Autonomous Reforestation Robot: ML Model for Crop Recommendation

PDE4433 – Machine Learning for Robotics

Coursework 01

N. G. Jayashanka Anushan
MISIS – M01037028
MSc in Robotics
Middlesex University Dubai

Table of Contents

List of figures

List of tables

1). Introduction

This project focuses on developing an Autonomous Reforestation Robot powered by machine learning. The system is designed to comprehensively scan designated areas and collect critical environmental parameters—including soil texture, moisture, humidity, and nutrient levels (nitrogen, potassium, and phosphorus)—through integrated sensors. These data are analyzed to determine the most suitable crop type for reforestation or agricultural purposes.

The machine learning pipeline employs supervised learning to process sensor inputs, correlate them with agronomic requirements, and generate optimized crop recommendations. By systematically evaluating environmental conditions and aggregating multisource data, the model ensures precise, data-driven decisions. This approach aims to enhance reforestation accuracy and promote sustainable land management through autonomous, intelligent systems.

An additional robotic application, such as an automated seed-planting mechanism or drone-assisted terrain mapping, can significantly boost the efficiency of this project by enabling precise seed placement and expanded environmental coverage, further supporting large-scale agricultural and forestation initiatives.

2). Used Dataset

For the analysis propose the data collected from Kaggl.com. Below is the data sources that collected data for analysis.

1. Image datasets;
   - https://www.kaggle.com/datasets/prasanshasatpathy/soil-types
   - https://www.kaggle.com/datasets/jhislainematchouath/soil-types-dataset
2. Tabular datasets;
   - https://www.kaggle.com/datasets/varshitanalluri/crop-recommendation-dataset

As per the concept of the proposed development, two models are involved in the system, with each model interacting with the other. The first model is designed to process and predict based on image data, while the second model operates solely on tabular data. To align with the requirements of the second dataset, the image dataset had to be manually preprocessed before it could be utilized for training the image prediction model. The tabular dataset includes five distinct soil types: *'Black'*, *'Clayey'*, *'Loamy'*, *'Red'*, and *'Sandy'*. Accordingly, using the two datasets obtained from Kaggle, a customized dataset was created to train the first model. The image data was categorized into training, validation, and testing sets to facilitate effective model development and evaluation.

Table 1 Dataset Size source: (Author developed)

	Image dataset	Tabular dataset
Size	- 83 images total for training - 20 images total for validation	(Rows: 2200, Columns: 9)
Columns / Classes	*'Black'*, *'Clayey'*, *'Loamy'*, *'Red'*, *'Sandy'*	'Nitrogen', 'Phosphorus', 'Potassium', 'Temperature', 'Humidity', 'pH_Value', 'Rainfall', 'Crop', 'SoilType'

Table 2 Image Dataset Description source: (Author developed)

width								
class	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Black	17	297.470588	112.154535	225	265	275	275	728
Clayey	17	311	0	311	311	311	311	311
Loamy	17	504.823529	501.713219	199	259	315	474	1937
Red	17	512.529412	566.236823	193	259	275	640	2496
Sandy	15	441.066667	62.000538	300	450.5	474	474	475
height								
class	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
Black	17	218.647059	131.820399	162	183	183	190	728
Clayey	17	162	0	162	162	162	162	162
Loamy	17	481.235294	507.281176	200	258	301	360	1936
Red	17	363.411765	388.285182	57	183	183	480	1664
Sandy	15	416.8	101.853677	270	314.5	474	474	631

Table 3 Tabular Dataset Description source: (Author developed)

	Nitrogen	**Phosphorus**	**Potassium**	**Temperature**	**Humidity**	**pH_Value**	**Rainfall**
count	2200	2200	2200	2200	2200	2200	2200
mean	50.551818	53.362727	48.149091	25.616244	71.481779	6.46948	103.463655
std	36.917334	32.985883	50.647931	5.063749	22.263812	0.773938	54.958389
min	0	5	5	8.825675	14.25804	3.504752	20.211267
0.25	21	28	20	22.769375	60.261953	5.971693	64.551686
0.5	37	51	32	25.598693	80.473146	6.425045	94.867624
0.75	84.25	68	49	28.561654	89.948771	6.923643	124.267508
max	140	145	205	43.675493	99.981876	9.935091	298.560117

Final dataset sample view

Figure 2-1 Image Dataset Sample Source:(Author Developed)

	Nitrogen	Phosphorus	Potassium	Temperature	Humidity	pH_Value	Rainfall	Crop
1	90	42	43	20.87974371	82.00274423	6.502985292	202.9355362	Rice
2	85	58	41	21.77046169	80.31964408	7.038096361	226.6555374	Rice
3	60	55	44	23.00445915	82.3207629	7.840207144	263.9642476	Rice
4	74	35	40	26.49109635	80.15836264	6.980400905	242.8640342	Rice
5	78	42	42	20.13017482	81.60487287	7.628472891	262.7173405	Rice
6	69	37	42	23.05804872	83.37011772	7.073453503	251.0549998	Rice
7	69	55	38	22.70883798	82.63941394	5.70080568	271.3248604	Rice
8	94	53	40	20.27774362	82.89408619	5.718627178	241.9741949	Rice
9	89	54	38	24.51588066	83.5352163	6.685346424	230.4462359	Rice
10	68	58	38	23.22397386	83.03322691	6.336253525	221.2091958	Rice
11	91	53	40	26.52723513	81.41753846	5.386167788	264.6148697	Rice
12	90	46	42	23.97898217	81.45061596	7.50283396	250.0832336	Rice
13	78	58	44	26.80079604	80.88684822	5.108681786	284.4364567	Rice

Figure 2-2 Tabular dataset Sample Source:(Author Developed)

Data flow of the system via prediction models

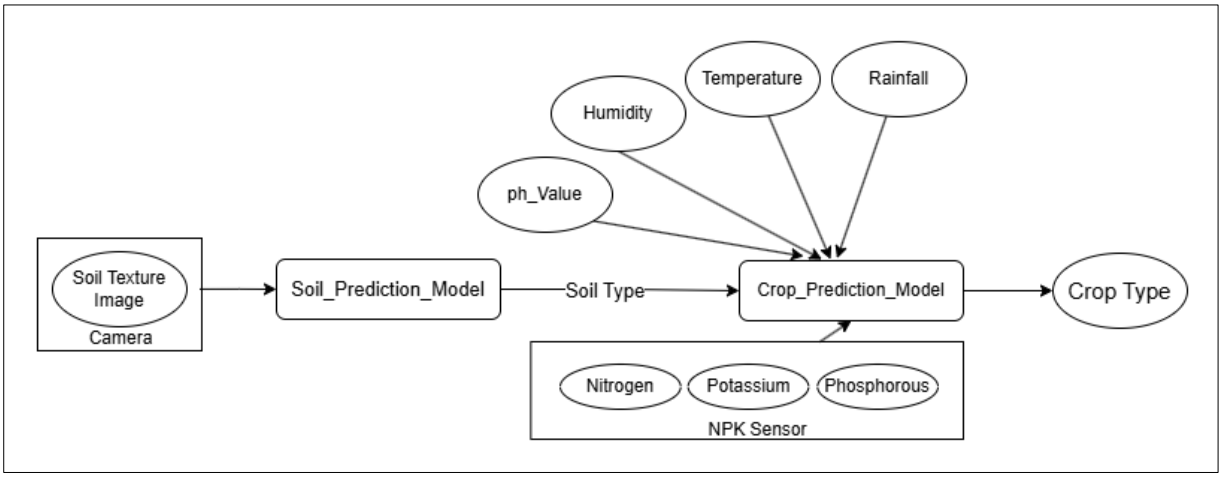

Figure 2-3 Data flow chat of the system source:(Author Developed)

3). Analysis

3.1). Machine Learning for Robotics

Main task of this project is to create a suitable machine learning model. Collecting real-time environmental data and processing the data to generate accurate predictions.

The collected data is analyzed to predict the most suitable crop type for the given area. To achieve this, a trained machine learning model is employed, capable of making accurate predictions based on the acquired data. Machine learning techniques are used to enhance the model's predictive capabilities, ensuring reliable and data-driven decision-making for the project.

By integrating machine learning into robotics, this project aims to develop an intelligent system that can autonomously collect, process, and analyze environmental data, contributing to more efficient and precise agricultural planning.

3.2). Methodology

This project employs a multimodal data approach, utilizing both image-based and tabular datasets to achieve its objectives. The methodology integrates heterogeneous data sources, including visual (image), textual, and numerical data, each processed through specialized machine learning pipelines. Image data undergoes computer vision-based analysis, while textual and numerical data are processed through structured machine learning models. This dual-stream architecture ensures comprehensive feature extraction and enables robust predictive performance by leveraging the complementary strengths of different data modalities.

3.3). Used ML model and effectiveness

Since the final target outcome is categorical data, decided to train model with Decision Tree and Random Forest architectures.

3.3.1). Soil Type Prediction Model

The model utilizes camera-captured images as input, processed through a Convolutional Neural Network (CNN) architecture. The implemented CNN sequentially processes standardized 224×224-pixel images through 17 layers to predict soil types following architecture:

1. Input layer accepting standardized 224×224-pixel RGB images
2. Four convolutional blocks, each containing:
    a. Conv2D layer with ReLU activation
    b. BatchNormalization layer
    c. MaxPooling2D layer for spatial down sampling
3. Flatten layer for feature vector conversion
4. Fully-connected Dense layer (256 units, ReLU activation)
5. Dropout layer (rate=0.5) for regularization
6. Output Dense layer with softmax activation for multi-class classification

```
model = Sequential()
model.add(InputLayer(input_shape=IMAGE_SIZE + [3]))

model.add(Conv2D(32, (3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dense(256, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(len(train), activation="softmax"))  # Multi-class classification

model.summary()
```

Figure 3-1 Model developed with CNN architecture Source:(Author Developed)

The architecture employs Batch Normalization after each convolutional layer to accelerate training convergence and improve gradient flow. The strategic combination of MaxPooling layers and a final Dropout layer (p=0.5) prevents overfitting while maintaining spatial feature hierarchies. This deep network structure enables robust feature extraction from soil texture images, with the 256-unit Dense layer serving as a high-dimensional feature space for final classification decisions.

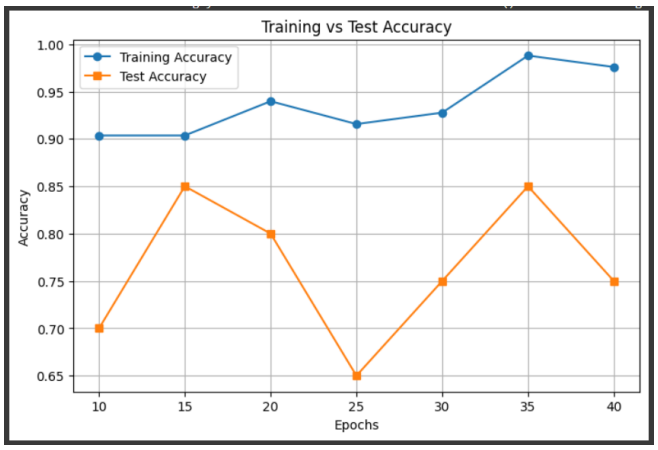

Figure 3-2 Training cycle testing training and test accuracy Source:(Author Developed)

Based on the distribution analysis, a 15-cycle training model was selected to ensure optimal performance. The selection was made considering the following factors:

1. **Avoidance of Overfitting**: The model maintained a balanced learning curve, preventing excessive variance between training and validation data.
2. **Prediction Accuracy**: The model's predictions were evaluated against the dataset, demonstrating satisfactory results without significant deviation.
3. **Performance Stability**: The training process showed consistent improvement across cycles without unnecessary complexity, making 15 cycles an ideal choice.

This approach ensures that the model generalizes well to unseen data while maintaining reliable predictive accuracy.

Testing Image model prediction;

Scenario 01: Expected result is Red.

```
predictions = test_model.predict(img_array)
print("Raw Predictions:", predictions)

print()

predicted_index = np.argmax(predictions)  # Get the class index
predicted_class = class_names[predicted_index]  # Get class name

print(f"Classes: {class_names}v \n")
print(f"Predicted Index: {predicted_index} \n")
print(f"Predicted Class: {predicted_class}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 992ms/step
Raw Predictions: [[1.3088236e-11 1.0612483e-08 5.5486649e-05 9.9994445e-01 2.1156596e-11]]

Classes: ['Black', 'Clayey', 'Loamy', 'Red', 'Sandy']v

Predicted Index: 3

Predicted Class: Red
```

Figure 3-3 Soil prediction testing code Source:(Author Developed)

```
plt.imshow(img)
plt.title(f'Image Path: {img_path} \n Predicted: {predicted_class}')
plt.show()
```

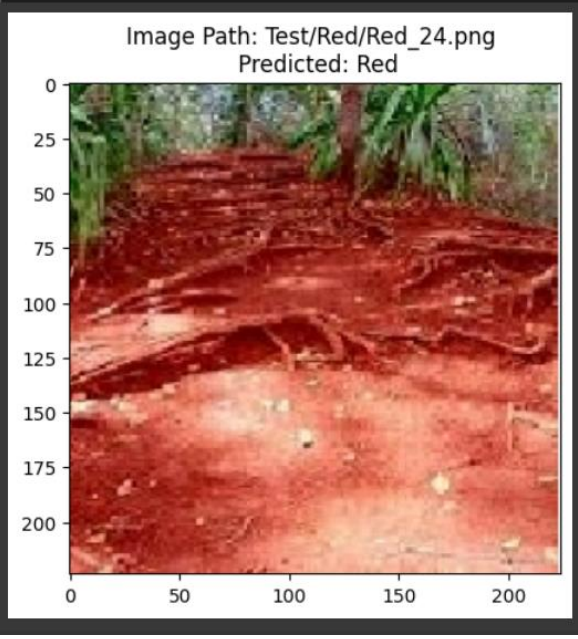

Figure 3-4 Scenario 01 test results Source:(Author Developed)

Scenario 02: Expected result is Sandy

```
Image File found at:  Test/Sandy/sandy 24.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
Raw Predictions: [[1.0883228e-10 6.4564001e-13 7.3335849e-04 2.3359820e-07 9.9926645e-01]]

Classes: ['Black', 'Clayey', 'Loamy', 'Red', 'Sandy']

Predicted Index: 4

Predicted Class: Sandy
```

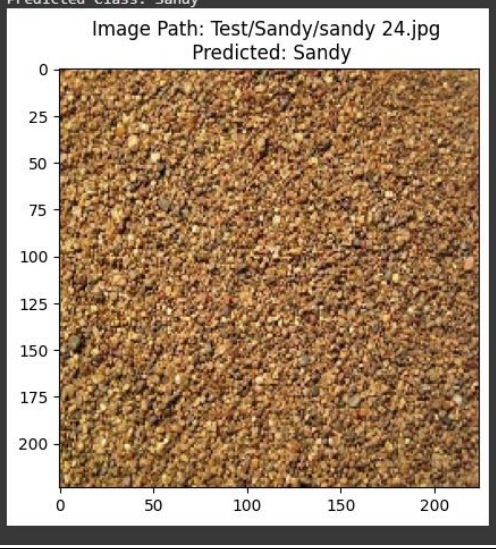

Figure 3-5 Scenario 02 test results Source:(Author Developed)

Scenario 03: Expected result is Loamy

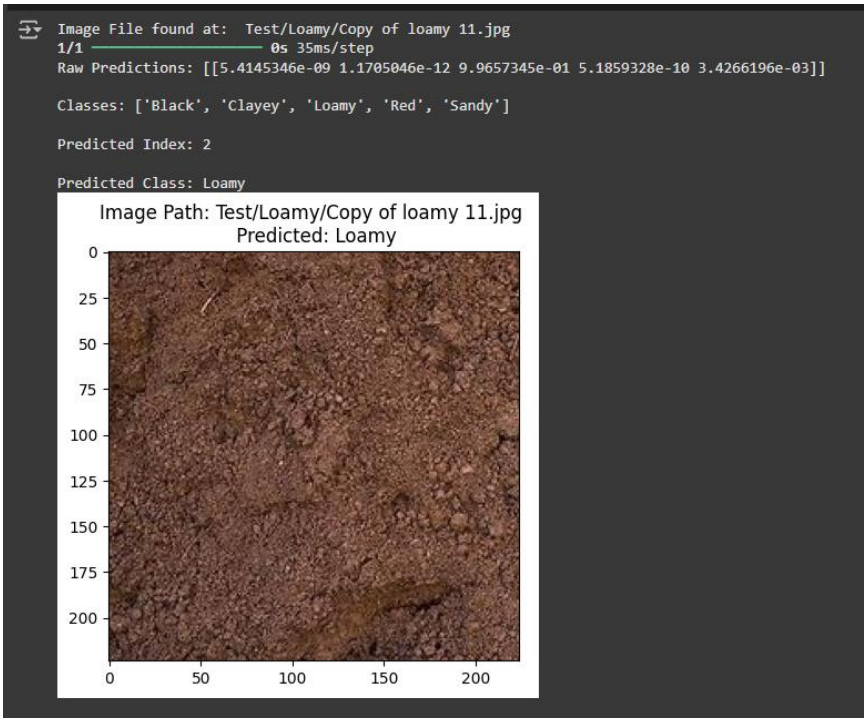

Figure 3-6 Scenario 03 test results Source:(Author Developed)

3.3.2). Crop Type Prediction Model

The second model is responsible for predicting the most suitable crop type based on the analyzed data. To achieve optimal performance, multiple machine learning algorithms will be tested, and the best-performing model will be selected for the final implementation. The models considered for this phase include:

1. Decision Tree
2. Random Forest

By evaluating these models, the most accurate and efficient algorithm will be integrated into the project to enhance prediction reliability.

Selected dataset has several fields as below;

```
[118]: # Check the columns
       df.columns

[118]: Index(['Nitrogen', 'Phosphorus', 'Potassium', 'Temperature', 'Humidity',
              'pH_Value', 'Rainfall', 'Crop', 'SoilType'],
             dtype='object')
```

Figure 3-7 Features of the dataset Source:(Author Developed)

Data scattering in tabular dataset (numerical data);

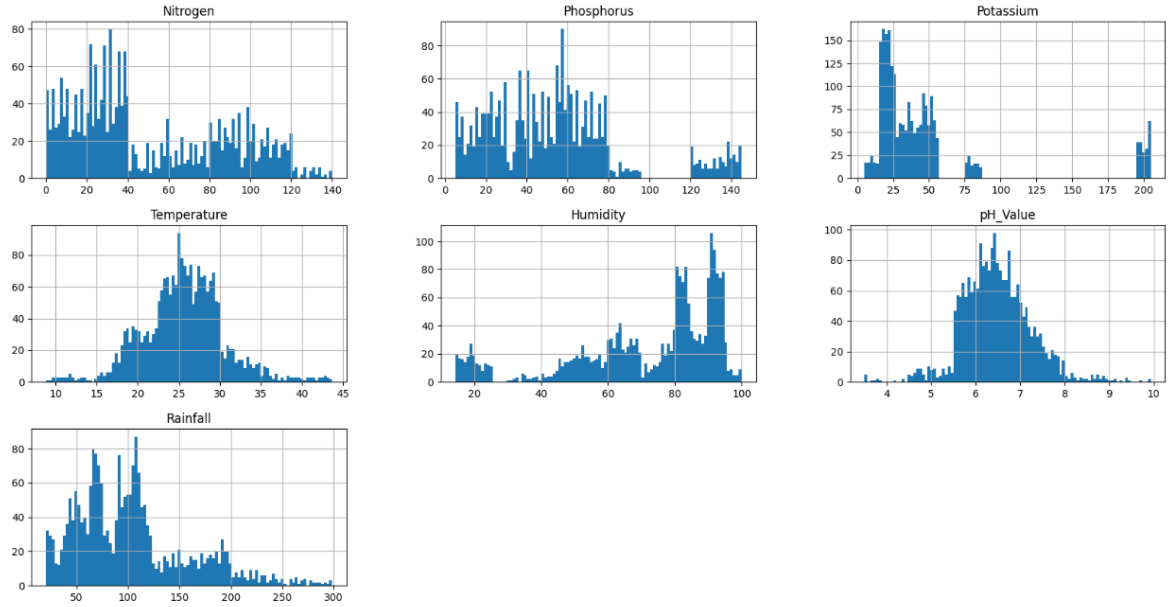

Figure 3-8 Numerical Data distribution Source:(Author Developed)

Features separated for model training as 'x1_feature' and 'y1_feature';

```
[8]: from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
df['SoilType'] = le.fit_transform(df['SoilType'])

x1_feature = ['Nitrogen','Phosphorus','Potassium','Temperature', 'Humidity', 'pH_Value', 'Rainfall', 'SoilType']
y1_feature = ['Crop']

print(x1_feature)
print(y1_feature)

x1 = df.loc[:, x1_feature].values
y1 = df.loc[:, y1_feature].values

print("X1 Shape: " , x1.shape)
print("Y1 Shape: " , y1.shape)
```

```
['Nitrogen', 'Phosphorus', 'Potassium', 'Temperature', 'Humidity', 'pH_Value', 'Rainfall', 'SoilType']
['Crop']
X1 Shape:  (2200, 8)
Y1 Shape:  (2200, 1)
```

Figure 3-9 Features separation Source:(Author Developed)

Split dataset into 70% for training and 30% for testing.

```
[9]: # Split data into train and test. trainin size decided to be 70% of data
X_train, X_test, Y_train, Y_test = train_test_split(x1, y1, test_size=0.3, random_state=0)

print("X_train shape : " , X_train.shape)
print("X_test shape : " , X_test.shape)
print("Y_train shape : " , Y_train.shape)
print("Y_test shape : " , Y_test.shape)
```

```
X_train shape :  (1540, 8)
X_test shape :  (660, 8)
Y_train shape :  (1540, 1)
Y_test shape :  (660, 1)
```

Figure 3-10 Split dataset Source:(Author Developed)

Testing with Decision Tree Model

The model, trained using a Decision Tree architecture, was tested across a range of depths to determine the optimal training cycles.

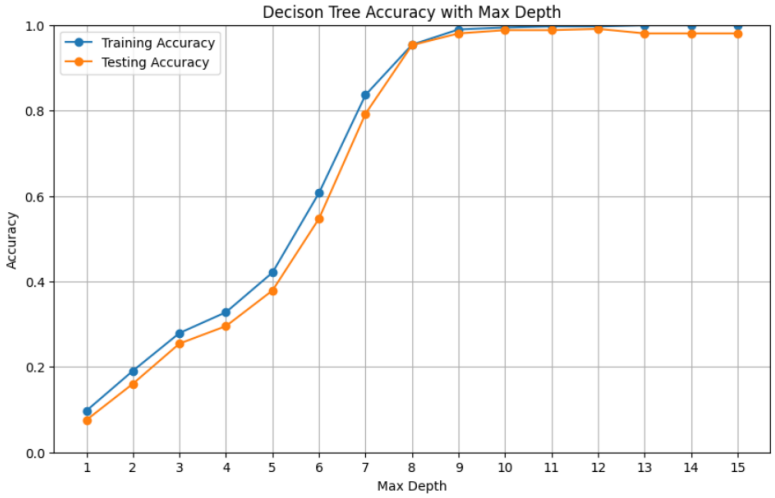

Figure 3-11 Decision Tree model training cycle summary Source:(Author Developed)

Based on the above chart, a maximum depth of 8 was selected for training the model. This depth was chosen to optimize performance while preventing overfitting or underfitting.

The model training resulted in the following accuracy metrics:

- Train Accuracy: 95.39%
- Test Accuracy: 95.00%

The selected depth ensured a balanced trade-off between complexity and generalization, leading to stable and reliable predictions.

Testing with Random Forest Model

The next model, trained using a Random Forest architecture, was tested across a range of depths to determine the optimal training cycles.

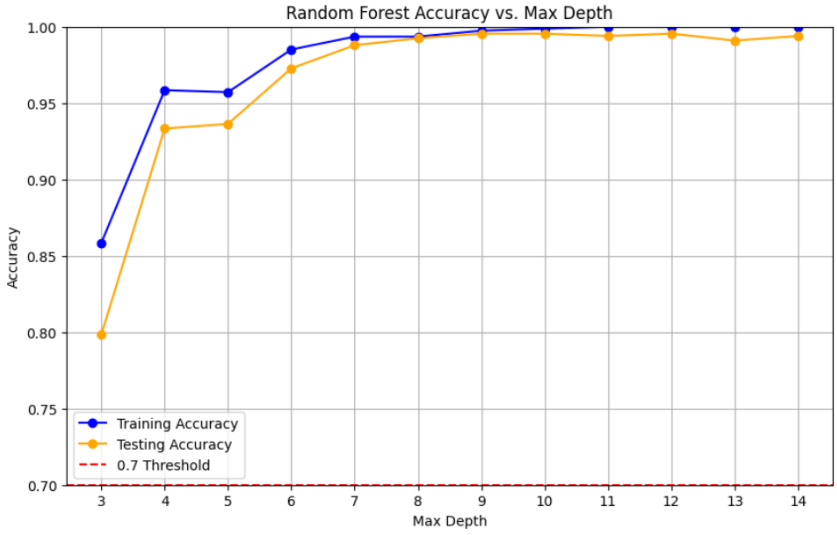

Figure 3-12 Random Forest model training cycle summary Source:(Author Developed)

Based on the above chart, a maximum depth of 8 was selected for training the model. This depth was chosen to optimize performance while preventing overfitting or underfitting.

The model training resulted in the following accuracy metrics:

- Train Accuracy: 99.35%
- Test Accuracy: 99.24%

The selected depth ensured a balanced trade-off between model complexity and generalization, leading to stable and reliable predictions. This configuration allowed the model to achieve high accuracy while maintaining robustness across unseen data.

3.4). Model Evaluation

To mitigate the risk of overfitting and enhance prediction accuracy, training was halted at an optimal point, with extensive fine-tuning applied to achieve the best model performance.

Both models delivered high accuracy during training, with the following results:

Model Performance Comparison

- Decision Tree Model
  - Train Accuracy: 95.39%
  - Test Accuracy: 95.00%
- Random Forest Model
  - Train Accuracy: 99.35%
  - Test Accuracy: 99.24%

Given the higher accuracy and robustness of the Random Forest model, it was selected for further robotic development to ensure optimal performance in real-world applications.

Final combined system is prediction is satisfied and provide satisfactory level recommendation as per input details.

Figure 3-13 Final system prediction Source: (Author Developed)

4). Conclusion

This project successfully developed an ML model for Autonomous Reforestation Robot powered by machine learning to analyze environmental parameters and provide data-driven crop recommendations. By integrating sensor-based data collection with machine learning algorithms, the system effectively determines the most suitable crops for reforestation and agricultural applications, ensuring sustainable land management.

A multimodal data approach was employed, incorporating both image-based and tabular datasets to enhance prediction accuracy. Image processing was conducted using a Convolutional Neural Network (CNN) for soil type classification, while Decision Tree and Random Forest models were tested for crop recommendation based on environmental attributes.

After extensive evaluation, the Random Forest model was selected for its superior accuracy (Train Accuracy: 99.35%, Test Accuracy: 99.24%), outperforming the Decision Tree model. The selected model ensures optimal generalization and robust performance, making it suitable for real-world deployment in autonomous robotics applications.

The findings of this project demonstrate the potential of AI-driven robotics in precision agriculture, offering an efficient and scalable solution for environmental sustainability and reforestation efforts. Future enhancements could involve real-time adaptive learning, autonomous navigation, and integration with drone-based monitoring systems to further improve accuracy and operational efficiency.

5). Future improvements.

While the current system demonstrates strong performance in crop recommendation and soil classification, further enhancements can improve its accuracy, efficiency, and adaptability to diverse environmental conditions. Below are some key areas for future improvement:

1. Expanding the Dataset
    a. To enhance model accuracy and robustness, a larger and more diverse image dataset should be incorporated, covering a wider variety of soil textures and conditions, particularly those found in desert regions.
    b. The tabular dataset should also be expanded to include detailed soil and climate attributes of desert areas, enabling the system to make better predictions in arid environments where reforestation is most challenging.
2. Advanced Machine Learning Techniques
    a. Implementing Deep Learning models such as Transformers or Hybrid CNN-RNN architectures could improve feature extraction and pattern recognition in complex terrains.
    b. Self-supervised learning and transfer learning can be integrated to adapt the model to new environmental conditions with minimal manual intervention.
3. Real-time Adaptive Learning
    a. The current model is pre-trained, meaning it does not learn from real-time data. Future versions should support continuous learning through online machine learning techniques, allowing the system to improve over time as new data is collected.
4. Integration with Remote Sensing & GIS Technologies

   a. Incorporating satellite imagery and drone-based data collection can provide a broader view of soil and vegetation conditions, helping the model make better-informed decisions.
   b. Geographic Information Systems (GIS) can be utilized to map soil health trends over time, improving the decision-making process for large-scale reforestation projects.
5. Autonomous Navigation and Deployment
   a. The robot can be further enhanced with autonomous navigation algorithms, allowing it to efficiently scan large desert areas without human intervention.
   b. Integrating LiDAR, GPS, and real-time sensor fusion would enable the system to move intelligently through rugged and remote environments, ensuring optimal seed placement in desert regions.
6. Integration with IoT and Cloud Computing
   a. Connecting the system to an IoT-based platform can enable real-time monitoring and remote control, allowing researchers and farmers to track soil conditions and model predictions remotely.
   b. Using cloud-based processing can allow the model to handle larger datasets and more complex computations, improving performance in large-scale deployments.
7. Energy Efficiency and Sustainability
   a. Future versions of the robot should incorporate solar power or energy-efficient components, enabling it to operate off-grid in remote desert areas for extended periods.

6). Live Demonstration

Live demonstration can be watch via below link;

YouTube uploaded video demonstration. (URL: https://youtu.be/1Edy1CHsnU4)

7). References

Dharmaraj, 2022. *Convolutional Neural Networks (CNN) — Architecture Explained.* [Online]
Available at: https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243
[Accessed 25 03 2025].

IBM, 2024. *What is a decision tree?.* [Online]
Available at: https://www.ibm.com/think/topics/decision-trees
[Accessed 25 03 2025].