various diseases. The system employed pre-processing techniques to enhance the data and a modified YOLO architecture for disease classification. The study demonstrated the effectiveness of YOLO for rice leaf disease detection with a processing time of around 40 seconds.

2.1 Comparison

Ref. No	Task	Dataset	Method	Accuracy
[6]	Detection and classification Of 6 rice leaf diseases	6,330 field-collecting images	YOLOv3 Mask R CNN FasterR-CNN RetinaNet	79.19% 75.92% 70.96% 36.11%
[16]	Performance Comparison between SVM and LS-SVM for Rice Leaf Disease Detection	DS1 (Two Datasets containing 4 disease classes) DS2	SVM LS-SVM SVM LS-SVM	83.3% 91.3% 98.75% 98.87%
[5]	Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning	Rice leaf Dataset Wheat Leaf Dataset	VGG-16	97.22% 98.75%
[18]	Lightweight multi-scale Convolutional Neural Network for Rice Leaf Disease Recognition	Kaggle dataset	Pre-trained VGG16 CNN	91.23% 97%
[10]	An Optimized Rice Leaf Disease Classification using Transfer Learning and Balanced Class Weight Distribution based on the Bandit Approach	Dataset containing Rice Blast, Hispa, and Brown spot diseases	Pre-trained CNN	98%
[7]	Detection and Prediction of Rice Leaf Disease Using a Hybrid CNN-SVM Model	Custom self-made dataset from capturing images to segregation using Grab-Cut algorithm	Hybrid CNN (Inception-ResNet)-SVM model	97%
[14]	DeepRice: A DL and deep feature-based classification of Rice leaf disease subtypes	Dataset containing 5932 self-generated images of rice leaves categorized into 9 classes	Custom VGG16	99.94%
[12]	Automatic early detection of rice leaf diseases using hybrid DL and ML methods	Dataset containing Bacterial leaf blight, Brown spot, and Leaf smut diseases	CNN based on VGG-16	97.3%

[2]	Automated Classification of rice leaf disease using DL Approach	Dataset containing 1045 images of Stem borer, Sheath Blight Rot Brown Spot, False Smut	KNN based on CNN	95%(Healthy) 90%(diseased)
[1]	YOLO Network-Based for Detection of Rice Leaf Disease	Public dataset of Indonesian rice leaves	Enhanced Yolo	94%
[3]	A Prediction of Rice Leaf Disease using KSC	Dataset was manually made and assembled by classifying diseased leaves into three groups. (bacterial leaf blight, brown spot, and leaf smut)	XceptionNet deep feature with SVM classifier	94.33%
[15]	Multiclass Classification of Rice Leaf Disease Using DL Based Model	Dataset from Kaggle	Proposed CNN with adam optimizer	95.72%
[17]	Transfer Learning based Rice Leaf Disease Classification with Inception-V3	Dataset containing 3 classes brown spot, leaf smut, and bacterial leaf blight.	Custom pre-trained InceptionV3	99.33%
[8]	Classification of Rice Leaves Diseases by Deep CNN-Transfer Learning Approach for Improved Rice Agriculture	Dataset containing leaf smut, bacterial leaf blight, and brown spot disease classes	Custom CNN transfer learning with VGG-16	97.22%
[11]	Semantic Segmentation of Rice Leaf Blast Disease using Optimized U-Net	Manually made dataset by observation, precisely in the Babadan sub-district, Ponorogo, East Java using Realme 6 phone	Optimized U-Net	98.5%
[4]	AlexNet CNN for Disease Detection and Classification of Tomato Leaf	Kaggle dataset	AlexNet-CNN	96%
[9]	DL-based segmentation and classification of leaf images for detection of tomato plant disease	Dataset containing 18,161 segmented and non-segmented tomato leaf images	Modified U-net segmentation InceptionNet1	98.66% 99.12%
[13]	ViT Based Models for Plant Disease Detection and Diagnosis	54,309 images of fruit and vegetable combinations including Apple, Blueberry, Cherry.	Vision transformer	96.7% 98.52% 99.1%

Table 2.1: Comparison across various Research Papers

2.2 Objectives

The goal of this project is to create a smartphone application that detects rice leaf disease correctly and instantly using DL. We want to achieve these aims by providing farmers with access to this cutting-edge device.

- Develop a mobile application using Flutter that integrates state-of-the-art DL models for the classification and object detection of rice leaf diseases.
- Utilize a ViT model for reliable classification of six rice leaf classes: healthy and five specific diseases.
- Employ YOLOv5 for precise object detection, pinpointing the location of diseased areas within the rice leaf image.
- Integrate weather data to provide insights into potential disease risk factors based on real-time weather conditions.
- Implement a shop locator feature to connect farmers with vendors offering appropriate pesticides and insecticides for the identified disease.
- Include a dedicated information section within the app for each disease, covering symptoms, causes, and most importantly, preventative measures.
- Design a user-friendly and intuitive interface for the mobile application, ensuring accessibility for farmers with varying levels of technical expertise.

This project aims to improve food security by improving early disease detection and better crop management practices.

- Improved farmer empowerment by making diagnostic resources and information more readily available.
- Encouragement of informed decision-making in relation to disease preventive strategies and ecologically sustainable farming methods.
- As a result of preventative and early management, the need for chemical applications has diminished.

This smartphone application has the potential to alter rice farming by providing farmers with the tools they need to create stable and healthy crops.

3. Exploratory Data Analysis

3.1 Dataset Description

Our study uses 3064 images from an image collection of rice leaf disease. This dataset was generated specifically for training and validating DL models for detecting rice leaf disease. The photographs were given by two sources:

a) **Publicly accessible sources:** A portion of the material was gathered from publicly available websites that focus on agricultural images. These archives contain a significant number of images of rice leaves displaying different diseases.

b) **Unaffiliated Gathering:** Aside from the information obtained from public sources, additional images were taken separately to capture local variances in the disease. Rice leaves from nearby farms and agricultural research facilities are shot as part of this independent gathering procedure.

3.1.1 Disease Classes:

The collection includes images of healthy rice leaves as well as five different types of rice leaf diseases, for a total of six categories to compare.

Fig 3.1: Categories of rice leaf diseases

These categories represent some of the most common and economically damaging rice diseases:

- **Bacterial Leaf Blight:** This bacterial infection causes rice plants to wilt and develop lesions, which reduces grain yield.
- **Brown Spot:** Fungal diseases that cause brown spots on leaves, reducing photosynthesis and lowering grain quality.
- **Healthy:** To discriminate between infected and healthy rice leaves, the model refers to images of healthy rice leaves.
- **Leaf Blast:** Fungus-induced oval or elliptical lesions on leaves can cause significant yield loss.
- **Leaf Scald:** Fungal infections that generate long, moist wounds on leaves may inhibit plant development and grain production.
- **Narrow Brown Spot:** Fungal disease that can reduce yield by creating dark, elongated, and thin lesions on leaves.

3.1.2 Data Split:

The dataset is divided into training and validation folders. The training folder contains the majority of the images used to train and the validation folder holds a smaller portion of the data used to assess the model's performance during the training process. This split ensures the model doesn't overfit the training data and generalizes well to unseen examples.

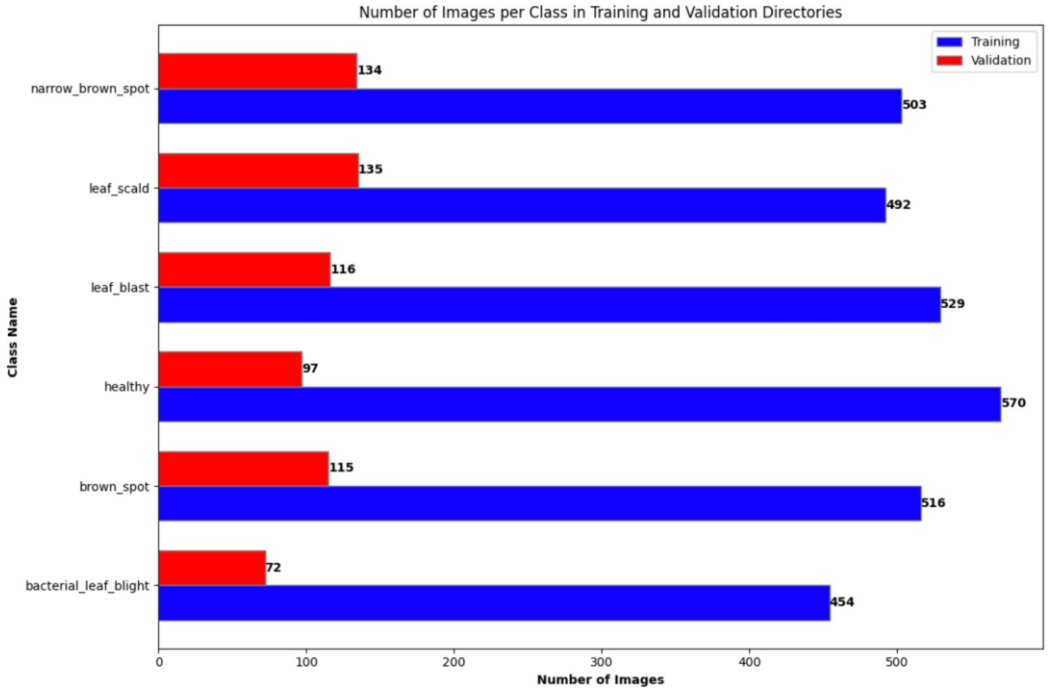


Fig 3.2: Imbalanced Distribution of Images in Training Dataset

In the training folder, initially, there were 3064 images distributed among 6 classes. However, the data was imbalanced, with varying numbers of images per class. To address this

imbalance, we balanced the data by reducing each class to 450 images, resulting in a total of 2700 images across all classes.

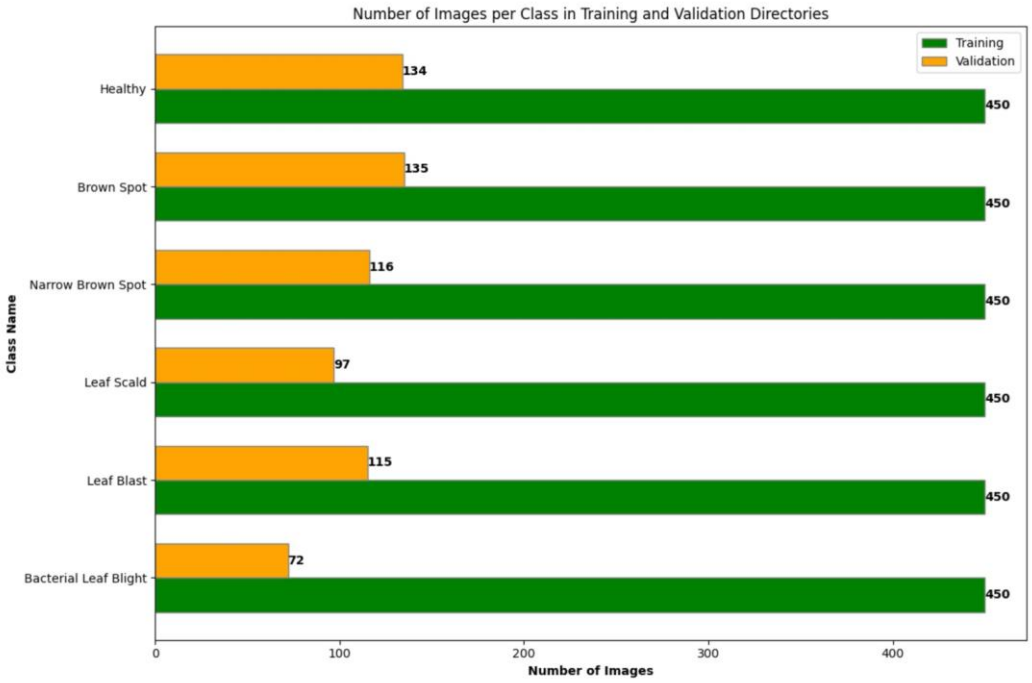


Fig 3.3: Balanced Distribution of Images in Training Dataset

4. Methodology

4.1 Introduction to Languages (Front-End and Back-End)

This project utilized a combination of programming languages to build a complete image classification system.

Front-End (Flutter with Dart):

Flutter is a mobile app development framework created by Google. It uses the Dart programming language, known for its speed, efficiency, and ability to compile for multiple platforms (Android and iOS) with a single codebase. Here we used it to build the user UI of the mobile application, allowing users to interact with the system.

Back-End (Python):

Python being a most versatile and widely used programming language known for its readability and extensive ecosystem of libraries became a main base for this project by serving as the primary language for building the core functionalities, including:

- **Vision Transformer Model Development:** Python, along with libraries like TensorFlow and Keras, facilitated the creation and training of the unique ViT model for image classification.
- **Model-Backend Communication:** Python played a role in establishing communication between the trained model and the mobile application's backend, potentially using cloud services like Azure for data exchange.

4.2 Supporting Languages/Packages

Beyond the core front-end and back-end languages, several other languages and packages were instrumental in building this system. Here are some main things that were very helpful in making our project a reality:

- **Firebase:** This Google service provides a backend-as-a-service (BaaS) solution, used for user authentication for login and sign-up of users in the application.
- **Google Cloud:** Google Cloud Platform was utilized for deploying the DL models, ensuring efficient hosting and deployment of the trained models, alongside facilitating communication between the front-end and back-end systems.

4.2.1 Additional Supporting Packages (Python):

In building the model several steps have taken place with the help of many libraries which played their role during the processing and evaluation.

a) Data Manipulation:

- **Pandas:** It is used for data analysis and exploration tasks related to the image dataset's labels.
- **NumPy:** Provides numerical computing capabilities for data manipulation and pre-processing.

b) Image Processing:

- **OpenCV:** It has been used for image pre-processing tasks like resizing or normalization.
- **PIL (Image):** This library has been used for image loading and manipulation during data preparation.

c) Data Augmentation:

- **ImageDataGenerator:** This Keras utility was used to facilitate data augmentation techniques to the dataset to improve model robustness.

d) Evaluation Metrics:

- **scikit-learn:** This is the library that provides functions for calculating metrics like confusion matrix and classification report to assess the model's performance. From this we imported a confusion matrix to evaluate the performance of our classification model.

4.3 User Characteristics

The project focuses on diseases which are widespread in rice fields causing major damage to the plants leading to a decline in production. The lesser yield leads to loss for farmers who have their entire livelihood dependent on these crops. So this application is designed for a primary user base consisting of

A) Target Users

- **Farmers:** This encompasses both small-scale and large-scale farmers who cultivate rice. The application should be accessible and user-friendly for individuals with varying levels of technical expertise.

- **Agricultural Extension Workers:** These agricultural-related officers of the Government are the ones who provide guidance and support to farmers. The application can be a valuable tool for them to diagnose rice diseases in the field and advise farmers on appropriate treatment strategies.

B) User Needs and Skills

1) Farmers:

I. Needs:

- **Easy to Use Interface:** The application should have a simple and intuitive interface that allows farmers to capture images of their rice plants and receive disease diagnoses quickly and efficiently.
- **Limited Technical Knowledge:** Many farmers may have limited experience with technology. The application should require minimal technical skills to operate.
- **Multilingual Support:** Catering to farmers in diverse regions might necessitate support for multiple languages.
- **Offline Functionality (Optional):** In areas with limited internet connectivity, the ability to use the application offline for basic functionality (disease identification from pre-downloaded reference images) could be beneficial.

II. Skills:

- Basic understanding of smartphone operation (if using a mobile application) and some good amount of knowledge on disease related to the plants.
- Ability to capture clear images of rice leaves using a smartphone camera.

2) Agricultural Extension Workers:

I. **Needs:**

- **Advanced Features:** Extension workers might require functionalities beyond basic disease identification, such as access to detailed information about different diseases, and recommended treatment options.

II. **Skills:**

- Proficiency with smartphones and mobile applications.
- Understanding of rice diseases and their management practices.

C) User Experience (UX) Considerations

- **Accessibility:** The application should be designed with accessibility in mind, considering major users would be with very little expertise with mobile usage in India. So, involving features like larger fonts, high-contrast color schemes, and simple processes to easily remember would be helpful.
- **Localization:** As we are targeting a local audience, the application should be able to work in remote areas where there would be weaker signals.
- **Training and Support:** Providing a user guide or a tutorial in the application can help users understand the application's functionalities and maximize its benefits.

4.4 Constraints

There are some limitations and challenges we encountered during the development of the rice leaf disease prediction application using DL. We categorized them in parts explaining the requirements and importance of overcoming them

a) Data Constraints:

- **Data Availability:** Obtaining large and diverse official datasets of rice leaf disease images is crucial for training an accurate DL model. Limited access to well-preprocessed data can restrict the model's ability to generalize to unseen disease patterns and variations.
- **Data Quality:** We all know that the quality of the training data highly impacts the model's performance as factors like blurry images, poor lighting conditions, or mislabeled data can lead to inaccurate diagnoses.
- **Data Class Imbalance:** Real-world different types of rice diseases might not be equally widespread. If the training data have a significant imbalance between disease classes and healthy leaves class, the model might struggle to accurately identify the less frequent diseases.

b) Computational Constraints:

- **Training Time and Resources:** DL models, especially complex architectures like ViT and Yolov5 require significant computational power and time for training. Limited access to powerful computing resources (GPUs, TPUs) can extend the development timeline.

c) User Constraints:

- **Technical Literacy:** As we mentioned in user characteristics the target users might have varying levels of technical skills and experience. A complex UI or lack of proper training materials might limit user engagement.
- **Internet Connectivity:** In areas with limited or unreliable internet access, the application's functionality might be restricted. Consideration of some offline capabilities like some basic details of each disease can be helpful.

d) Ethical Constraints:

- **Data Privacy:** Ensuring user data privacy is crucial as it is one of the main parts which if fails can lead to many high levels of violations. Implementing secure data storage practices and obtaining user consent for data collection and usage are needed.
- **Bias and Fairness:** The model's training data can introduce biases that might lead to inaccurate diagnoses for certain rice varieties or environmental conditions. As these types of issues are common during building the model, taking the steps which can overcome them using techniques like data augmentation and careful selection of training data can help mitigate these biases.

4.5 Flow Chart

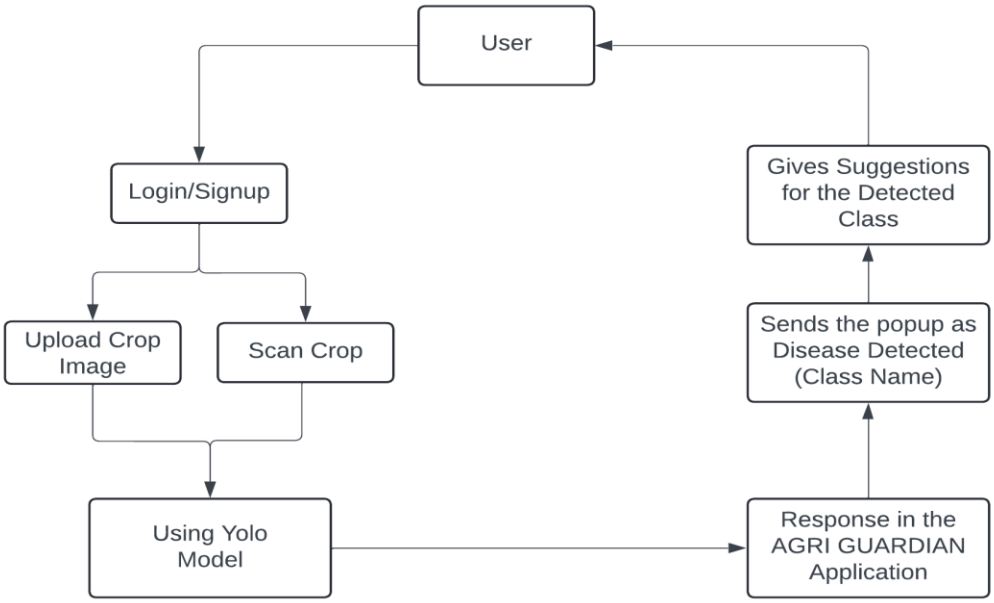


Fig 4.5.1: Modular work Flow

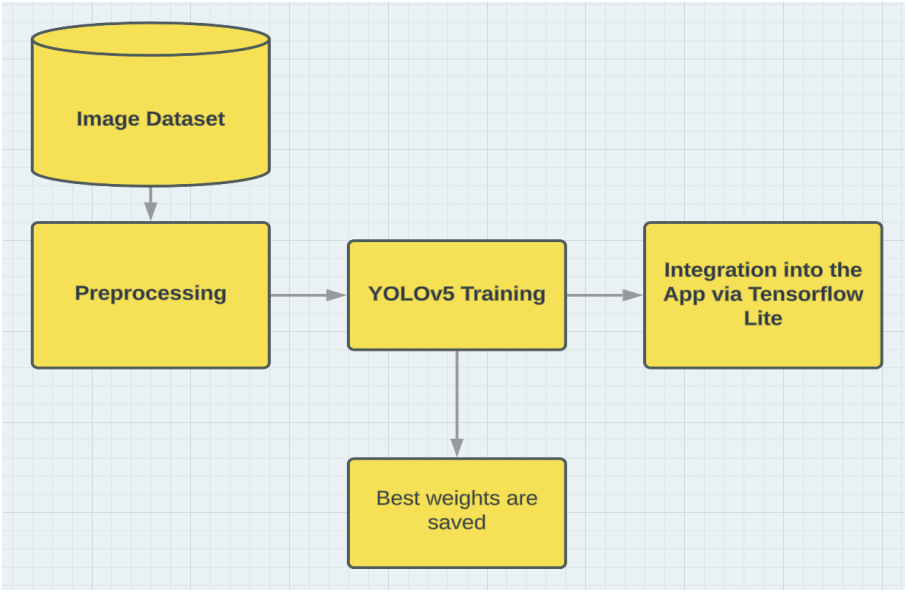


Fig 4.5.2: Flowchart for the Yolov5 model architecture

The process of gathering task-related photos. These are downloadable, capturable, or available from other sources. We adjust the photographs for consistency, resizing, or normalizing them in order to get them ready for training. A YOLOv5 model is then trained using this preprocessed data to identify and locate objects within the pictures. To use the trained model in our application on low-

resource devices, it is finally compressed into a smaller size. The model that performs the best on a different validation set is referred to by the "best weights" when we store checkpoints during training to monitor progress.

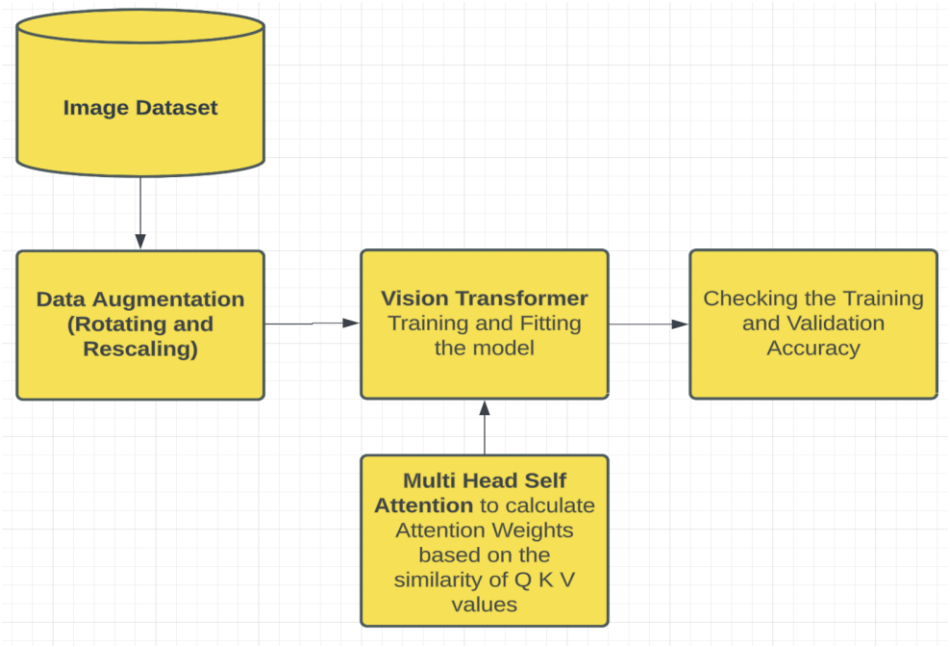


Fig 4.5.3: Flowchart for the ViT model architecture

4.6 Database Design

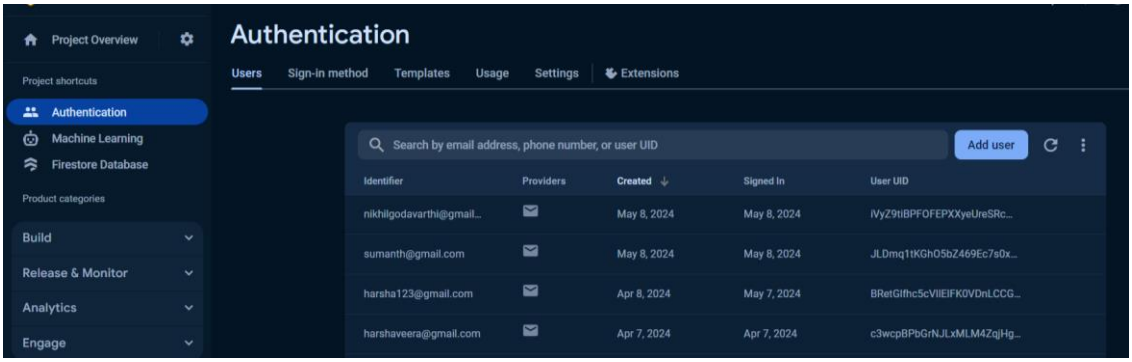


Fig 4.6: Firebase Database for Agri Guardian

In the above Figure 6, illustrates users information storage structure in Firebase for the Agri Guardian application, which enables login and signup functionalities.

These are the functionalities we used to store users information:

Users Collection: This collection stores a separate document for each user who signups in the app.
Unique User ID (uid): It helps to identify the user and retrieves the data.
Timestamps: Records of creation and update times.

4.7 Algorithm Discussion

4.7.1 Vision Transformer

The system architecture of a ViT model for image classification can be explained in a step by-step process providing a detailed breakdown of its components and functionalities.

Input Processing Pipeline:

a) **Image Ingestion:** The system begins by receiving an image as input. For RGB images, for easier processing we made the system assume a certain format (e.g., 224x224x3). By doing this, the training process is guaranteed to receive consistent data feeding.
b) **Patch Extraction:** To leverage the power of transformers, the image undergoes a process called patching. Here, the image is divided into smaller, fixed-size squares (e.g., 16x16 pixels) known as patches. This creates a grid-like representation of the original image, breaking it down into manageable units.
c) **Patch Embedding:** Next, a lower-dimensional vector representation of each extracted patch is generated. This embedding technique enables the model to use the transformer architecture's capacity to process numerical inputs. To accomplish this transition, the system employs a convolutional layer, which essentially pulls the main properties from each patch.
d) **Positional Encoding:** Transformers do not have the same internal capacity as conventional CNNs, which can automatically detect spatial relationships within images. To do this, additional information must be supplied to each patch's embedded vector to show its relative position within the original image. This step is critical for the model to understand the context and relationships between several patches.

Transformer Encoder: The Core Engine

The stacked Transformer encoder blocks are the foundation of the ViT architecture. These building components do the real lifting, repeatedly evaluating the data and extracting increasingly complex features. Every block consists of two main sub-layers:

- **Multi-head Self-Attention (MHSA):** This sublayer enables the model to pay attention to connections between many patches. Here, the model focuses on specific patches while also considering their interactions with other patches in the image. This method allows the model to capture long-range relationships, providing it a significant advantage over traditional CNNs, which focus mostly on local neighborhoods.
- **Feed-forward Network (FFN):** This sublayer introduces nonlinearity into the network, allowing it to learn more complicated feature representations. This structure is often created by inserting a non-linear activation function (Swish) between two convolutional layers. With this FFN, the model can identify more complex patterns in the data than simply basic linear correlations.

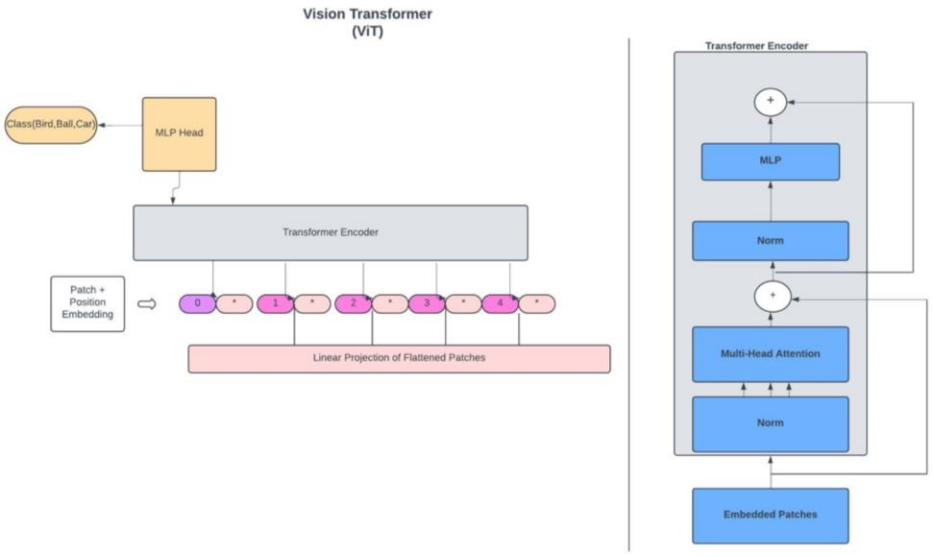


Fig 4.7.1: ViT Model architecture

Residual Connections and Layer Normalization:

To ensure stability and prevent vanishing gradients during training, the system employs residual connections and layer normalization. Residual connections add the input of each block to its output, allowing the model to learn from the original information alongside the transformed features. Layer normalization, on the other hand, normalizes the activations within each block, facilitating faster convergence and improved training efficiency.

The system stacks multiple identical encoder blocks, allowing the model to progressively refine its understanding of the image through a series of attention and feed-forward operations. With each iteration, the model extracts higher-level features that capture the intricate relationships between different parts of the image.

Post-processing and Classification:

a) **Global Average Pooling:** After passing through the encoder stack, the system employs a global average pooling layer. In order to create a single feature vector, this layer compiles the data from every other element in the feature map. This vector summarizes the learned features and captures the essence of the whole image.

b) **Classification Head:** This final stage involves the classification head, responsible for predicting the class probabilities for the input image. This head typically comprises:

- **Dropout layer:** This layer introduces a degree of randomness by dropping a certain percentage of activations during training to prevent overfitting and improve the model's generalization capabilities.
- **Dense layer with non-linear activation:** This layer transforms the feature vector into a higher-dimensional space, allowing for the creation of more complex decision

boundaries. The code utilizes a Swish activation function for this purpose.

- **Output layer with softmax activation:** The final dense layer, equipped with a softmax activation function, outputs a vector with class probabilities. Each element in this vector represents the probability of the image belonging to a specific class.

Training and Evaluation:

The system utilizes the Adam optimizer with a learning rate to train the model. This optimizer efficiently navigates the loss function landscape, updating model parameters to minimize the classification error.

4.7.2 Yolov5

YOLOv5 from Ultralytics is used for object detection. This pre-trained model is fine-tuned on our custom dataset having rice leaf diseases. YOLO is a family of object detection models. The primary framework of YOLOv5 consists of stacking multiple modules of Conv + BatchNorm + SiLu and C3. Lastly, an SPPF module is linked to improve the capability of feature expression.

- **Backbone:** CSPDarknet53

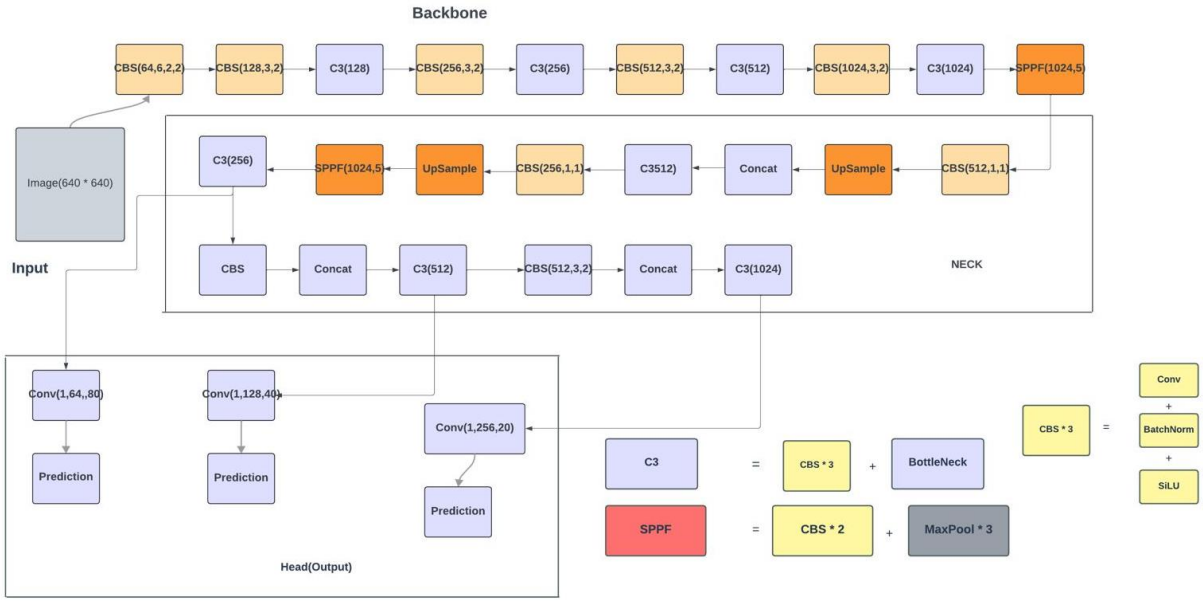


Fig 4.7.2: Yolov5 Model architecture

- **Neck:** FPN (Feature Pyramid Network) and PAN (Pyramid Attention Network for Semantic Segmentation). To create multiple new feature maps (P3, P4, and P5) for the purpose of detecting different scale targets, FPN up-samples the output feature map (C3, C4, and C5) produced by multiple convolutions down sampling operations from the feature extraction network.
- **Head:** The feature map's upper left corner's coordinate value is set to (0, 0). The predicted center point's unadjusted coordinates are rx and ry. The values of gx, gy, gw,

and gh indicate the information in the modified prediction box. PW and Ph are for the previous anchor's information. The offsets determined by the model are denoted by sx and sy. the procedure of changing the preset prior anchor's center coordinate and size to match the final prediction box's center coordinate and size.

The dataset was obtained from Roboflow. The images in the dataset were of size 640x640. Each and every image is annotated using roboflow tool. The images of The YOLOv5 pretrained model was cloned from the official GitHub repository. The pretrained weights and the required packages were downloaded. The paths to the train and validation image directories and number of classes were updated in the 'data.yaml' file. Images were fed and trained using the file 'train.py' with the following parameters:

- Image Size: 640x640
- Batch Size: 16
- Epochs: 10
- Data: 'data.yaml' file
- Weights: 'yolov5s.pt' file

There were three loss components: box_loss (Bounding Box Regression Loss), obj_loss (Objectness Loss) and cls_loss (Classification Loss).

- **box_loss (Bounding Box Regression Loss):** This loss measures how well the predicted bounding box around an object aligns with the actual bounding box of the object in the training data (ground truth). It is typically calculated using a distance metric like mean squared error (MSE). A lower box_loss indicates that the model is predicting tighter bounding boxes that accurately enclose the objects.
- **obj_loss (Objectness Loss):** This loss assesses the model's confidence in predicting whether a specific region in the image contains an object or not (background). It helps the model distinguish between regions with objects and empty backgrounds. A lower obj_loss signifies that the model is confident in its predictions about the presence or absence of objects in different image regions.
- **cls_loss (Classification Loss):** This loss evaluates how well the model classifies the type of object detected within a bounding box. It is often calculated using the cross-entropy loss function, which penalizes the model for incorrect class predictions. A reduced cls_loss indicates that the objects found inside the bounding boxes are appropriately categorized by the model.

The mAP50 statistic refers to the mAP at an IoU threshold of 0.5. It focuses on how well the model works when the predicted and real object's bounding boxes overlap by at least 50% (IoU > 0.5).

5. Results

A training path for the YOLOv5 DL model is shown in Figure 4, along with important metrics like training loss, object loss, class loss, recall, accuracy, and mean average precision, or mAP. 10 epochs were used to train the model.

Figure 5.5 illustrates precision as the proportion of correctly anticipated positive outcomes among all the positive predictions produced by the YOLOv5 model. Across all classes, the model has an average confidence level of 0.457.

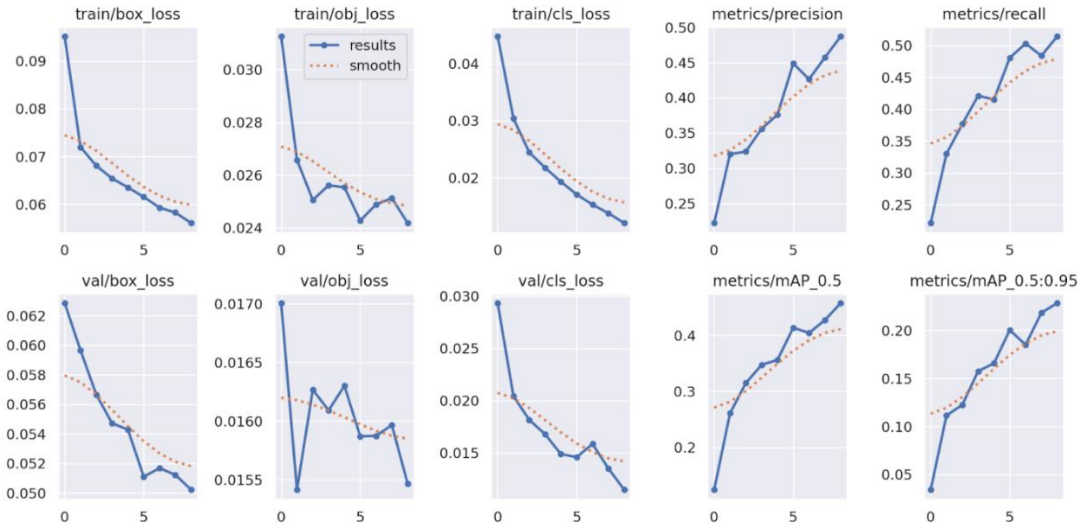


Fig 5.1: Plots of Evaluation metrics of Yolov5

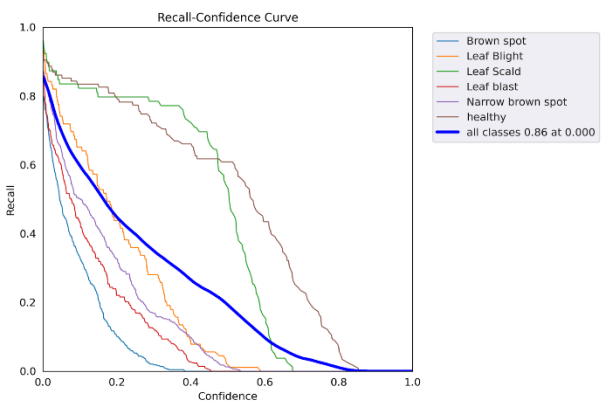


Fig 5.2: Plot of Recall vs Confidence

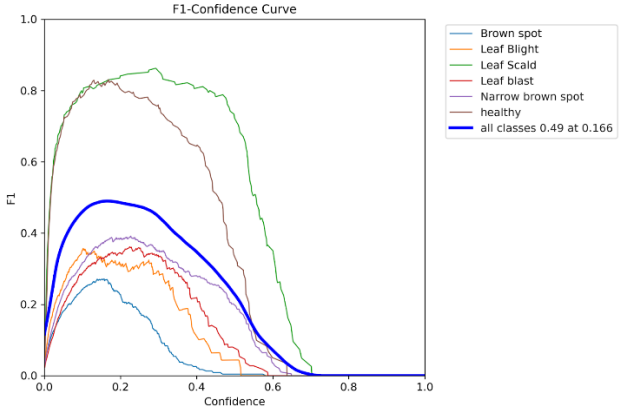


Fig 5.3: Plot of F1 Score vs Confidence

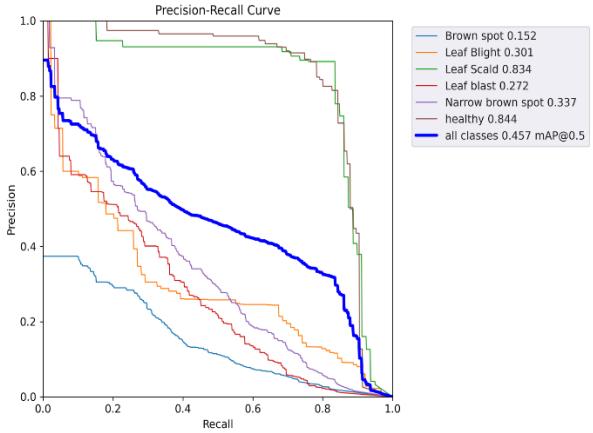


Fig 5.4: Plot of Recall vs Precision

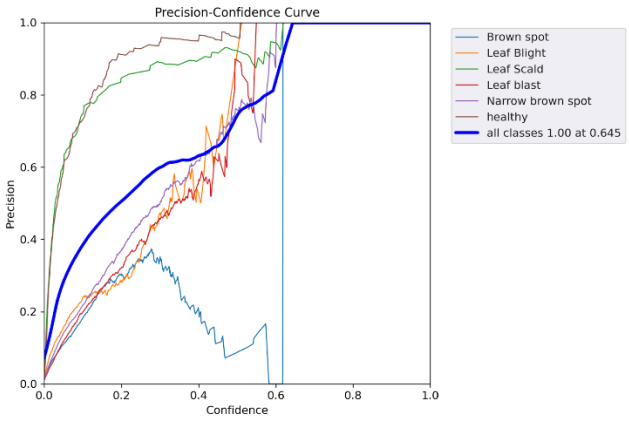


Fig 5.5: Plot of Confidence vs Precision

Fig 5.6: Results of some random samples

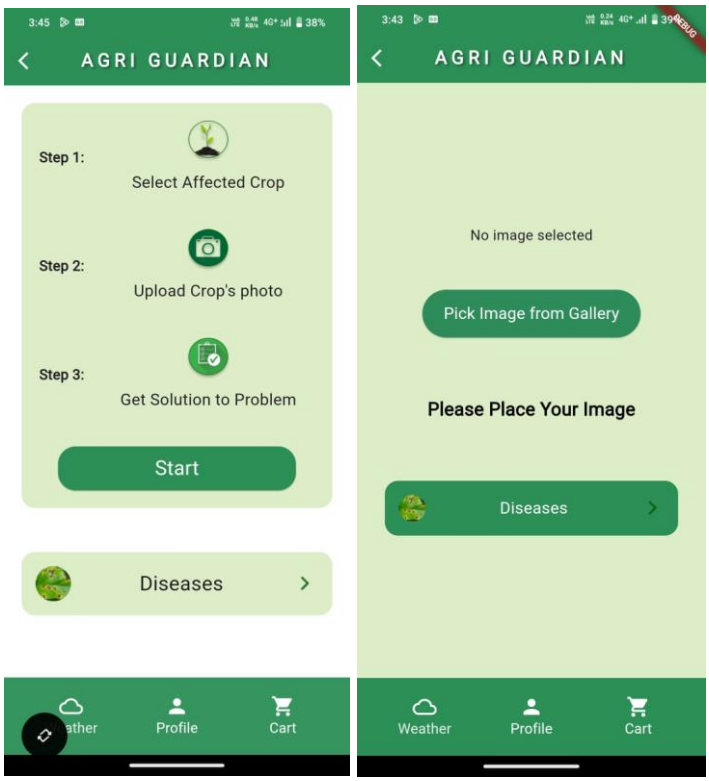


Fig 5.7: Screenshots of Agri Guardian Application

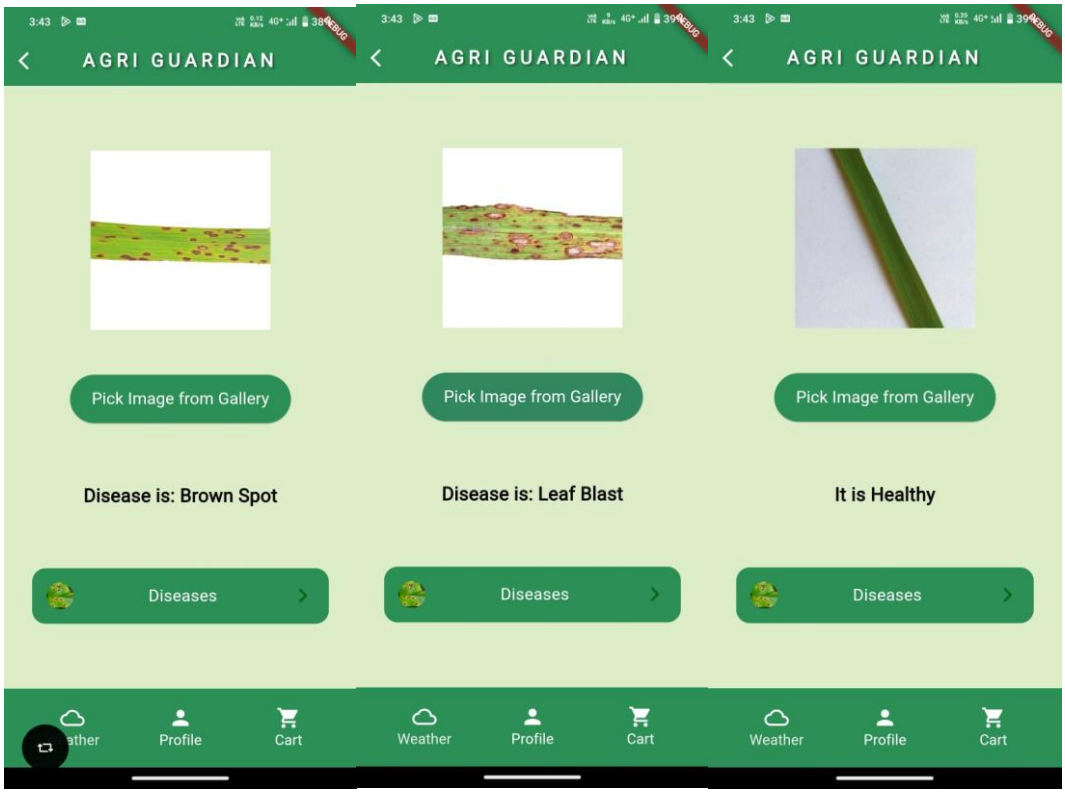


Fig 5.8: Results of some random samples from Agri Guardian

6. Conclusion and Future Scope

6.1 Conclusion

We have successfully developed a mobile application for rice leaf disease detection for multiple diseases utilizing a combination of DL techniques: ViT classification for accurate disease identification and YOLOv5 for real-time object detection. This integrated approach offers reliable and user-friendly solutions for farmers.

Weather data and a store option for buying insecticides and pesticides are two further features that the application added to improve usefulness. Farmers are better able to identify diseases, decide on the best course of action based on the weather, and easily obtain the resources they need which can make the application pretty successful among the users.

In conclusion, this mobile application presents a valuable tool for rice farmers and agriculture officers promoting early disease detection, informed treatment strategies, and ultimately improving crop health and yield.

6.2 Future Scope

1. Build and train a robust YOLOv5 model on balanced and several other datasets.
2. Integrate local languages into app interface and disease information, for better user accessibility.
3. Perform Data Augmentation and build a pipeline of Segmentation, Object Detection and Classification.
4. Enable in-app consultations with agricultural specialists.

7. References:

[1] Aziz, F., Ernawan, F., Fakhreldin, M., & Adi, P. W. (2023, August). YOLO Network-Based for Detection of Rice Leaf Disease. In 2023 International Conference on Information Technology Research and Innovation (ICITRI) (pp. 65-69). IEEE.

[2] Cherukuri, N., Kumar, G. R., Gandhi, O., Thotakura, V. S. K., NagaMani, D., & Basha, C. Z. (2021, December). Automated Classification of rice leaf disease using Deep Learning Approach. In 2021 5th International conference on electronics, communication, and aerospace technology (ICECA) (pp. 1206-1210). IEEE.

[3] Shanuja, C. C., Arvind, R., & Ganga, B. M. (2023, August). A Prediction of Rice Leaf Disease using KSC. In 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 1050-1054). IEEE.

[4] Chen, H. C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C. W., Chen, L., & Weng, C. E. (2022). AlexNet convolutional neural network for disease detection and classification of tomato leaf. Electronics, 11(6), 951.

[5] Jiang, Z., Dong, Z., Jiang, W., & Yang, Y. (2021). Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. Computers and Electronics in Agriculture, 186, 106184.

[6] Kiratiratanapruk, K., Temniranrat, P., Kitvimonrat, A., Sinthupinyo, W., & Patarapuwadol, S. (2020, September). 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.

[7] Chaudhari, D. J., & Malathi, K. (2023). Detection and Prediction of Rice Leaf Disease Using a Hybrid CNN-SVM Model. Optical Memory and Neural Networks, 32(1), 39-57.

[8] Moiz, M., Akmal, M., Ishtiaq, M. S., & Javed, U. (2022, December). Classification of Rice Leaves Diseases by Deep CNN-Transfer Learning Approach for Improved Rice Agriculture. In 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE) (pp. 1-6). IEEE.

[9] Shoaib, M., Hussain, T., Shah, B., Ullah, I., Shah, S. M., Ali, F., & Park, S. H. (2022). Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. Frontiers in Plant Science, 13, 1031748.

[10] Putra, O. V., Trisnaningrum, N., Puspitasari, N. S., Wibowo, A. T., & Rachmawaty, E. (2022, August). An optimized rice leaf disease classification using transfer learning and balanced class

weight distribution based on bandit approach. In 2022 5th International Conference on Information and Communications Technology (ICOIACT) (pp. 417-422). IEEE.

[11] Putra, O. V., Annafii, M. N., Harmini, T., & Trisnaningrum, N. (2022, November). Semantic segmentation of rice leaf blast disease using optimized U-Net. In 2022 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM) (pp. 43-48). IEEE.

[12] Rajpoot, V., Tiwari, A., & Jalal, A. S. (2023). Automatic early detection of rice leaf diseases using hybrid deep learning and machine learning methods. Multimedia Tools and Applications, 82(23), 36091-36117.

[13] Boukabouya, R. A., Moussaoui, A., & Berrimi, M. (2022, November). Vision Transformer Based Models for Plant Disease Detection and Diagnosis. In 2022 5th International Symposium on Informatics and its Applications (ISIA) (pp. 1-6). IEEE.

[14] Ritharson, P. I., Raimond, K., Mary, X. A., Robert, J. E., & Andrew, J. (2024). DeepRice: A deep learning and deep feature based classification of Rice leaf disease subtypes. Artificial Intelligence in Agriculture, 11, 34-49.

[15] Saini, A., Guleria, K., & Sharma, S. (2023, August). Multiclass Classification of Rice Leaf Disease Using Deep Learning Based Model. In 2023 3rd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-6). IEEE.

[16] Acharya, S., Kar, T., Samal, U. C., & Patra, P. K. (2023). Performance comparison between svm and ls-svm for rice leaf disease detection. EAI Endorsed Transactions on Scalable Information Systems, 10(6).

[17] Teja, K. U. V. R., Reddy, B. P. V., Kesara, L. R., Kowshik, K. D. P., & Panchaparvala, L. A. (2021, October). Transfer Learning based Rice Leaf Disease Classification with Inception-V3. In 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON) (pp. 1-6). IEEE.

[18] Zhang, C., Ni, R., Mu, Y., Sun, Y., & Tyasi, T. L. (2023). Lightweight Multi-scale Convolutional Neural Network for Rice Leaf Disease Recognition. Computers, Materials & Continua, 74(1).

8. Plagiarism Check Report:

8.1 Similarity

Submission date: 13-May-2024 10:41AM (UTC+0530)
Submission ID: 2377965425
File name: P3REPORT123.pdf (3.73M)
Word count: 6212
Character count: 37552

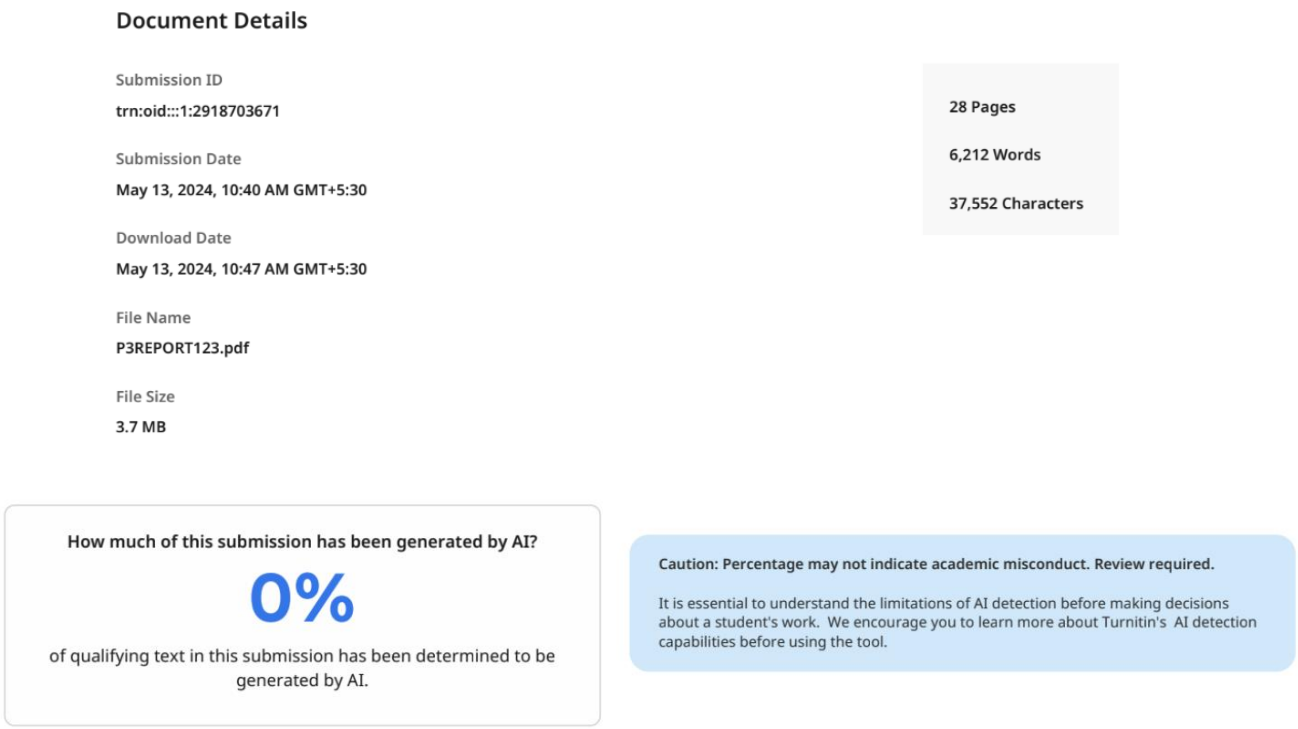
ORIGINALITY REPORT

2%	1%	1%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Minh Dang, Hanxiang Wang, Yanfen Li, Tri-Hai Nguyen, Lilia Tightiz, Nguyen Xuan-Mung, Tan N. Nguyen. "Computer Vision for Plant Disease Recognition: A Comprehensive Review", The Botanical Review, 2024 Publication	<1%
2	www.ijisae.org Internet Source	<1%
3	Ghazanfar Latif, Sherif E. Abdelhamid, Roxane Elias Mallouhy, Jaafar Alghazo, Zafar Abbas Kazimi. "Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model", Plants, 2022 Publication	<1%
4	Mehwish Moiz, Muhammad Akmal, Muhammad Shakeel Ishtiaq, Usman Javed. "Classification of Rice Leaves Diseases by Deep CNN-Transfer Learning Approach for Improved Rice Agriculture", 2022 International Conference on Emerging Trends	<1%

8.2 AI Check



Document Details

Submission ID
trn:oid:::1:2918703671

Submission Date
May 13, 2024, 10:40 AM GMT+5:30

Download Date
May 13, 2024, 10:47 AM GMT+5:30

File Name
P3REPORT123.pdf

File Size
3.7 MB

28 Pages

6,212 Words

37,552 Characters

How much of this submission has been generated by AI?

0%

of qualifying text in this submission has been determined to be generated by AI.

Caution: Percentage may not indicate academic misconduct. Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.