Crop yield prediction using machine learning – Paddy Harvest Prediction

24-25J-125

2024

Team Members

Student Name	Student ID	Specialization
Jayathilaka D.H.R. A	IT21308352	IT
Piyumani K.V. P	IT21227868	IT
Amarasinghe A.I.S. A	IT21225192	IT
Jayasekara S.S. D	IT21227318	IT

Supervisor : Mr.Kanishka Yapa
Co - Supervisor : DR.Harinda Fernando

Overall Project Description

- Sri Lanka is experiencing an economic crisis due to unsustainable debt and persistent deficits, leading to a severe shortage of foreign currency. Agriculture, a vital sector, plays a crucial role in the economy and provides livelihoods for a significant portion of the population.

- Our research project aims to enhance paddy cultivation by leveraging advanced machine learning and image processing techniques. This includes predicting paddy yield, recommending optimal paddy varieties, and managing pests and weeds efficiently.

- By providing real-time, data-driven recommendations, the platform will help farmers optimize their practices, reduce losses, and promote sustainable farming, contributing to the economic stability and growth of Sri Lanka's agricultural sector.

Research Objectives

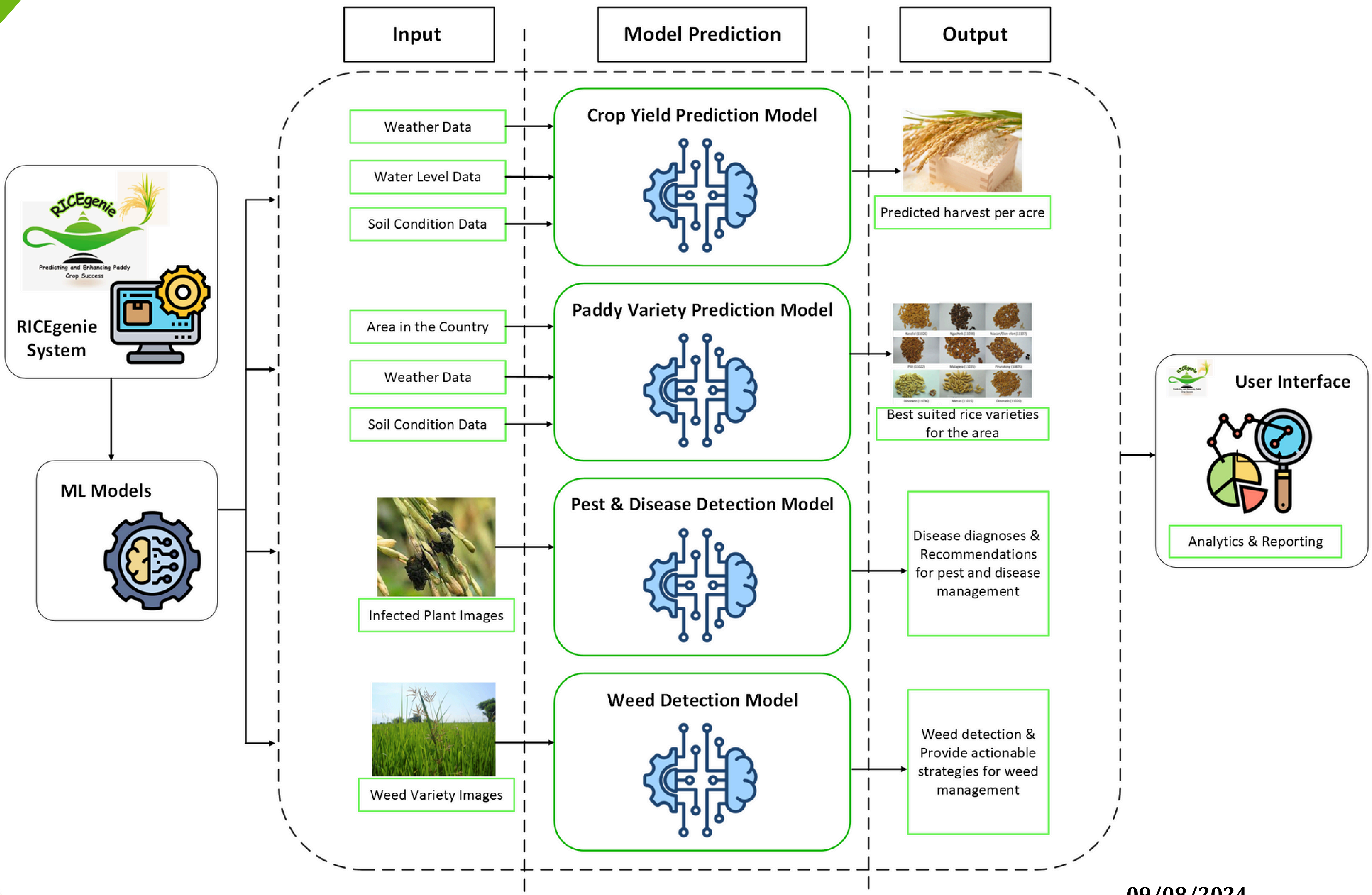
Main Objective:

- Develop an comprehensive platform to Enhance paddy cultivation in Sri Lanka by leveraging predictive models and image processing techniques.

Sub Objectives:

1. **Predict Harvest Yield:** Forecast paddy yields under varying conditions.
2. **Recommend Optimal Paddy Varieties:** Suggest suitable paddy types based on local conditions.
3. **Detect and Manage Pests and Diseases:** Use image processing for timely identification and treatment.
4. **Identify and Control Weeds:** Detect weed varieties and offer control strategies.

Overall System Diagram



REFERENCES

1. CropWat - https://www.fao.org/land-water/databases-and-software/cropwat/en/
2. STICS - https://www.quantitative-plant.org/model/STICS
3. APSIM - https://www.apsim.info/
4. CERES - https://ceresglobalagcorp.com/about/who-we-are/

IT21308352 | Jayathilaka D.H.R.A

Crop Yield Prediction System

Information Technology

SLIIT FACULTY OF COMPUTING

INTRODUCTION

- **Objective**: Develop a robust crop yield prediction model for paddy crops using machine learning.
- **Data Integration**: Analyze a comprehensive dataset, including historical yield data, weather conditions, soil health, and irrigation patterns.
- **Recommendations**: Incorporate a decision support framework to provide actionable insights and recommendations.
- **Benefits**: Enhance yield predictions, optimize agricultural practices, and improve resource management for increased productivity.

Challenges Encountered

- **Lack of Integration in Existing Models :**Current models do not integrate critical factors such as weather, water, and soil conditions effectively.

- **Feature Selection:** Determining which features are most relevant to yield prediction.

- **Model Accuracy:** This lack of comprehensive data integration leads to inaccuracies in yield predictions.

- **Scalability issues:** Many approaches do not scale well for large agricultural fields.

- **Lack of user-friendly interfaces and lack of Harvest prediction System:** There is a need for more emotionally and easy-to-use interfaces for farmers .

Specific and Sub Objectives

- **Specific Objectives :**
  - To provide the yield per hectare and provide suitable recommendations for farmers.

- **Sub Objectives :**
  - Collect historical data on paddy crop yields from Rice Research and Development Institute Bathalagoda.
  - Develop key features and train a machine learning model for accurate yield predictions.
  - Integrate the model into a decision support system with a user-friendly interface for farmers.

Methodology

- **Data Collection:** Gather historical data on paddy yields, weather conditions, soil health, and irrigation patterns from reliable sources.
- **Data Analysis**: Analyze the collected data to identify patterns and relationships between soil/climatic factors to predict the harvest
- **Pre-processing** :Prepare the dataset for model training by handling missing values, normalizing features, and encoding categorical variables.
- **Model Training :**Train a machine learning model using the processed dataset and fine-tune it for optimal performance.
- **Model Validation :** Split the dataset into training, validation, and test sets. Evaluate model performance using metrics like RMSE, MAE, and R^2.
- **Implementation:** Develop a user-friendly interface for farmers to input their local conditions and provide the harvest that they got and provide recommendations to increase the harvest

Data Overview

- **Dataset:** 10,000 entries, containing key features for paddy harvest prediction.
- **Features:** Likely include attributes such as area (in acres), seed variety, district, soil quality, weather conditions, and historical yields.
- **Goal:** Increase diversity and robustness of the model.

Rainfall (mm)		Relative Hu	Sunshine H	Wind Spee	Soil Type	Irrigation T	Water Sou	Paddy Vari	Fertilizer U	Area (hect	Soil Nitrog	Soil Phospl	Soil Potass	Pest Sever	Season	District	Yield (kg)
1407.3	32.2	60.2	9.1	11.3	Sandy	Rainfed	Well	BG 450	34	6.2	46	49	35	High	Maha	Polonnaru	38018.4
1412.7	30.7	66	7	10.9	Clay	Rainfed	Rainwater	BG 360	51	10.9	49	10	130	Medium	Yala	Jaffna	39752.3
1211	33.2	63.9	8.1	6.1	Loam	Tube Well	Well	BG 350	50.7	12.7	74	40	123	High	Maha	Batticaloa	38912.8
940.3	30.4	86.9	8.7	14.8	Loam	Rainfed	Well	BG 450	71	10.9	13	8	88	High	Yala	Batticaloa	38041
899.8	32.5	86.5	11	14.5	Loam	Rainfed	Well	BG 450	69.9	8.9	53	28	175	Medium	Maha	Polonnaru	39035.4
994.5	27.6	61.4	7.4	2.9	Loam	Canal	River	BG 250	94.2	15.2	32	20	122	Medium	Yala	Anuradhap	39884.8
900.4	32.9	64.8	11.2	12.4	Clay	Tube Well	Rainwater	BG 450	39.8	6	76	10	87	Medium	Maha	Hambanto	36228
1344.1	32.8	73.5	9.9	12	Loam	Rainfed	Rainwater	BG 350	22.9	17.2	98	49	51	High	Maha	Polonnaru	38029.2
1805.5	33.2	62.2	7.9	16.3	Loam	Rainfed	Rainwater	BG 250	81.1	14.3	96	8	76	Low	Maha	Mannar	42799.9
1196.5	29.3	79.2	11.7	7.7	Clay	Canal	River	BG 450	98.9	8.2	61	25	134	Low	Yala	Trincomale	41779
1004.3	25.1	84.6	10.2	13.5	Sandy	Canal	River	BG 350	27.6	18.2	20	45	72	Medium	Yala	Trincomale	36946
1350.7	33.5	89	7.6	13.8	Sandy	Tube Well	Well	BG 360	96.9	18.5	35	6	157	Low	Yala	Trincomale	42013.5
1150.1	25.6	72.8	7	13.6	Clay	Tube Well	River	BG 360	75.2	5.5	31	47	15	Medium	Yala	Hambanto	39336
1918.7	27.3	60.8	10.2	1.1	Loam	Rainfed	River	BG 350	51.7	17.4	99	39	183	Low	Maha	Polonnaru	43882.8
976.8	30.1	88.8	8.6	14.4	Sandy	Canal	Well	BG 250	31.5	18.4	29	35	17	High	Yala	Anuradhap	36027.2
1062.1	32.9	61.6	8.8	17.7	Clay	Canal	Well	BG 360	81.3	11.2	54	42	62	High	Maha	Jaffna	38897.6
800.6	26.4	83.2	11.5	15.6	Sandy	Tube Well	Rainwater	BG 350	42.8	5	67	26	186	Medium	Yala	Trincomale	37955
1975.9	32.9	84.8	11.8	1.9	Loam	Rainfed	Rainwater	BG 350	25.7	9.7	76	48	150	Low	Maha	Batticaloa	42311.4
1792.5	28.1	68	10.9	16.8	Clay	Canal	Rainwater	BG 450	53	9.4	49	31	188	Medium	Yala	Polonnaru	42845.2
1400	29.2	86.5	8.5	16.1	Clay	Canal	Well	BG 360	80	12.8	35	48	62	Medium	Maha	Batticaloa	41254.4
1200.7	33.9	65.2	10	9.1	Loam	Tube Well	Well	BG 250	45	5.5	23	27	180	High	Yala	Anuradhap	38467
1007.7	25.5	66.3	10.8	11	Loam	Canal	Well	BG 450	95	9.2	73	34	180	Medium	Maha	Hambanto	41436.8
896.6	30.3	87.3	10.8	1.4	Sandy	Tube Well	River	BG 250	76.1	15.2	53	37	109	High	Maha	Jaffna	38988
1299.7	32.9	81.3	7.1	8.9	Clay	Canal	Well	BG 450	21.1	15.9	98	10	183	Low	Maha	Ampara	38796
1611.5	30.7	86	8.1	17.7	Loam	Rainfed	River	BG 360	66.8	2.7	52	42	82	High	Maha	Anuradhap	4180.68

Data Analysis

```
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Rainfall (mm)           10000 non-null  float64
 1   Temperature (°C)        10000 non-null  float64
 2   Relative Humidity (%)   10000 non-null  float64
 3   Sunshine Hours (hrs)    10000 non-null  float64
 4   Wind Speed (km/h)       10000 non-null  float64
 5   Soil Type               10000 non-null  object
 6   Irrigation Type         10000 non-null  object
 7   Water Source            10000 non-null  object
 8   Paddy Variety           10000 non-null  object
 9   Fertilizer Usage (kg)   10000 non-null  float64
 10  Area (hectare)          10000 non-null  float64
 11  Soil Nitrogen (mg/kg)   10000 non-null  int64
 12  Soil Phosphorus (mg/kg) 10000 non-null  int64
 13  Soil Potassium (mg/kg)  10000 non-null  int64
 14  Pest Severity           10000 non-null  object
 15  Season                  10000 non-null  object
 16  District                10000 non-null  object
 17   Yield (kg)             10000 non-null  float64
dtypes: float64(8), int64(3), object(7)
memory usage: 1.4+ MB
```

	Rainfall (mm)	Temperature (°C)	Relative Humidity (%)	Sunshine Hours (hrs)	Wind Speed (km/h)	Soil Type	Irrigation Type	Water Source	Paddy Variety	Fertilizer Usage (kg)	Area (hectare)	Soil Nitrogen (mg/kg)	Soil Phosphorus (mg/kg)	Soil Potassium (mg/kg)	Pest Severity	Season	District	Yield (kg)
0	1407.3	32.2	60.2	9.1	11.3	Sandy	Rainfed	Well	BG 450	34.0	6.2	46	49	35	High	Maha	Polonnaruwa	38018.4
1	1412.7	30.7	66.0	7.0	10.9	Clay	Rainfed	Rainwater	BG 360	51.0	10.9	49	10	130	Medium	Yala	Jaffna	39752.3
2	1211.0	33.2	63.9	8.1	6.1	Loam	Tube Well	Well	BG 350	50.7	12.7	74	40	123	High	Maha	Batticaloa	38912.8
3	940.3	30.4	86.9	8.7	14.8	Loam	Rainfed	Well	BG 450	71.0	10.9	13	8	88	High	Yala	Batticaloa	38041.0

Pre-processing

```python
categorical_features = ['Soil Type', 'Irrigation Type', 'Water Source',
                        'Paddy Variety', 'Pest Severity', 'Season', 'District']

numerical_features = X.select_dtypes(include=['float64', 'int64']).columns.tolist()
print("numerical_features", numerical_features)

numerical_features = [col for col in numerical_features if col not in categorical_features]
print("numerical_features =", numerical_features)

# Preprocessing: Encode categorical features
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ],
    remainder='passthrough',  # Leave numerical features untouched
    force_int_remainder_cols=False  # Future-proof behavior
)

print("\nPreprocessor Details:")
print(preprocessor)
```

```
numerical_features ['Rainfall (mm)', 'Temperature (°C)', 'Relative Humidity (%)', 'Sunshine Hours (hrs)', 'Wind Speed (km/h)', 'Fertilizer Usage (kg)', 'Area (hectare)',
numerical_features = ['Rainfall (mm)', 'Temperature (°C)', 'Relative Humidity (%)', 'Sunshine Hours (hrs)', 'Wind Speed (km/h)', 'Fertilizer Usage (kg)', 'Area (hectare)'

Preprocessor Details:
ColumnTransformer(force_int_remainder_cols=False, remainder='passthrough',
                  transformers=[('cat', OneHotEncoder(handle_unknown='ignore'),
                                 ['Soil Type', 'Irrigation Type',
                                  'Water Source', 'Paddy Variety',
                                  'Pest Severity', 'Season', 'District'])])
```

Create Pipeline for Preprocessing and Model

Create Pipeline for Preprocessing and Model

```
[ ]  pipeline = Pipeline(steps=[
         ('preprocessor', preprocessor),  # Data preprocessing step (scaling, encoding, etc.)

         ('model', RandomForestRegressor(
             n_estimators=150,              # Number of trees in the forest (reduces variance and prevents overfitting by averaging predictions)
             max_depth=18,                  # Limits the depth of each tree to prevent overfitting by controlling model complexity
             min_samples_split=10,          # Requires at least 10 samples to split a node (prevents overly specific splits)
             min_samples_leaf=4,            # Ensures that each leaf node has at least 4 samples, preventing overfitting to small data variations
             random_state=42                # Ensures reproducibility of results (ensures the same splits each time for consistency in testing)
         ))
     ])
```

3. Model Development

Algorithm Selection

1. Random Forest Regressor

```
MAE: 510.0285261251583
MSE: 4755389.907179633
RMSE: 2180.685650702465
R-squared: 0.9730830528786836
Mean Absolute Percentage Error (MAPE): 3.49%
Regression Accuracy: 96.51%
```

Accuracy : 96.51%%

2. Linear Regression

```
MAE: 8505.219688917525
MSE: 104963650.44756976
R-squared: 0.40587394852902126
Mean Absolute Percentage Error (MAPE): 81.04%
Regression Accuracy: 18.96%
```

Accuracy : 18.96%

3. Gradient Boosting Regressor

```
MAE: 617.559723035567
MSE: 5151988.029699174
R-squared: 0.9708381873890726
Mean Absolute Percentage Error (MAPE): 4.88%
Regression Accuracy: 95.12%
```

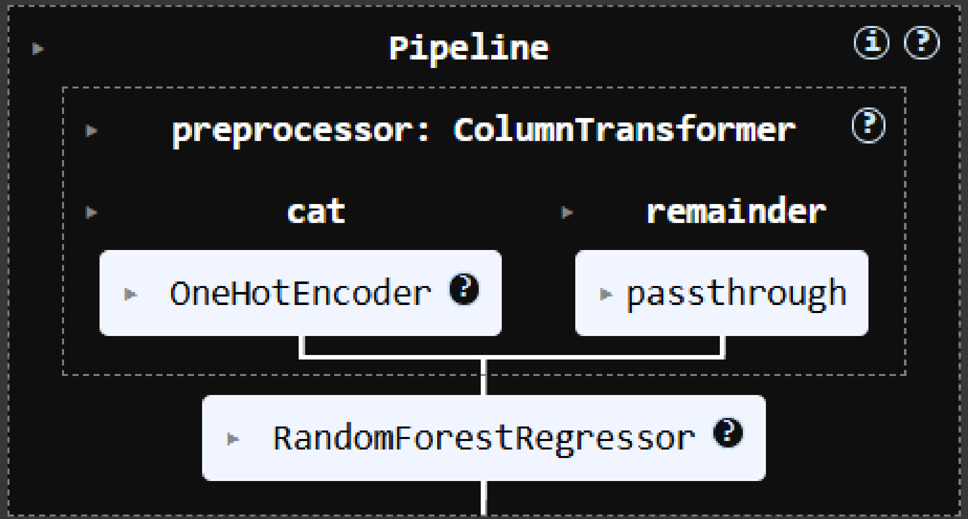
Accuracy : 95.12%

- So comparing the accuracy of 3 Machine Learning Algorithms, with a Accuracy of 96.51% Random Forest Regressor has choosen to develop the Paddy Harvest Prediction model.

Train the model

Train the Model

```
pipeline.fit(X_train, y_train)
```



Accuracy of the model

Accuracy of the model

```
[ ] import numpy as np
    from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

    # Predict using the trained model
    y_pred = pipeline.predict(X_test)

    # Example metric calculations
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    # Calculate RMSE
    rmse = np.sqrt(mse)

    # Print the metrics
    print(f"MAE: {mae}")
    print(f"MSE: {mse}")
    print(f"RMSE: {rmse}")
    print(f"R-squared: {r2}")

    # MAPE and Accuracy
    mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100
    accuracy = 100 - mape
    print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")
    print(f"Regression Accuracy: {accuracy:.2f}%")
```

```
MAE: 510.0285261251583
MSE: 4755389.907179633
RMSE: 2180.685650702465
R-squared: 0.9730830528786836
Mean Absolute Percentage Error (MAPE): 3.49%
Regression Accuracy: 96.51%
```

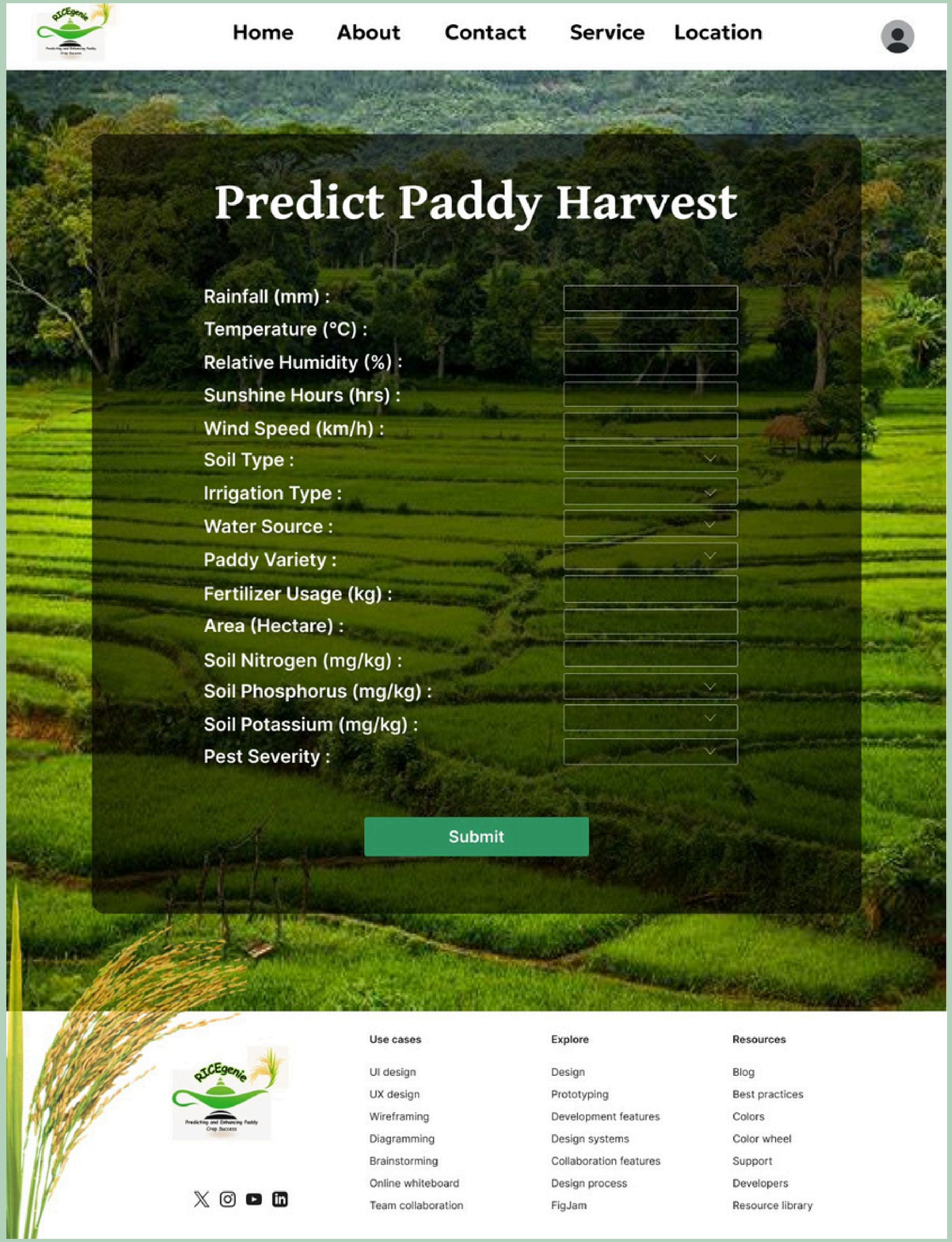
Random Forest Regressor

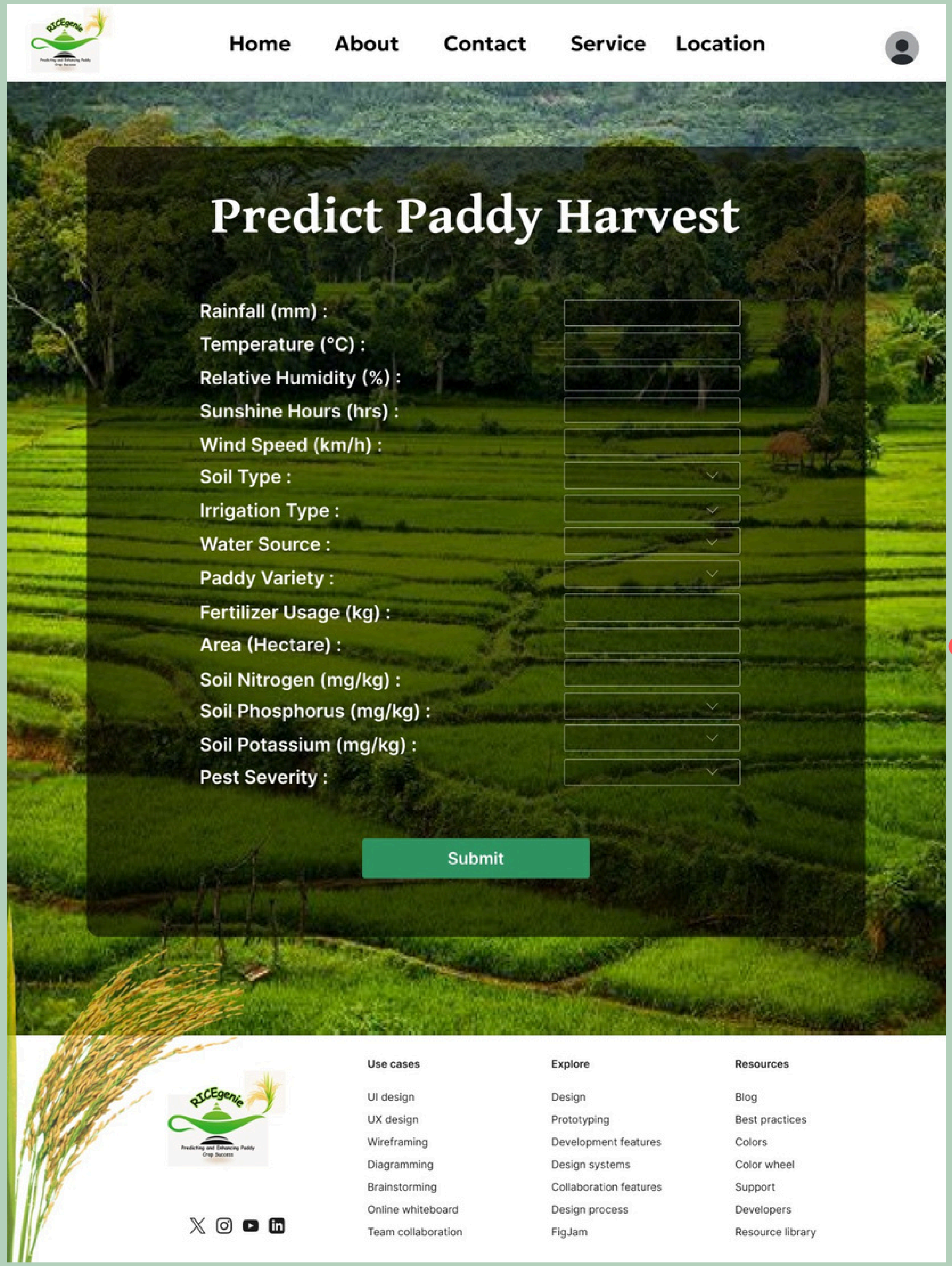
3. Model Development

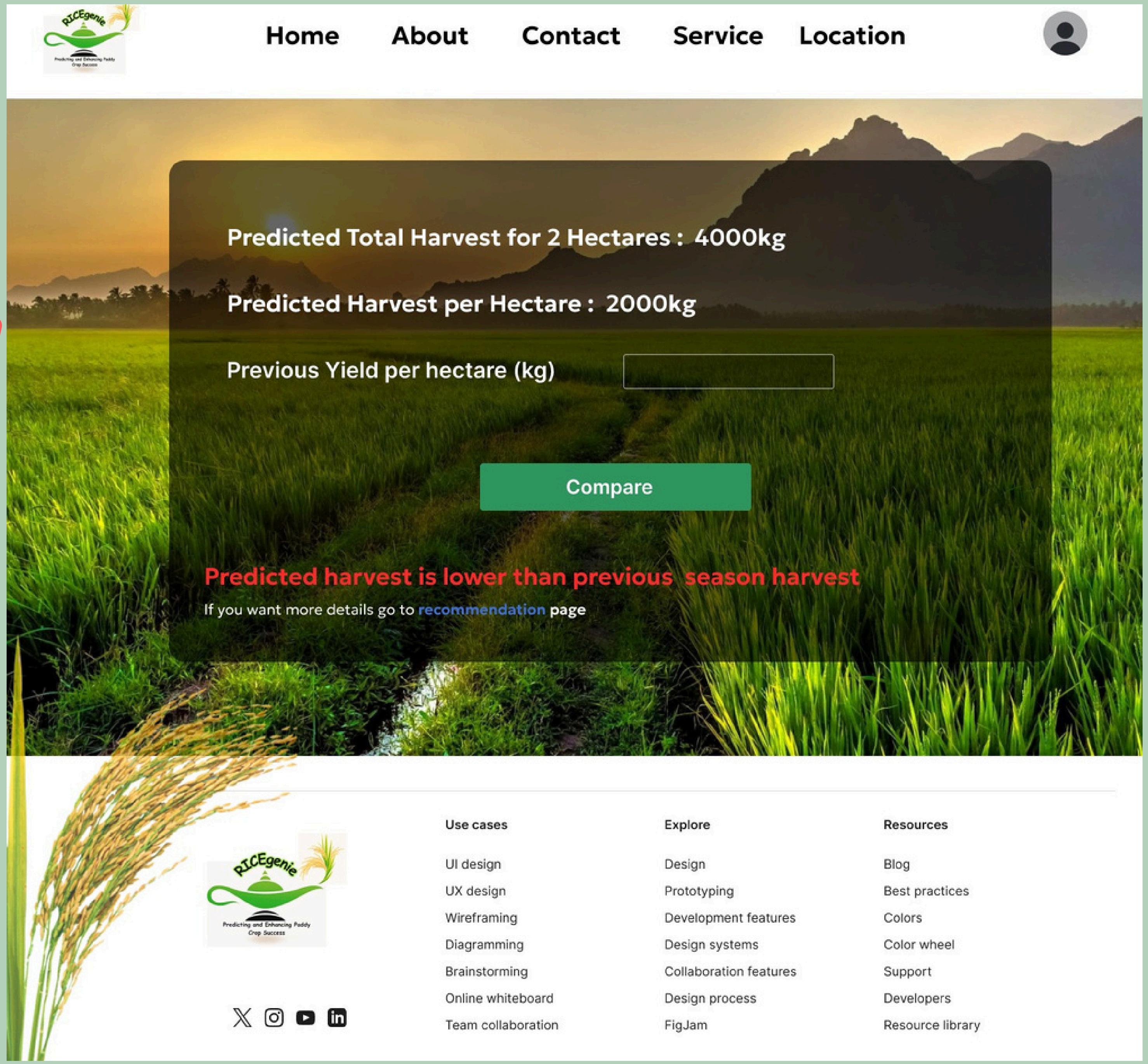
```python
# Function to predict harvest based on input acreage and other factors, including recommendations based on yield comparison
def predict_total_harvest():
    # Step 1: Farmer inputs various factors
    print("=== Enter Farm Data ===")
    area = float(input("Enter the total area of land in hectares: "))
    rainfall = float(input("Enter the total rainfall in mm: "))
    temperature = float(input("Enter the temperature in °C: "))
    humidity = float(input("Enter the relative humidity percentage: "))
    sunshine_hours = float(input("Enter the total sunshine hours per day: "))
    wind_speed = float(input("Enter the average wind speed in km/h: "))
    soil_type = input("Enter the soil type (e.g., Loam, Sandy, Clay): ")
    irrigation_type = input("Enter the irrigation type (e.g., Rainfed, Canal, Tube Well): ")
    water_source = input("Enter the water source (e.g., Well, River, Rainwater): ")
    paddy_variety = input("Enter the paddy variety (e.g., BG 450, BG 250 , BG 350 , BG 360 ): ")
    fertilizer_usage = float(input("Enter the fertilizer usage in kg: "))
    soil_nitrogen = float(input("Enter the soil nitrogen level in mg/kg: "))
    soil_phosphorus = float(input("Enter the soil phosphorus level in mg/kg: "))
    soil_potassium = float(input("Enter the soil potassium level in mg/kg: "))
    pest_severity = input("Enter the pest severity (e.g., Low, Medium, High): ")
    season = input("Enter the season (e.g., Maha, Yala): ")
    district = input("Enter the district: ")

    # Step 2: Input for previous yield per acre
    previous_yield_per_hectare = float(input("Enter the previous yield per hectare in kg: "))

    # Step 3: Prepare the data for prediction
    new_data = pd.DataFrame([{
        'Rainfall (mm)': rainfall,
        'Temperature (°C)': temperature,
        'Relative Humidity (%)': humidity,
        'Sunshine Hours (hrs)': sunshine_hours,
```

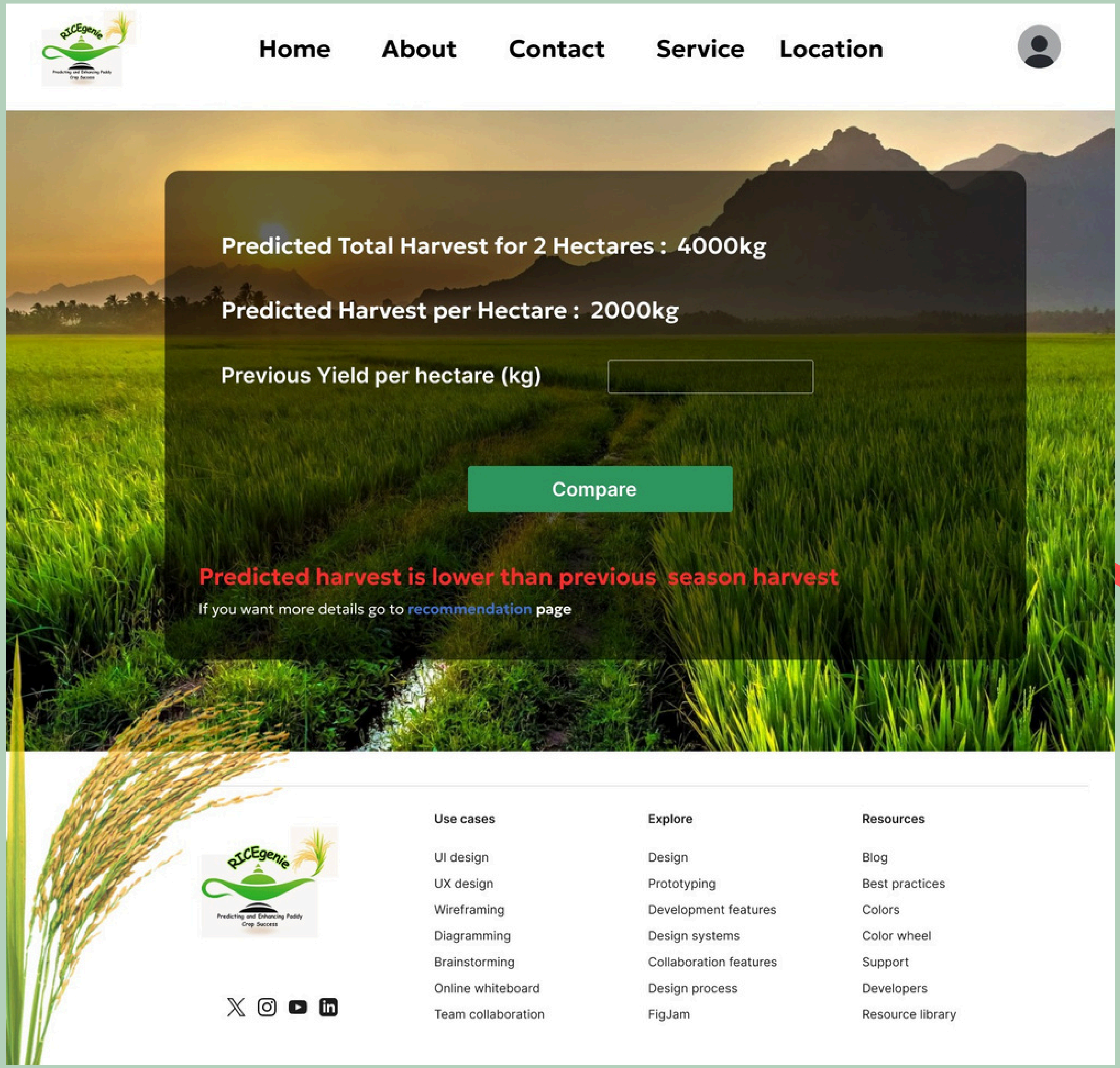
```
Enter the total area of land in hectares: 6.2
Enter the total rainfall in mm: 1407.3
Enter the temperature in °C: 32.2
Enter the relative humidity percentage: 60.2
Enter the total sunshine hours per day: 9.1
Enter the average wind speed in km/h: 11.3
Enter the soil type (e.g., Loam, Sandy, Clay): Sandy
Enter the irrigation type (e.g., Rainfed, Canal, Tube Well): Rainfed
Enter the water source (e.g., Well, River, Rainwater): Well
Enter the paddy variety (e.g., BG 450, BG 250 , BG 350 , BG 360 ): BG 450
Enter the fertilizer usage in kg: 34
Enter the soil nitrogen level in mg/kg: 46
Enter the soil phosphorus level in mg/kg: 49
Enter the soil potassium level in mg/kg: 35
Enter the pest severity (e.g., Low, Medium, High): High
Enter the season (e.g., Maha, Yala): Maha
Enter the district: Polonnaruwa
Enter the previous yield per hectare in kg: 1200

=== Predicted Results ===
Total Harvest for 6.2 hectare: 38092.36 kg
Predicted Yield per hectare: 6143.93 kg
=========================

=== Recommendations Based on Yield Comparison ===
Condition: Predicted Yield > Previous Yield

Predicted Yield is HIGHER than the Previous Yield.
Recommendations for Maintaining or Further Improving Yield:
- Continue current agricultural practices.
- Monitor soil health regularly and apply nutrients accordingly.
- Use precision farming tools to track crop progress.
- Plan for seasonal crop rotation to maintain soil fertility.
```


Home
About
Contact
Service
Location
Enhancing Paddy Cultivation in Sri Lanka
Leveraging predictive models and image processing techniques for sustainable farming.
Main Objective
Develop a comprehensive platform to enhance paddy cultivation in Sri Lanka by leveraging predictive models and image processing techniques.
Predict Harvest Yield
Detect & Manage Diseases
Recommend Optimal Paddy Varieties
Identify & Control Weeds


Home
About
Contact
Service
Location
Predict Paddy Harvest
Rainfall (mm) :
Temperature (°C) :
Relative Humidity (%) :
Sunshine Hours (hrs) :
Wind Speed (km/h) :
Soil Type :
Irrigation Type :
Water Source :
Paddy Variety :
Fertilizer Usage (kg) :
Area (Hectare) :
Soil Nitrogen (mg/kg) :
Soil Phosphorus (mg/kg) :
Soil Potassium (mg/kg) :
Pest Severity :
Submit
Use cases
UI design
UX design
Wireframing
Diagramming
Brainstorming
Online whiteboard
Team collaboration
Explore
Design
Prototyping
Development features
Design systems
Collaboration features
Design process
FigJam
Resources
Blog
Best practices
Colors
Color wheel
Support
Developers
Resource library


Home
About
Contact
Service
Location
Predict Paddy Harvest
Rainfall (mm) :
Temperature (°C) :
Relative Humidity (%) :
Sunshine Hours (hrs) :
Wind Speed (km/h) :
Soil Type :
Irrigation Type :
Water Source :
Paddy Variety :
Fertilizer Usage (kg) :
Area (Hectare) :
Soil Nitrogen (mg/kg) :
Soil Phosphorus (mg/kg) :
Soil Potassium (mg/kg) :
Pest Severity :
Submit
Use cases
UI design
UX design
Wireframing
Diagramming
Brainstorming
Online whiteboard
Team collaboration
Explore
Design
Prototyping
Development features
Design systems
Collaboration features
Design process
FigJam
Resources
Blog
Best practices
Colors
Color wheel
Support
Developers
Resource library

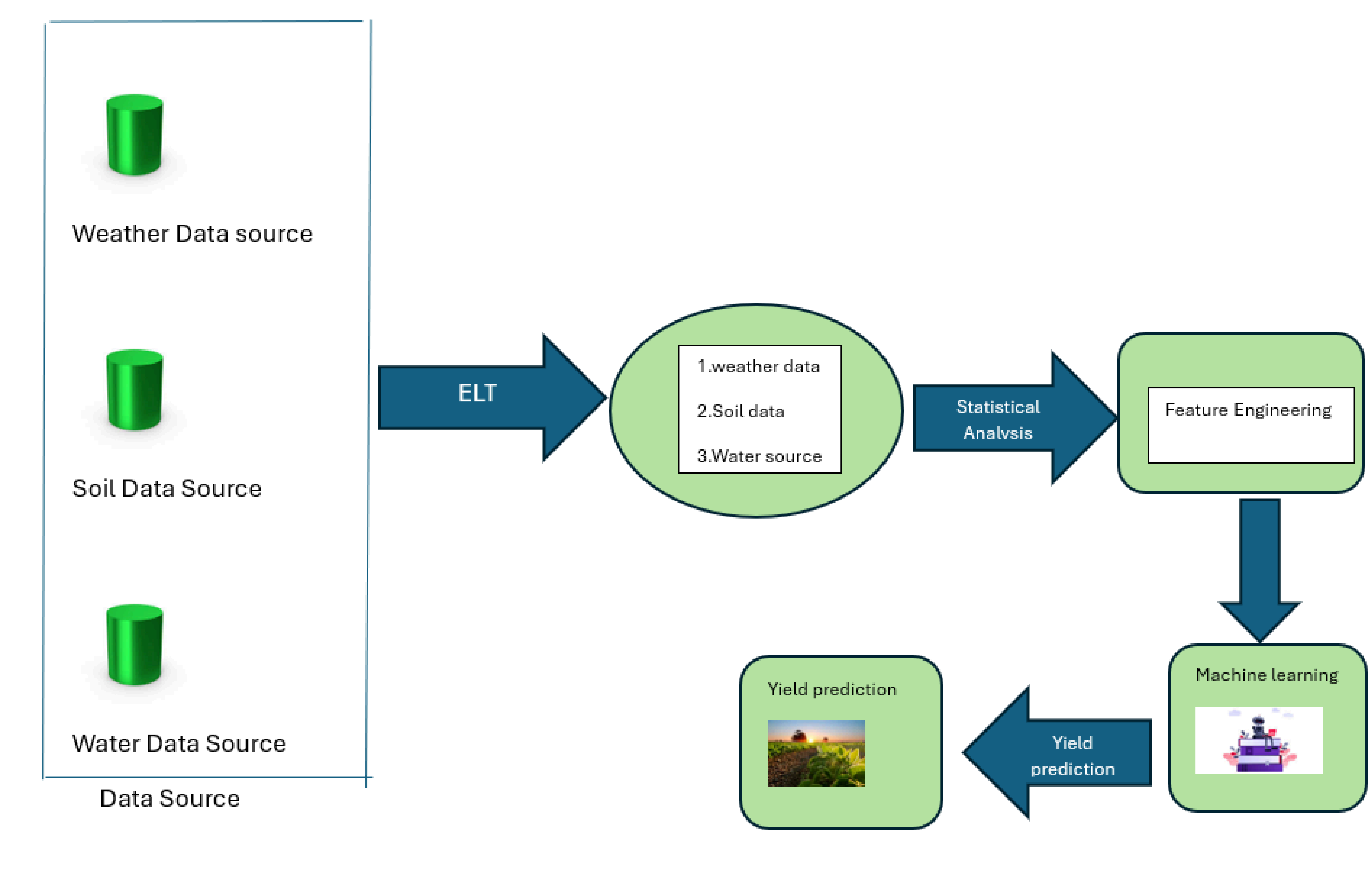

Home
About
Contact
Service
Location
Predicted Total Harvest for 2 Hectares : 4000kg
Predicted Harvest per Hectare : 2000kg
Previous Yield per hectare (kg)
Compare
Predicted harvest is lower than previous season harvest
If you want more details go to recommendation page
Use cases
UI design
UX design
Wireframing
Diagramming
Brainstorming
Online whiteboard
Team collaboration
Explore
Design
Prototyping
Development features
Design systems
Collaboration features
Design process
FigJam
Resources
Blog
Best practices
Colors
Color wheel
Support
Developers
Resource library



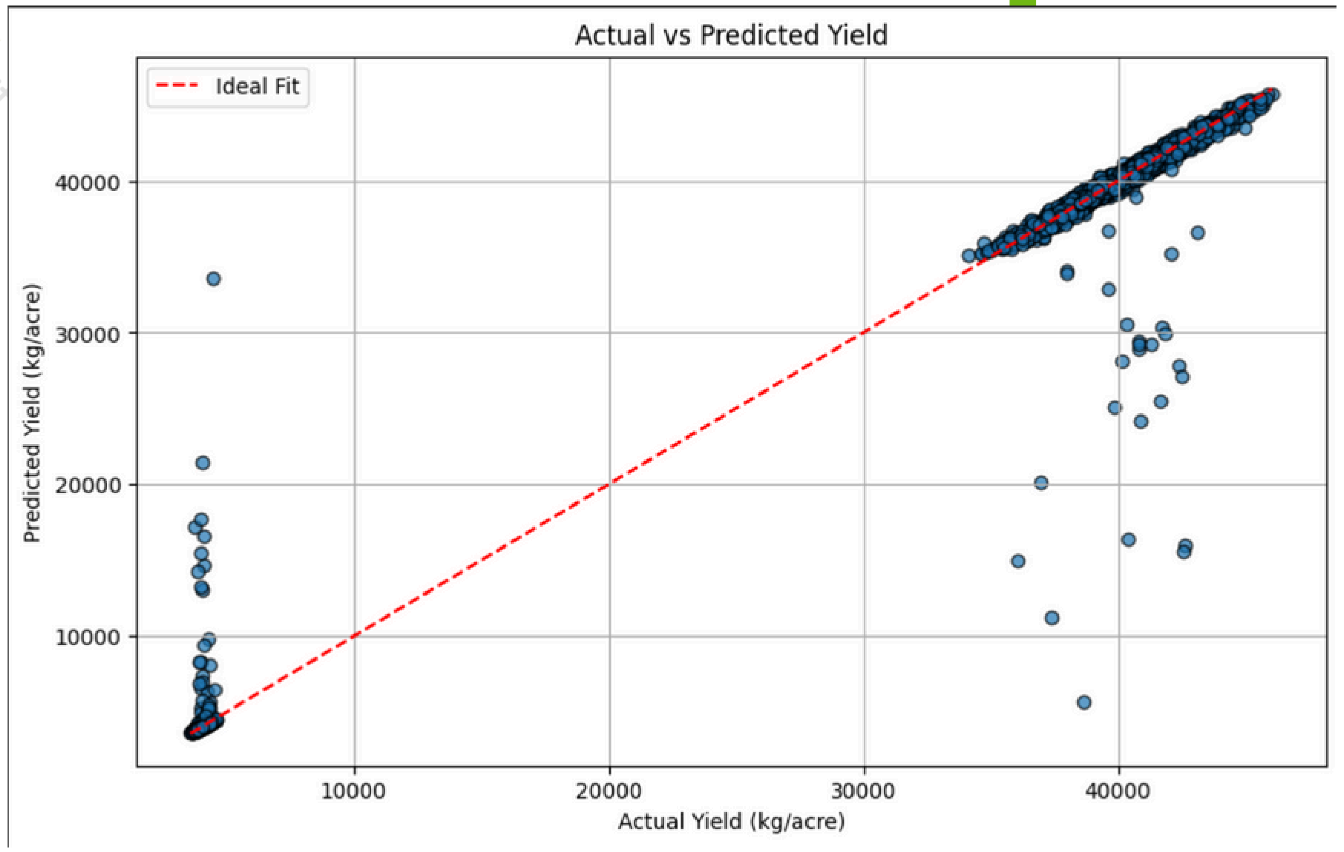
Home
About
Contact
Service
Location
Predicted Total Harvest for 2 Hectares : 4000kg
Predicted Harvest per Hectare : 2000kg
Previous Yield per hectare (kg)
Compare
Predicted harvest is lower than previous season harvest
If you want more details go to recommendation page
Use cases
UI design
UX design
Wireframing
Diagramming
Brainstorming
Online whiteboard
Team collaboration
Explore
Design
Prototyping
Development features
Design systems
Collaboration features
Design process
FigJam
Resources
Blog
Best practices
Colors
Color wheel
Support
Developers
Resource library

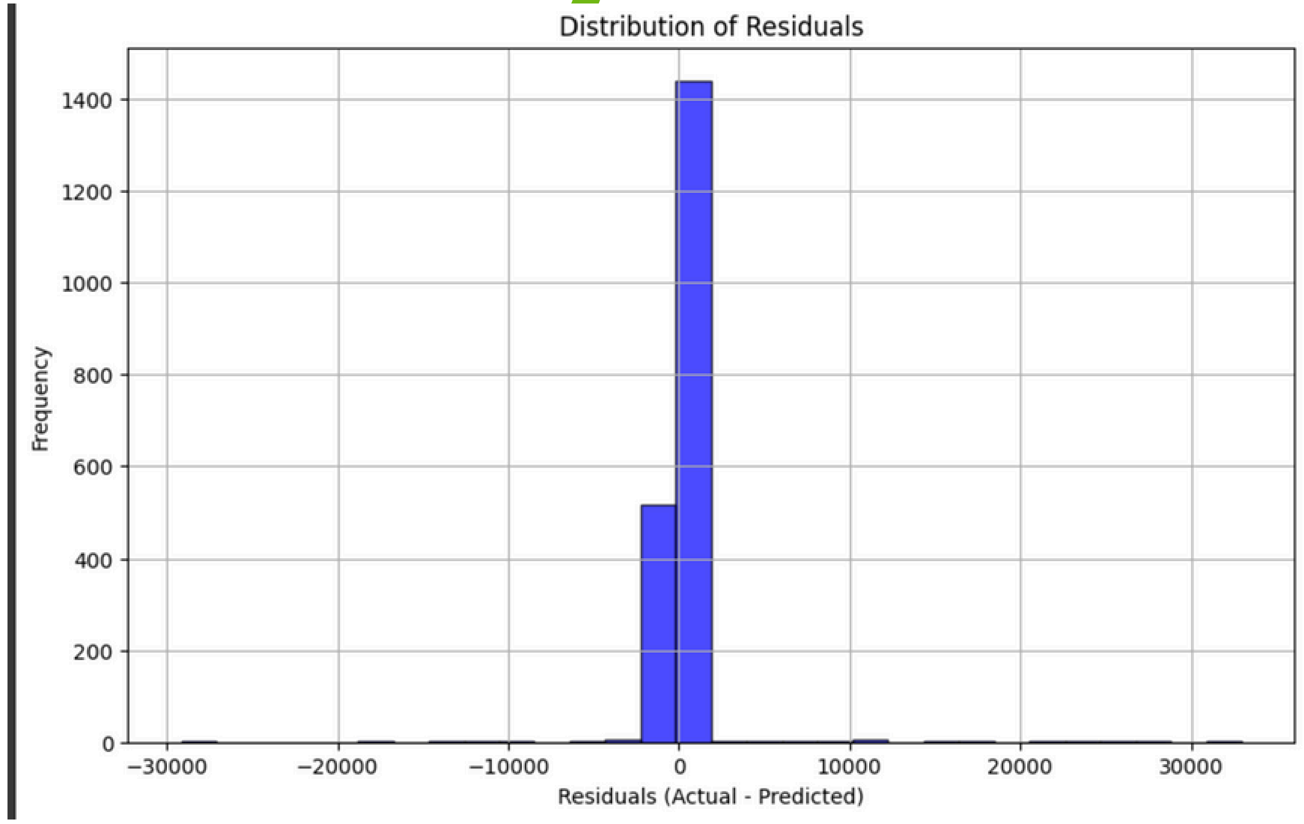
Recommendation Actions to Improve Yield
• Optimize fertilizer usage based on soil tests.
Optimizing fertilizer usage based on soil tests is essential for sustainable paddy cultivation in Sri Lanka. Conducting soil tests helps farmers understand the nutrient levels and pH of their soil, enabling them to apply fertilizers more efficiently and cost-effectively. By identifying the specific nutrients that the soil lacks, such as nitrogen, phosphorus, or potassium, farmers can avoid over-application or under-application of fertilizers.
This practice not only reduces production costs but also minimizes environmental pollution caused by excessive fertilizer runoff into water bodies. Additionally, tailored fertilizer application improves crop yields and enhances soil health over time. Farmers in Sri Lanka are encouraged to work with agricultural extension services or local agrarian centers to carry out soil testing and receive recommendations for the appropriate type and amount of fertilizers.
• Improve irrigation practices to ensure consistent water supply
Improving irrigation practices is vital to ensure a consistent water supply for paddy cultivation in Sri Lanka. Efficient irrigation systems, such as properly maintained canals, drip irrigation, or alternate wetting and drying (AWD) techniques, help regulate water usage and reduce wastage.
Farmers can benefit from adopting water management practices like scheduling irrigation based on crop growth stages and weather conditions, ensuring that fields are neither over-flooded nor water-stressed. Rainwater harvesting and storage can also be implemented to provide a backup water source during dry periods.
These practices not only ensure a steady water supply but also improve yields, conserve resources, and promote sustainable agriculture. Farmers can seek guidance from local agricultural officers to adopt modern irrigation techniques suited to their region.
Use cases
UI design
UX design
Wireframing
Diagramming
Brainstorming
Online whiteboard
Team collaboration
Explore
Design
Prototyping
Development features
Design systems
Collaboration features
Design process
FigJam
Resources
Blog
Best practices
Colors
Color wheel
Support
Developers
Resource library

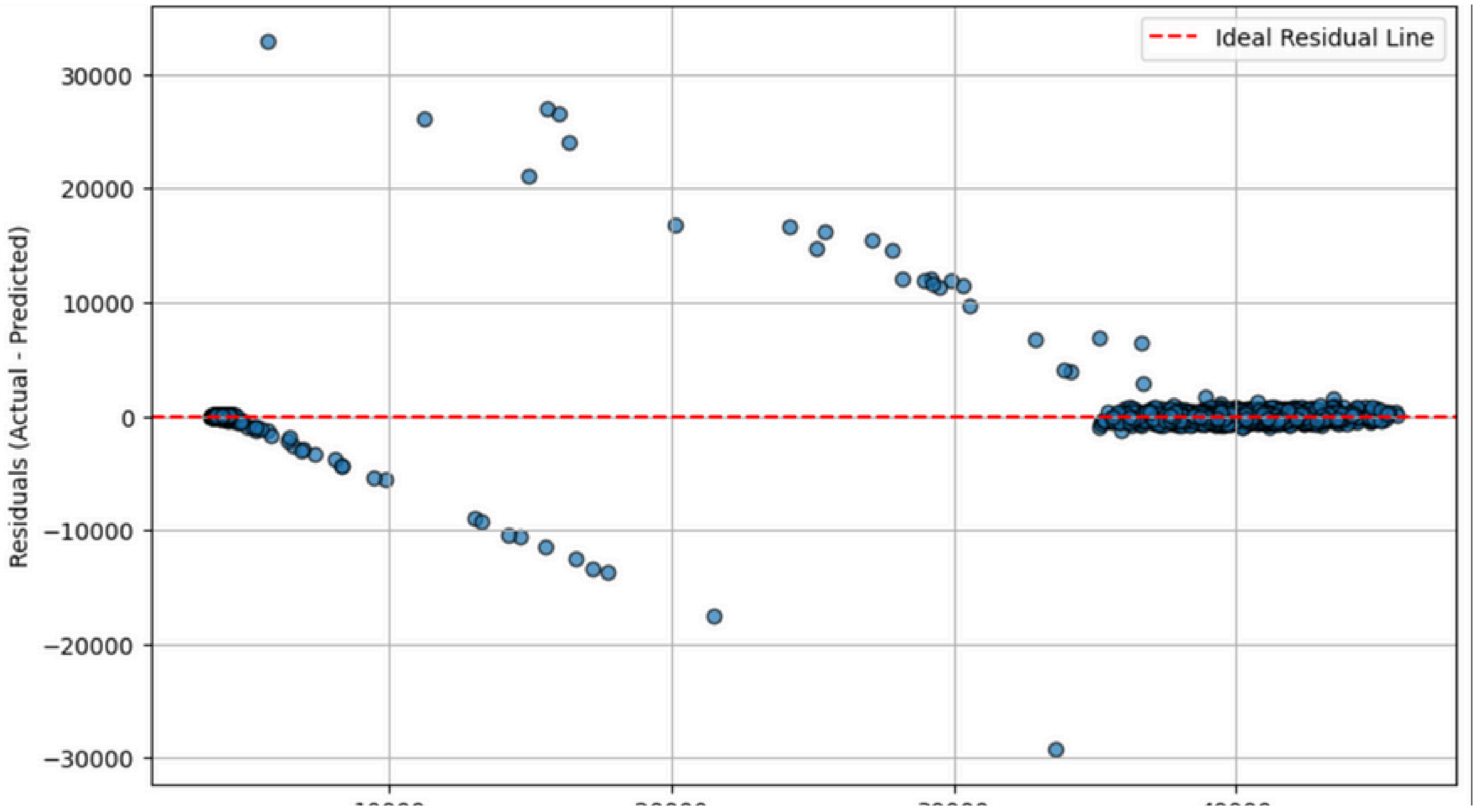
System Diagram



Model performance Analysis







Random Forest Regressor

Requirements

Non-functional requirements

- Availability
- Usability
- Performance
- Accuracy

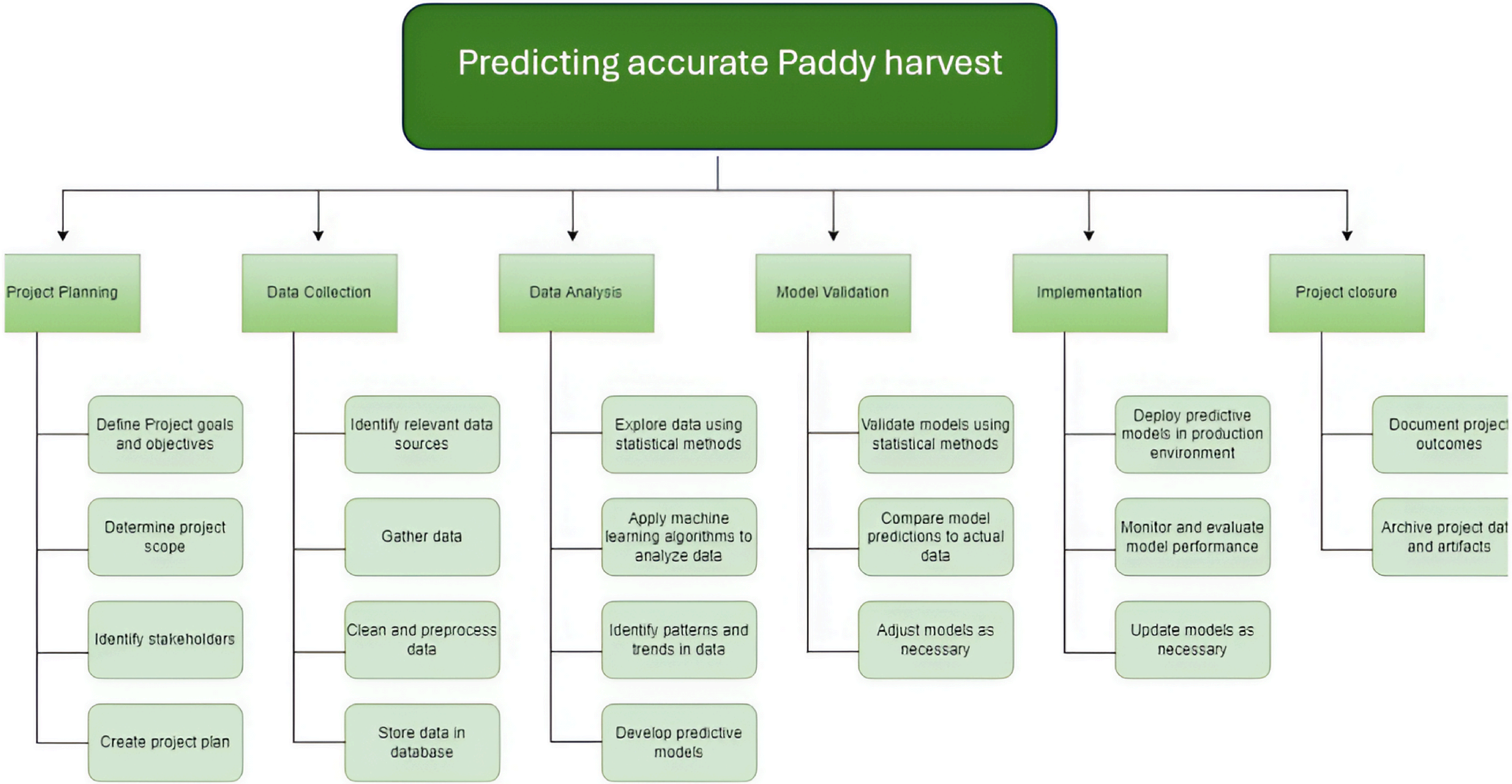
Functional requirements

- Yield Prediction
- Decision Support

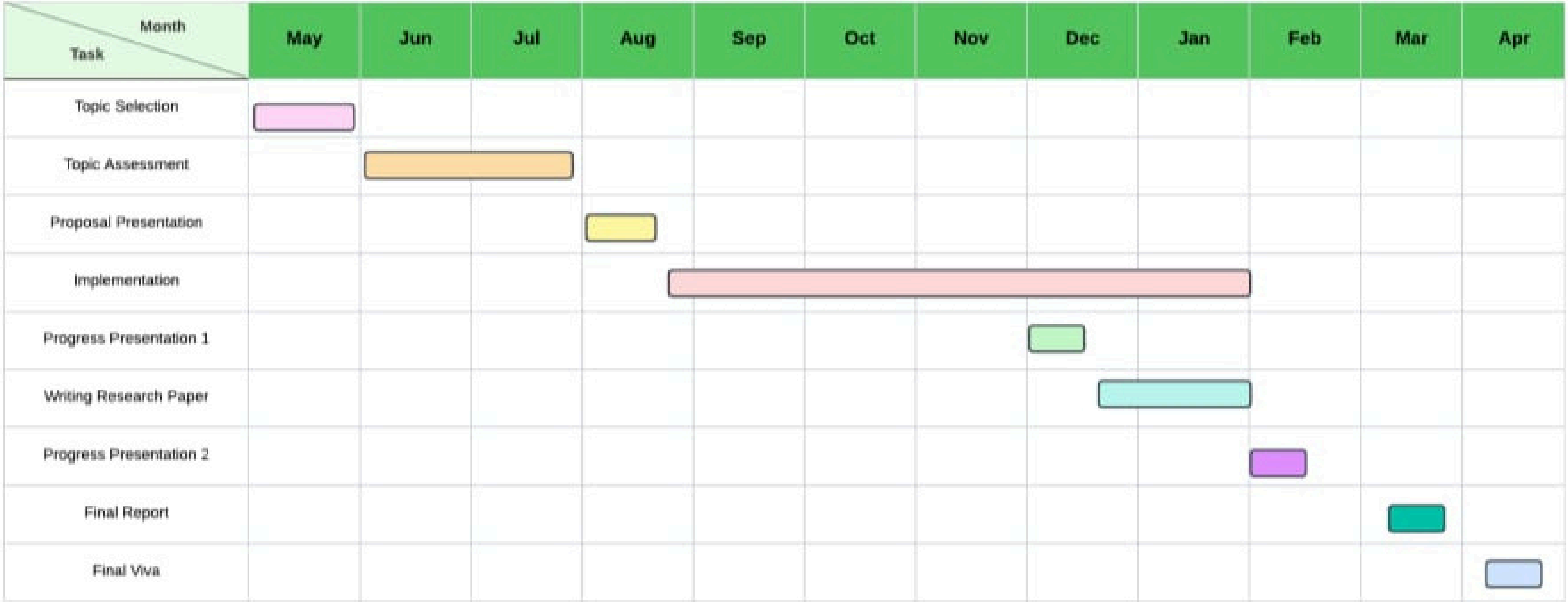
Technologies

- Python (Back end)
- ML (Regression)
- ReactJS (Front end)
- Tailwind (for styles)
- Fast API
- Google Colab
- MySql
- Git Hub (Version control system)
- Trello(Project Management)

Work Breakdown Structure



Gantt chart

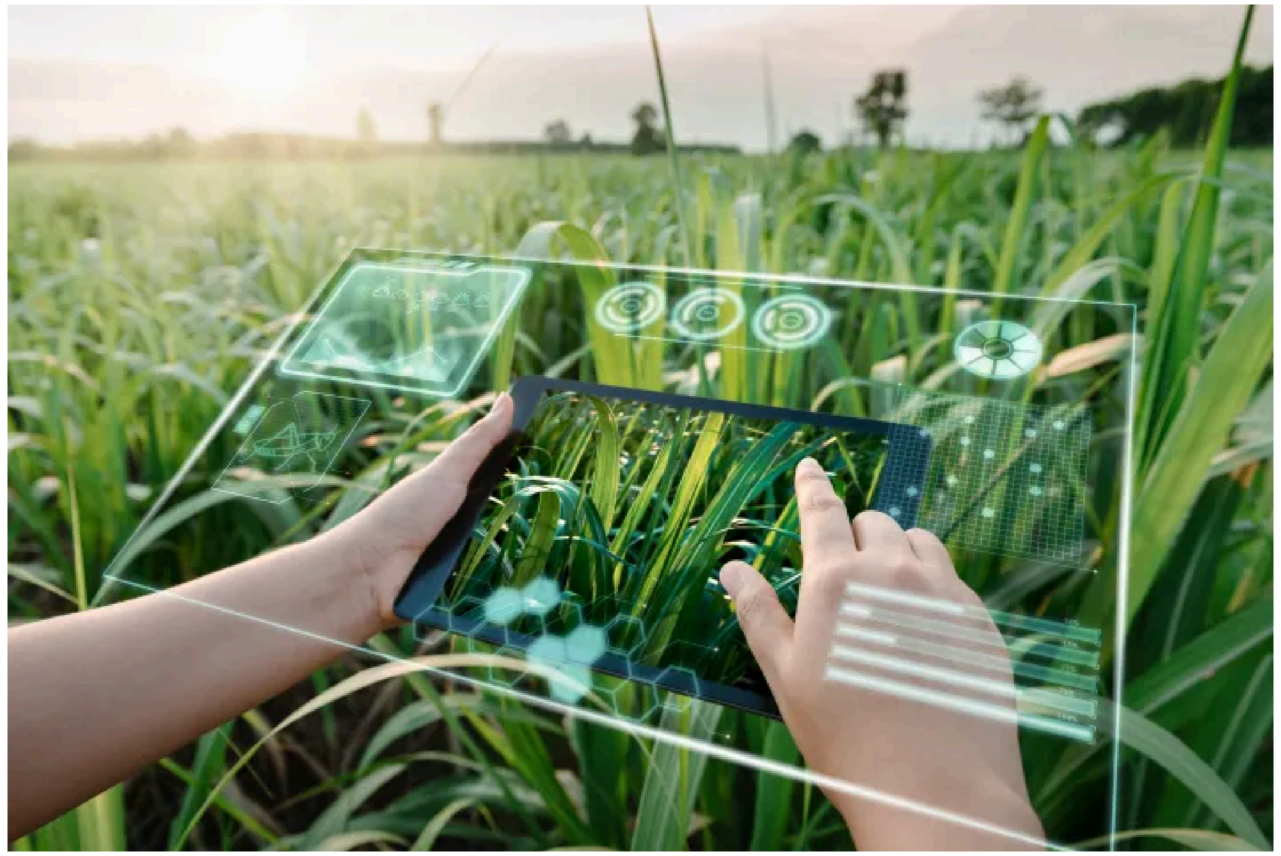

Month
Task
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
Topic Selection
Topic Assessment
Proposal Presentation
Implementation
Progress Presentation 1
Writing Research Paper
Progress Presentation 2
Final Report
Final Viva

REFERENCES

1. Smith, J., & Doe, A. (2020). Soil Health Monitoring Using Advanced Machine Learning Techniques. Journal of Agricultural Research, 45(3), 234-245.
2. Lee, B., & Kim, H. (2019). Integrating Real-Time Irrigation Metrics with Predictive Models for Crop Yield Enhancement. International Journal of Smart Agriculture, 12(2), 150-165.
3. Patel, R., & Kumar, S. (2018). Utilizing Localized Weather Patterns for Accurate Crop Predictions. Agricultural Systems, 33(1), 98-110.

IT21225192 | Amarasinghe A.I.S.A

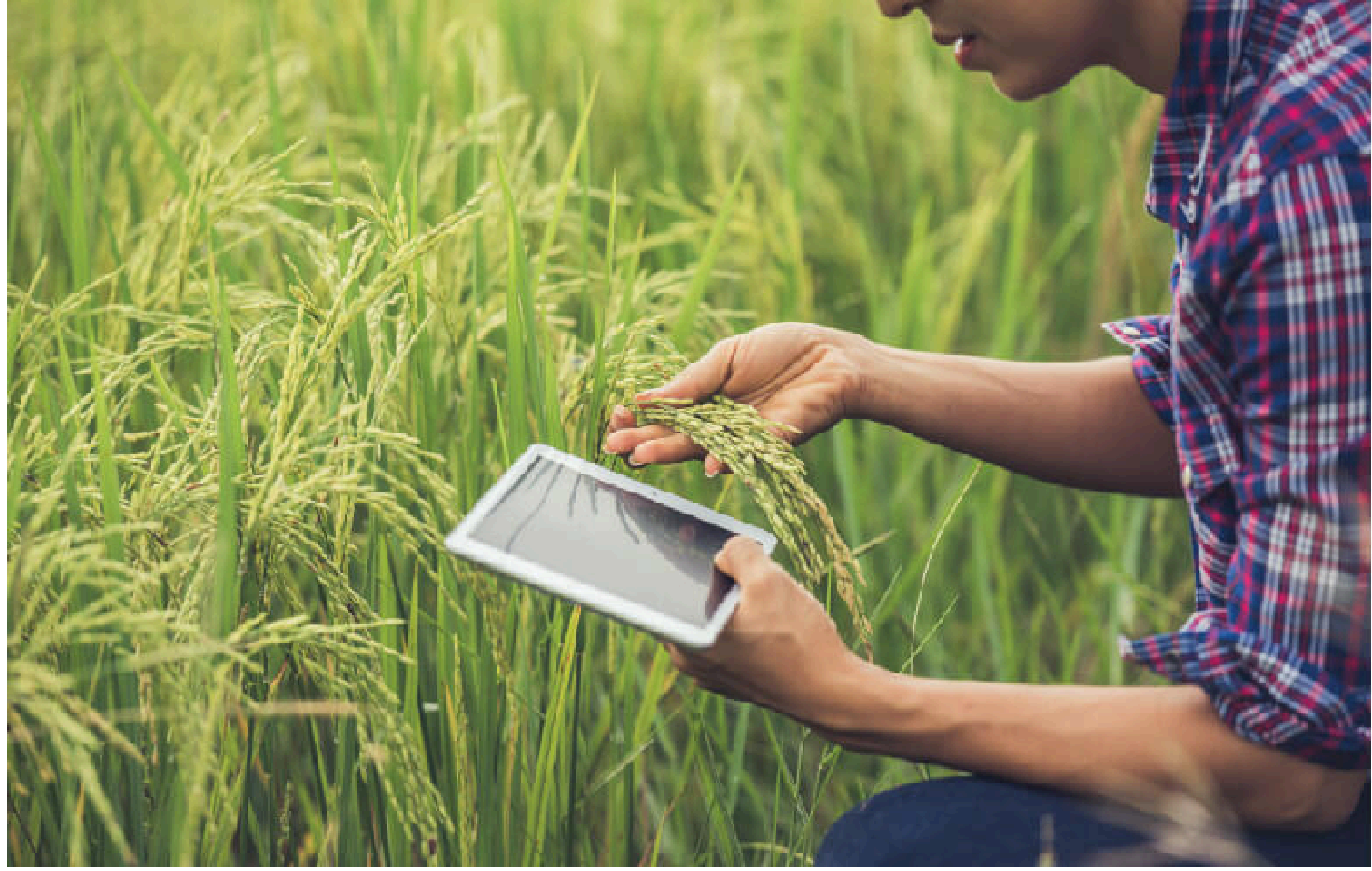
Pre Harvesting Diseases Detection & Mitigation System



External supervisor - DR.Rukmali Gunapala

Information Technology

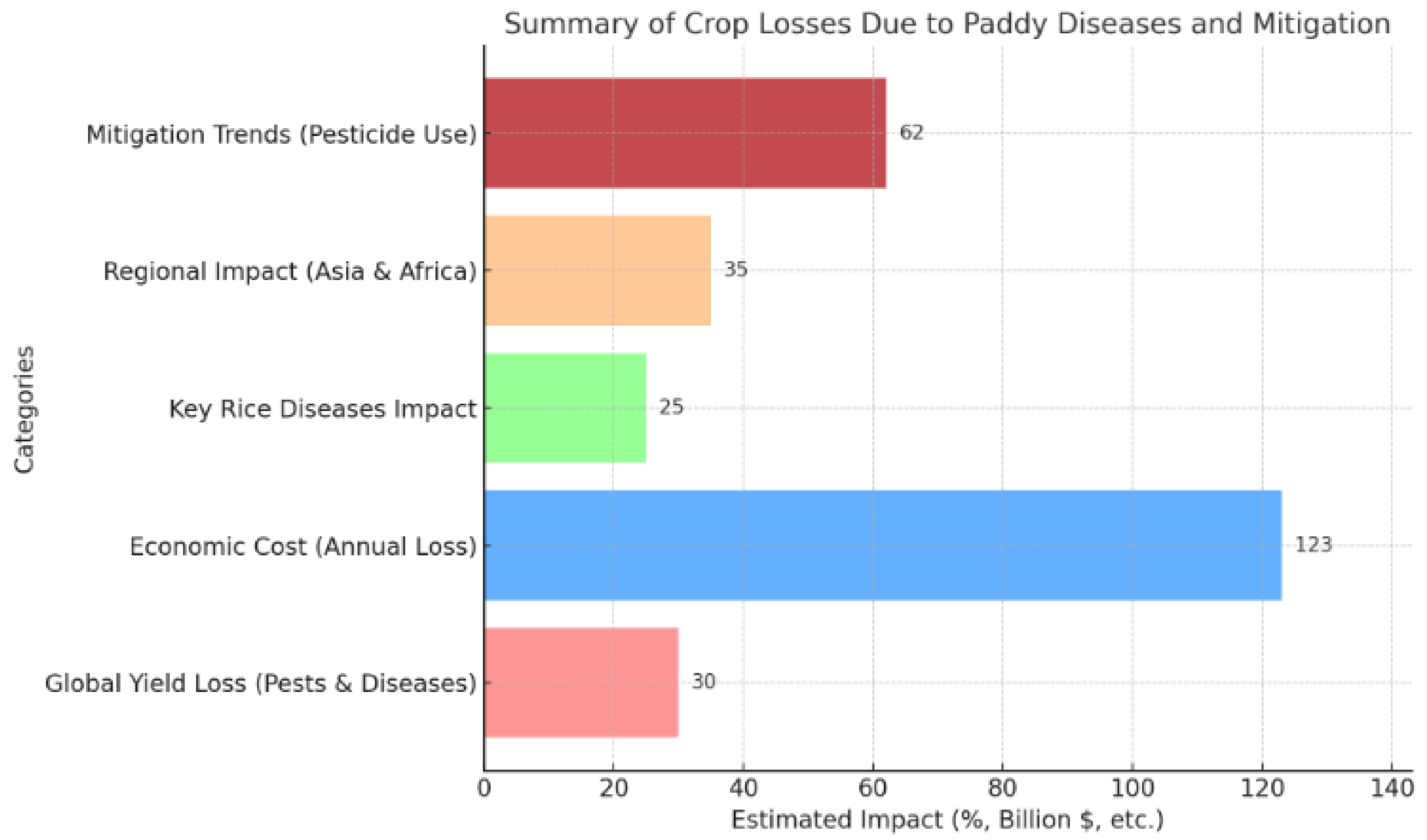
INTRODUCTION



Importance of Paddy Disease Identification ,

- **Pre harvesting Paddy diseases** can have a significant impact on crop yield and quality.
- It is important to identify the paddy diseases at an **early stage to prevent their spread and minimize** damage.
- **Image processing techniques** can be used to analyze digital images of plants and identify signs of disease.
- Image processing offers a promising approach for rapid and accurate detection of plant diseases, which can help to support more effective disease management and control strategies.

Statistics on crop losses due to paddy diseases



Research Gap

1. Limited focus on integrated solutions: Most existing studies focus only on disease detection, not on providing treatment recommendations.

2. Lack of real-time processing: Many methods do not offer real-time disease detection and treatment suggestions.

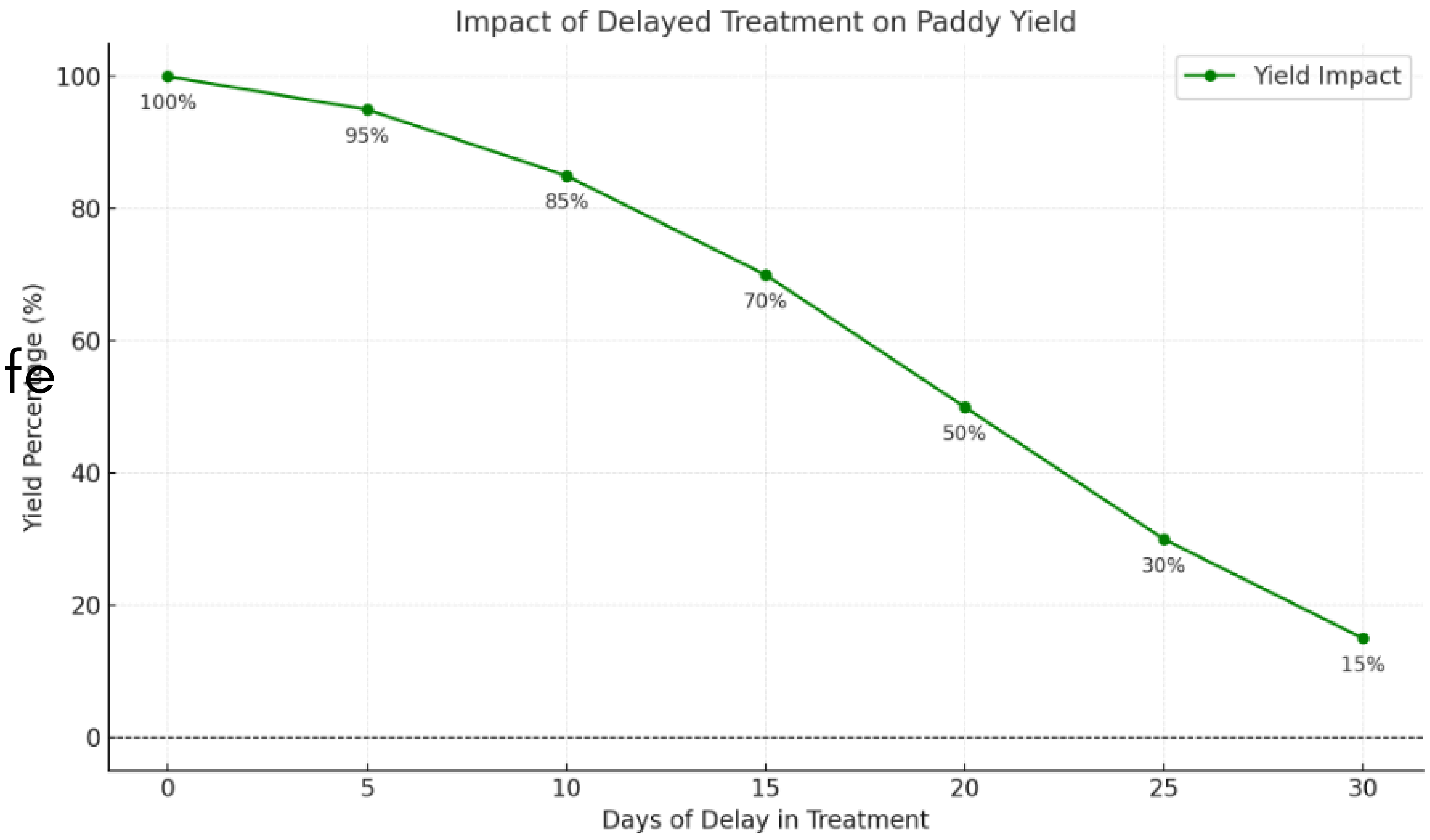
3.Insufficient accuracy: Current systems may not be accurate enough in identifying specific paddy diseases.

4. Scalability issues: Many approaches do not scale well for large agricultural fields.

5. Lack of user-friendly interfaces: There is a need for more emotionally and easy-to-use interfaces for farmers.

Research Problem

- Reduced crop yield
- Quality degradation and low marketability
- Higher production costs
- Crop loss due to reduced shelf life
- Economic losses



Specific and Sub Objectives

Specific Objectives :

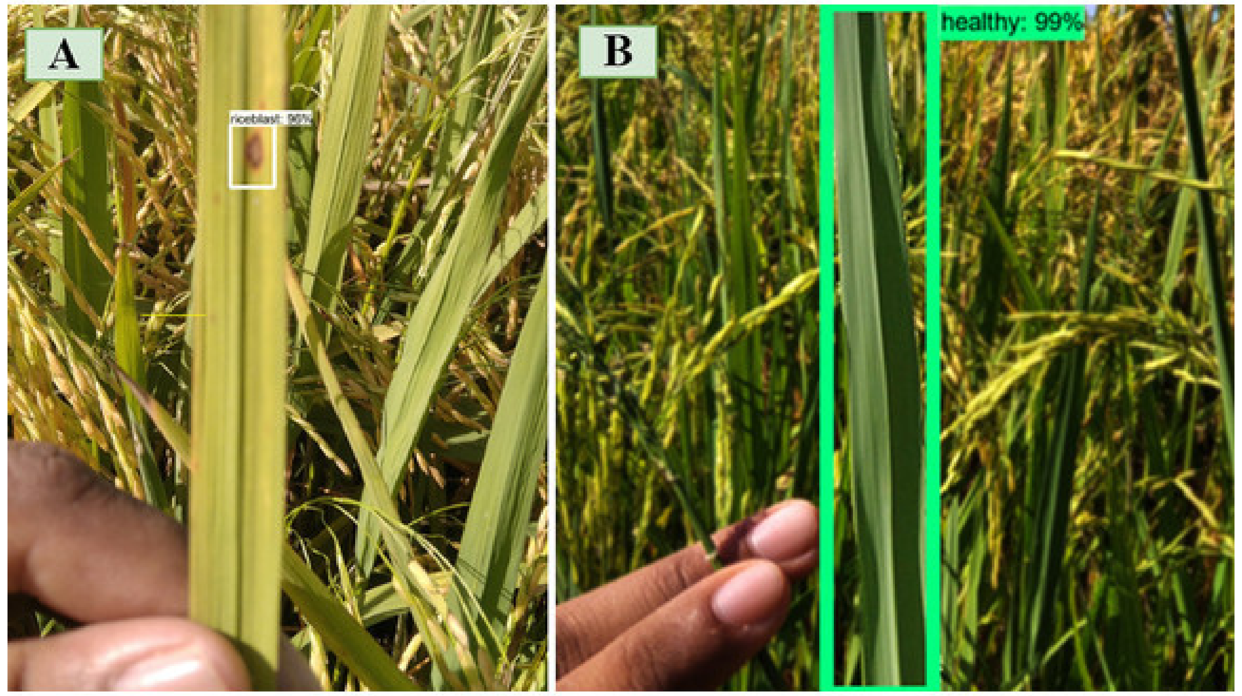
a. **Monitor and maintain healthy growth** of commercially viable paddies by paddy
b. **Integrated Disease Detection and Treatment System**: While many systems focus solely on disease detection, our project stands out by integrating both detection and treatment recommendations in a single platform.

Sub Objectives :

- Provide **sustainable treatments** for paddy diseases
- **Generate report** of plant diseases
- **Customized Treatment Recommendations**: The treatment suggestions are tailored to the specific disease detected, considering factors like severity and local agricultural practices.

Methodology

- **Image Acquisition** : capturing high quality images o f plants using digital cameras.
- **Pre-processing** : captured images are preprocessed to improve image
- **quality Segmentation:** separating the plant parts from the background and isolating the region of interest.
- **Feature Extraction :** Features are extracted from image ,these features may include color, texture, shape, or any other relevant features that can help distinguish between healthy and diseased plants.
- **Classification :** classifying the plant as either healthy or diseased. This involves using machine learning algorithms to train a model that can accurately predict the presence of disease in a given plant.
- **Visualization:** The results are visualized to provide a clear and concise output to the user. This includes highlighting the infected area in the image.
- **Diagnosis :** The system identifies the type of plant disease and suggests appropriate treatment measures, Based on the classification result



Import Libraries

```python
# Import libraries
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
```

Image Processing:

- os, cv2 (OpenCV): For handling files and image processing.

Data Handling & Visualization:

- numpy: For numerical computations.
- matplotlib: To visualize data and results.

Deep Learning:

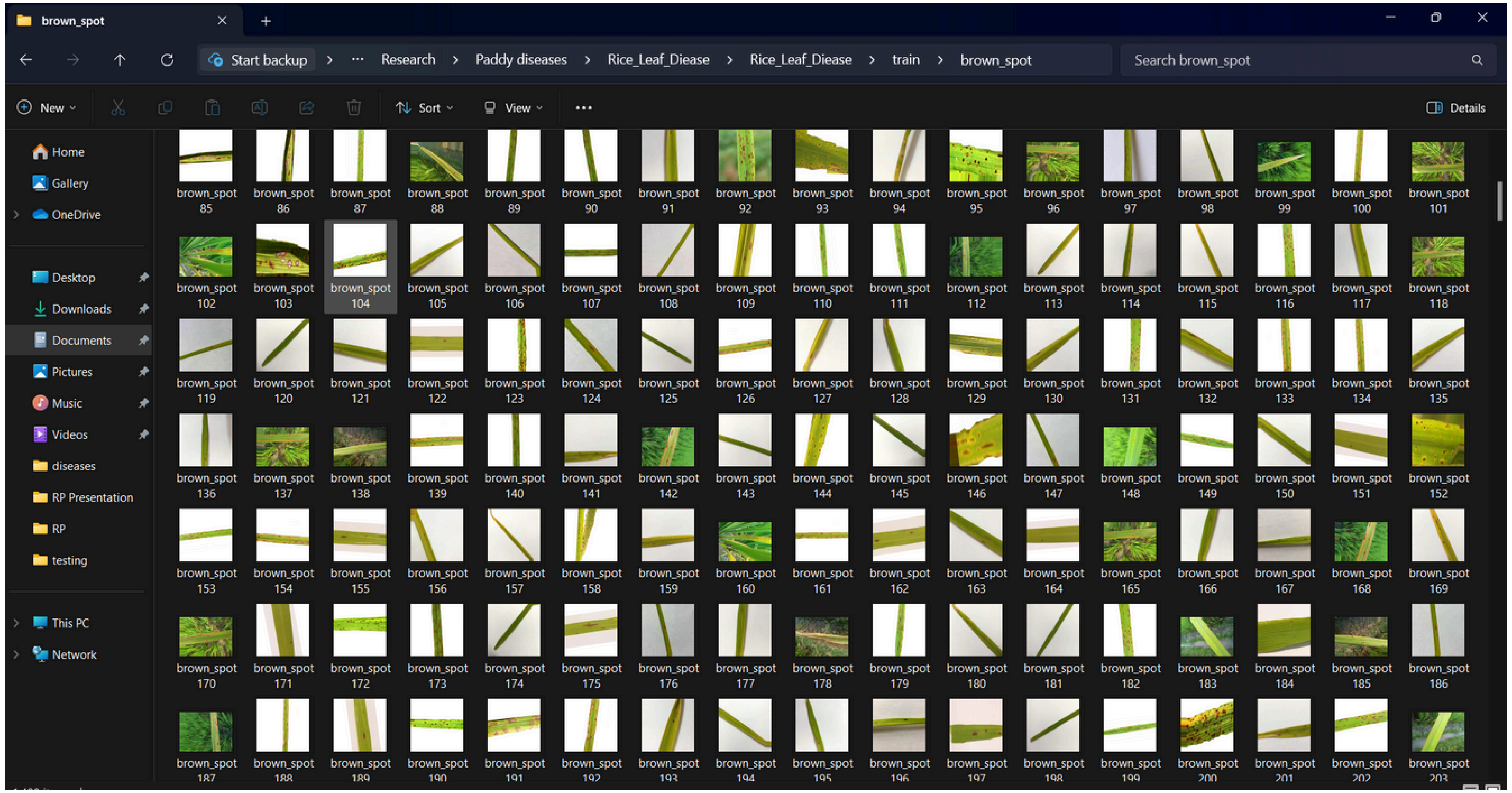
- tensorflow.keras: For building and training the CNN model.
  - Layers: Conv2D, MaxPooling2D, Dense, Dropout, Flatten.
  - Optimizer: Adam.
- to_categorical: Converts labels for classification tasks.

Data Splitting:

- sklearn: Splits data into training and testing sets.

Data Collection

- **Dataset:** 5 classes with near 3000 images per class (augmented and non-augmented).
- **Goal:** Increase diversity and robustness of the model.



Model Training

- **Model:** CNN (Convolutional Neural Network) selected because of high accuracy.(88.93%)

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 126, 126, 32)	896
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_1 (Conv2D)	(None, 61, 61, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_2 (Conv2D)	(None, 28, 28, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 128)	3,211,392
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 5)	645

Total params: 3,305,285 (12.61 MB)
Trainable params: 3,305,285 (12.61 MB)
Non-trainable params: 0 (0.00 B)

	precision	recall	f1-score	support
bacterial_leaf_blight	1.00	0.95	0.98	288
brown_spot	0.90	0.79	0.84	305
healthy	0.88	0.93	0.90	274
leaf_blast	0.83	0.82	0.82	348
sheath_blight	0.86	0.98	0.92	294
accuracy			0.89	1509
macro avg	0.89	0.89	0.89	1509
weighted avg	0.89	0.89	0.89	1509

Model Training

Image preprocessing:

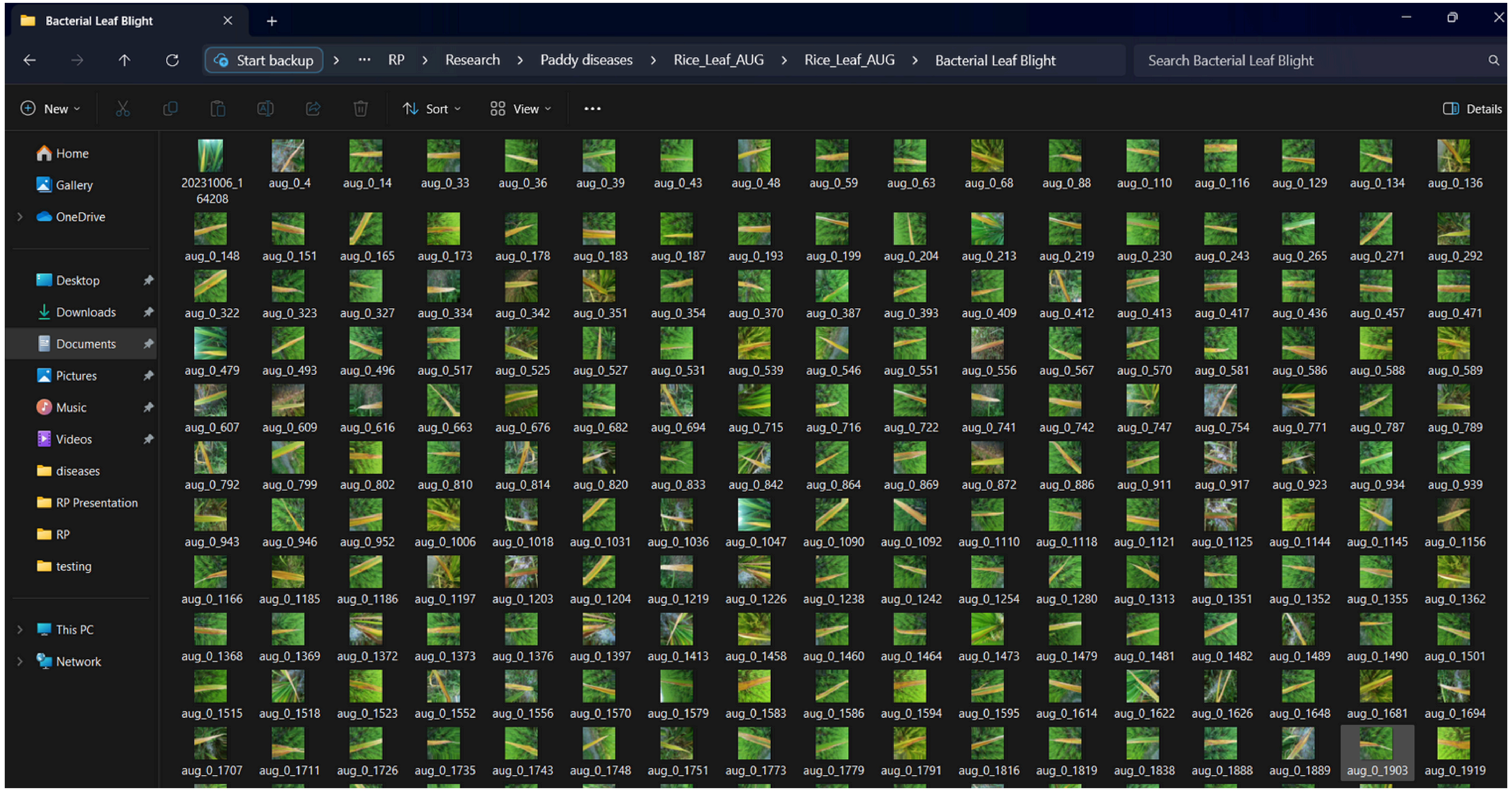
- Preprocessing: Resizing and normalizing images.

```python
# Image Preprocessing
def preprocess_image(img_path):
    try:
        # Load the image with reduced memory consumption
        img = cv2.imread(img_path, cv2.IMREAD_UNCHANGED)
        if img is None:
            raise ValueError(f"Image at {img_path} could not be read. Check the file.")

        # Resize the image to 128x128 pixels
        img_resized = cv2.resize(img, (128, 128))

        # Normalize pixel values to [0, 1]
        img_normalized = img_resized / 255.0
        return img_normalized
    except Exception as e:
        print(f"Error processing image {img_path}: {e}")
        return None  # Skip this image
```

Model Training

Training Process:

- Augmentation: Flip, rotate, and adjust brightness.



Reduce overfitting

```
# Add convolutional layers with MaxPooling
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(128, 128, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))

# Flatten the output and add Dense layers
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))  # Regularization to reduce overfitting
model.add(Dense(num_classes, activation='softmax'))  # Output layer

# Compile the Model
model.compile(optimizer=Adam(learning_rate=0.001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

- Input Shape: 128x128x3 (RGB images)

Feature Extraction:

- Conv2D + ReLU: Extract features with filters of sizes 32, 64, and 128.
- MaxPooling: Downsample to reduce spatial dimensions.

Classification:

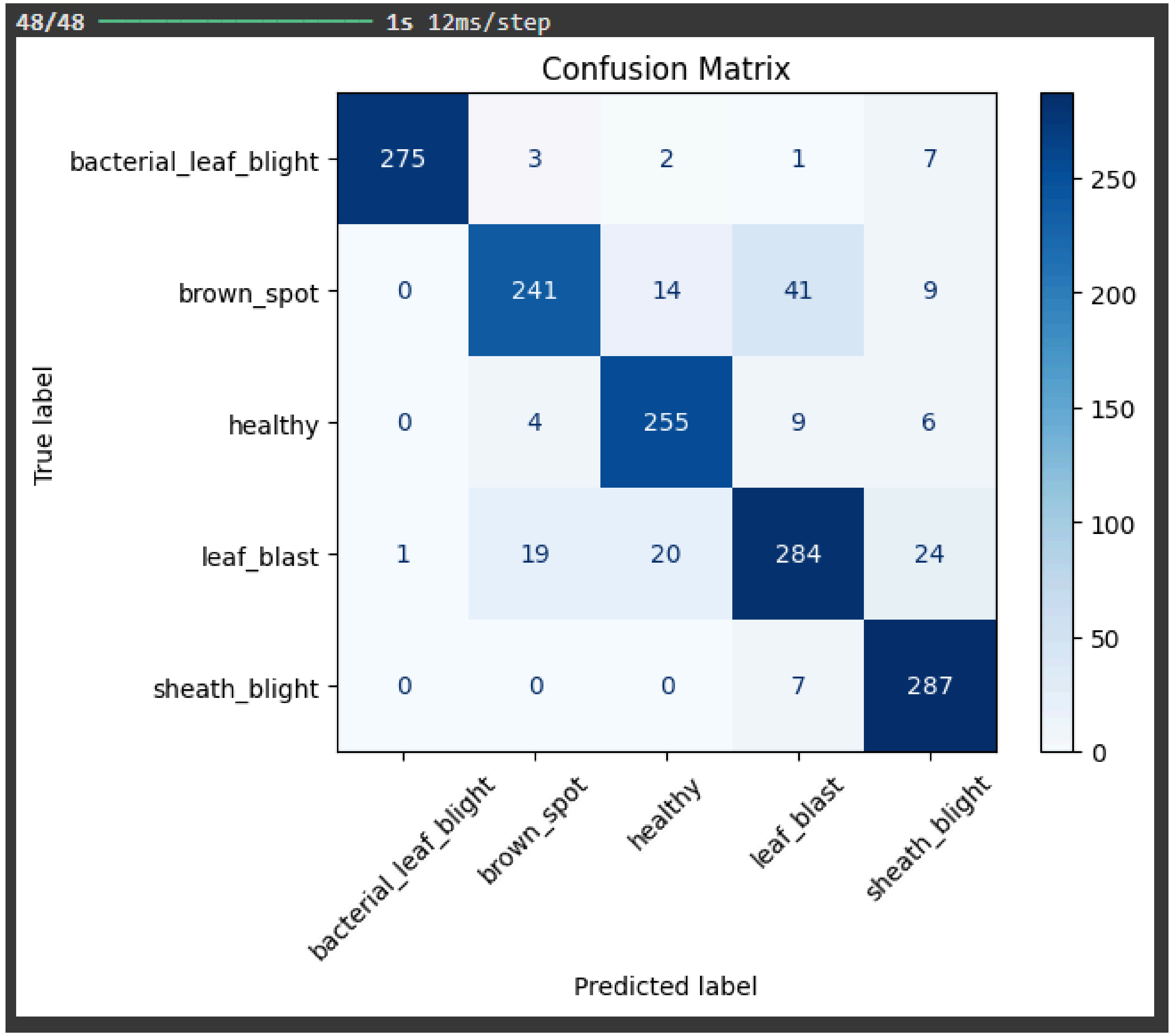
- Flatten: Convert feature maps into a 1D vector.
- Dense Layers: Fully connected layers for prediction.
- Dropout: 50% regularization to **reduce overfitting**.
- Output Layer: **Softmax** activation for multi-class classification.

Compilation:

- Optimizer: **Adam** (learning rate = 0.001).
- Loss Function: **Categorical Crossentropy.**
- Evaluation Metric: Accuracy.

Design

- **matrix visualization of CNN model**



Treatment Recommendation

Process:

- Match predicted class with predefined treatment plans.

```
[ ] import pickle

    # Define the data in a dictionary format
    treatment_data = {
        'bacterial_leaf_blight': {
            'Name': 'Bacterial Leaf Blight',
            'Symptoms': 'Yellowing of leaves, water-soaked lesions, leaf wilting, and dieback.',
            'Treatment (Brand Names)': 'Kocide 3000, Agrimycin 100',
            'How to Use': 'Spray when symptoms appear; repeat every 7-10 days.',
            'Recommendations': 'Use resistant rice varieties. Improve drainage. Maintain proper hygiene in the field.'
        },
        'brown_spot': {
            'Name': 'Brown Spot',
            'Symptoms': 'Brown, circular spots on leaves, yellowing around spots, stunted growth.',
            'Treatment (Brand Names)': 'Tilt, Folicur',
            'How to Use': 'Apply at the first signs of disease; follow label instructions.',
            'Recommendations': 'Rotate crops and practice good field management. Use resistant varieties.'
        },
        'healthy': {
            'Name': 'Healthy',
            'Symptoms': 'No visible symptoms; healthy leaf color and structure.',
            'Treatment (Brand Names)': 'N/A',
            'How to Use': 'Continue regular monitoring and maintain good practices.',
            'Recommendations': 'Regularly inspect fields for early detection of diseases.'
        },
        'leaf_blast': {
            'Name': 'Leaf Blast',
            'Symptoms': 'Elliptical, greenish-gray lesions with a white center; can cause rapid plant death.',
            'Treatment (Brand Names)': 'Blast-Off, Tricyclazole (Beam)',
            'How to Use': 'Apply fungicide at the first sign of symptoms; repeat every 10-14 days if needed.',
            'Recommendations': 'Plant resistant varieties; monitor environmental conditions.'
        },
        'sheath_blight': {
            'Name': 'Sheath Blight',
            'Symptoms': 'Irregular, water-soaked lesions on the leaf sheath; can lead to lodging.',
            'Treatment (Brand Names)': 'Raxil, Headline',
            'How to Use': 'Apply at tillering stage or at first signs; follow label recommendations.',
            'Recommendations': 'Maintain proper irrigation and avoid excessive nitrogen fertilization.'
        }
    }
```

Treatment Recommendation

Process:

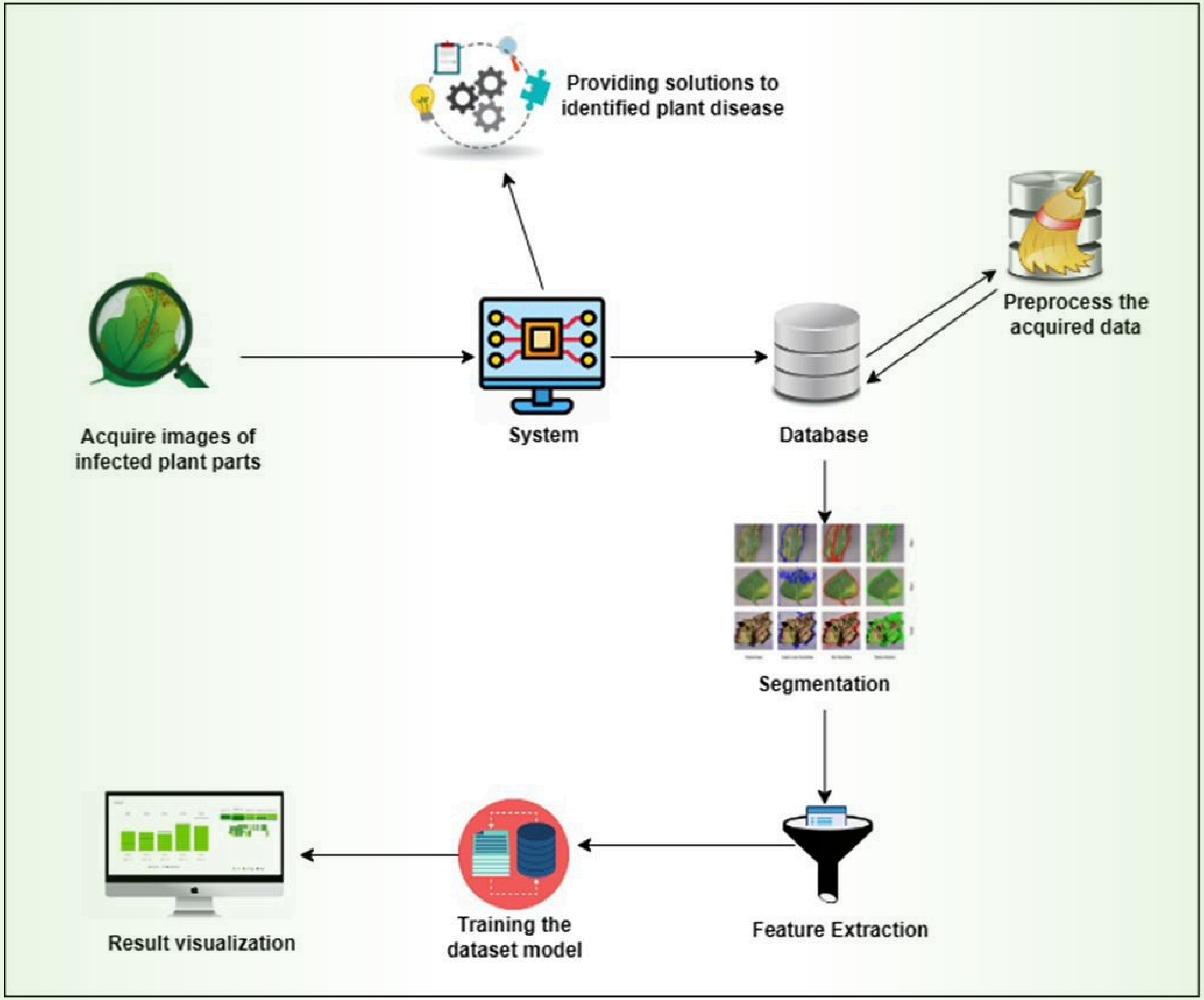
- Display treatments, including disease names, application methods, and prevention tips.

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
Model loaded successfully.
Treatment data loaded successfully.
Available Treatment Classes: ['bacterial_leaf_blight', 'brown_spot', 'healthy', 'leaf_blast', 'sheath_blight']
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step
```

Predicted: sheath_blight (99.91%)

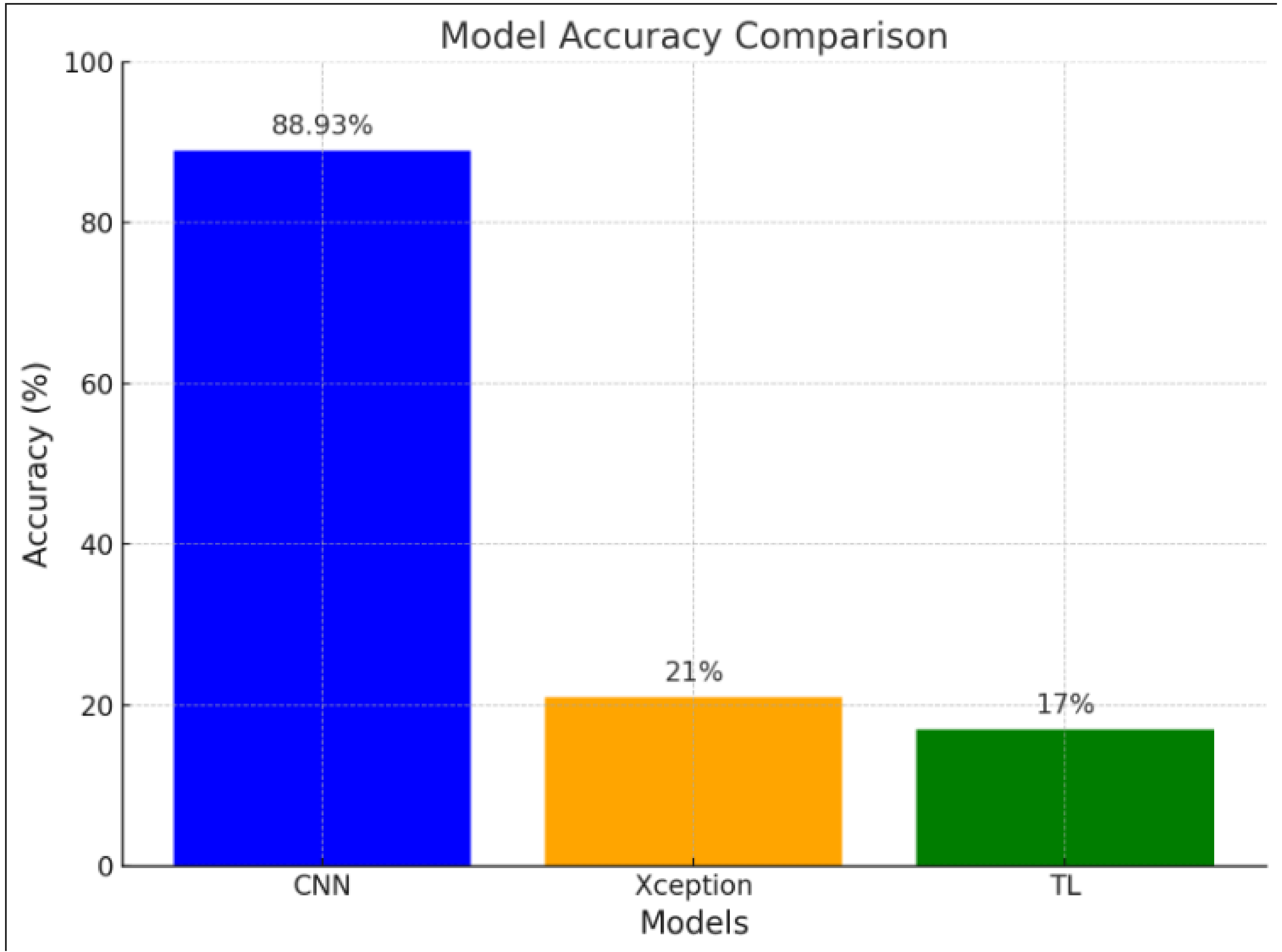
```
--- Treatment Suggestions ---
Disease Class: sheath_blight
Name: Sheath Blight
Symptoms: Irregular, water-soaked lesions on the leaf sheath; can lead to lodging.
Treatment (Brand Names): Raxil, Headline
How to Use: Apply at tillering stage or at first signs; follow label recommendations.
Recommendations: Maintain proper irrigation and avoid excessive nitrogen fertilization.
```

System Diagram

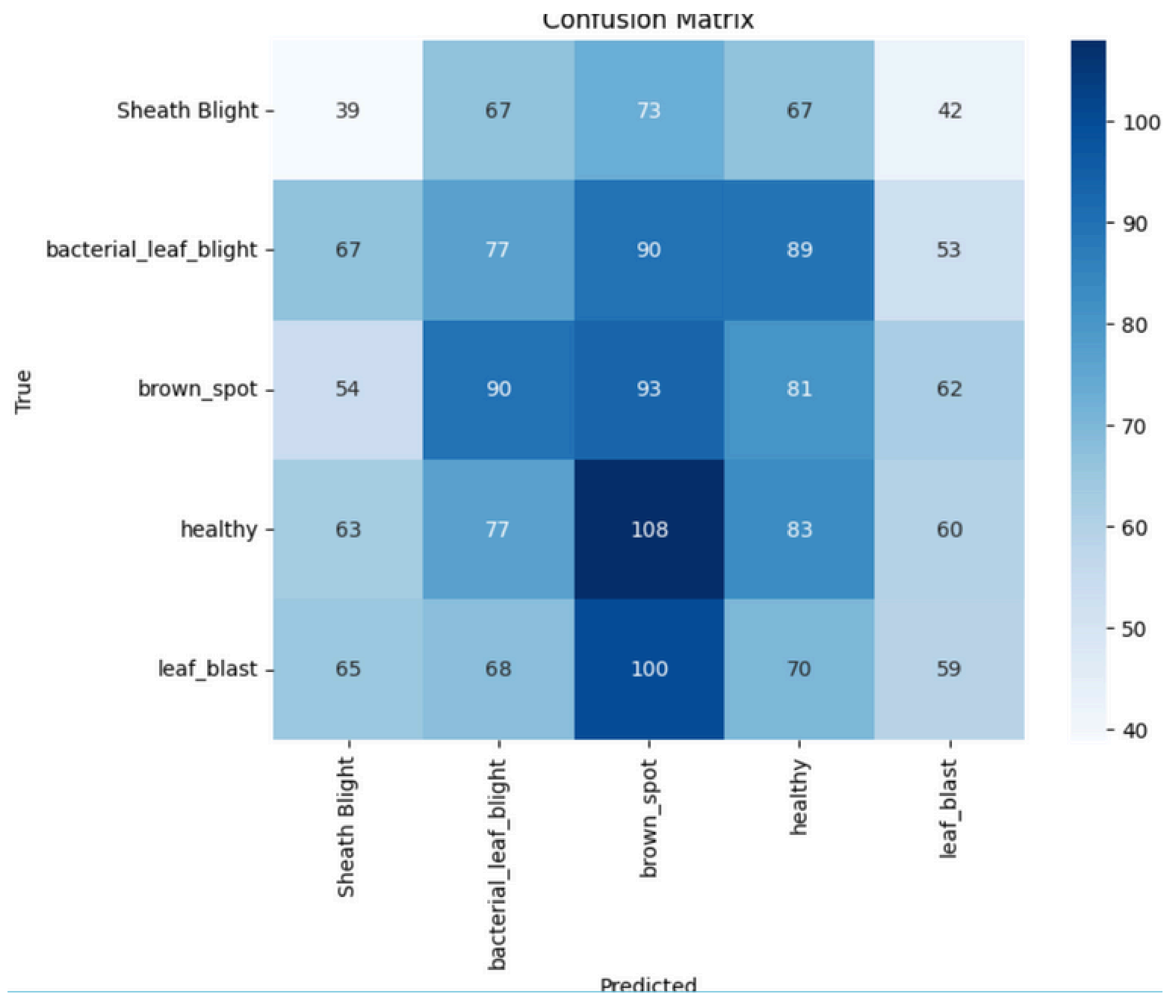


Performance Metrics

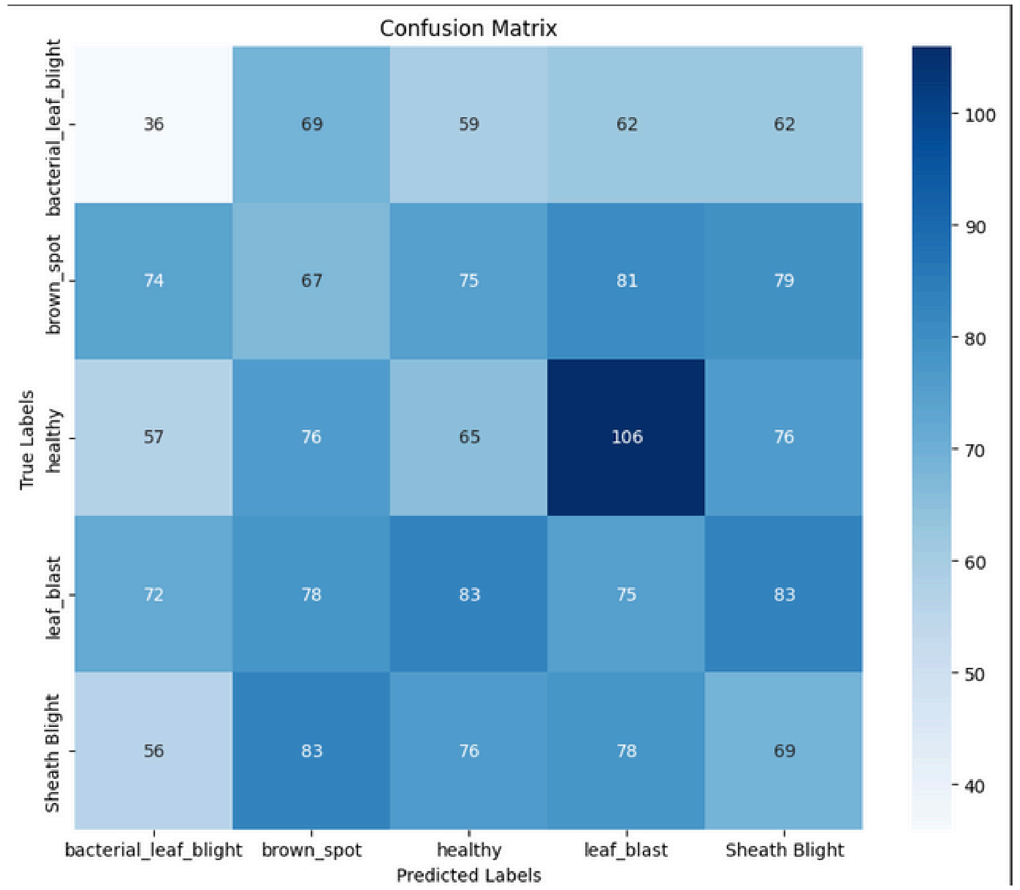
- CNN significantly outperforms other models with an accuracy of 88.93%, showcasing its robust architecture and effective data augmentation techniques.
- Xception and TL models demonstrate limited performance, achieving 21% and 17% accuracy, respectively.
- The bar chart underscores the superiority of CNN, making it the most reliable model for paddy disease identification.



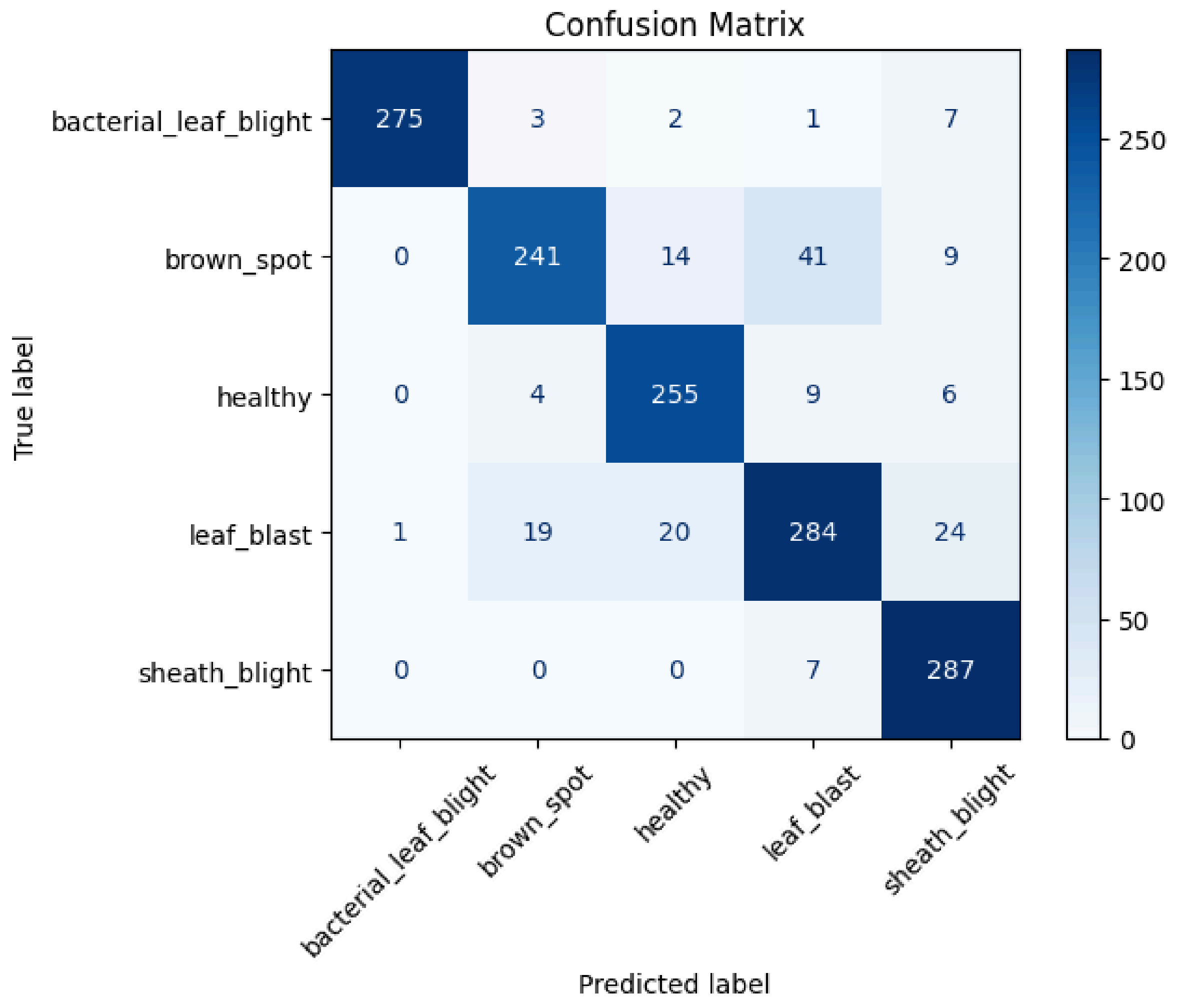
Model performance Analysis



TL



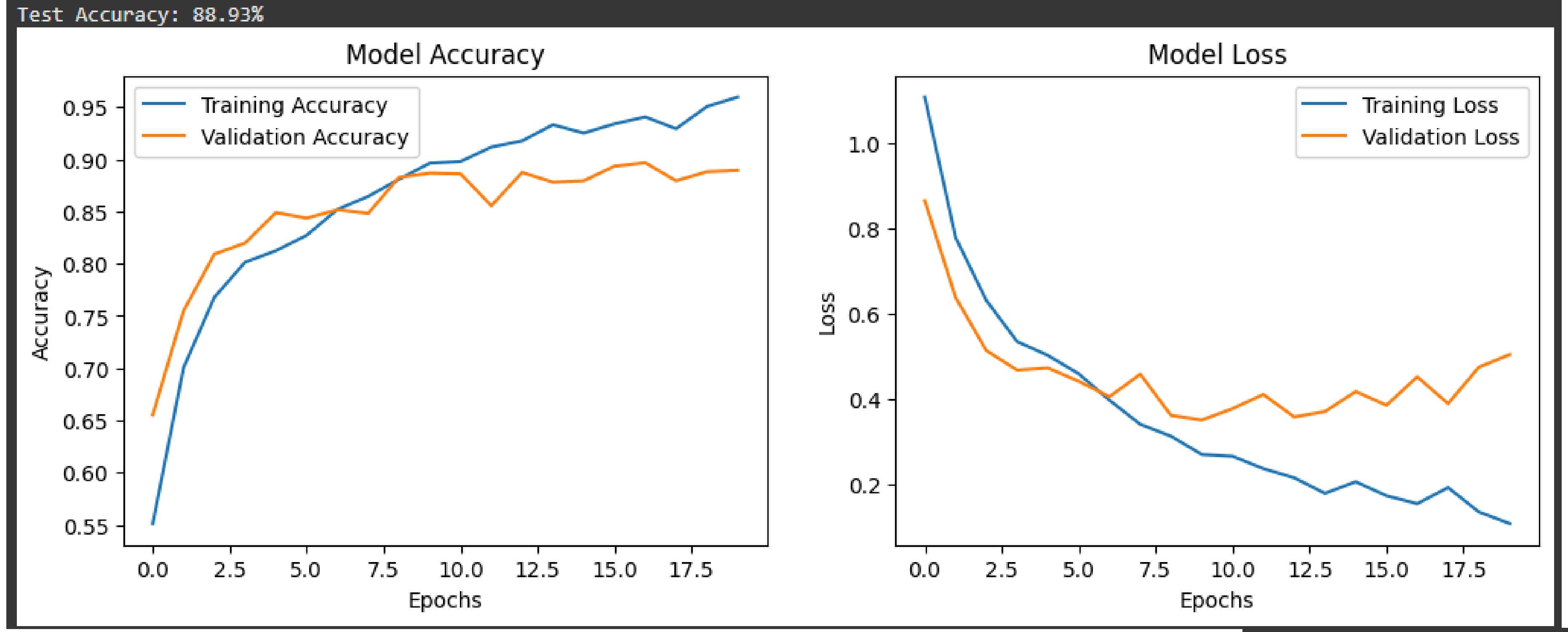
Xception



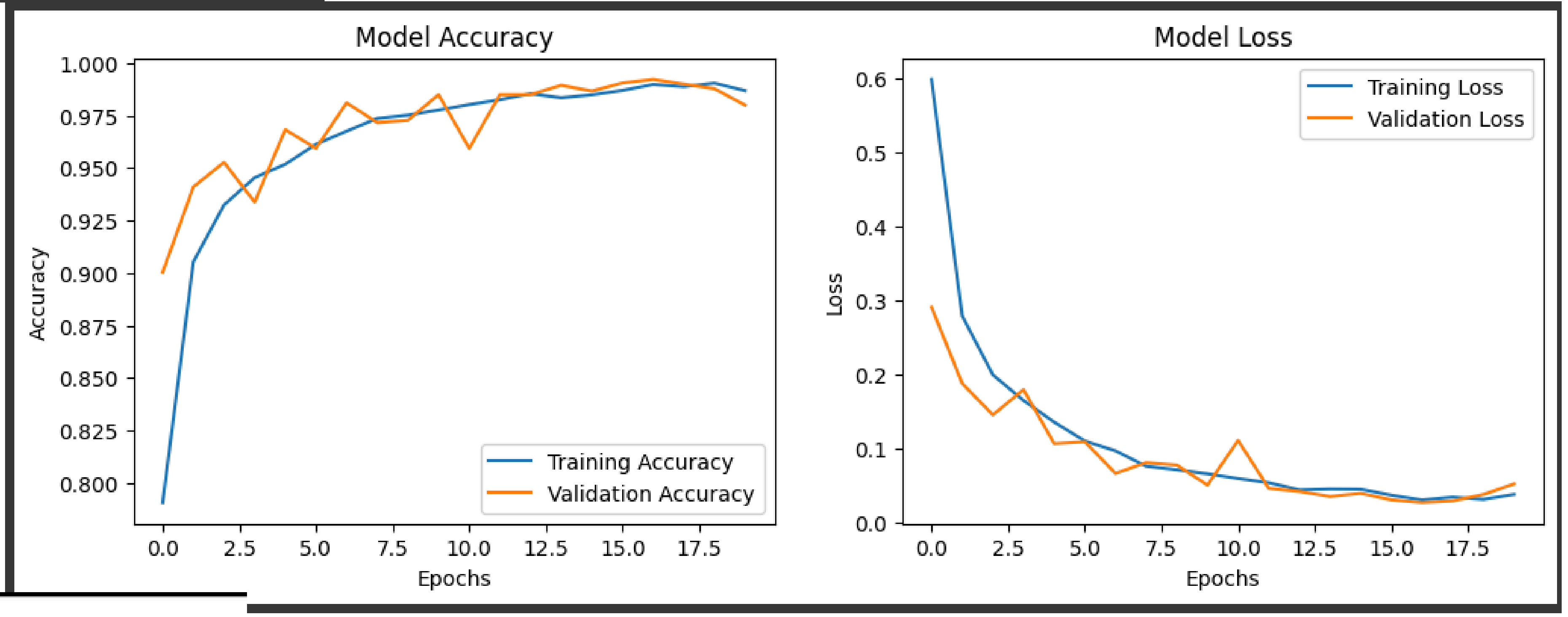
CNN

The confusion matrices highlight the effectiveness of CNN compared to TL and Xception, emphasizing its superior accuracy and reliability for paddy disease identification. Further optimization in TL and Xception may improve their performance.

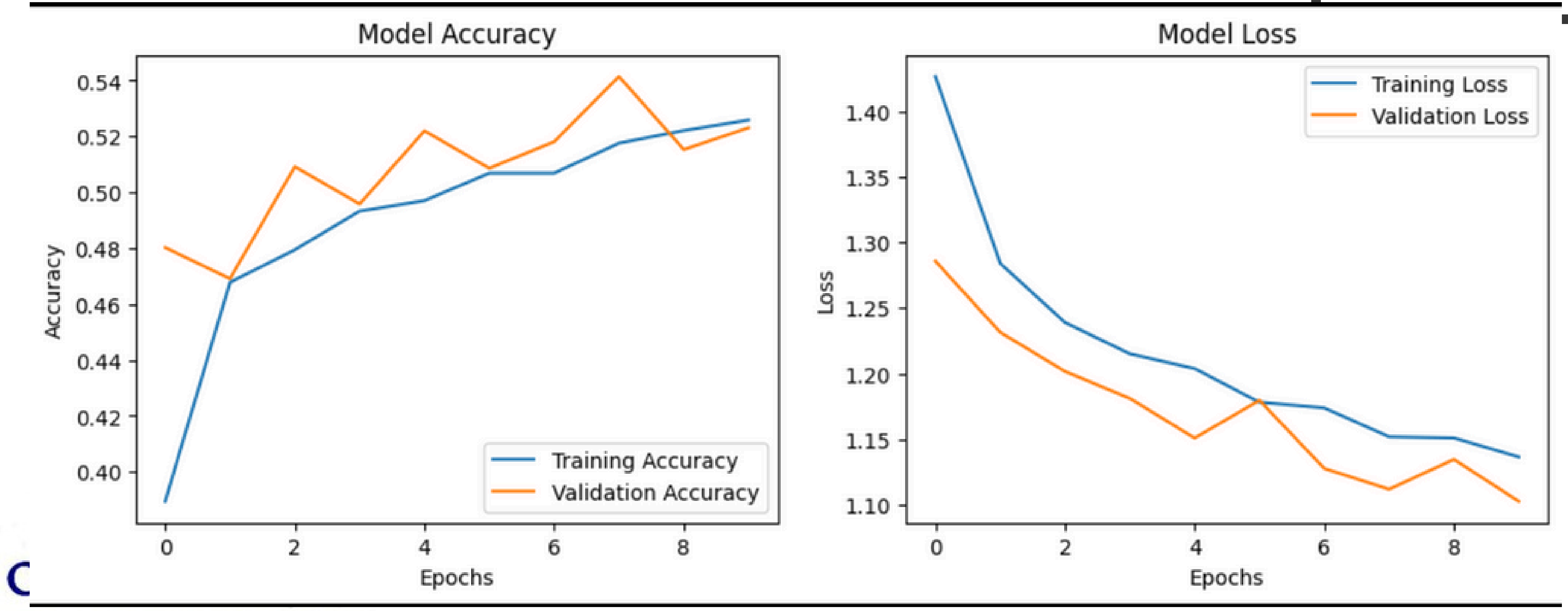
CNN



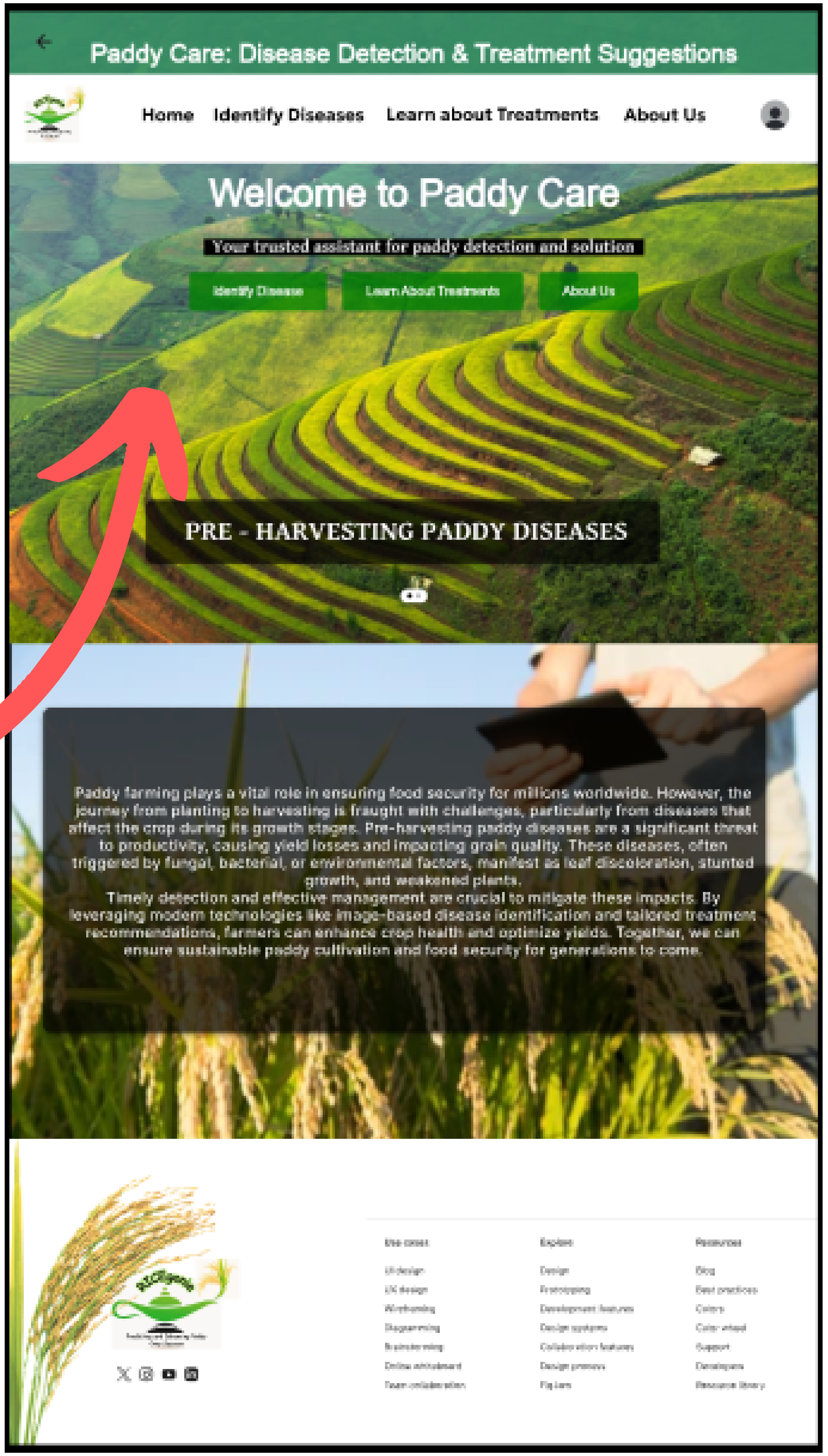
Xception



TL

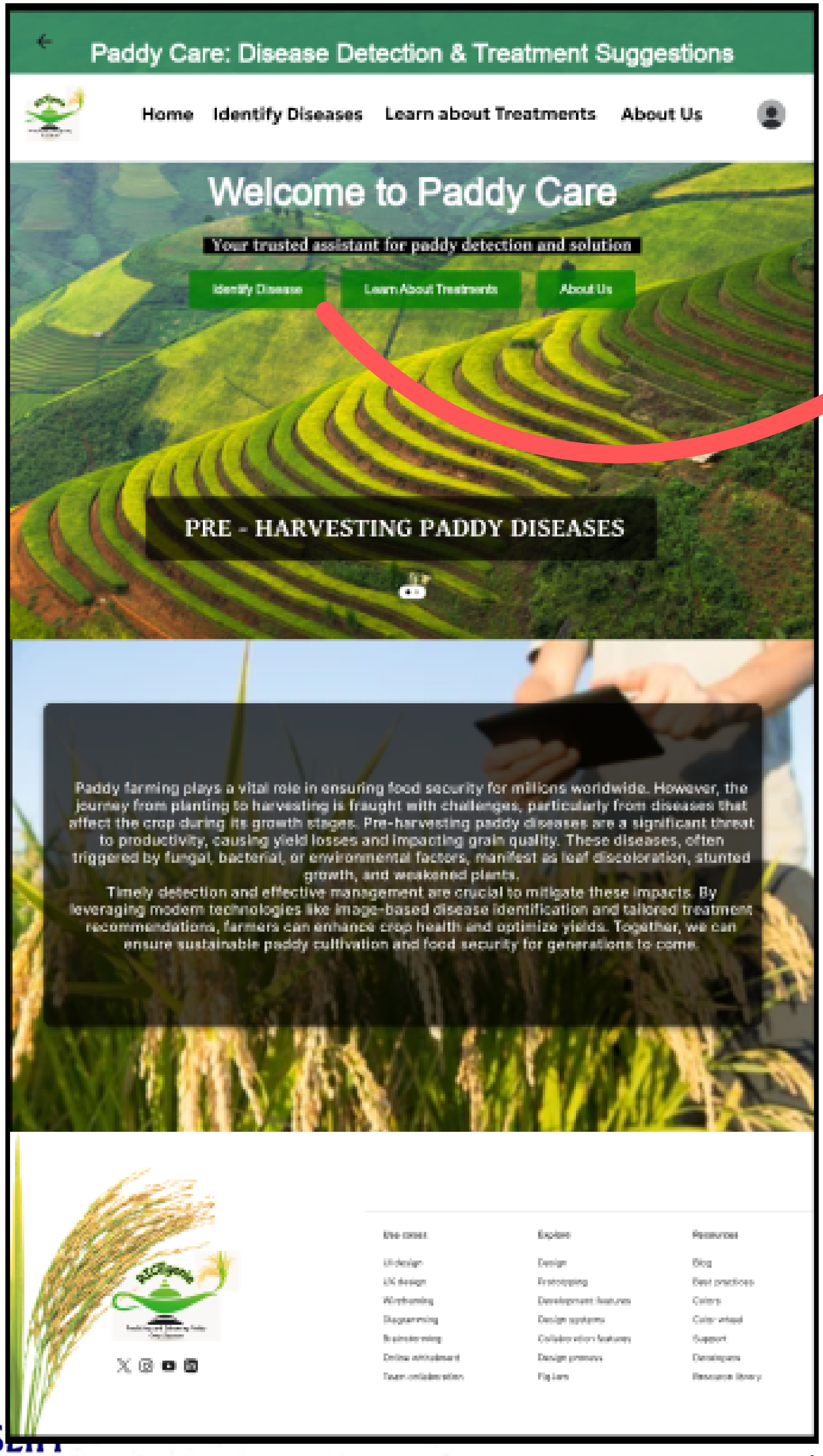


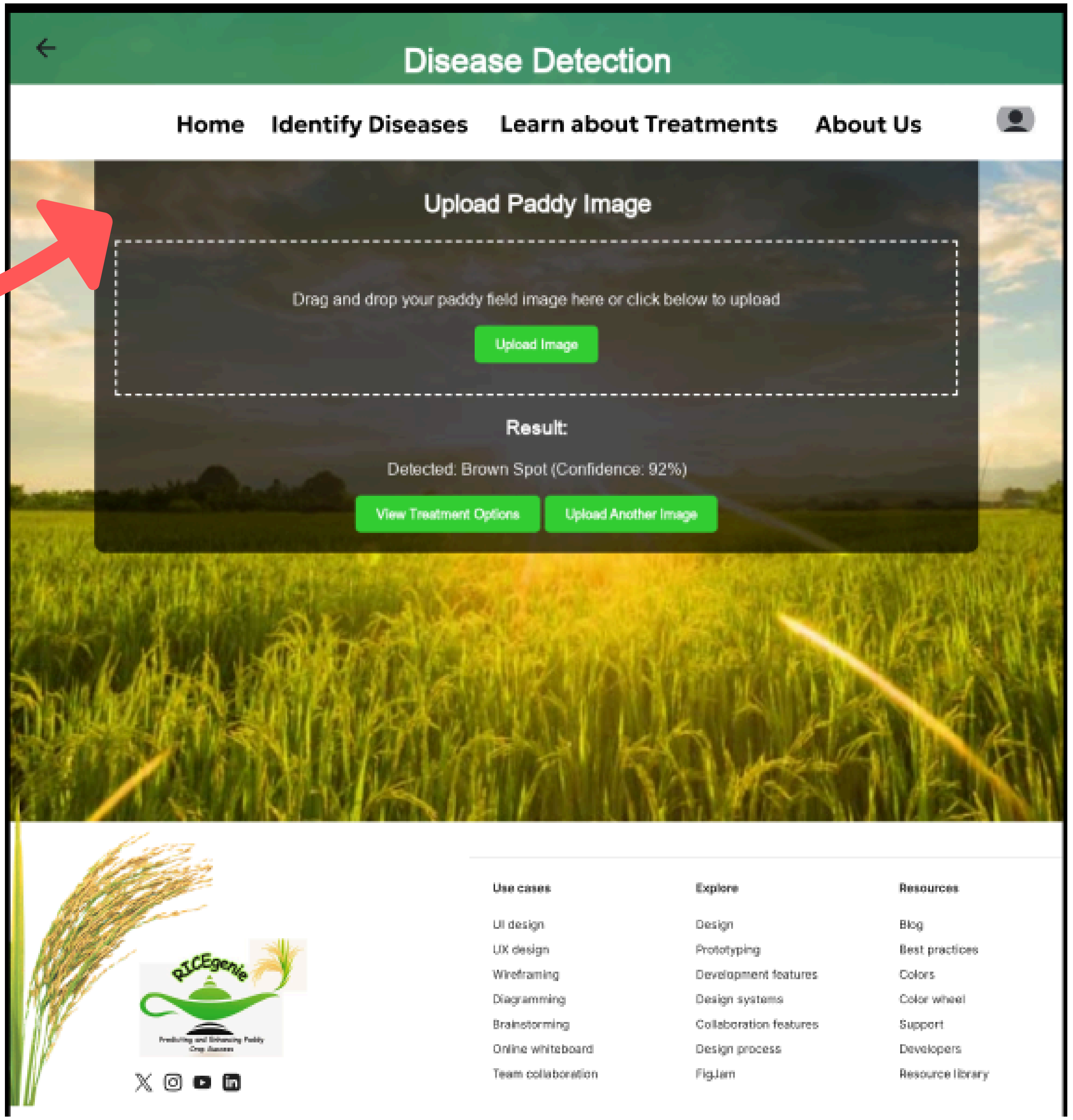
Effective performance relies on selecting the right architecture and balancing model capacity with dataset complexity.



Requirements

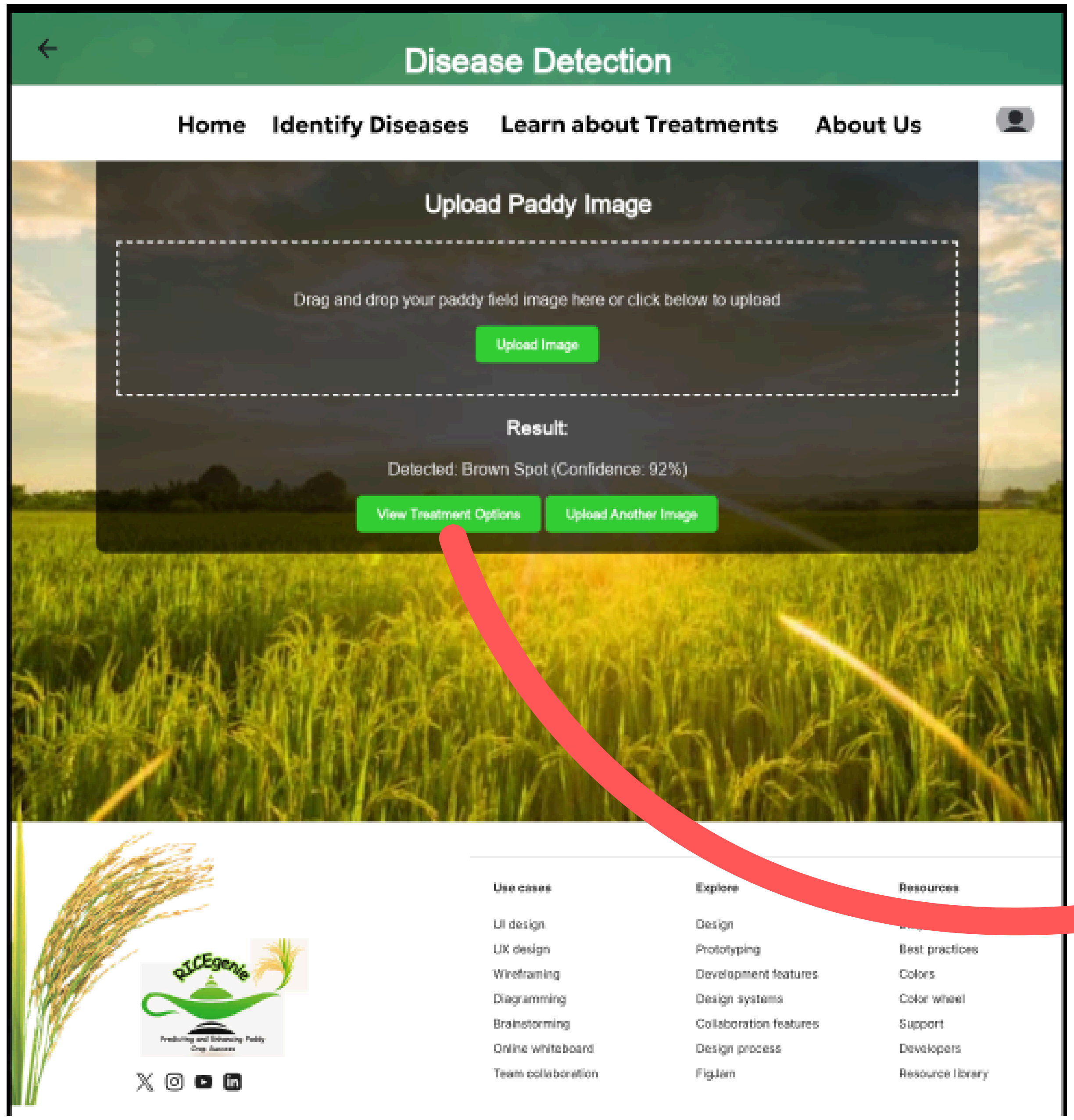
- **Log:** Farmers log to the Detect &Manage Diseases tab which user-friendly app interface.

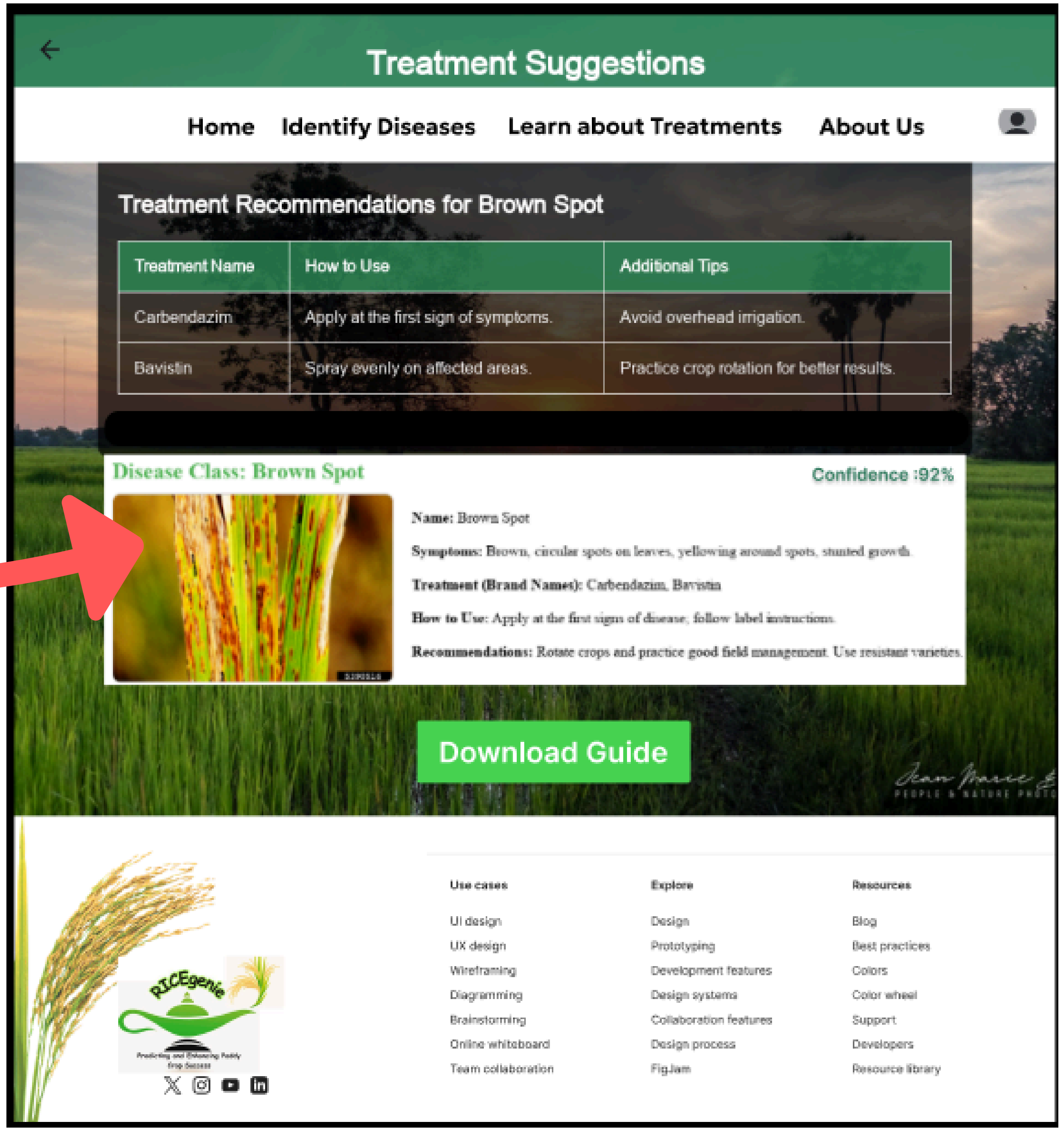




Requirements

Input: Farmer-uploaded images via a user-friendly app interface.





Requirements

Output: Disease identification and treatment recommendations.

Technologies

- Python (Back end)
- Tensor Flow (Framework)
- Deep learning (Classificatior
- ReactJS (Front end)
- Tailwind (for styles)
- Fast API
- Google Colab
- MySql
- Git Hub (Version control system)
- Trello(Project Management)

Requirements

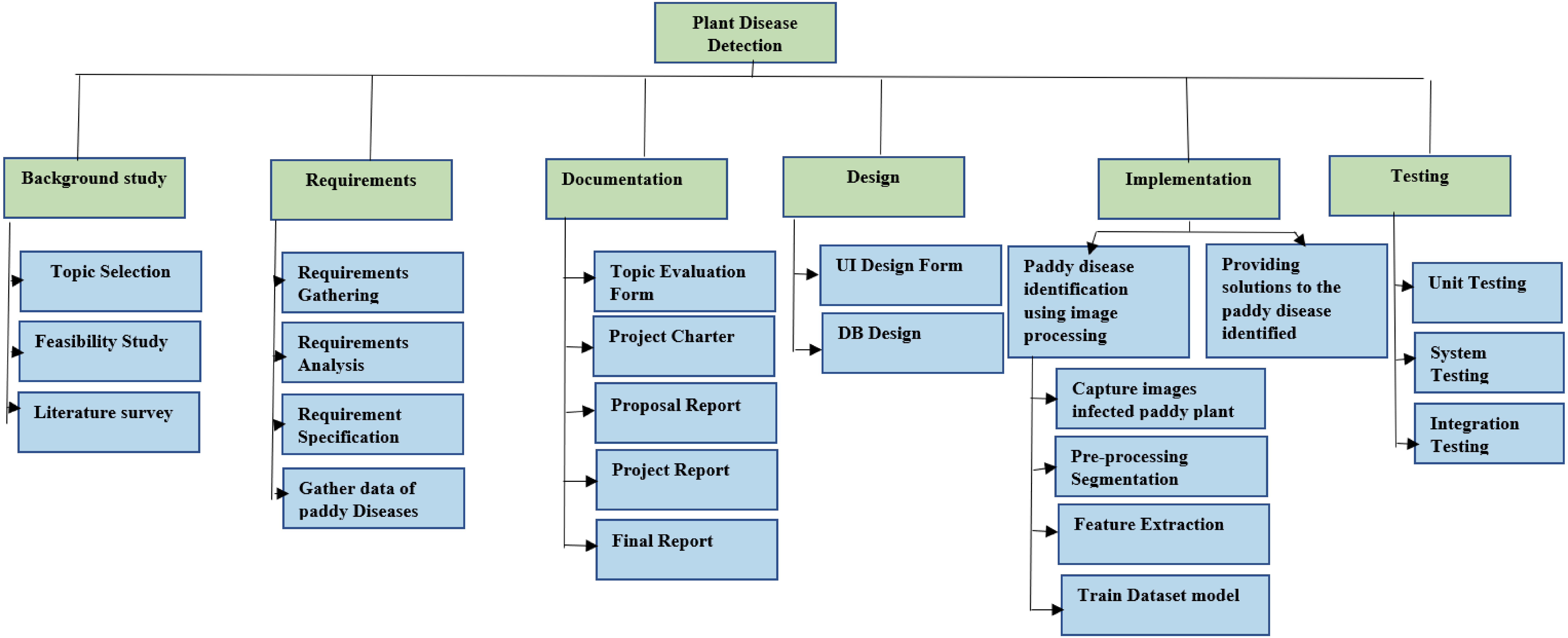
Non-functional requirements

- Availability – Accessible
- Reliability – perform without Errors
- Performance – quickly
- Usability – easy to use

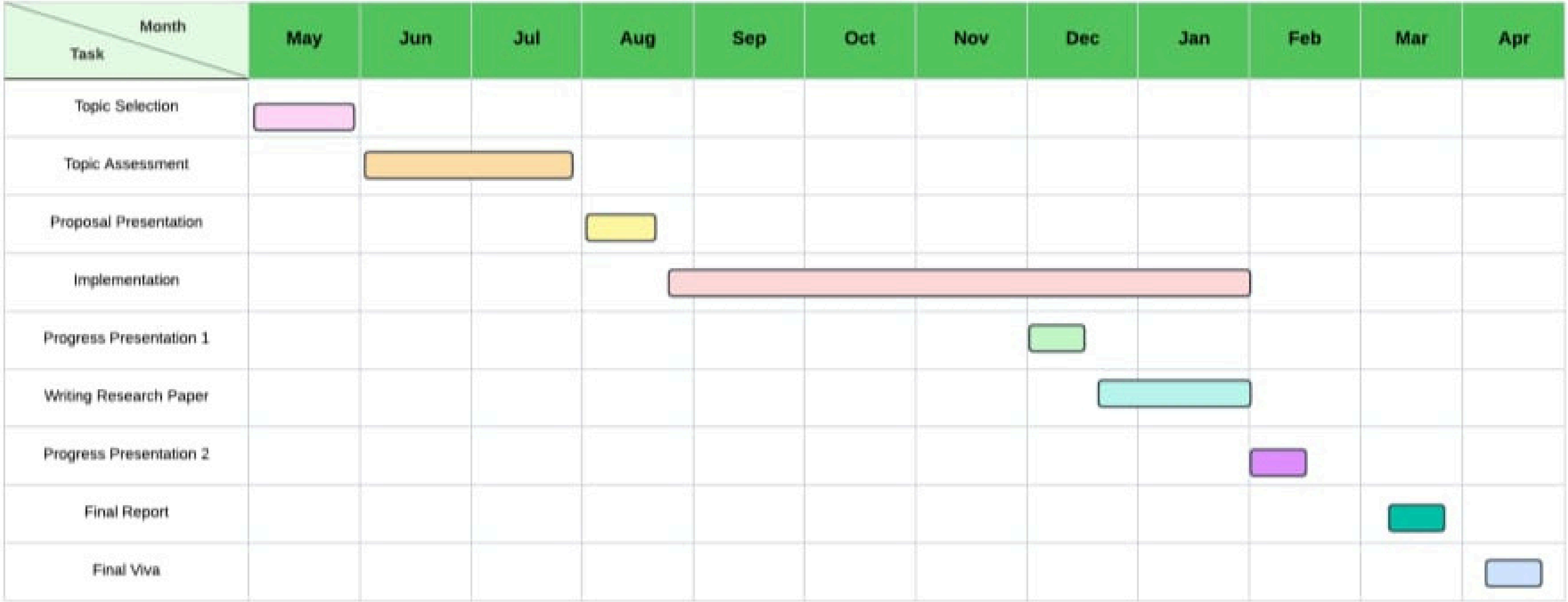
Functional requirements

- Ability to identify plant diseases
- Ability to suggest treatment to diseases

Work Breakdown Structure



Gantt chart



Month
Task
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
Topic Selection
Topic Assessment
Proposal Presentation
Implementation
Progress Presentation 1
Writing Research Paper
Progress Presentation 2
Final Report
Final Viva

REFERENCES

[1]https://www.researchgate.net/publication/344933690_Detection_and_Recognition_of_Paddy_Plant_Leaf_Diseases_using_Machine_Learning_Technique

[2] Kumar, R. et al. (2022) A systematic analysis of machine learning and deep learning based approaches for Plant Leaf Disease Classification: A Review, Journal of Sensors. Hindawi. Available at: https://doi.org/10.1155/2022/3287561 (Accessed: March 27, 2023).

[3] Kumar, R. et al. (2023) A systematic analysis of machine learning and deep learning based approaches for Plant Leaf Disease Classification: A Review, Journal of Sensors.

IT21227868 | PIYUMANI K.V.P

Paddy Variety Prediction System

Information Technology

INTRODUCTION

This Research component which The Paddy Variety Prediction Model aims to ,

- **Identify the most suitable paddy varieties for specific regions in Sri Lanka**

- Based on local soil and weather conditions. By leveraging data analytics and machine learning techniques, this model seeks to optimize paddy cultivation, enhance crop yield, and improve economic outcomes for farmers.

Research Gap

- Currently, there is a lack of integrated systems that consider local environmental conditions for predicting the best-suited paddy varieties for different regions in the contry. While various studies have explored crop yield prediction, specific focus on paddy variety suitability under diverse environmental conditions is limited. This project fills this gap by providing a comprehensive model that aids farmers in selecting the optimal paddy variety for their fields.

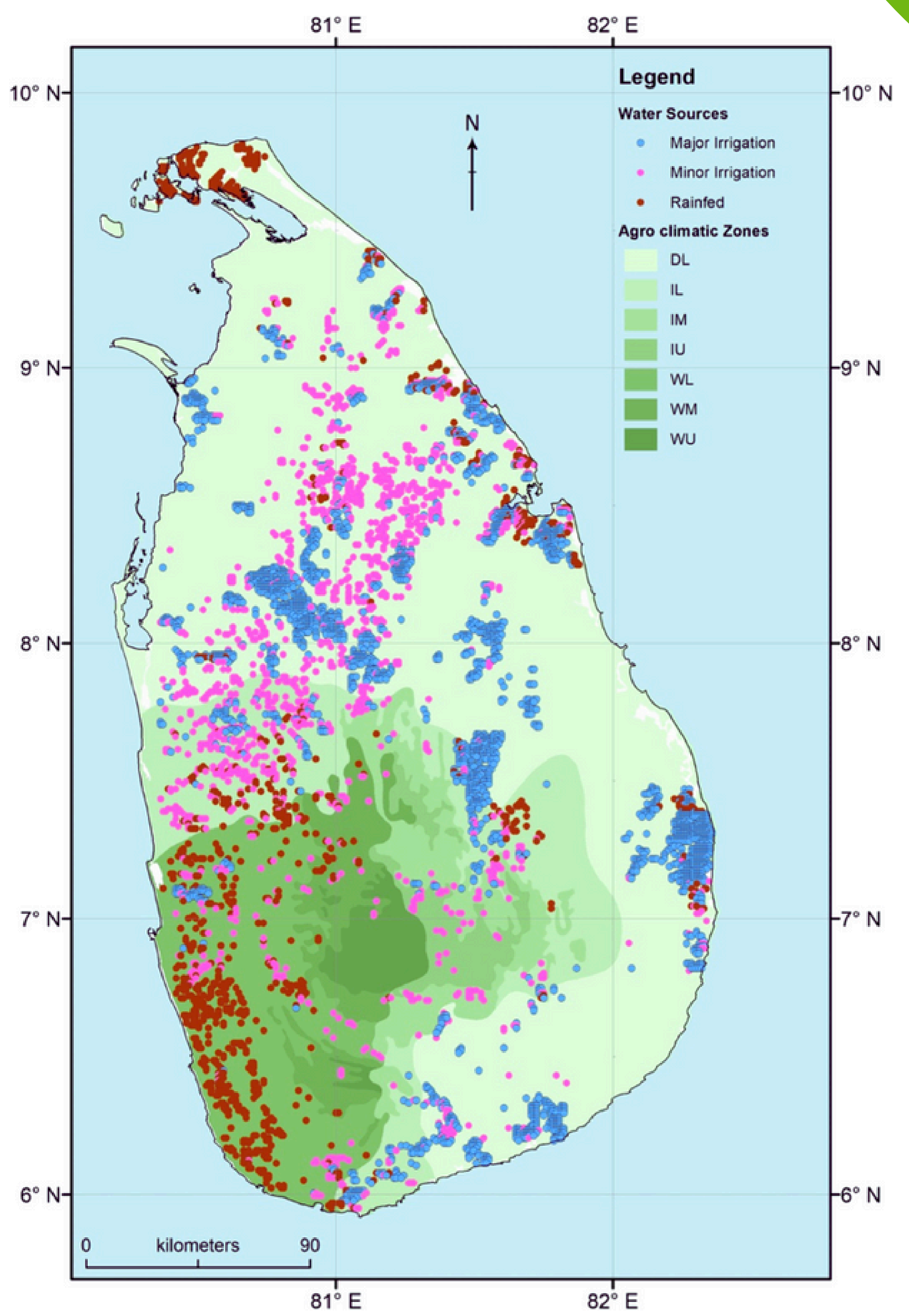
Research Problem

The primary research problem is to establish a predictive relationship between different paddy varieties and the local environmental conditions, such as soil type, pH, moisture, temperature, and nutrient levels. The challenge lies in accurately predicting which paddy varieties can thrive in specific conditions to maximize yield and resilience to environmental stressors.

Specific and Sub Objectives

- **Specific Objectives :**
  - To develop a predictive model for determining the suitability of paddy varieties based on environmental data.

- **Sub Objectives :**
  - Collect and analyze data on paady varieties, various soil parameters (pH, moisture, nutrient content) and weather conditions (temperature, rainfall).
  - Identify the key factors influencing the growth and yield of different paddy varieties.
  - Develop a machine learning model to predict the most suitable paddy variety for an area of the country under given set of environmental conditions.



Mapping Productivity-related Spatial Characteristics in Rice-based Cropping Systems in Sri Lanka

Methodology

- **Data Collection:**Gather data on soil properties (pH, NPK levels, soil moisture), and climatic conditions (temperature, humidity, rainfall) from relevant agricultural departments and field studies.

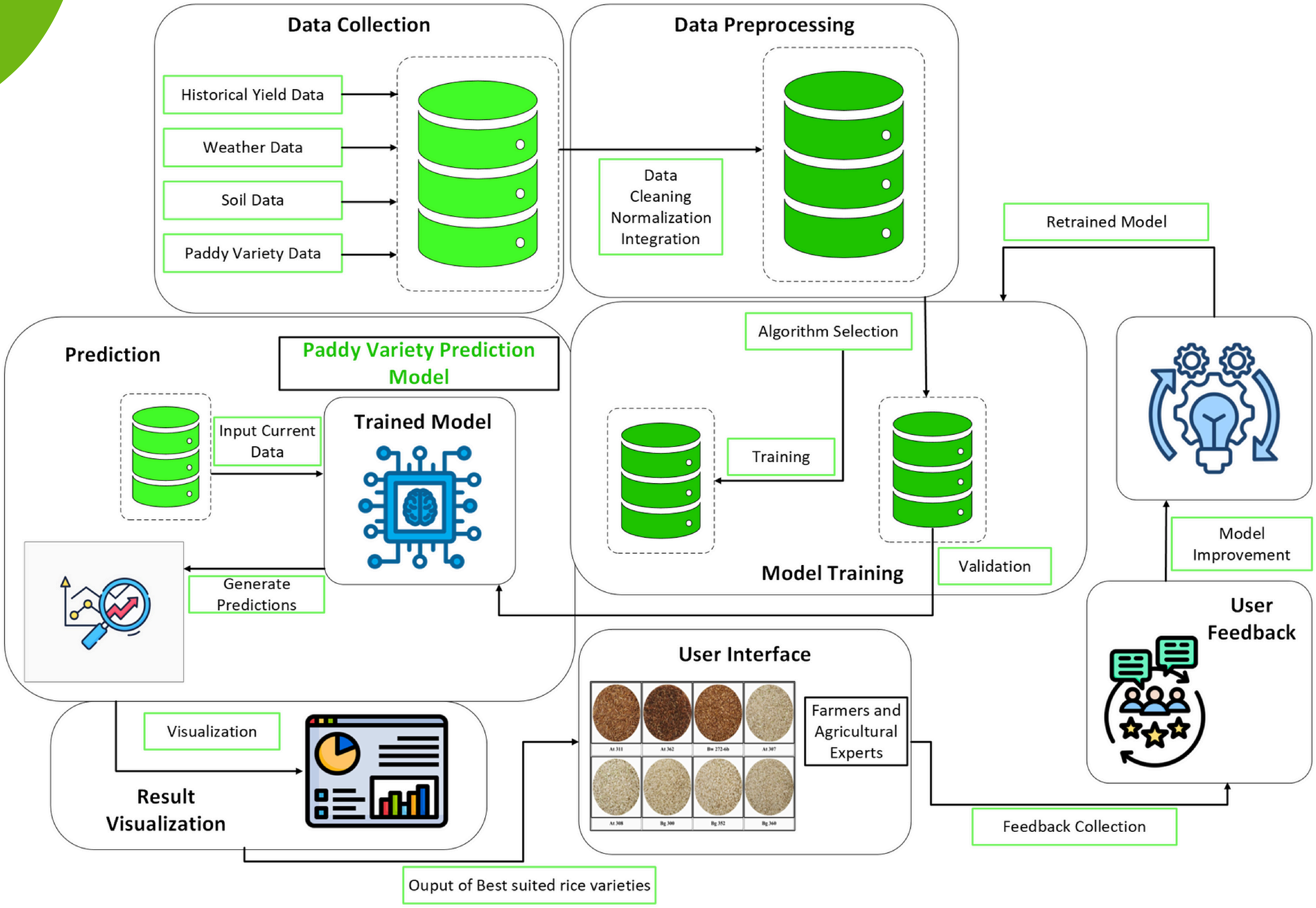
- **Data Analysis:**Analyze the collected data to identify patterns and relationships between soil/climatic factors and paddy variety performance.

- **Model Development:**Use machine learning techniques, such as decision trees, support vector machines, or neural networks, to develop a predictive model.
- Train the model on historical data to learn the relationship between environmental conditions and paddy variety yield.

- **Model Validation:**Validate the model using a separate dataset to assess its accuracy and reliability in predicting suitable paddy varieties.

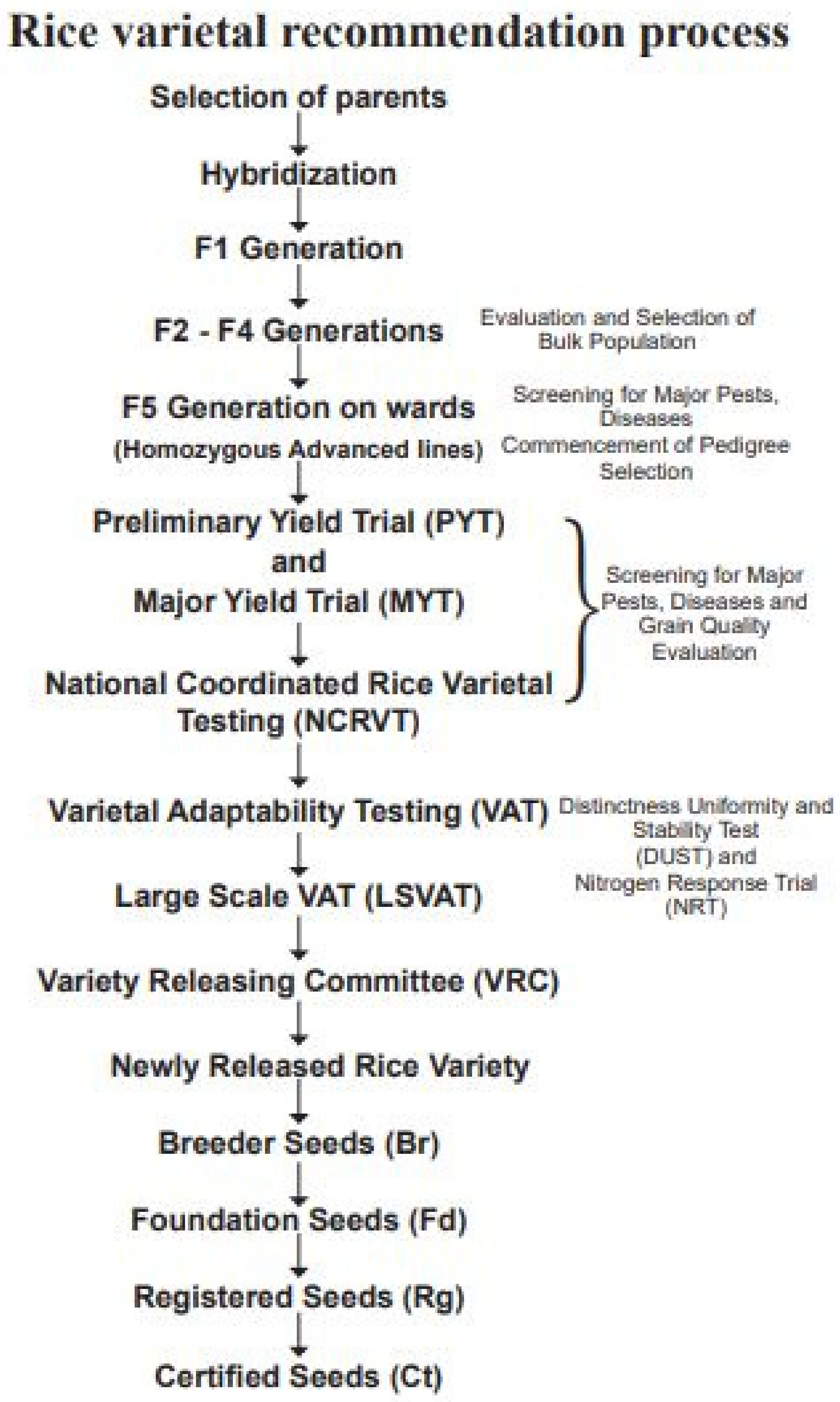
- **Implementation:**Develop a user-friendly interface for farmers to input their local conditions and receive recommendations on the best-suited paddy varieties.

System Diagram

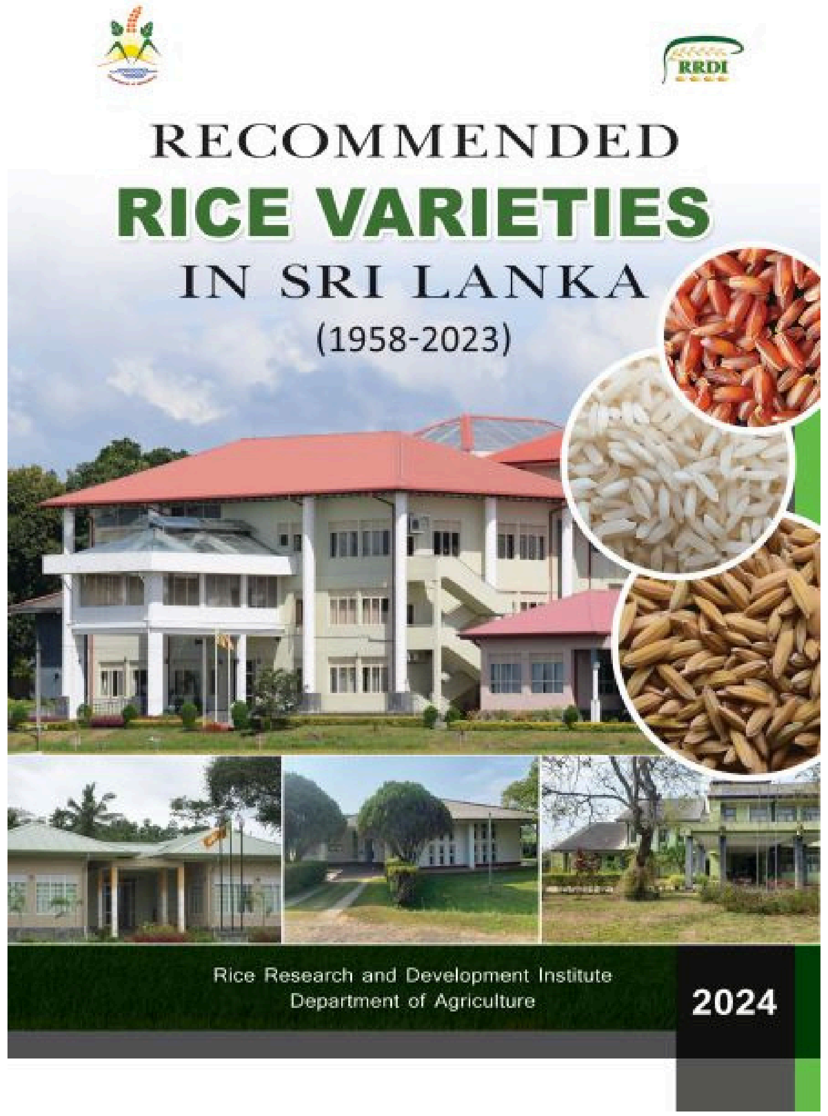


1. Data Collection

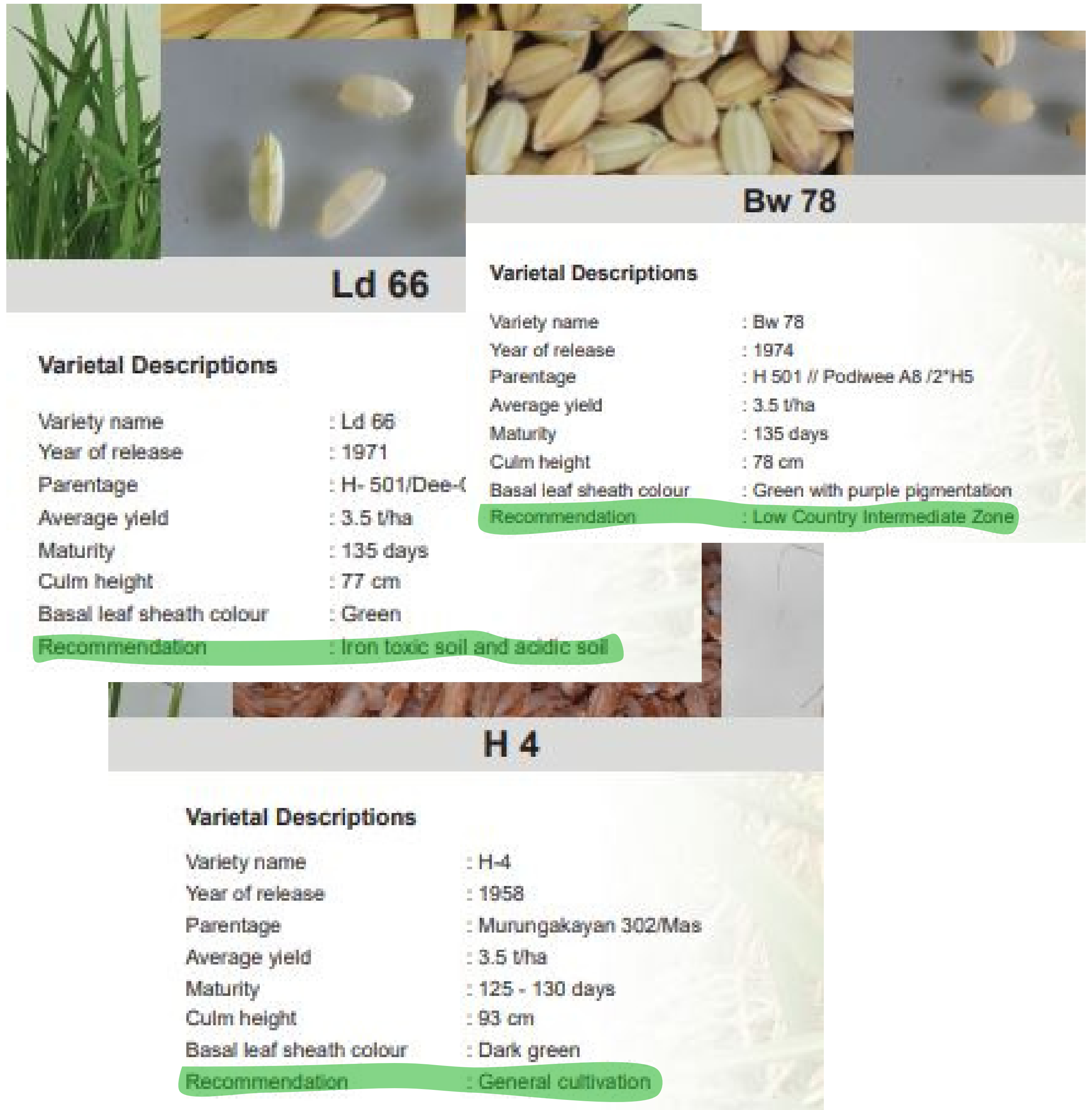
- Sri Lanka's Rice cultivation has a process of recommending the Rice varieties with a scientific aspect.



- After the completion of the process they categorized the rice varieties with the recommendations that are favourable for the growth of a particular rice variety.



Recommended rice varieties book published by Rice Research and Development Institute – Bathlagoda, Sri Lanka



Variety Name	Year o	Parentage	Average	Maturity	Age Gro	Basal Leaf She	Recommendation	Brown R	Milling Re	Head Ric	Gelatinization	1000 Grain	Grain Shape	Pericar	Bushe	Reaction to Pest and
H 4	1958	Murungakayan 302/	3.5	125-130	4-4.5	Dark green	General cultivation	80	72.7	61.4	High	28.3	Long Medium	Red	20.8	Brown Planthopper:
H 7	1964	Pachchaperumal/M	3.6	105	3.5	Green	General cultivation	79.4	72.3	60	Intermediate	23.6	Intermediate Me	White	21.2	Blast: R, Bacterial Le
H 8	1966	H 4/Podiwee A8	3.7	135	4-4.5	Green	General cultivation	76.61	73.4	52	High	18.2	Short Round	White	22.3	Blast: MR, Bacterial L
H 9	1968	C104/Mas/Panduru	3	155	5-6	Green	General cultivation	80.4	74	70.7	High	22.5	Intermediate Me	White	20.4	Blast: Susceptible, B
H 10	1968	Pachchaperumal/M	3	90	3	Green	General cultivation	76.61	73.4	52	High	26	Intermediate Me	Red	19.5	Blast: Susceptible, B
62-355	1968	Pachchaperumal/H	3.9	90-95	3	Green	Rainfed areas	75.6	72	57.5	Intermediate	29	Intermediate Bo	Red	17.5	Brown Planthopper:
Bg 11-11	1970	Engkatek/2*H 8	4.5	125-130	4-4.5	Purple	General cultivation	79	72.2	62.3	Intermediate	13.8	Short Round	White	21.7	Brown Planthopper:
Bg 34-6	1971	IR 8-246///Pachchap	5.5	105	3	Green	General cultivation	79.2	72.2	67.1	Intermediate	25.5	Intermediate Bo	Red	21.3	Blast: Susceptible, B
Bg 34-8	1971	IR-246///Pachchape	6.1	96	3	Green	General cultivation	80.4	74	70.7	Intermediate	26	Intermediate Bo	White	21.7	Brown Planthopper:
Ld 66	1971	H-501/Dee-Geo-Wc	3.5	135	4-4.5	Green	Iron toxic soil and acidic	79.3	72.9	63.8	High	21.6	Intermediate Me	White	22.1	Blast: Susceptible, B
MI 273	1971	Gamma Irradiated H	4.2	135	4-4.5	Green	General cultivation	80.4	74	70.7	Intermediate	28.4	Long Medium	Red	22.5	Brown Planthopper:
Bg 3-5	1973	BG 94-1/BG 350	3.8	100	3	Green	General cultivation	78.2	70.5	60.8	Intermediate	24.5	Medium Slender	White	22.5	Brown Planthopper:
Bg 94-1	1975	IR 262/Ld 66	4.1	105	3	Green	General cultivation	79.3	72.9	63.8	High-Intermed	28.3	Long Medium	White	20.8	Brown Planthopper:
At 16	1977	IR 8/H4	3.8	105	4-4.5	Green	Southern province	79	72.2	62.3	High	28.4	Intermediate Me	Red	19.1	Blast: Moderately Re
Bg 90-2	1975	IR 262/Remadja	6.5	120	4-4.5	Green	General cultivation	78.6	72.9	63.7	Intermediate	29.3	Long Medium	White	22.1	Brown Planthopper:
Bw 78	1974	H 501 // Podiwee A	3.5	135	4-4.5	Green with pu	Low Country Intermedia	79.3	72.9	63.8	High	20.2	Short Round	White	22.2	Brown Planthopper:
Bg 94-2	1978	IR 262/Ld 66	5.9	105	3	Green	General cultivation	81.4	76.7	71.1	High	25.6	Long Medium	Red	22.3	Brown Planthopper:
Bw 361	2002	IR 36 / Bw 267-3-11	4.5	105	3.5	Green	General cultivation	81.3	74.4	64.1	Low	21.4	Intermediate Me	Red	21	Brown Planthopper:
Bw 361	2002	IR 36 / Bw 267-3-11	4.5	105	3.5	Green	Iron toxic soil and acidic	81.3	74.4	64.1	Low	21.4	Intermediate Me	Red	21	Brown Planthopper:
At 362	2002	At 85-2/Bg 380	6	110	3.5	Green	General cultivation	78.3	68.6	54.2	Low	25.7	Long Medium	Red	20.6	Brown Planthopper:
Bg 310	2014	Bg 300/Pokkali	5.6	95-98	3	Green	Saline prone areas	81.4	76.7	71.1	Intermediate	27.5	Intermediate Bo	White	21.3	Brown Planthopper:
At 373	2014	IR 70422-66-5-2/Bg	4.9	103	3.5	Green	General cultivation	80.1	73.3	67.6	High-Intermed	10.5	Short Oblong	White	19.8	Brown Planthopper:

Sheet1

- So by extracting the data reguarding the rice varietis it has created a comprehensive data set of currently recommended rice varieties to grow in Sri Lanka.

- And with using the data on different climate zones and environmental conditions in Sri Lanka inco-operate with the data of Department of Census and Statistics - Sri Lanka, a comprehensive and robust data set has created reguarding 25 districs of Sri Lanka.

Province	District	Annual Temperature	Annual Humidity	Annual Rainfall	Recommendation
Western Province	Colombo	High	Normal	High	General cultivation
Western Province	Gampaha	High	Normal	High	General cultivation
Western Province	Gampaha	High	Normal	High	Low Country Wet Zone
Western Province	Kalutara	Normal	Normal	High	General cultivation
Western Province	Kalutara	Normal	Normal	High	Low Country Wet Zone
Western Province	Kalutara	Normal	Normal	High	Coastal saline areas
Western Province	Kalutara	Normal	Normal	High	Kalutara and Galle Districts
Central Province	Kandy	Normal	Normal	Normal	General cultivation
Central Province	Kandy	Normal	Normal	Normal	High potential area
Central Province	Matale	Low	High	Normal	General cultivation
Central Province	Matale	Low	High	Normal	Rainfed areas
Central Province	Nuwara Eliya	Low	High	Normal	General cultivation
Central Province	Nuwara Eliya	Low	High	Normal	Rainfed areas
Southern Province	Galle	High	High	Normal	General cultivation
Southern Province	Galle	High	High	Normal	Wet Zone
Southern Province	Galle	High	High	Normal	Kalutara and Galle Districts
Southern Province	Matara	Normal	High	Normal	General cultivation
Southern Province	Matara	Normal	High	Normal	Wet Zone
Southern Province	Hambantota	High	Normal	Low	General cultivation
Southern Province	Hambantota	High	Normal	Low	Saline prone areas
Southern Province	Galle	High	High	Normal	Southern province
Southern Province	Matara	Normal	High	Normal	Southern province
Southern Province	Hambantota	High	Normal	Low	Southern province

SriLankaDistricts

2. Data Analysis

- Load the both data sets into dataframes and combine both into single dataframe using inner join under referrence of "Recommendation" column and create a unified dataset.

```python
# Load the datasets
varieties_file_path = '/content/drive/MyDrive/Rice Genie/RiceVarietiesData.xlsx'
district_file_path = '/content/drive/MyDrive/Rice Genie/SriLankaDistricts.csv'

varieties_data = pd.read_excel(varieties_file_path, sheet_name='Sheet1')
district_data = pd.read_csv(district_file_path)

#combined_data = pd.merge(varieties_data, district_data, how='cross')
combined_data = pd.merge(varieties_data, district_data, how='inner', on='Recommendation')
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Variety Na	Year of Re	Parentage	Average Yi	Maturity (c	Age Group	Basal Leaf	Recomme	Brown Ric	Milling Re	Head Rice	Gelatiniza	1000 Grair	Grain Sha	Pericarp C	Bushel We	Reaction t	Province	District	Annual Te	Annua
2	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Western P	Colombo	High	Norm
3	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Western P	Gampaha	High	Norm
4	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Western P	Kalutara	Normal	Norm
5	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Central Pr	Kandy	Normal	Norm
6	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Central Pr	Matale	Low	High
7	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Central Pr	Nuwara El	Low	High
8	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Southern	Galle	High	High
9	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Southern	Matara	Normal	High
10	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Southern	Hambantc	High	Norm
11	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Northern	Jaffna	High	Norm
12	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Northern	Kilinochcl	Normal	Norm
13	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Northern	Mannar	High	Norm
14	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Northern	Vavuniya	High	Norm
15	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Northern	Mullaitivu	Normal	Norm
16	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Eastern Pr	Trincomal	High	Norm
17	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Eastern Pr	Batticaloa	High	Norm
18	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	Eastern Pr	Ampara	High	High
19	H 4	1958	Murungak	3.5	125-130	4-4.5	Dark greer	General cu	80	72.7	61.4	High	28.3	Long Medi	Red	20.8	Brown Pla	North Wes	Kurunegal	High	Low

Data Pre-Processing

- Convert "Matuarity" column values into numeric averages which we will use as a feature and also uses to categorize the life span of paddy varieties into age groups.

```
def convert_maturity_to_numeric(value):
    if isinstance(value, str) and '-' in value:
        parts = value.split('-')
        return (int(parts[0]) + int(parts[1])) / 2
    try:
        return float(value)
    except ValueError:
        return np.nan

# Convert 'Maturity (days)' to numeric averages
combined_data['Maturity (days)'] = combined_data['Maturity (days)'].apply(convert_maturity_to_numeric)
combined_data.dropna(subset=['Maturity (days)'], inplace=True)
```

```
# Create a list of recommendation keywords/phrases that indicate suitability
recommendation_keywords = [
    "Low Country Wet Zone",
    "High potential area",
    "Rainfed areas",
    "Wet Zone",
    "Saline prone areas",
    "Northern region",
    "Saline areas",
    "Iron toxic soil and acidic soil",
    "Major irrigation in Dry Zone and Intermediate Zone",
    "Southern province",
    "High potential areas in Low Country Wet Zone",
    "Dry Zone",
    "Rainfed areas of Dry and Intermediate Zone",
    "General cultivation"
]

# Creating a new column 'Suitability' based on multiple recommendation criteria
combined_data['Suitability'] = combined_data.apply(
    lambda row: 1 if (any(keyword.lower() in str(row['Recommendation']).lower() for keyword in recommendation_keywords)
                  and row['Average Yield (t/ha)'] >= 5.0) else 0,
    axis=1
)
```

- Using the Recommendation keywords a new column named "Suitability" has created base on multiple recommendation criteiria.

3. Model Development

Algorithm Selection

1. Random Forest Classifier

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       141
           1       1.00      1.00      1.00       147

    accuracy                           1.00       288
   macro avg       1.00      1.00      1.00       288
weighted avg       1.00      1.00      1.00       288

Accuracy: 100.00%
```

Accuracy : 100%

2. Gradient Boosting Classifier

```
Accuracy (Gradient Boosting): 0.9965277777777778
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       284
           1       0.80      1.00      0.89         4

    accuracy                           1.00       288
   macro avg       0.90      1.00      0.94       288
weighted avg       1.00      1.00      1.00       288

Mean Cross-Validation Score (Gradient Boosting): 0.9128992450638792
```

Accuracy : 99.65%

3. Logistic Regression

```
Accuracy (Logistic Regression): 0.9895833333333334
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       141
           1       0.98      1.00      0.99       147

    accuracy                           0.99       288
   macro avg       0.99      0.99      0.99       288
weighted avg       0.99      0.99      0.99       288
```

Accuracy : 98.95%

4. Support Vector Classifier

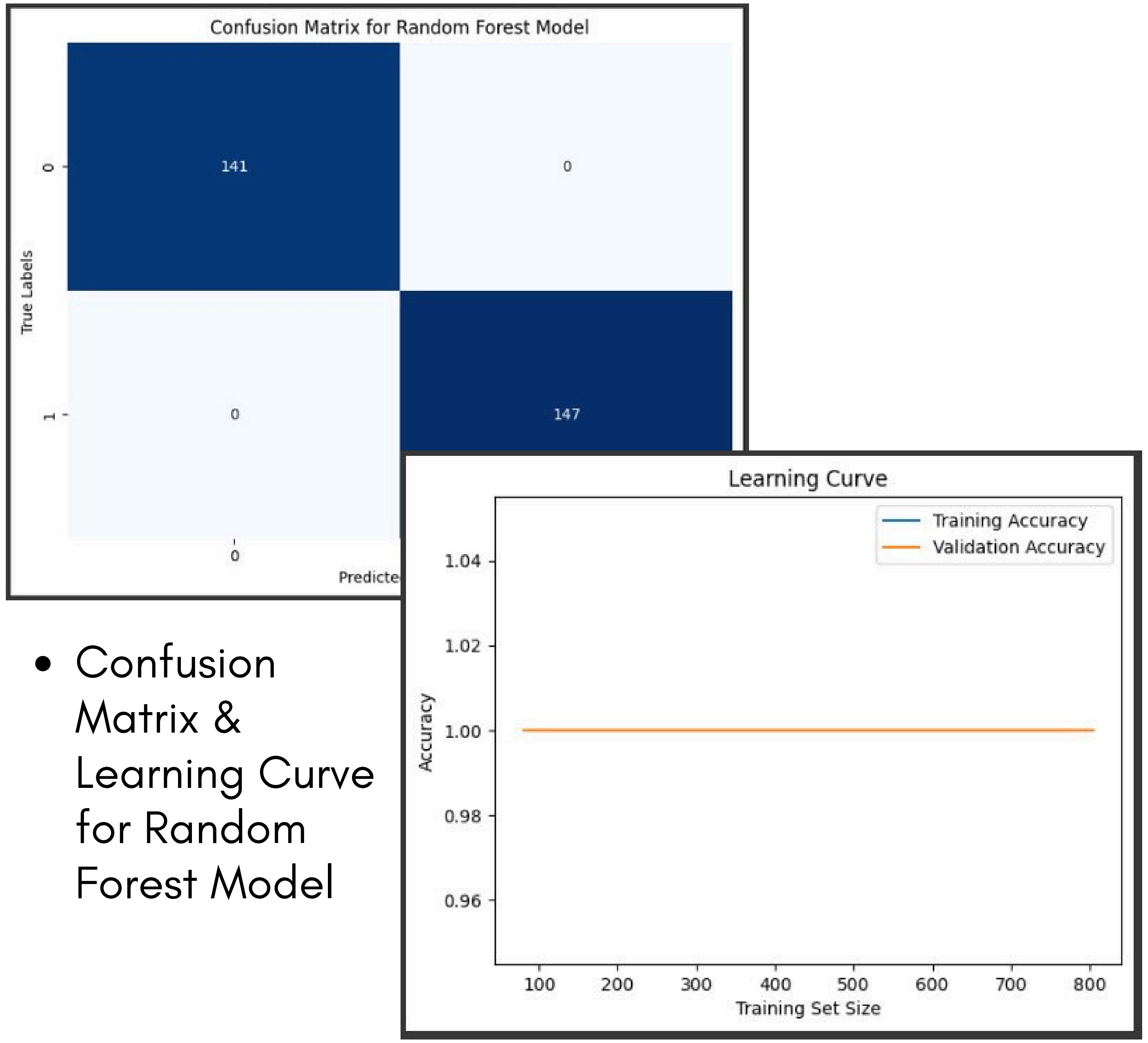
```
Accuracy (Support Vector Classifier): 0.6111111111111112
              precision    recall  f1-score   support

           0       0.89      0.23      0.37       141
           1       0.57      0.97      0.72       147

    accuracy                           0.61       288
   macro avg       0.73      0.60      0.54       288
weighted avg       0.73      0.61      0.55       288
```

Accuracy : 61.11%

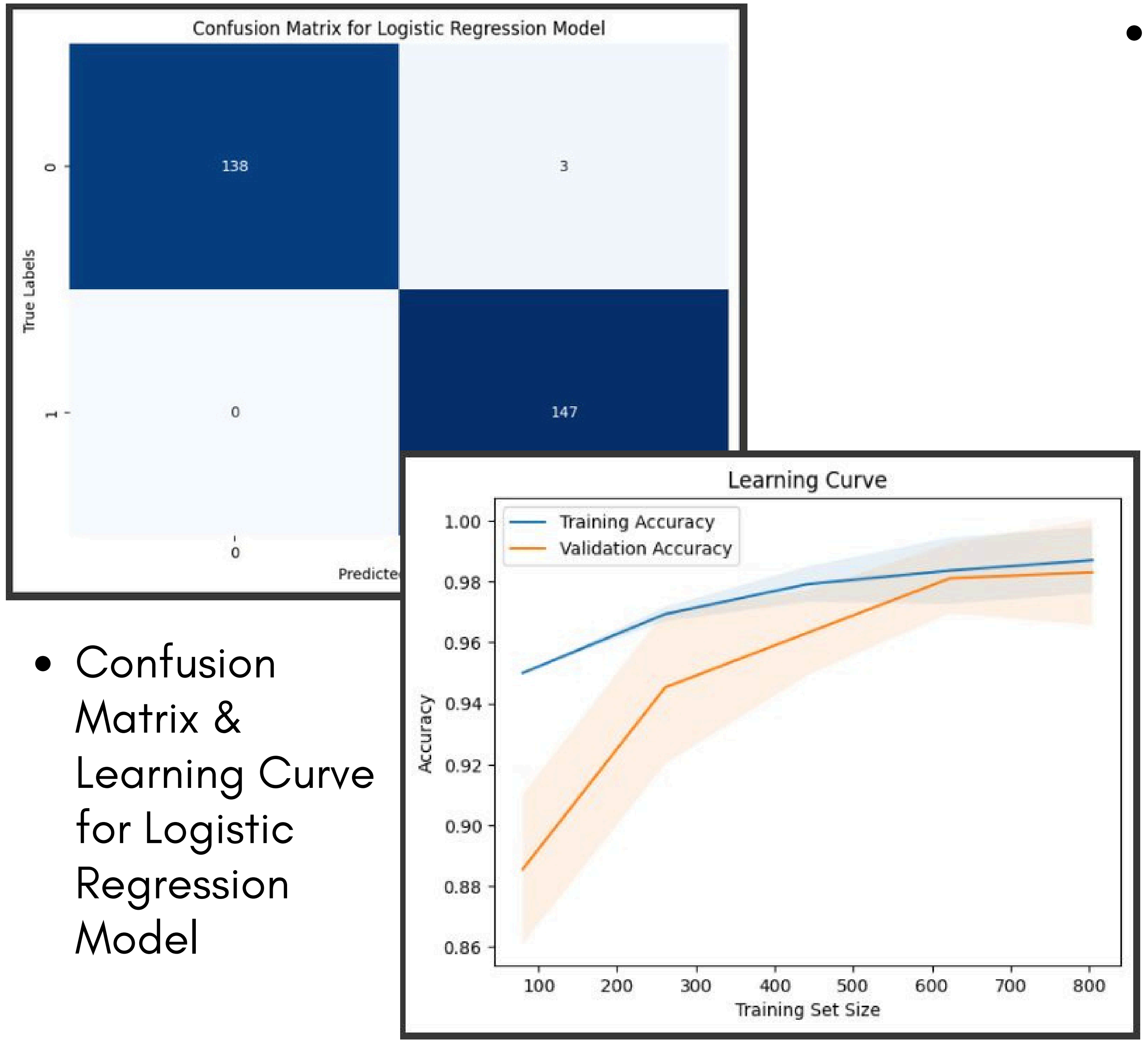
- When comparing the accuracy of 4 Machine Learning Algorithms, with a Accuracy of 100% Random Forest Classifier hold the lead. BUT,

3. Model Development

Algorithm Selection



- Confusion Matrix & Learning Curve for Random Forest Model



- Confusion Matrix & Learning Curve for Logistic Regression Model

- Eventhough Random Forest Model has the highest accuracy, Logistic Regression is less prone to overfitting compared to complex models like Random Forest or Gradient Boosting and also It is computationally efficient, a simple and interpretable model, Logistic regression has choosen to develop the Rice Variety Prediction Model.

3. Model Development

- Logistic Regression Model is trained using defined features and classify the Suitability of rice varieties.

```python
from sklearn.preprocessing import LabelEncoder

le_temperature = LabelEncoder()
le_rainfall = LabelEncoder()

# Encoding 'AnnualTemperature' and 'Annual Rainfall' into numerical values
combined_data['Annual Temperature'] = le_temperature.fit_transform(combined_data['Annual Temperature'])
combined_data['Annual Rainfall'] = le_rainfall.fit_transform(combined_data['Annual Rainfall'])

# Define features (X) and target (y)
features = [
    'Average Yield (t/ha)', 'Maturity (days)', 'Annual Temperature', 'Annual Rainfall'
]
X = combined_data[features]
y = combined_data['Suitability']
```

```python
from sklearn.model_selection import train_test_split

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

from sklearn.linear_model import LogisticRegression

# Train the Logistic Regression model
logreg_model = LogisticRegression(max_iter=1000, random_state=42)
logreg_model.fit(X_train, y_train)

from sklearn.metrics import classification_report, accuracy_score

# Predict and evaluate the model
y_pred = logreg_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print("Accuracy (Logistic Regression):", accuracy)
print(report)
```

Logistic Regression model Accuracy:

```
Accuracy (Logistic Regression): 0.9895833333333334
              precision    recall  f1-score   support

           0       1.00      0.98      0.99       141
           1       0.98      1.00      0.99       147

    accuracy                           0.99       288
   macro avg       0.99      0.99      0.99       288
weighted avg       0.99      0.99      0.99       288
```

4. Model Validation

Mean Cross vaidation score for Logistic Regression model :

```
from sklearn.model_selection import cross_val_score
import numpy as np

# Perform 5-fold cross-validation
cv_scores = cross_val_score(logreg_model, X, y, cv=10, scorir
mean_cv_score = np.mean(cv_scores)
print("Mean Cross-Validation Score (Logistic Regression):", m
```

```
Mean Cross-Validation Score (Logistic Regression): 0.98125
```

Training accuracy & Validation accuracy for Logistic Regression model :

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)


# Training accuracy
y_train_pred = logreg_model.predict(X_train)
train_accuracy = accuracy_score(y_train, y_train_pred)

# Validation accuracy
y_val_pred = logreg_model.predict(X_val)
val_accuracy = accuracy_score(y_val, y_val_pred)

print(f"Training Accuracy: {train_accuracy * 100}")
print(f"Validation Accuracy: {val_accuracy * 100}")
```

```
Training Accuracy: 98.30845771144278
Validation Accuracy: 98.6046511627907
```

5. Model Implementation

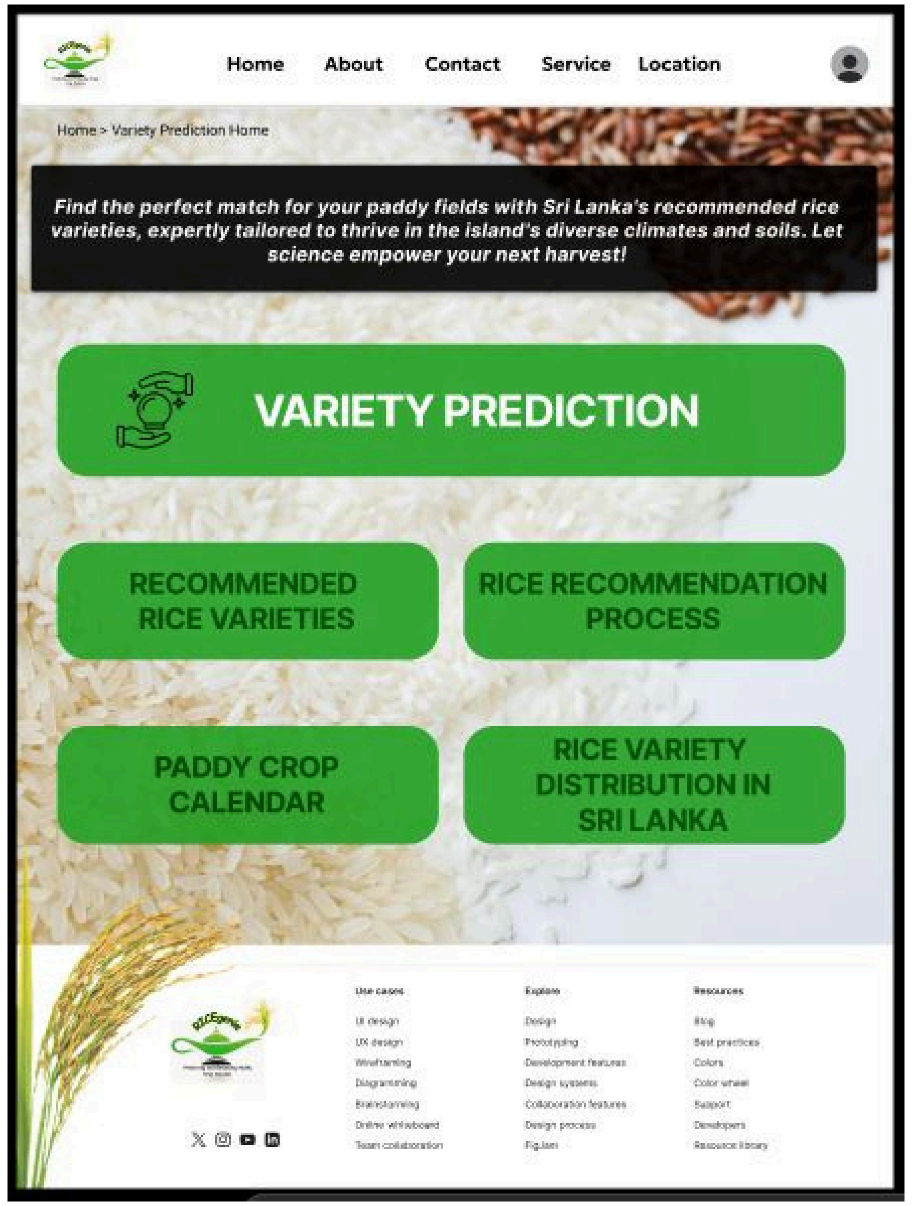
- Logistic Regression model implementation:

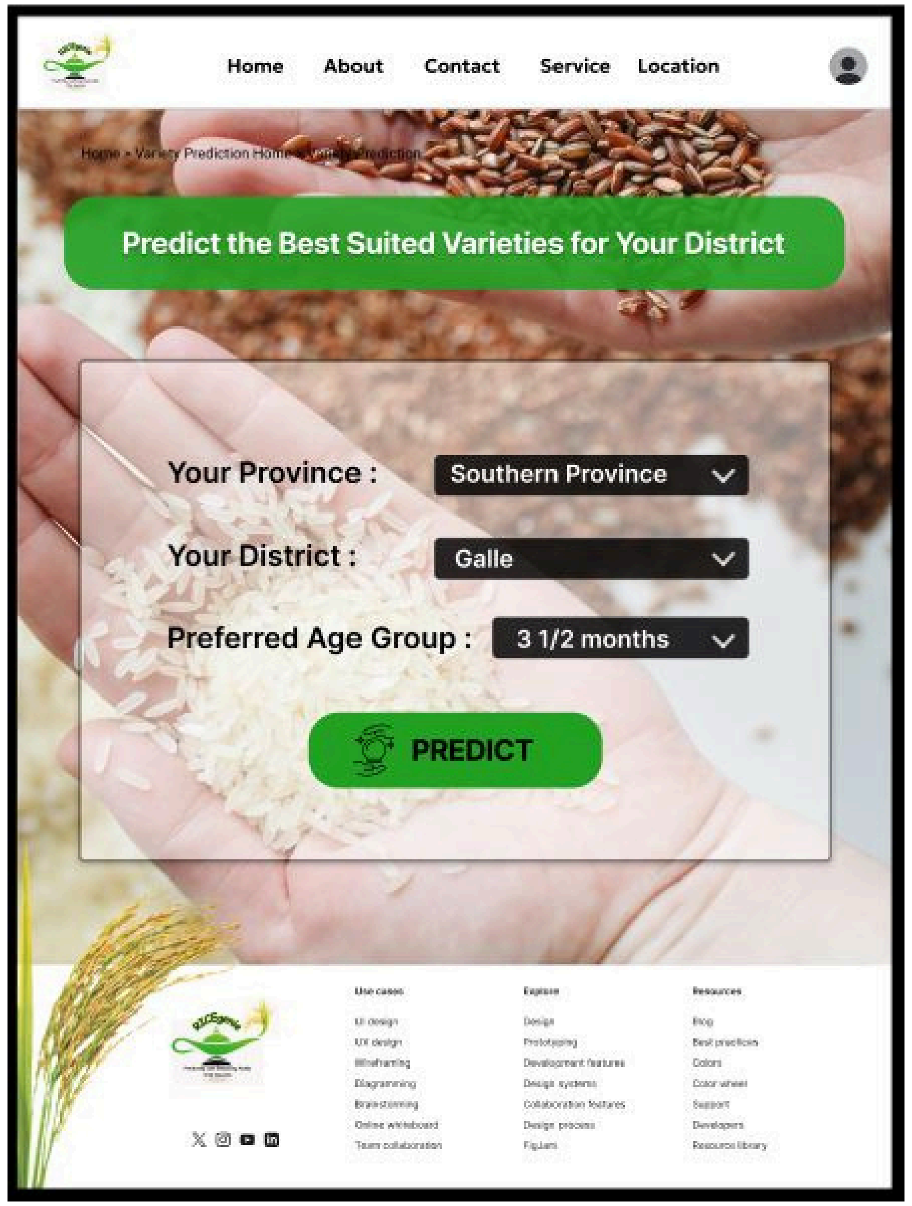
```
def predict_suitable_varieties_for_district(province_name, district_name, district_data, varieties_data, model):
    # Filter the district data by the specified province and district
    district_conditions = district_data[
        (district_data['Province'] == province_name) &
        (district_data['District'] == district_name)
    ]

    if district_conditions.empty:
        raise ValueError(f"No data found for district '{district_name}' in province '{province_name}'.")

    # Extract the environmental conditions for prediction
    district_conditions = district_conditions.iloc[0]
    annual_temperature = le_temperature.transform([district_conditions['Annual Temperature']])[0]
    annual_rainfall = le_rainfall.transform([district_conditions['Annual Rainfall']])[0]

    # Prepare the prediction data
    prediction_data = varieties_data.copy()
    prediction_data['Annual Temperature'] = annual_temperature
    prediction_data['Annual Rainfall'] = annual_rainfall

    # Ensure 'Maturity (days)' is numeric
    prediction_data['Maturity (days)'] = prediction_data['Maturity (days)'].apply(convert_maturity_to_numeric)
    prediction_data.dropna(subset=['Maturity (days)'], inplace=True)

    # Add 'Age Group' if not already present
    if 'Age Group' not in prediction_data.columns:
        prediction_data['Age Group'] = prediction_data['Maturity (days)'].apply(map_to_age_group)

    # Prepare features for prediction
    X_pred = prediction_data[['Average Yield (t/ha)', 'Maturity (days)', 'Annual Temperature', 'Annual Rainfall']]

    # Predict suitability for each variety
    prediction_data['Suitability'] = model.predict(X_pred)
```

✓ 35s completed at 11:14 PM

5. Web app Implementation

- UI Designed for the Paddy Variety Prediction interface of the RICEgenie web app.





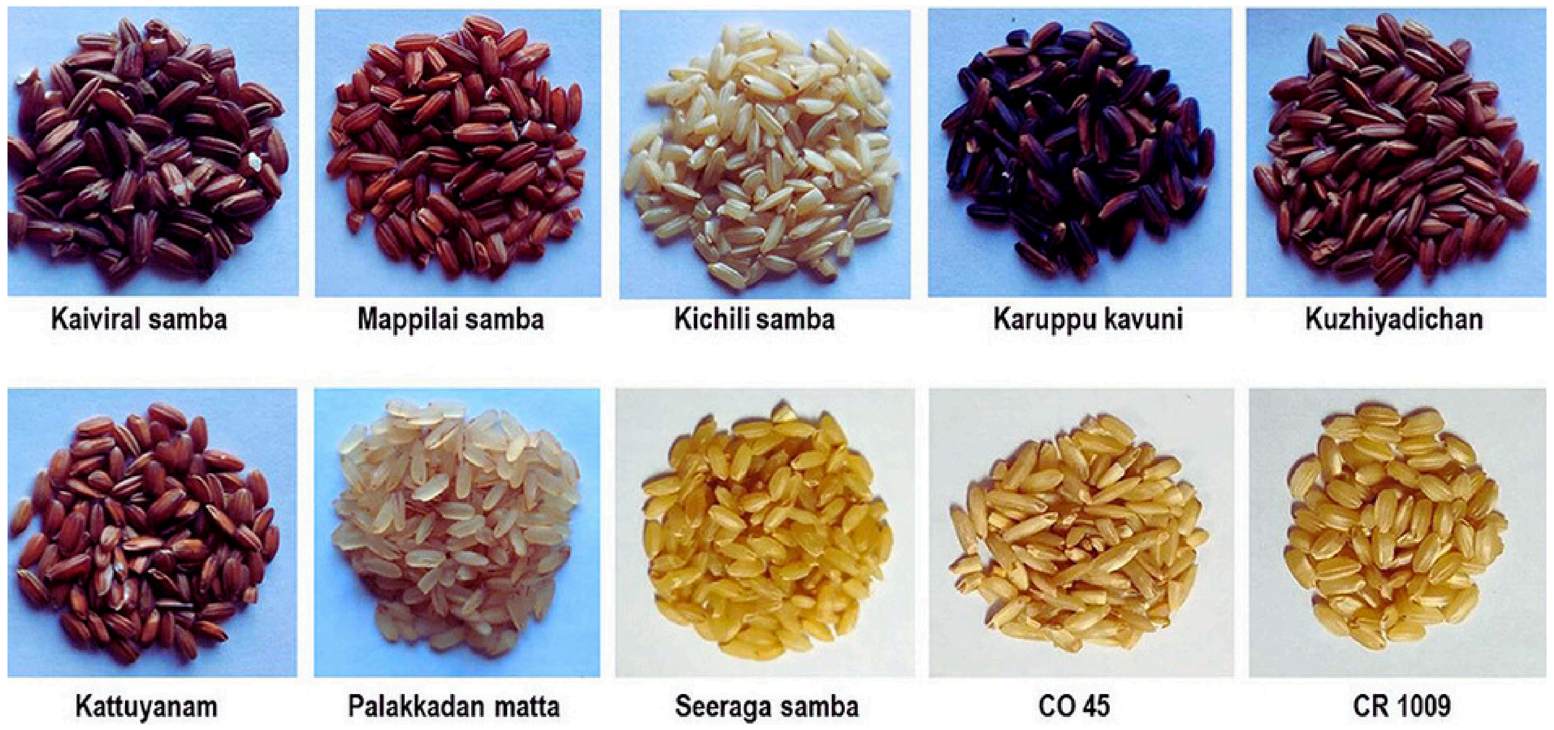
Work Breakdown Structure

- **Paddy Variety Prediction Model**
  - Requirement Analysis
    - Define Project Objectives and Scope
    - Identify Data Sources and Requirements
    - Specify Model Requirements
    - Resource Allocation and Timeline
  - Data Collection and Preprocessing
    - Data Collection
    - Data Cleaning and Preprocessing
    - Data Integration
  - Model Development
    - Algorithm Selection
    - Model Training
    - Model Validation and Testing
    - Model Optimization
  - System Integration
    - Develop User Interface
    - Integrate Prediction Model
    - Database Setup
  - Testing and Validation
    - System Testing
    - User Testing and Feedback
    - Model Validation with Real-world Data
  - Deployment and Maintenance
    - Deployment
    - Monitoring and Performance Evaluation
    - Maintenance and Updates
    - Documentation and Training

Requirements

Non-functional requirements

- Accessibility: Ensure the system is accessible to all users, including farmers with limited technical expertise.
- Reliability: The model should provide consistent and accurate predictions.
- Performance: The system should deliver predictions promptly.
- Usability: The interface should be intuitive and easy to use.



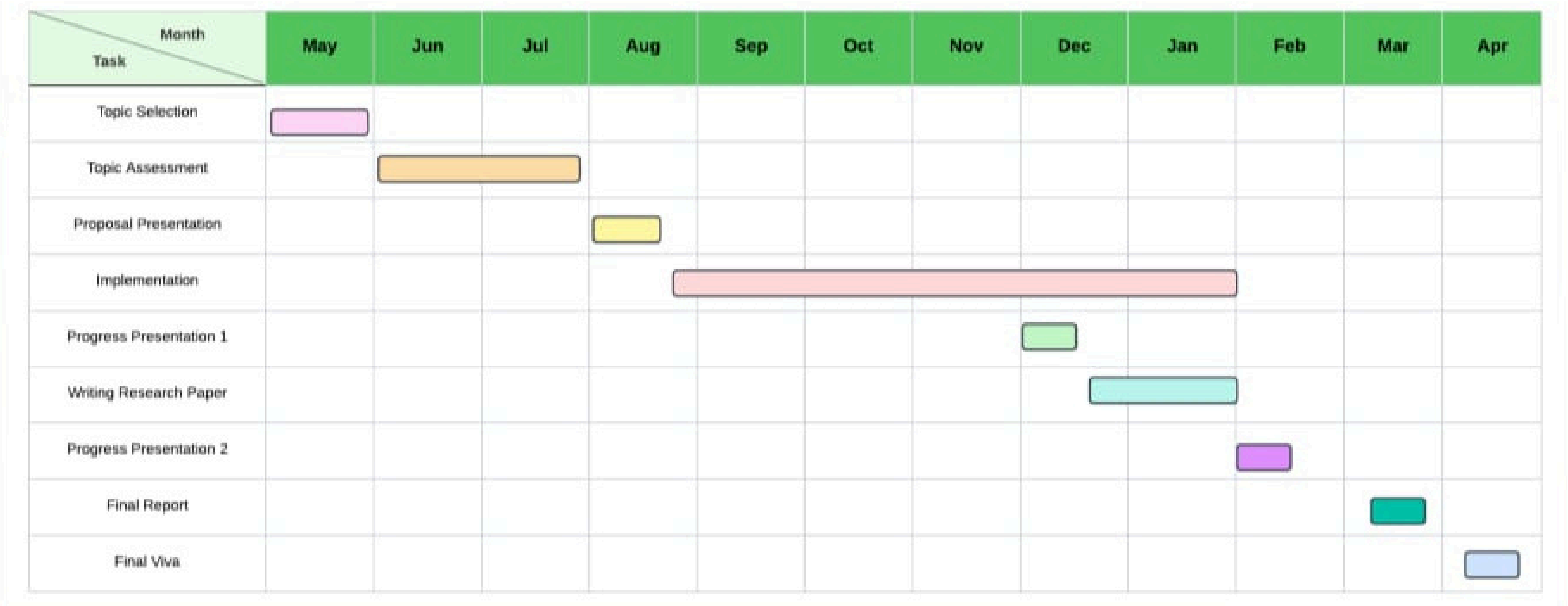
Functional requirements

- Ability to analyze environmental data.
- Ability to predict the best paddy variety for specific conditions.

Technologies

- Python (Back end)
- ML (Regression)
- ReactJS (Front end)
- Tailwind (for styles)
- Fast API
- Google Colab
- MySql
- Git Hub (Version control system)
- Trello(Project Management)

Gantt chart



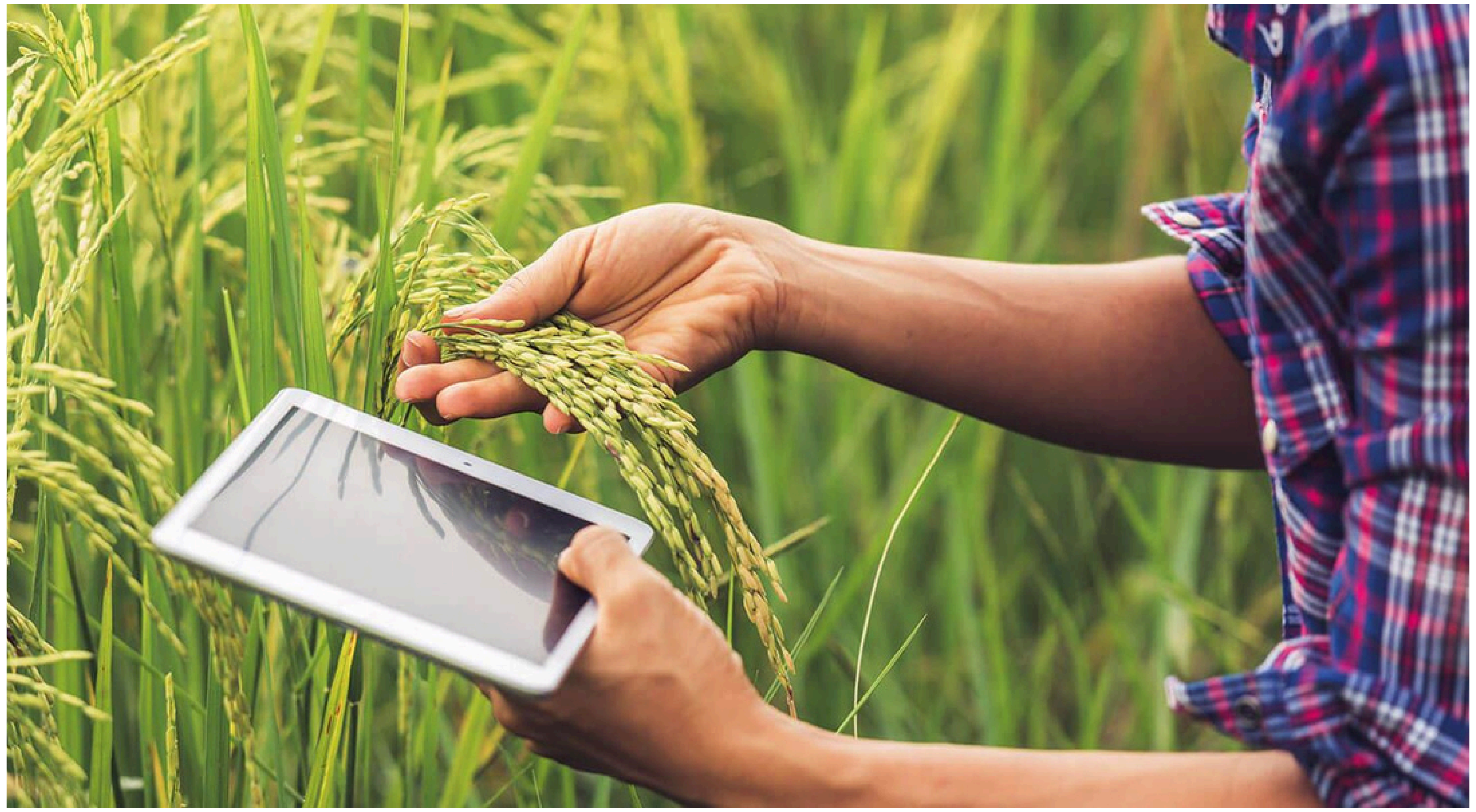
Month
Task
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
Topic Selection
Topic Assessment
Proposal Presentation
Implementation
Progress Presentation 1
Writing Research Paper
Progress Presentation 2
Final Report
Final Viva

REFERENCES

1. H. A. R. T. I. "Rice Varieties and Their Characteristics," Hector Kobbekaduwa Agrarian Research and Training Institute, Colombo, Sri Lanka, Research Report No. 186, Accessed: Aug. 08, 2024. [Online]. Available: H. A. R. T. I. "Rice Varieties and Their Characteristics," Hector Kobbekaduwa Agrarian Research and Training Institute, Colombo, Sri Lanka, Research Report No. 186, Accessed: Aug. 08, 2024. [Online]. Available:https://www.harti.gov.lk/images/download/reasearch_report/new1/186.pdf
2. "Climate Change Ready Rice," International Rice Research Institute (IRRI), Accessed: Aug. 08, 2024. [Online]. Available: http://www.knowledgebank.irri.org/step-by-step-production/pre-planting/rice-varieties/item/climate-change-ready-rice
3. A. Gunawardena, H. Munasinghe, and W. Wickramasinghe, "Assessment of the suitability of temperature and relative humidity for rice cultivation in rainfed lowland paddy fields in Kurunegala district," Accessed: Aug. 08, 2024. [Online]. Available: https://www.researchgate.net/publication/319227475_Assessment_of_the_suitability_of_temperature_and_relative_humidity_for_rice_cultivation_in_rainfed_lowland_paddy_fields_in_Kurunegala_district

IT21227318 | Jayasekara S.S.D

Weed Detection & Mitigation System



Information Technology

INTRODUCTION

- **This Research component is about,**
  - Identify and mitigate weed varieties in paddy fields
- **An Accurate weed detection and mitigation system can enhance productivity and reduce losses.**
- **Machine learning and image processing offer innovative solutions for real-time weed management.**

RESEARCH GAP

- Traditional weed detection relies on manual identification, which is time-consuming and less accurate.

- Bad mitigation strategies can cause huge negative impacts not only on paddy yield but also on properties like soil and water !!

- Lack of integrated solutions that provide both detection and mitigation recommendations.

RESEARCH PROBLEM

Building a system that offers accurate mitigation strategies based on detected weed species is needed.

- Build an automated system to detect and classify weed species accurately and recommend mitigation strategies with a user-friendly user interface.

SPECIFIC AND SUB OBJECTIVES

- **Specific Objectives :**
  - Enhance paddy crop yield by developing a robust weed detection and mitigation system using machine learning.

- **Sub Objectives :**
  - Develop an Image Classification Model for Weed Detection: Train a model to identify and classify weeds.
  - Implement a Weed Mitigation Recommendation System: Suggest effective control strategies for detected weeds.

METHODOLOGY

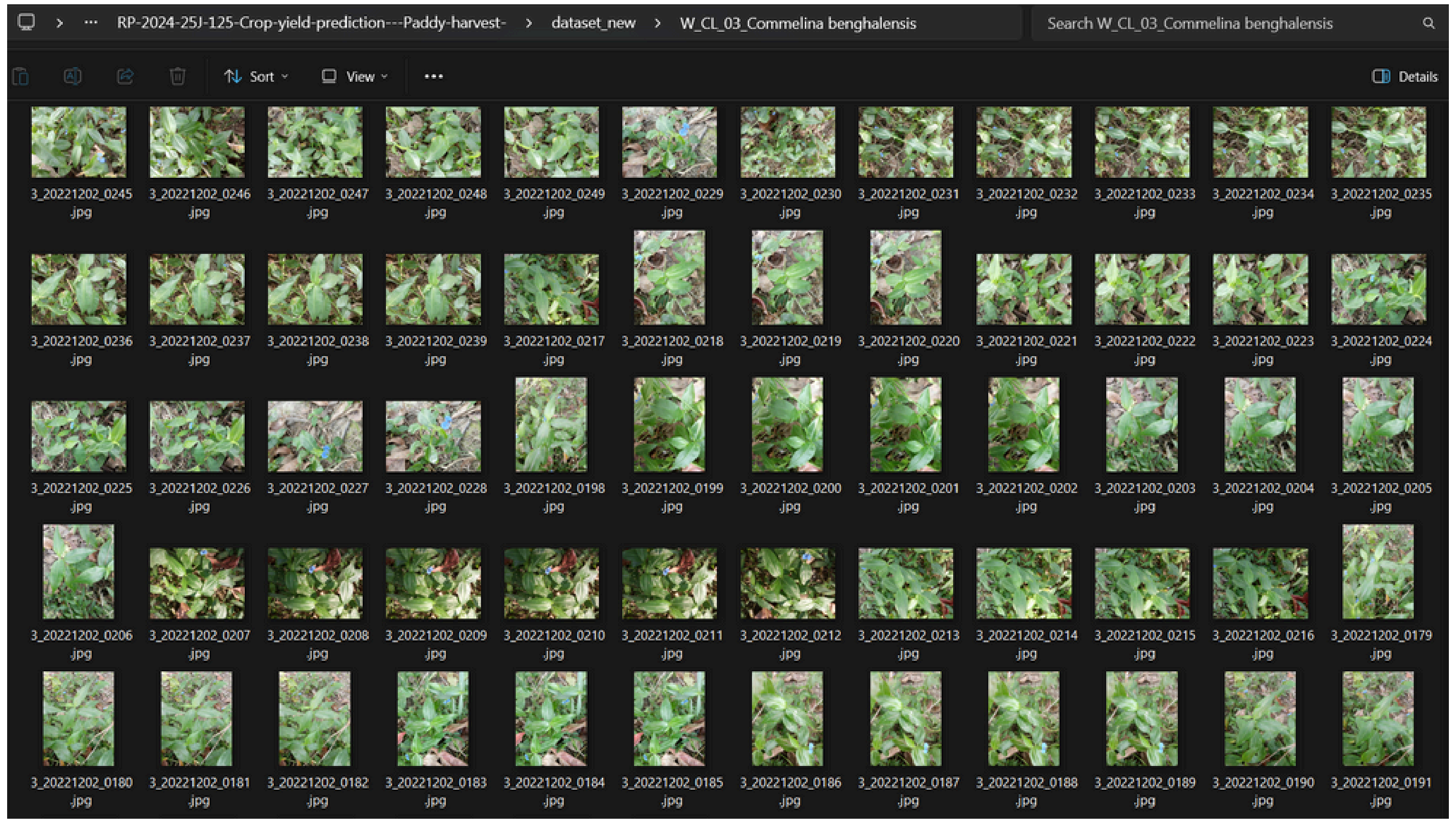
- **Dataset Collection and Preprocessing**
  - Collect Images
  - Preprocess Images

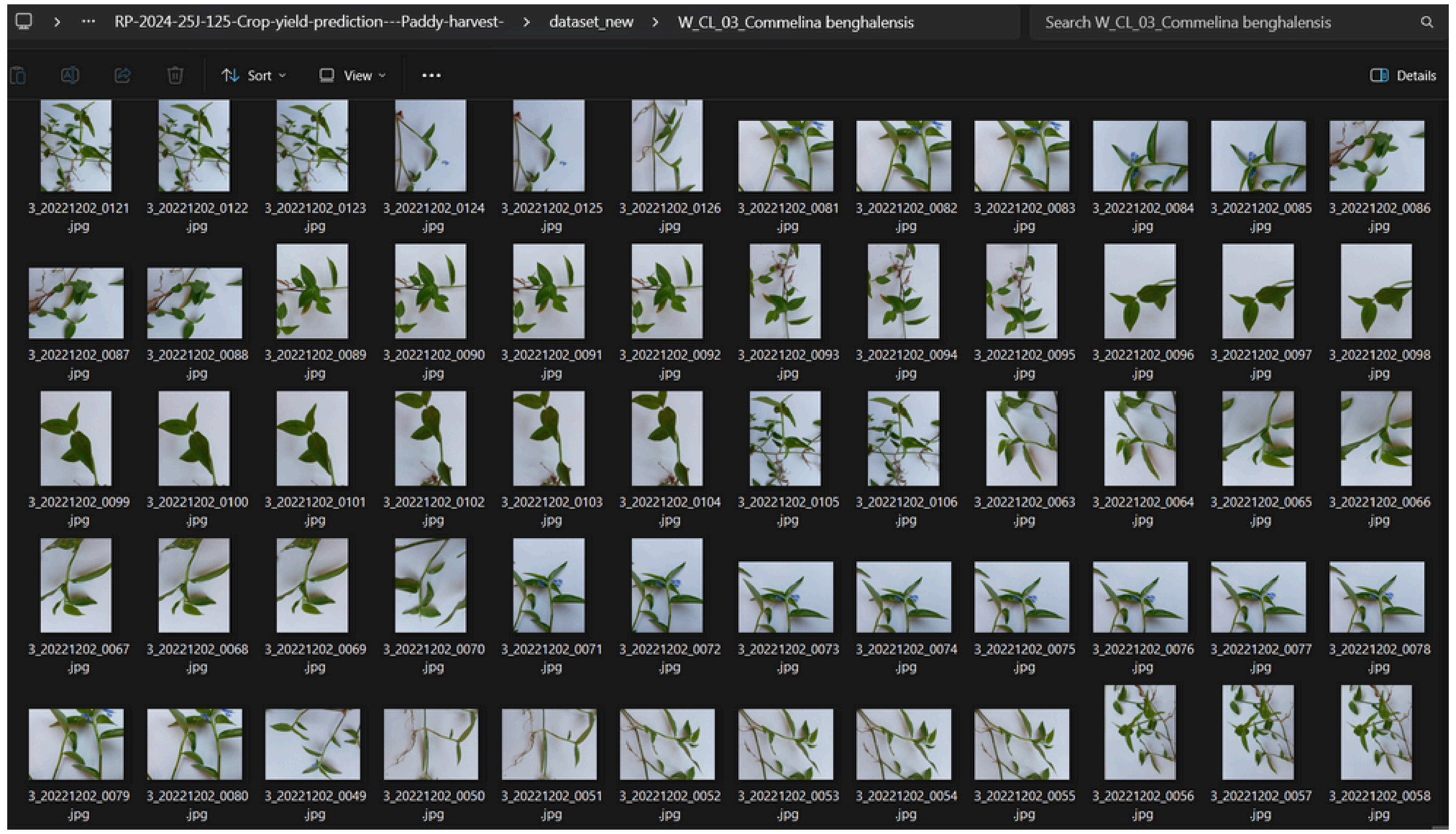
- **Model Training and Validation**
  - Split Dataset: Divide into training, validation, and test sets.
  - Choose Model: Select and train a CNN for image classification.
  - Evaluate Model: Assess performance

- **Development and Integration**
  - Document Details(DB) : Include methods, techniques, and dosages.
  - Develop Algorithm: Link detected weeds to mitigation strategies.
  - User Interface Development: Design an interface to upload images and receive mitigation methods.

- **Field Testing and Feedback Collection**

DATA COLLECTION





- 3000 images for six most common paddy weed classes (without augmented images)
- Natural and controlled enviroments

W_CL_11_Synedrella nodiflora	12/4/2024 9:25 PM	File folder
W_CL_10_Pteris vittata	12/4/2024 9:25 PM	File folder
W_CL_09_Paspalum scrobiculatum	12/4/2024 9:25 PM	File folder
W_CL_07_Marsilea minuta	12/4/2024 9:25 PM	File folder
W_CL_06_Ipomoea aquatic	12/4/2024 9:25 PM	File folder
W_CL_03_Commelina benghalensis	12/4/2024 9:25 PM	File folder

MODEL SELECTION

```
[10]
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")

20/20 ━━━━━━━━━━━━━━━━━━━━ 4s 189ms/step - accuracy: 0.2576 - loss: 24.8933
Test Accuracy: 25.57%
```

```
[11]
plt.grid(True)
plt.show()
```

- Trained few models with different accuracy levels
- Fixed the imbalance in the classes of the dataset
- Increased the image count by image augmentation

MODEL SELECTION

```
train_data_dir = augmented_dir
test_data_dir = test_dir

train_datagen = ImageDataGenerator(rescale=1.0/255)
test_datagen = ImageDataGenerator(rescale=1.0/255)

train_generator = train_datagen.flow_from_directory(
    train_data_dir, target_size=(150, 150), batch_size=32, class_mode='categorical'
)
test_generator = test_datagen.flow_from_directory(
    test_data_dir, target_size=(150, 150), batch_size=32, class_mode='categorical'
)
```

[5]

```
Found 9363 images belonging to 6 classes.
Found 464 images belonging to 6 classes.
```

```
test_loss, test_accuracy = model.evaluate(test_generator)
print(f"Test Accuracy: {test_accuracy * 100:.2f}%")
```

[9]

```
15/15 ━━━━━━━━━━━━━━━━━━━━ 3s 177ms/step - accuracy: 0.4620 - loss: 4.1965
Test Accuracy: 46.98%
```

```
base_model = VGG16(weights='imagenet', include_top=False, input_shape=(150, 150, 3))

model = Sequential([
    base_model,
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(train_generator.num_classes, activation='softmax')
])

base_model.trainable = False
model.compile(optimizer=Adam(learning_rate=0.0001), loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit(train_generator, epochs=20, validation_data=test_generator)
```

[7]

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
Epoch 1/20
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your `PyDataset` class should call `super().
  self._warn_if_super_not_called()
293/293 ━━━━━━━━━━━━━━━━━━━━ 61s 168ms/step - accuracy: 0.8712 - loss: 0.4363 - val_accuracy: 0.4483 - val_loss: 2.3112
Epoch 2/20
293/293 ━━━━━━━━━━━━━━━━━━━━ 37s 125ms/step - accuracy: 1.0000 - loss: 0.0082 - val_accuracy: 0.4504 - val_loss: 2.6057
```

MODEL SELECTION

```
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(len(class_names), activation='softmax')  # Output layer for 6 classes
])
```

- Selected a Custom CNN model built using the Sequential API
- Accuracy level - 86%

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 128)	4,735,104
dense_1 (Dense)	(None, 6)	774

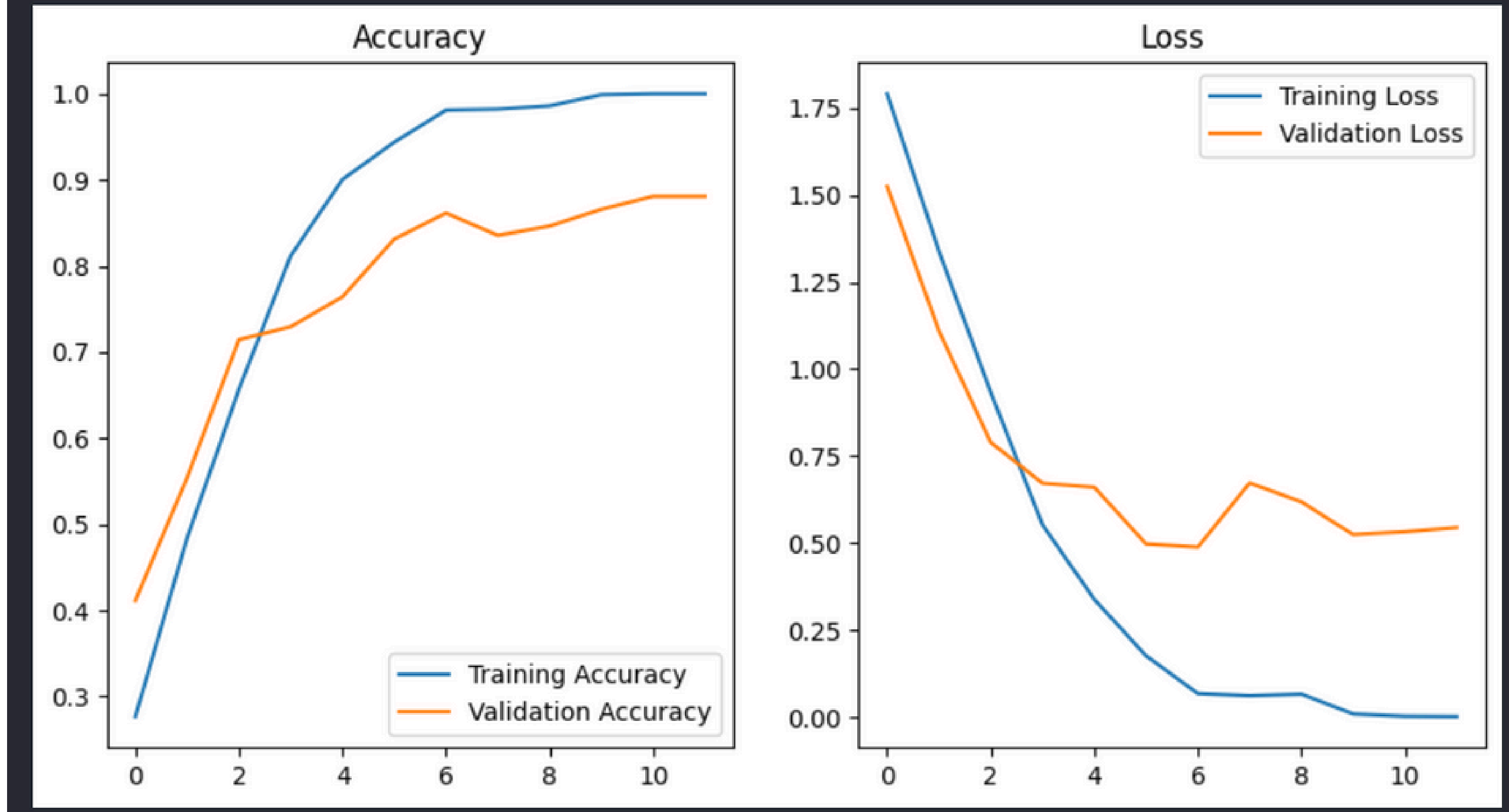
Total params: 4,829,126 (18.42 MB)

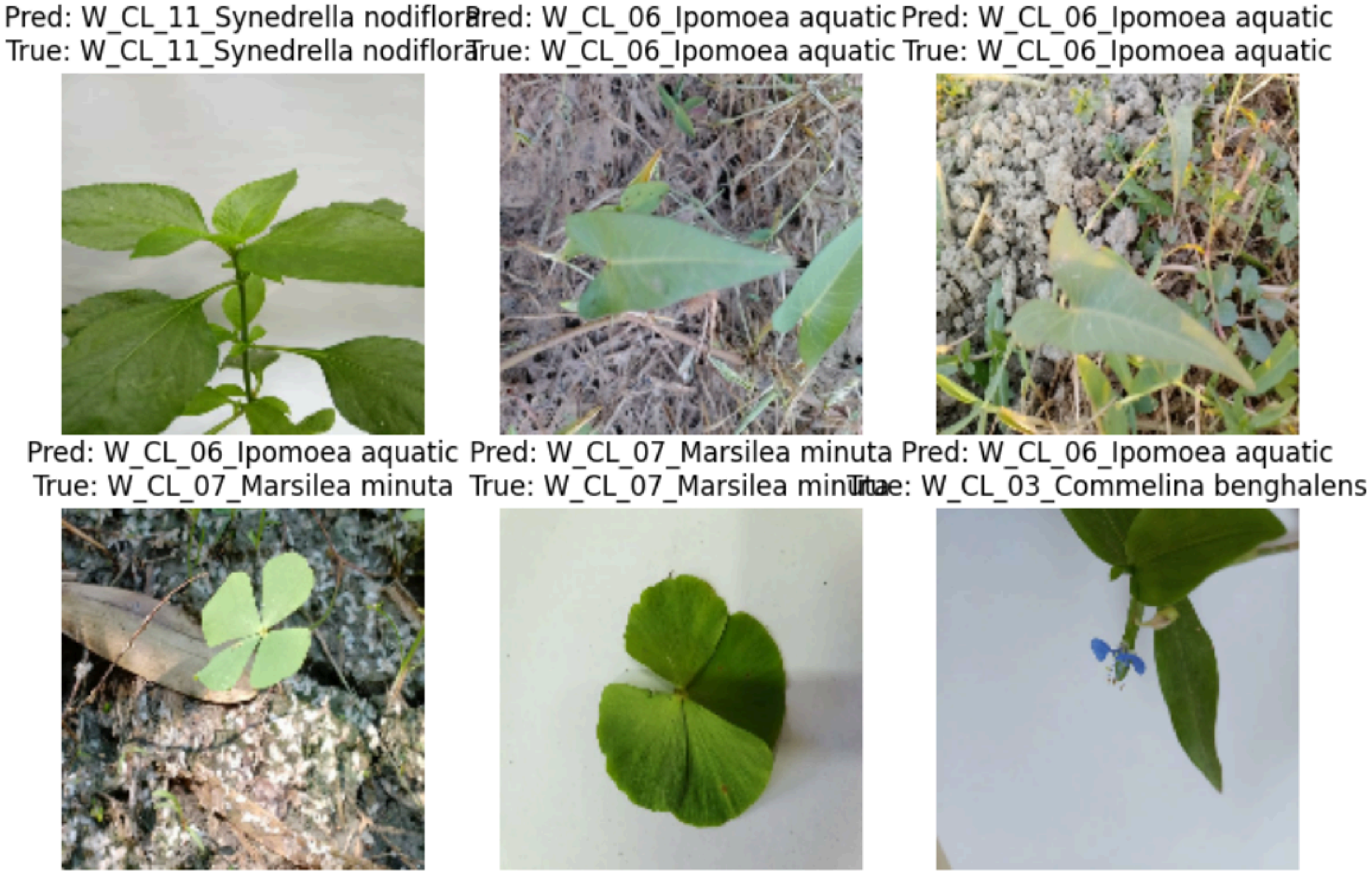
Trainable params: 4,829,126 (18.42 MB)

Non-trainable params: 0 (0.00 B)

MODEL SELECTION

```python
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Loss')
plt.show()
```



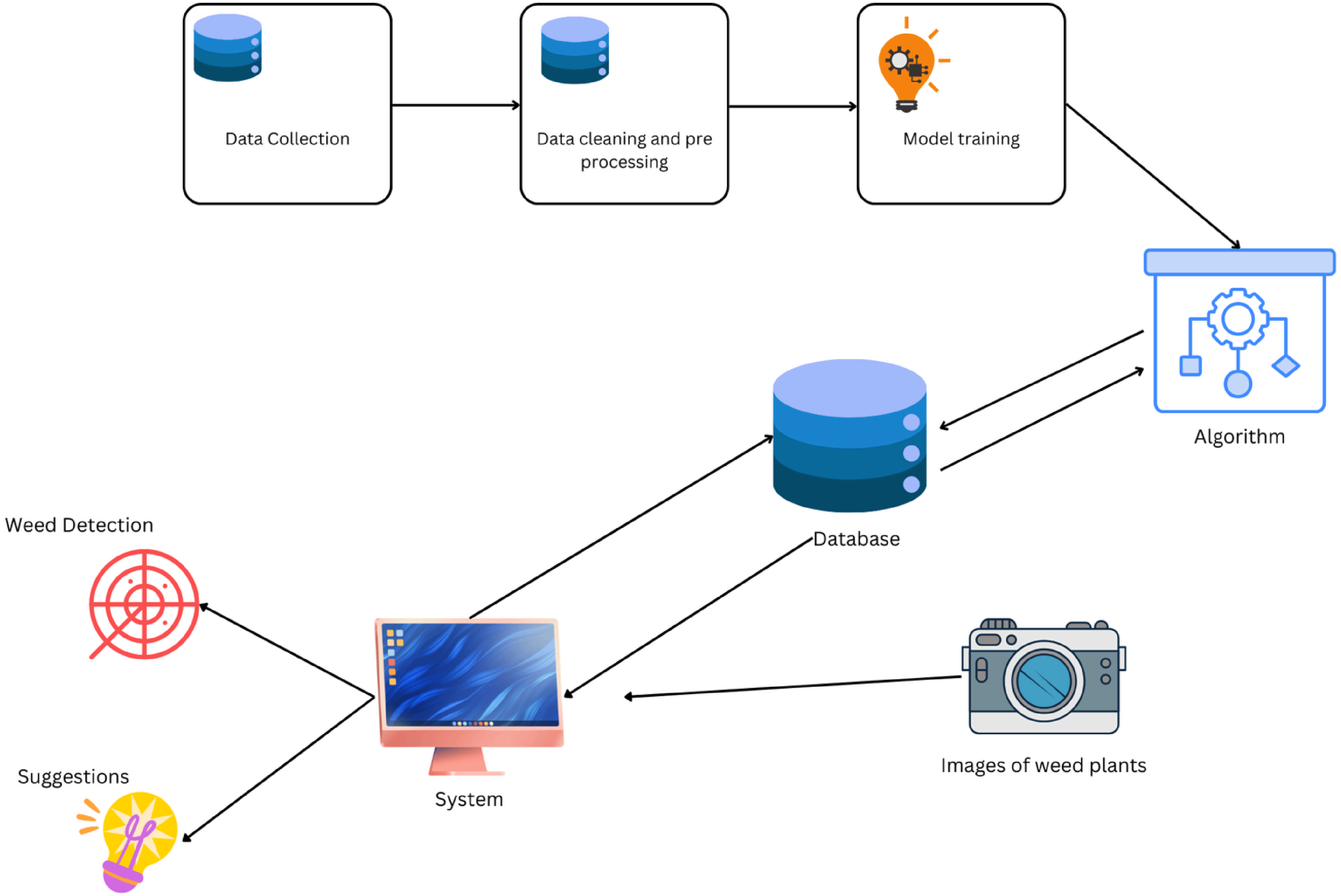



```python
test_loss, test_acc = model.evaluate(val_dataset)
print(f"Test Accuracy: {test_acc:.2f}")
```

[18]

```
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.8345 - loss: 0.5735
Test Accuracy: 0.86
```

SYSTEM DIAGRAM



TECHNOLOGIES

- Python (Back end)
- Tensor Flow (Framework)
- ML (Classifications)
- ReactJS (Front end)
- HTML (Front end)
- CSS (Front end)
- Bootstrap (Front end)
- Google Colab
- Git Hub (Version control system)
- Trello(Project Management)

Requirements

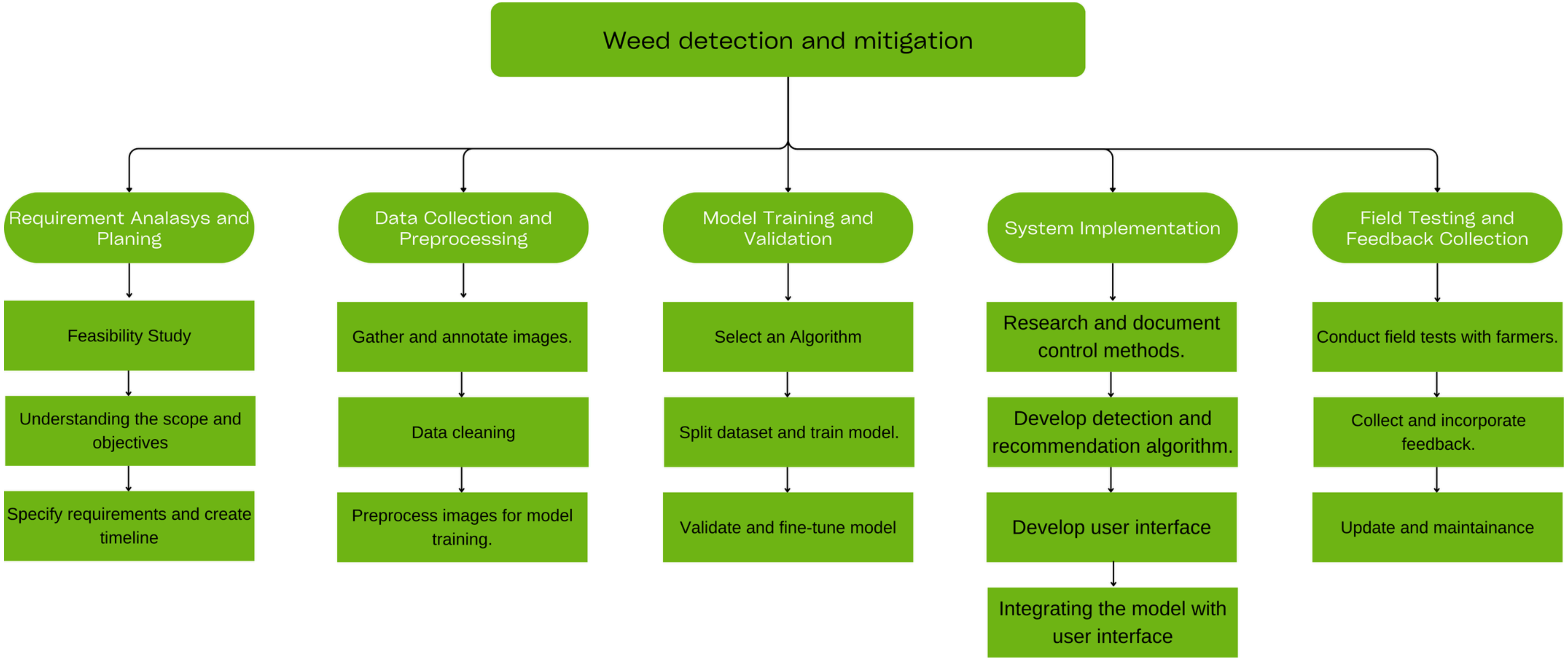
Non-functional requirements

- Availability
- Reliability
- Performance
- Usability

Functional requirements

- Ability to identify weed varieties
- Ability to suggest treatments to mitigate weed plants

Work Breakdown Structure



Gantt chart


Month
Task
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Apr
Topic Selection
Topic Assessment
Proposal Presentation
Implementation
Progress Presentation 1
Writing Research Paper
Progress Presentation 2
Final Report
Final Viva

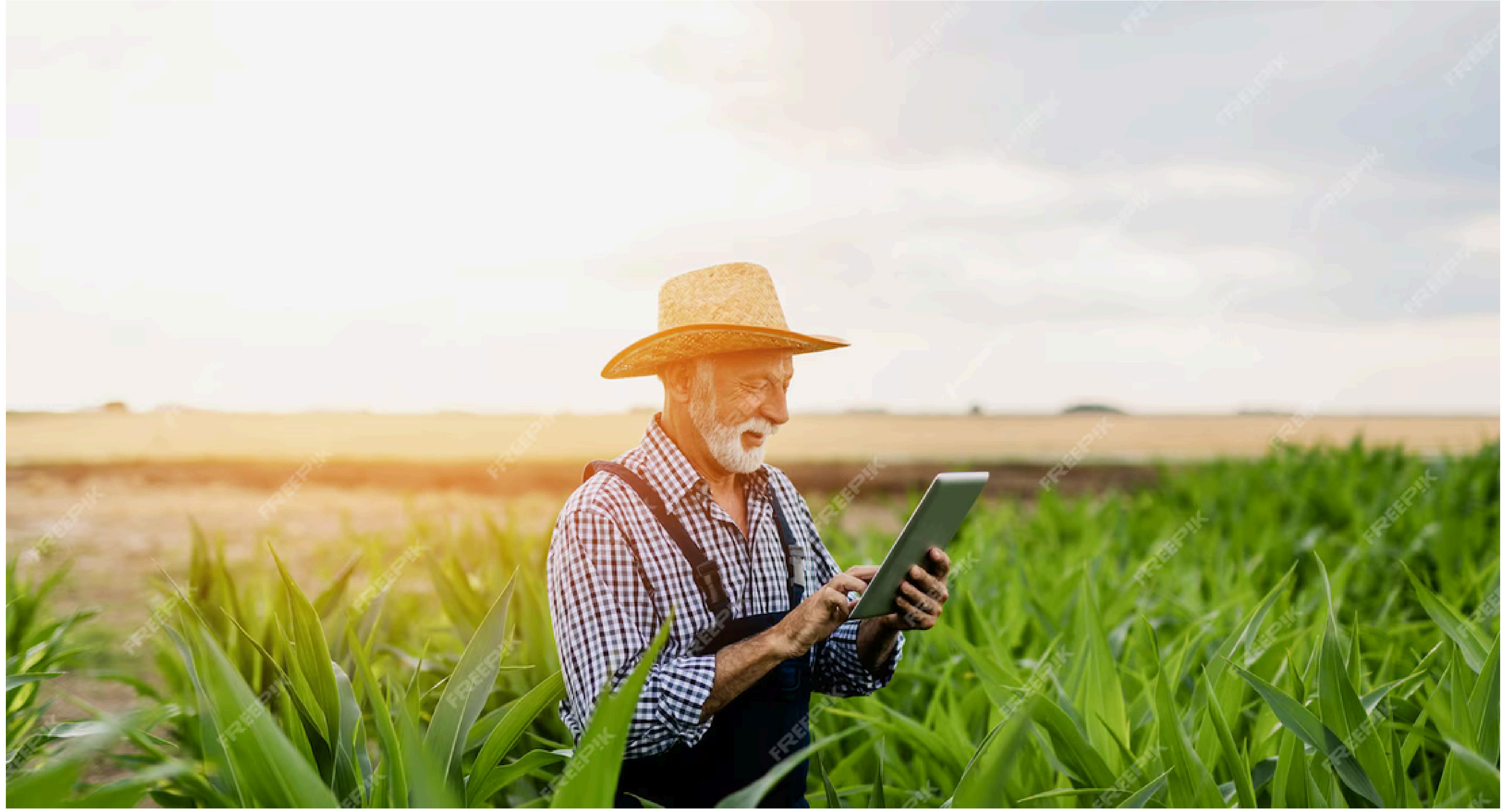
References

[1]Elakya, R. & Vignesh, U. & Valarmathi, P. & Chithra, N. & Sigappi, S.. (2022). A Novel Approach for Identification of Weeds in Paddy By using Deep Learning Techniques. International Journal of Electrical and Electronics Research. https://www.researchgate.net/publication/366698172_A_Novel_Approach_for_Identification_of_Weeds_in_Paddy_By_using_Deep_Learning_Techniques

[2] Kamath, Radhika & Balachandra, Mamatha & Prabhu, Srikanth. (2020). Paddy Crop and Weed Discrimination: A Multiple Classifier System Approach. International Journal of Agronomy. 2020. 1-14. 10.1155/2020/6474536.
Available at: https://doi.org/10.1155/2022/3287561 (Accessed: August 3, 2024).

[3] Radhika Kamath, Mamatha Balachandra, Amodini Vardhan & Ujjwal Maheshwari | (2022) Classification of paddy crop and weeds using semantic segmentation,
https://doi.org/10.1080/23311916.2021.2018791

Supportive information



Commercialization

- **Target Market Sectors:**
  - Farmers, agricultural extension officers, and other stakeholders involved in the agriculture industry.
  - Government agencies responsible for agriculture policies.

- **Marketing Plan :**
  - Developing and maintaining a website to showcase features, benefits, and usage guidelines
  - Conducting live demonstrations and training workshops for farmers and agricultural extension officers.

Budget

	Cost (in LKR)
Domain Hosting space (If needed per month)	7500
Travelling cost	2000 * 4* 11
Total	95,500.00

- The budget mentioned is a rough estimate and may vary according to the purpose when developing the system.


THANKS FOR
Your Attention