CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

This project utilizes machine learning to aid farmers in making informed decisions regarding crop selection, fertilizer application, and disease control. It analyzes various factors such as soil nutrients, rainfall patterns, crop types, and leaf images to provide personalized recommendations. By leveraging advan This project utilizes machine learning to aid farmers in making informed decisions regarding crop selection, fertilizer application, and disease control. It analyzes various factors such as soil nutrients, rainfall patterns, crop types, and leaf images to provide personalized recommendations. By leveraging advanced algorithms like Random Forest and Naive Bayes, the system aims to achieve high accuracy in its predictions. The goal is to optimize agricultural practices for increased productivity and sustainability. Through the integration of diverse data sources, the project seeks to address key challenges faced by farmers. It aims to enhance crop yield while minimizing resource wastage and environmental impact. The developed system offers a user-friendly interface for easy access and interpretation of recommendations. By empowering farmers with actionable insights, it contributes to the advancement of precision agriculture. The project's approach combines cutting-edge technology with practical solutions to real-world agricultural problems. It aims to bridge the gap between traditional farming methods and modern data-driven approaches. Through collaboration with agricultural experts and stakeholders, the project ensures relevance and applicability to the farming community. It emphasizes simplicity and usability to cater to the diverse needs of farmers worldwide. The project is driven by a commitment to improving food security and promoting sustainable farming practices. It envisions a future where technology serves as a powerful tool for agricultural development. In summary, the project strives to revolutionize farming through data-driven decision-making and innovative solutions.ced algorithms like Random Forest the system aims to achieve high accuracy in its predictions.

The goal is to optimize agricultural practices for increased productivity and sustainability. Through the integration of diverse data sources, the project seeks to address key challenges faced by farmers. It aims to enhance crop yield while minimizing resource wastage and environmental impact. The developed system offers a user-friendly interface for easy access and interpretation of recommendations. By empowering farmers with actionable insights, it contributes to the advancement of precision agriculture. The project's approach combines cutting-edge technology with practical solutions to real-world agricultural problems. It aims to bridge the gap between traditional farming methods and modern data-driven approaches. Through collaboration with agricultural experts and stakeholders, the project ensures relevance and applicability to the farming community. It emphasizes simplicity and usability to cater to the diverse needs of farmers worldwide. The project is driven by a commitment to improving food security and promoting sustainable farming practices. It envisions a future where technology serves as a powerful tool for agricultural development. In summary, the project strives to revolutionize farming through data-driven decision-making and innovative solutions.

1.2 PROBLEM STATEMENT

Agriculture plays a pivotal role in ensuring global food security and sustaining livelihoods. However, farmers encounter significant challenges in maximizing crop productivity, primarily stemming from uncertainties in crop selection, optimal fertilizer usage, and effective crop disease management. Traditional farming methodologies often rely on subjective decision-making processes, resulting in inefficient resource allocation and suboptimal agricultural outcomes. Consequently, there exists a critical necessity for an intelligent decision support system that harnesses the power of machine learning techniques. This system aims to empower farmers by providing them with data-driven insights and recommendations for informed decision-making concerning crop selection, fertilizer application, and disease management strategies.

1.3 OBJECTIVES

- To Raita Mitra aims to revolutionize agriculture by providing farmers with personalized crop recommendations, fertilizer advice, and early plant disease detection.
- To reducing the hardware requirements and improving the productivity.
- To classify the plant image into “Healthy” and “Disease” category based on leaf condition.

1.4 EXISTING SYSTEM

One existing system that aligns with the provided description is the traditional agricultural advisory services often found in rural areas of developing countries. These services typically involve local agricultural experts who rely on manual analysis and their traditional knowledge to provide recommendations to farmers. These experts offer generalized advice based on their experience and understanding of local soil types, climate conditions, and common crop diseases. However, these recommendations may not always be personalized to the specific needs of individual farmers or based on comprehensive historical data.

As a result, farmers may encounter challenges in optimizing crop selection for maximum yield and profit, as they must navigate through the limitations of relying on outdated methods and generalized advice.

1.5 LIMITATIONS OF EXISTING SYSTEMS

1. Reliance on historical data may not account for sudden environmental changes or emerging diseases, leading to less accurate predictions.

2. Inadequate consideration of local variations in soil composition and climate conditions may result in suboptimal recommendations for specific regions.

3. Dependency on image recognition technology for disease prediction may encounter challenges in accurately identifying complex or rare diseases, potentially leading to misdiagnosis.

1.6 PRPOSED SYSTEM

The proposed system seamlessly integrates ResNet-9, a convolutional neural network architecture widely recognized for its efficacy in image recognition tasks, alongside machine learning and deep learning algorithms. This integration aims to deliver personalized recommendations to farmers regarding crop selection, fertilizer application, and disease prevention strategies.

By harnessing agricultural data encompassing both historical and real-time information on soil conditions, weather patterns, and pest outbreaks, the system furnishes farmers with precise and timely insights. The ResNet-9 model is specifically employed to discern diseases in plants by scrutinizing images of afflicted crops. Concurrently, machine learning and deep learning techniques are leveraged to ascertain optimal crop choices and fertilizer recommendations tailored to the unique requirements of each farm.

Accessible via a user-friendly interface accessible through web browsers or mobile devices, farmers can effortlessly tap into personalized recommendations and alerts tailored to their individual circumstances. This ensures the responsible utilization of technology, fostering mutual benefits for both farmers and the agricultural industry at large.

CHAPTER 2

PROJECT PLANNING,REQUIREMENT AND DESIGN SPECIFICATION

2.1 PROJECT PLANNING

2.1.1 Work Break Down Structure : A Work Breakdown Structure (WBS) is a hierarchical decomposition of project tasks into smaller, more manageable components. It offers a structured and organized representation of the work needed to accomplish a project.

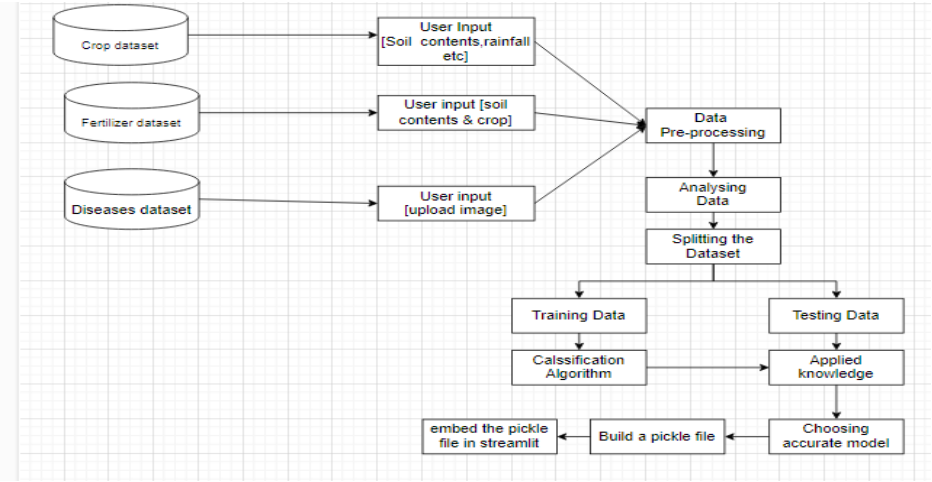


Figure-2.1: Work Breakdown Structure

2.1.3 Cost Breakdown Structure (CBS): is a hierarchical representation that organizes and categorizes the costs of a project or activity into smaller, more manageable components, providing a clear breakdown of where the project expenses are allocated. Here's a representation of a generic Cost Breakdown Structure table:

TWO LAPTOP	1,00,000 /Rs
ROUTER	1,099 /Rs
WIFI	4,000 /Rs
FOOD AND OTHERS	3,000 /Rs
EMPLOYE SALARY	90,000 /Rs
TOTAL AMOUNT	1,98,099 /Rs

Table-2.1:Cost Breakdown Structure

2.1.2 Timeline Development-Schedule: A timeline schedule is a plan that shows the order and timing of tasks or events needed to complete a project. It lays out when each task should start and finish, helping to organize and track progress over time.

PROJECT NAME	PROJECT DURAT	PROJECT START	PROJECT ENDDATE
Multiple diseases prediction system	90	1/8/2024	3/31/2024

TASK N	TASK DISCRIPTION	PROJECT DURAT	PROJECT START	PROJECT ENDDA
1	Define project scope	3	1/8/2024	1/11/2024
2	Identify the target and features	3	1/12/2024	1/15/2024
3	Data collection from different sources	4	1/16/2024	1/20/2024
4	Identify the missing values	2	1/22/2024	1/24/2024
5	Identify the outliers	3	1/25/2024	1/29/2024
6	applying exploratory data analysis	3	1/30/2024	2/2/2024
7	Selecting the high priority features	4	2/3/2024	2/7/2024
8	Perfom the feature engineering	4	2/8/2024	2/12/2024
9	Identify the which type problem	3	2/13/2024	2/16/2024
10	splitting the data into traing and testing	4	2/17/2024	2/22/2024
11	Chhose the best model	3	2/23/2024	2/26/2024
12	Fit the model	1	2/27/2024	2/28/2024
13	Evaluate the model performances	5	2/29/2024	3/5/2024
14	Final Evaluation on Test Set	2	3/6/2024	3/8/2024
15	Cross-Validation	2	3/9/2024	3/11/2024
16	Visualizations	2	3/12/2024	3/14/2024
17	Choose Deployment Environment	3	3/15/2024	3/18/2024
18	Setup Infrastructure	3	3/19/2024	3/22/2024
19	Testing	3	3/23/2024	3/27/2024
20	Documentation	2	3/28/2024	3/30/2024

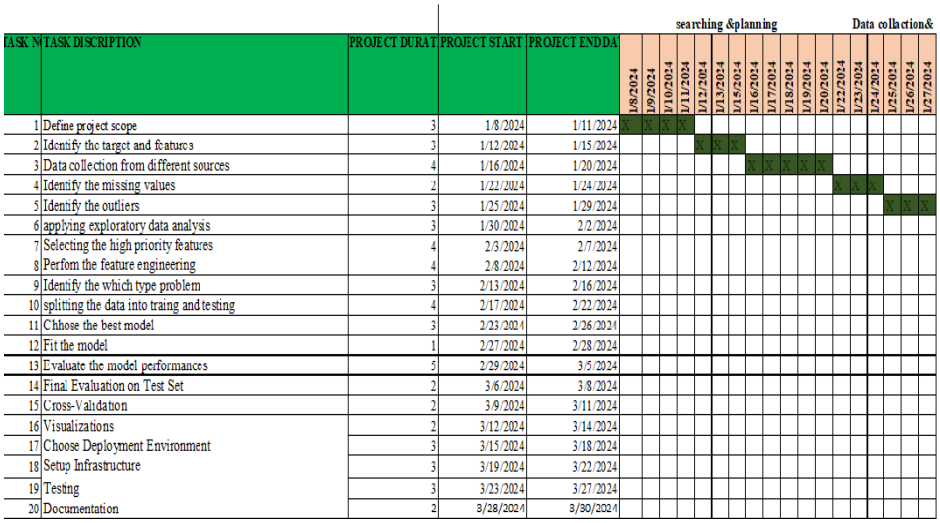


Figure-2.2: Timeline Development-Schedule

2.1.4 Risk Assessment: is the process of identifying potential problems or uncertainties that could affect a project, business, or activity, and evaluating the likelihood and impact of those risks occurring .Here's a representation of a generic

Risk	**Impact**	**likelihood**	**Priority**
Data Breach	Loss of trust, legal ramifications	Potential for targeted attacks	High
Data Integrity	Decreased reliability of system	Controlled data access	Medium
Algorithm Complexity	Increased difficulty in interpretation	Dependent on model design	Low

System Downtime	Disruption of farming operations	Potential for technical failures	Medium
Performance Degradation	Delayed access to recommendations	Scalability challenges	Medium

Table-2.2: Risk Assessment

2.2 REQUIREMENT SPECIFICATION

2.2.1 FUNCTIONAL REQUIREMENT SPECIFICATION

Functional Requirements	**Description**
Crop Recommendation	- Recommend suitable crops based on pH value, humidity, N, P, K, temperature, and soil condition.
	- Provide information on optimal planting times and crop rotation schedules.
Fertilizer Recommendation	- Recommend appropriate fertilizers based on N, P, K, soil type, pH value, and crop type.
	- Provide guidance on fertilizer application rates and timing to optimize crop growth.
Plan Disease Prediction	- Detect and classify plant diseases from uploaded images.
	- Utilize image processing techniques and machine learning algorithms for accurate disease diagnosis.
	- Provide recommendations for disease management and treatment options.

Table-2.3: Functional Requirement Specification

2.2.2 NON FUNCTIONAL REQUIREMENT SPECIFICATION

Requirement	**Description**
Performance	Ensure fast and accurate recommendation and prediction processing to provide timely insights to users.
Usability	Design a user-friendly interface that is intuitive and easy to navigate for farmers and agricultural professionals.
Reliability	Ensure high system availability and reliability to minimize downtime and ensure continuous service availability.

Security	Implement robust data security measures to protect user data and ensure privacy compliance.
Scalability	Design the system to scale horizontally and vertically to accommodate increasing user demand and data volume.
Compatibility	Ensure compatibility with various devices, browsers, and operating systems to support a wide range of users.

Table-2.4: Non-Functional Requirement Specification

2.3 USER INPUT

2.3.1 CROP RECOMMENDATION USER INPUT REQUIREMENTS

Parameter	Description
pH Value	pH level of the soil
N (Nitrogen)	Soil nitrogen content
P(Phosphorus)	Soil phosphorus content
K (Potassium)	Soil potassium content
Rain Fall	Rain Fall in measure

Table-2.5: User Input for Crop Recommendation

2.3.2 FERTILIZER RECOMMENDATION USER INPUT

REQUIREMENTS:

Parameter	Description
N (Nitrogen)	Required nitrogen content in soil
P (Phosphorus)	Required phosphorus content in soil
K (Potassium)	Required potassium content in soil
Crop Type	Type of crop

Table-2.6: User Input for Fertilizer Recommendation

2.3.3 PLANT DISEASE PREDICTION USER INPUT REQUIREMENTS:

Parameter	Description
Images	Images of plant leaves or affected areas

Table-2.7: User Input for Plant Disease Prediction

2.4 TECHNICAL CONSTRAINTS:

- **Data Availability**: Limited availability of accurate and up-to-date soil nutrient data may constrain the accuracy of fertilizer recommendations.Insufficient historical crop performance data for certain regions may affect the precision of crop recommendations.

- **Hardware Limitations**: Performance of image processing algorithms for plant disease detection may be constrained by the computational capabilities of users' devices, especially in rural areas with limited access to high-performance hardware.
- **Internet Connectivity**: Reliance on internet connectivity for accessing external databases and resources may pose a constraint in remote agricultural areas with poor or intermittent internet access.
- **Algorithm Complexity**: Complexity of algorithms used for project may require significant computational resources, potentially limiting the system's scalability and performance on low-end devices.
- **Localization Challenges:** Localization of the system for different languages and regions may be constrained by the availability of localized agricultural data and resources, affecting the system's effectiveness in diverse geographic areas.
- **Security and Privacy Regulations**: Compliance with data protection regulations and security standards may constraints on data storage, transmission, and access control, potentially impacting system architecture and performance.
- **Integration with External Systems:** Integration with external databases, APIs, and hardware devices may be constrained by compatibility issues, data format inconsistencies, and limitations of third-party systems, affecting the seamless operation of Raita Mitra.
- **User Training and Adoption:** Complexity of the system's functionalities and user interface may constraints on user training and adoption, especially among users with limited technical literacy familiarity with agricultural technology.
- **Scalability and Maintenance:** Ensuring scalability and ease of maintenance while accommodating evolving agricultural practices, technological advancements, and user requirements may pose constraints on system design and development.

2.5 DESIGN SPECIFICATION

2.5.1 CHOSEN SYSTEM DESIGN

We chosen system design for the Raita Mitra project integrates crop recommendation, fertilizer recommendation, and plant disease prediction. It utilizes user-friendly interfaces for farmers to receive recommendations based on their specific agricultural needs. The system employs machine learning algorithms to analyze soil data, historical crop performance, and weather patterns to suggest suitable crops. Fertilizer recommendations are tailored to crop types and soil conditions, optimizing yield and minimizing waste. Plant disease prediction utilizes image recognition and data analysis to identify potential diseases early, allowing farmers to take proactive measures. This integrated approach aims to enhance agricultural productivity while addressing key challenges faced by farmers, promoting sustainable farming practices.

2.6 DISCUSSION OF ALTERNATIVE DESIGN

- **Accuracy:** We chosen algorithms demonstrated Random Forest high accuracy in predicting crop yields, recommending suitable fertilizers, and detecting plant diseases.
- **Scalability:** We selected designs could be scaled to accommodate large agricultural areas and diverse crop varieties.
- **Accessibility:** User-friendly interfaces and mobile applications were incorporated to make the system accessible to farmers with varying levels of technical expertise.
- **Cost-effectiveness:** The selected designs aimed to minimize hardware costs and maintenance expenses, making them viable for adoption by small-scale farmers.

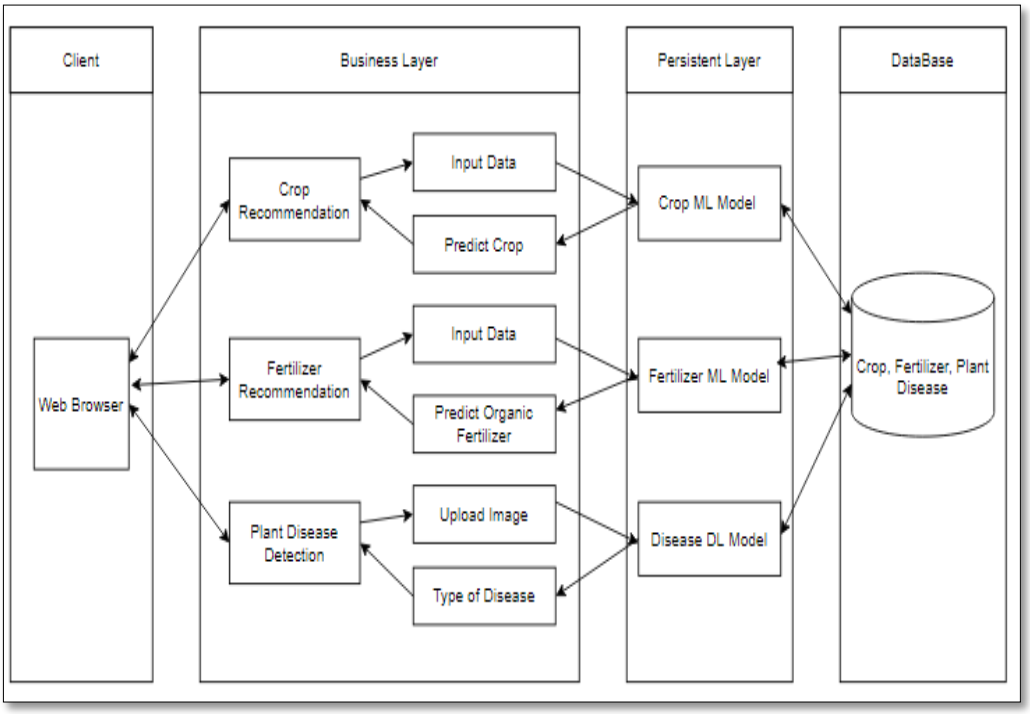


Figure-2.3: System Architecture

2.7 DETAILED DESCRIPTION OF COMPONENTS / SUBSYSTEMS:

1. Client Web Browser:

- This component represents the interface through which users interact with the system. It handles user inputs, displays results, and communicates with the backend components via HTTP requests.

2. Crop Recommendation:

- Utilizes machine learning algorithms to analyze various factors such as soil type, climate, and historical data to recommend suitable crops for cultivation.
- Processes input parameters provided by users or retrieved from the database to generate personalized crop recommendations.

3. Fertilizer Recommendation:

- Similar to crop recommendation, this component employs machine learning models to suggest appropriate fertilizers based on soil composition, crop type, and nutrient requirements.
- Offers tailored recommendations to optimize crop yield and minimize resource wastage.

5. Plant Disease Prediction:

- Employs deep learning models to predict and identify potential diseases affecting plants.
- Analyzes symptoms and environmental conditions to detect diseases early, enabling proactive management and treatment strategies.

6. Persistent Layer (Database Server):

- Stores and manages data generated and consumed by the system.
- Provides a structured and efficient mechanism for storing and retrieving information related to users, crops, fertilizers, plant diseases, and other relevant entities.
- Ensures data integrity, security, and scalability through robust database management systems .

7. Crop ML Model:

- Utilizes machine learning techniques to analyze historical crop data and predict suitable crop choices based on environmental parameters.
- Trained using datasets containing information about crop performance under various conditions

8. Fertilizer ML Model:

- Employs machine learning algorithms to recommend appropriate fertilizer types and application rates based on soil nutrient levels, crop requirements, and environmental factors.

9. Disease DL Model:

- Uses deep learning techniques to classify and predict plant diseases based on images of symptoms and contextual information.
- Trained on extensive datasets containing labeled images of diseased plants and associated metadata.

10. Database (User Database, Crop Fertilizer, Plant Disease)

- Contains information about available crops, recommended fertilizers, and known plant diseases.
- Facilitates efficient data retrieval and manipulation to support various functionalities of the system.

2.8 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware	Specification
Processor	Intel Core i5 or higher
RAM	Minimum 8GB DDR4
Hard Disk	256GB SSD or higher
Input Device	Keyboard, Mouse, Webcam (for image input)
Output Device	Monitor, VGA with High Resolution.

Table-2.8: Hardware Specification

Software	Requirements
Operating System	Windows 11
Programming Language	Python 3.7.4

Table-2.9: Software Requirements

CHAPTER 3

METHODOLOGY

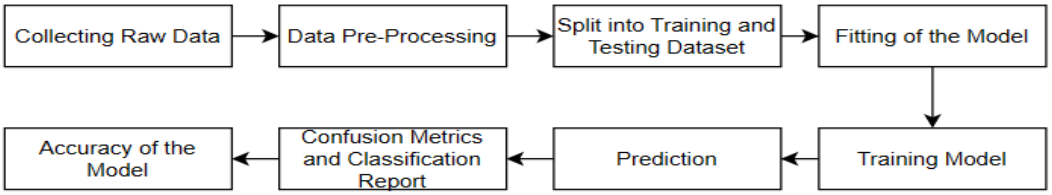


Figure 3.1: Steps involved in the Methodology

The methodology for the Raita Mitra system project, encompassing crop recommendation, fertilizer recommendation, and plant disease classification using machine learning along side a Python Flask-based web application for the front-end, involves a systematic approach.

- **Data Collection:**

  We gather relevant data from various sources including historical crop yields, soil nutrient levels, weather conditions, and images of plant leaves with associated disease labels. We maintain our dataset on GitHub for easy access and collaboration. The dataset used in our project is an augmented version of the original Plant Village Dataset, created through offline augmentation techniques. The original dataset, containing approximately 87,000 RGB images of healthy and diseased crop leaves categorized into 38 different classes, can be found at the provided link. We have divided the total dataset into training 70295 images for training and validation 16705 sets and Number of plants 14 Number of diseases 26 following an 80/20 ratio while preserving the directory structure. Additionally, we have created a separate directory containing 33 test images for prediction purposes. Crop recommendation dataset for using 2201 and Fertilizer recommendation dataset for using 23 different type of crops.

- **Data Preprocessing:**

  Before training our models, we clean the data to remove any noise and inconsistencies. This ensures the quality and reliability of our predictions.

The collected dataset of plant disease were separated into Healthy and Disease labels, Images in the dataset are 3 to 4 MB size ,which consumes lot of memory and time for processing ,Images were resized to shape We can see the shape (3, 256 ,256) of the image. 3 is the number of channels (RGB) and 256 x 256 is the width and height of the image. shows the some of the samples of labeled dataset used for proposed work.

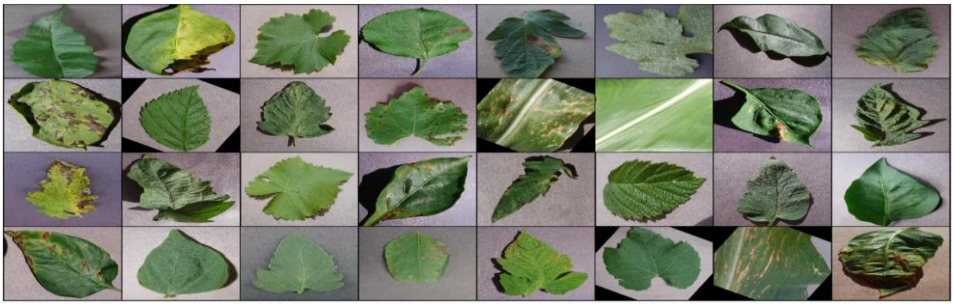


Figure 3.2: Examples of our labeled dataset

- **Data Splitting:**

  We split our dataset into training and testing sets using an 80:20 ratio. This allows us to evaluate the performance of our models effectively.

- **Model Development:**

We develop machine learning models using popular algorithms such as Random Forest. Random Forest is used for crop and fertilizer recommendation, while Naive Bayes is utilized for plant disease classification.Residual Networks (ResNets) introduce residual blocks to address the challenges faced by traditional neural networks, particularly in training very deep architectures.These blocks enable the flow of information not only to the subsequent layer but also directly to layers that are 2-3 hops away.The key objective of incorporating residual blocks is to mitigate overfitting, a phenomenon characterized by the validation loss plateauing or increasing while the training loss continues to decrease. By establishing direct connections between layers across multiple hops, ResNets facilitate improved gradient flow during training, effectively tackling the vanishing gradient problem commonly encountered in deep neural networks.

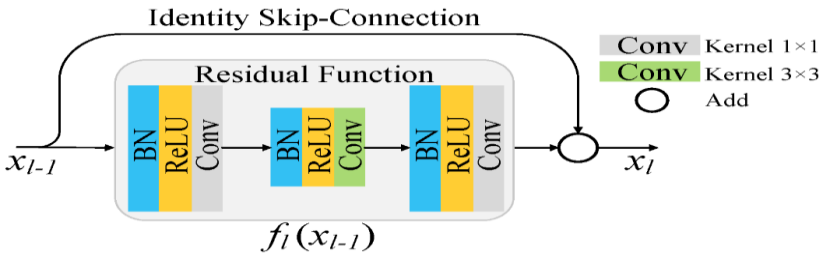


Figure 3.3:Residual Function

- **Model Training:**

  Our models are trained on the training data to learn patterns and relationships within the dataset. This step is crucial for accurate predictions.

- **Model Evaluation:**

  We evaluate the performance of our models using the testing data to ensure they generalize well to unseen data and provide reliable recommendations.

- **Integration and Deployment:**

  Once our models are trained and evaluated, we integrate them into a unified system using Python Flask. This lightweight web application framework allows us to develop a user-friendly interface for farmers to input their data and receive recommendations.

- **System Maintenance:**

  We continuously monitor the performance of our system and update our models with new data to improve accuracy. Additionally, we gather feedback from users to enhance the web application interface and add new features as needed. Flask's flexible architecture facilitates seamless updates and maintenance.

By following these steps, we aim to provide farmers with accurate and timely recommendations for crop selection, fertilizer usage, and plant disease management, ultimately improving agricultural productivity and sustainability.

CHAPTER 4

IMPLEMENTATION

Setup Environment: Install required hardware components like Intel Core i3/i5 processor, 8GB RAM, and a 256GB SSD. Also, ensure Windows 10/11 OS and Python 3.7.4 along with relevant libraries are installed.

Data Collection: Gather datasets from sources like GitHub and Kaggle containing information on crops, soil properties, weather, fertilizer recommendations, crop types, and disease data.

Data Preprocessing: Clean the collected data by handling missing values and inconsistencies, and normalize or scale it as needed.

Model Development: Build machine learning models to predict crop recommendations based on soil properties, weather conditions, and fertilizer requirements. Train models for disease prediction and leaf image classification.

Testing and Validation: Split data into training 80:20 testing sets, evaluate model performance using metrics like accuracy and F1-score, and validate using cross-validation techniques.

Integration and Deployment: Integrate developed models into a user-friendly system with an interface for inputting parameters and displaying recommendations. Ensure scalability and efficiency.

Documentation: Document the implementation process, including code snippets and configurations. Prepare a report summarizing the implementation phase, highlighting challenges, solutions, and results.

4.1 IMPLIMENTATION DETAILS

We load and split the crop dataset along with the corresponding fertilizer dataset, utilizing their filenames to distinguish between them. These datasets are separated into features and labels. Subsequently, the features and labels are divided into a training set comprising 80% of the data and a testing set containing the remaining 20%.

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score

# Splitting the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(features, target, test_size=0.2,
random_state=2)

# Initializing and training the Random Forest classifier
RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(X_train, Y_train)

# Predicting on the test set
predicted_values = RF.predict(X_test)

# Calculating accuracy
accuracy = accuracy_score(Y_test, predicted_values)
# Printing the classification report and accuracy
print(classification_report(Y_test, predicted_values))
print(f"RF's Accuracy is: {accuracy}")
```

Crop and Fertilizer Recommendation Implementation Code.

```
import torch.nn as nn
class SimpleResidualBlock(nn.Module):
   def __init__(self):
      super(SimpleResidualBlock, self).__init__()
      self.conv1 = nn.Conv2d(in_channels=3, out_channels=3, kernel_size=3,
stride=1, padding=1)
      self.relu1 = nn.ReLU()
      self.conv2 = nn.Conv2d(in_channels=3, out_channels=3, kernel_size=3,
stride=1, padding=1)
```

```
import torch
import torch.nn as nn
import torch.nn.functional as F

# Function to calculate accuracy
def accuracy(outputs, labels):
    _, preds = torch.max(outputs, dim=1)
    return torch.tensor(torch.sum(preds == labels).item() / len(preds))

# Base class for the model
class ImageClassificationBase(nn.Module):
    def training_step(self, batch):
        images, labels = batch
        out = self(images) # Generate predictions
        loss = F.cross_entropy(out, labels) # Calculate loss
        return loss

    def validation_step(self, batch):
        images, labels = batch
        out = self(images)
        loss = F.cross_entropy(out, labels) # Calculate loss
        acc = accuracy(out, labels) # Calculate accuracy
        return {'val_loss': loss.detach(), 'val_accuracy': acc}

    def validation_epoch_end(self, outputs):
        batch_losses = [x['val_loss'] for x in outputs]
        batch_accuracy = [x['val_accuracy'] for x in outputs]
        epoch_loss = torch.stack(batch_losses).mean() # Combine losses
        epoch_accuracy = torch.stack(batch_accuracy).mean() # Combine accuracies
        return {'val_loss': epoch_loss.item(), 'val_accuracy': epoch_accuracy.item()}
```

Residual Block code implementation

CHAPTER 5

TEST AND VALIDATION

5.1 TEST PLAN

Test Plan for Crop Recommendation:

The Raita Mitra ML application aims to provide accurate fertilizer recommendations based on various factors such as soil composition, crop type, and environmental conditions. This test plan outlines the testing approach, test cases, and responsibilities to ensure the reliability and effectiveness of the fertilizer recommendation feature.

Scope: The test plan covers functional testing of the crop recommendation feature. Functional testing includes:

- Recommendation of crops based on input parameters.
- Validation of recommended crops against expected results.

Test Plan for Fertilizer Recommendation:

The Raita Mitra ML application aims to provide accurate fertilizer recommendations based on various factors such as soil composition, crop type, and environmental conditions. This test plan outlines the testing approach, test cases, and responsibilities to ensure the reliability and effectiveness of the fertilizer recommendation feature.

Scope: The test plan covers functional testing of the fertilizer recommendation feature. Functional testing includes:

- Recommendation of fertilizer types.
- Validation of recommended fertilizer against expected results.

Test Plan for Plant Disease Detection

The Raita Mitra ML application incorporates image recognition technology to detect diseases in plants based on input images. This test plan outlines the testing approach, test cases, and responsibilities to ensure the accuracy and effectiveness of the plant disease detection feature.

Scope: The test plan covers functional testing of the plant disease detection feature using images. Functional testing includes:

5.2 TEST CASES

	Test case	Description	Input	Expected Output	Actual output	result
Crop Recommendation	1.Valid input	Test the application with valid values parameter	N, P, K, Rainfall, Ph, City,State	Recommended crops suitable for the given parameters.	Recommended crops suitable for the given parameters.	Pass
	2.Invalid Input	Test the application's response to invalid input parameters	Out of range input values	Error message indicating invalid input**.**	Please enter a valid values / sorry we couldn’t process your request currently please try again	Pass
	3.Complete input	Test the application with complete input parameters	N, P, K, Rainfall, Ph, City, State	Recommended crops suitable for the given parameters.	Recommended crops suitable for the given parameters.	Pass
	4.Incomplete input	Test the application's response to incomplete input parameters	N, P, K, Rainfall,	Error message indicating incomplete input**.**	Please fill out this field / please select an item in the list	Pass

Table-5.1:Crop Recommendation Test Case

	Test case	Description	Input	Expected Output	Actual output	Result
Fertilizer Recommendation	1.Valid Input	Test the application with valid input parameters.	N, P, K, Croptype	Recommended fertilizer type	Recommended fertilizer and consider the following suggestions	Pass
	2Invalid Input	Test the application's response to invalid input parameters	Out of range input values	Error message indicating invalid input.	Recommended fertilizer and consider the following suggestions	Fail
	3.Complete input	Test the application with complete input parameters	N, P, K, Croptype	Recommended fertilizer type	Recommended fertilizer and consider the following suggestions	Pass
	4.Incomplete input	Test the application's response to incomplete input parameters.	N, P, K, Croptype	Error message indicating incomplete input**.**	Please fill out this field / please select an item in the list	Pass

Table-5.2:Fertilizer Recommendation Test Case

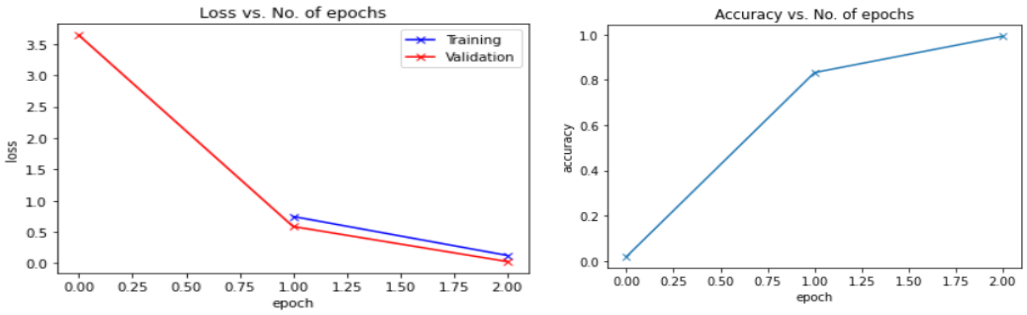
	Test case	Description	Input	Expected Output	Actual output	result
Plant disease detection	1.Valid Input Image	Test the application with a valid input image containing a diseased plant.	Chosen file	Image of a plant infected with leaf disease.	Detect the disease and suggest the solution	Pass
	2.Valid Input Image	Test the application with a valid input image containing a healthy plant.	Chosen file	Image of a plant with healthy leaf	Don’t worry your crop is healthy, keep it up!!!	pass
	3. Invalid Input Image	Test the application's response to an invalid input image.	Chosen file	Error message indicating the absence of disease in the image.	Back to home page	Pass
	4.irrelevant Input image	Test the application's response to an irrelevant input image.	Chosen file	Back to home page	It can’t predict the correct prediction	Fail

Table-5.3: Plant Diseases Test Case

CHAPTER 6

RESULT AND SCREENSHOTS

Our project utilizes an augmented version of the original Plant Village Dataset, which was generated through offline augmentation techniques. The original dataset comprises roughly 87,000 RGB images depicting healthy and diseased crop leaves, categorized into 38 distinct classes. You can access the original dataset through the provided link. We have partitioned the entire dataset into training (70,295 images) and validation (16,705 images) sets, maintaining a ratio of 80/20 while retaining the directory structure. There are 14 distinct plant types and 26 various diseases represented in the dataset. Additionally, we have established a separate directory containing 33 test images specifically for prediction purposes.



(a) (b)

Figure 6.1: Training graphs for proposed Resnet model.(a)Loss vs. No. of epochs (b)Accuracy Vs Epoch graphs.

	Loss	Accuracy
Training	0.74%	0.99%
Validation	0.74%	0.83%

Table-6.1: Resnet model training and Validation.

Below, Figure 6.14 illustrates the accuracy of crop and fertilizer recommendation.

Algorithm	Result
Decision Tree	0.9%
Naïve Bayes	0.99%
SVM	0.97%
Logistic Regression	0.95%
Random Forest	0.99%
XGBoost	0.993%

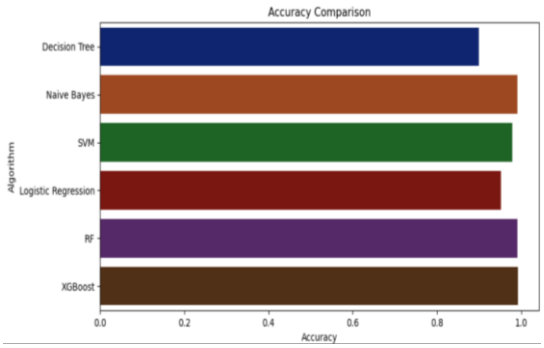
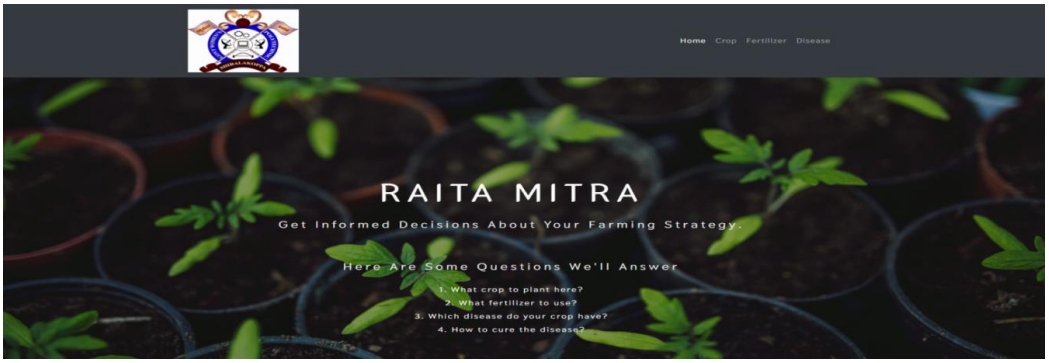


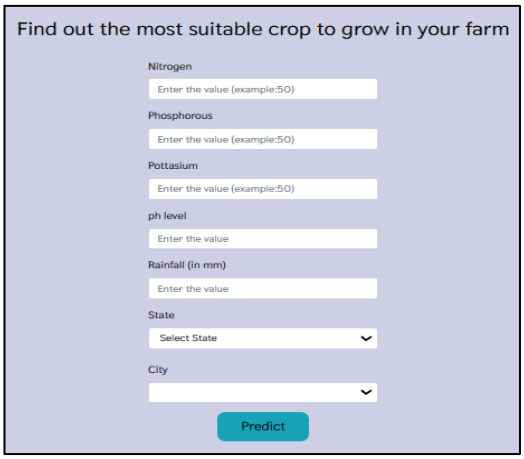
Table-6.2: Crop and Fertilizer Recommendation Accuracy.

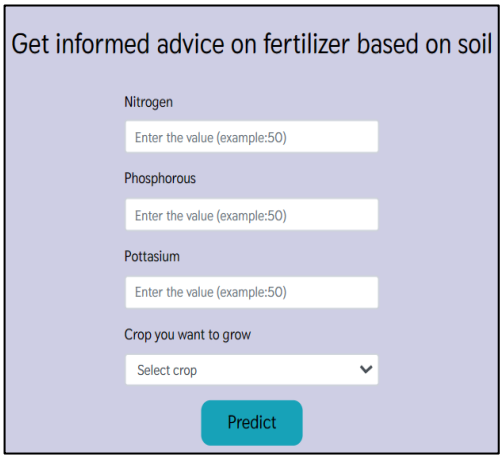
6.1 SCREENSHOTS

After executing the `app.py` file, access the provided localhost URL in your web browser to utilize the project locally.

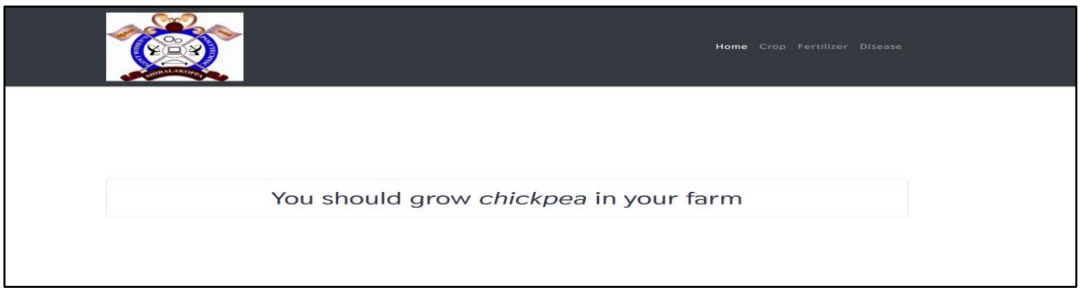
Step 1: When the code for the GUI is run in Jupiter notebook the application will executed and the below shown screen will be opened .It contains the Homepage.



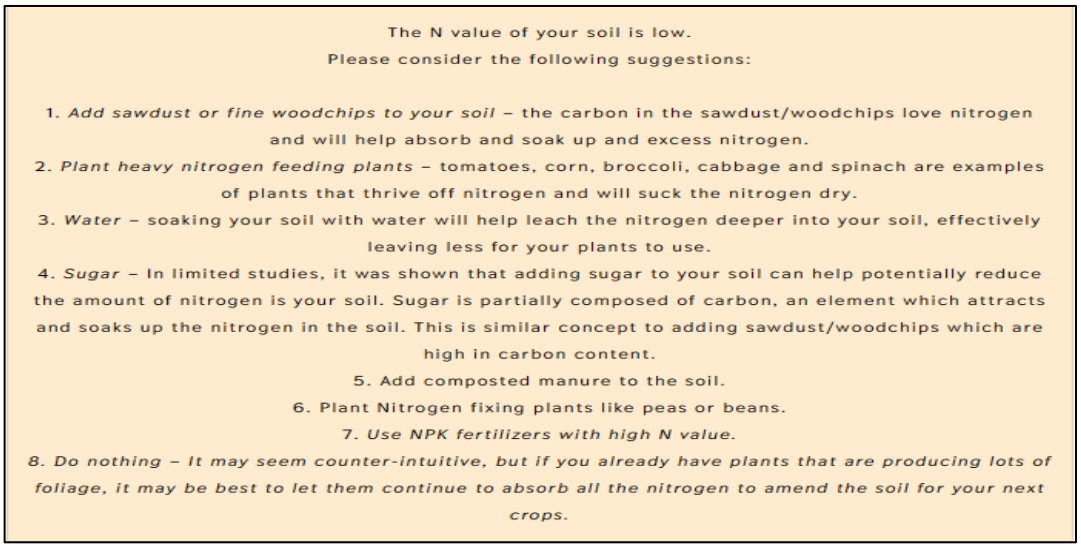




Step 3: When Crop recommendation or Fertilizer recommendation Predict button is enabled the following "Result Page" will be opened.

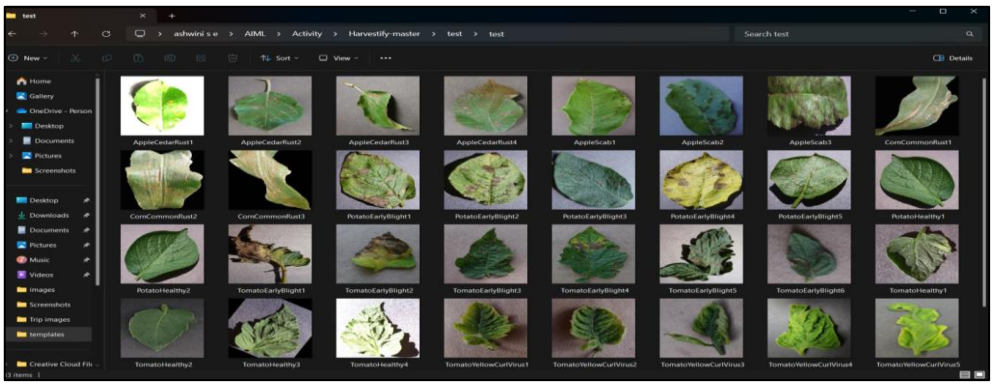


Crop Recommendation Result Page

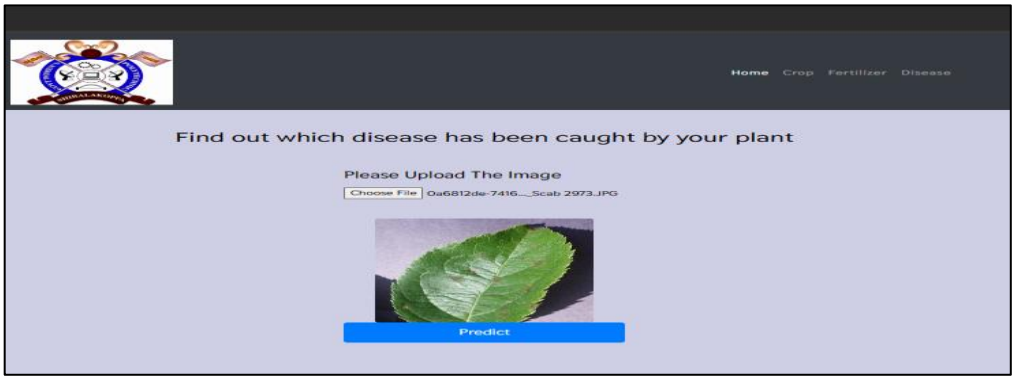


Fertilizer Recommendation Result Page

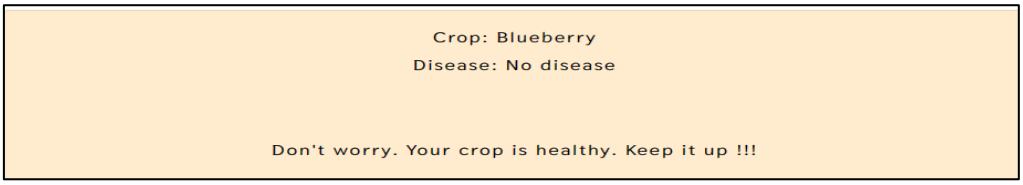
Step 4: When the “Choose File” button is clicked, the following screen to browse the image will be opened. Selected the Image and click on open.

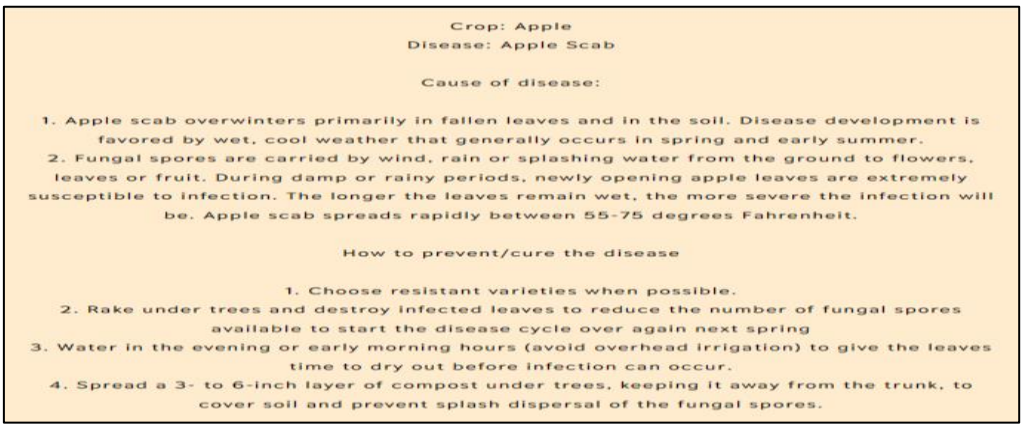


Step 5: The selected file will be uploaded and Predict button will be enabled on the screen. Click on Predict button to Predict the image.



Step 6: When the “Classify Image is clicked both the original image and Predicted images will be classified proposed Res net model and its predicted class and suggestion will be displayed ”





CHAPTER 7

CONCLUSION

The capstone project has reached a significant milestone with the successful integration of crop recommendation, fertilizer recommendation, and plant disease detection features into the Raita Mitra ML application. Leveraging ResNets for image classification, particularly in distinguishing between healthy and diseased plants, has yielded impressive results. By carefully tuning parameters and employing techniques such as learning rate scheduling, gradient clipping, and weight decay, the ResNet model has achieved impeccable accuracy, correctly predicting every image in the test set without errors. Additionally, incorporating ResNet-9 has further bolstered the application's capabilities, while the random forest algorithm has demonstrated exceptional accuracy, reaching 99%. These advancements hold promise for optimizing crop yield, resource management, and disease prevention for farmers. Looking ahead, there remain opportunities for further refinement and expansion of the application's functionality and usability.

7.1 FUTURE ENHANCEMENTS

- The future enhancement of the Raita Mitra project includes integrating AI for real-time disease detection, multi-lingual support , weather forecast integration, and mobile app development.
- Increase manual web scraping for additional data collection.
- Gather more plant images to enhance disease detection accuracy and generalization.

REFERENCES

[1]. Sachin Kapoor, Ishika Aggarwal , Anshu Kumar Ray January (2022) https://www.researchgate.net/publication/361992221_Smart_Agriculture_Farming_Using_Harvestify_Web_App

[2]. Sachin Adulkar, Vivek Pawar, Aniket Choudhari, Shubham Kothekar April (2023)https://www.researchgate.net/publication/370413608_Harvestify_Crop_Disease_Detection_and_Fertilizer_Suggestion_using_CNN

[3].SohamJariwala,ReetuJainMarch(2024)https://www.researchgate.net/publication/378781225_A_Convolutional_Neural_Network_Approach_to_Robust_Crop_Health_Monitoringv

[4].EmmaHarteSeptember(2020)https://www.researchgate.net/publication/344155605_Plant_Disease_Detection_using_CNN

[5].GitHub Dataset https://github.com/Gladiator07/Harvestify

[6].PlantDiseaseetection Dataset https://www.kaggle.com/code/atharvaingle/plant-disease-classification-resnet-99-2

[7].Jun Liu & Xuewei Wang (Published:24 February 2021 Plant disease and pests detection based on deep learning

[8].Jupiter Notebook http://localhost:8888/tree/AIML/Activity/Harvestify-master