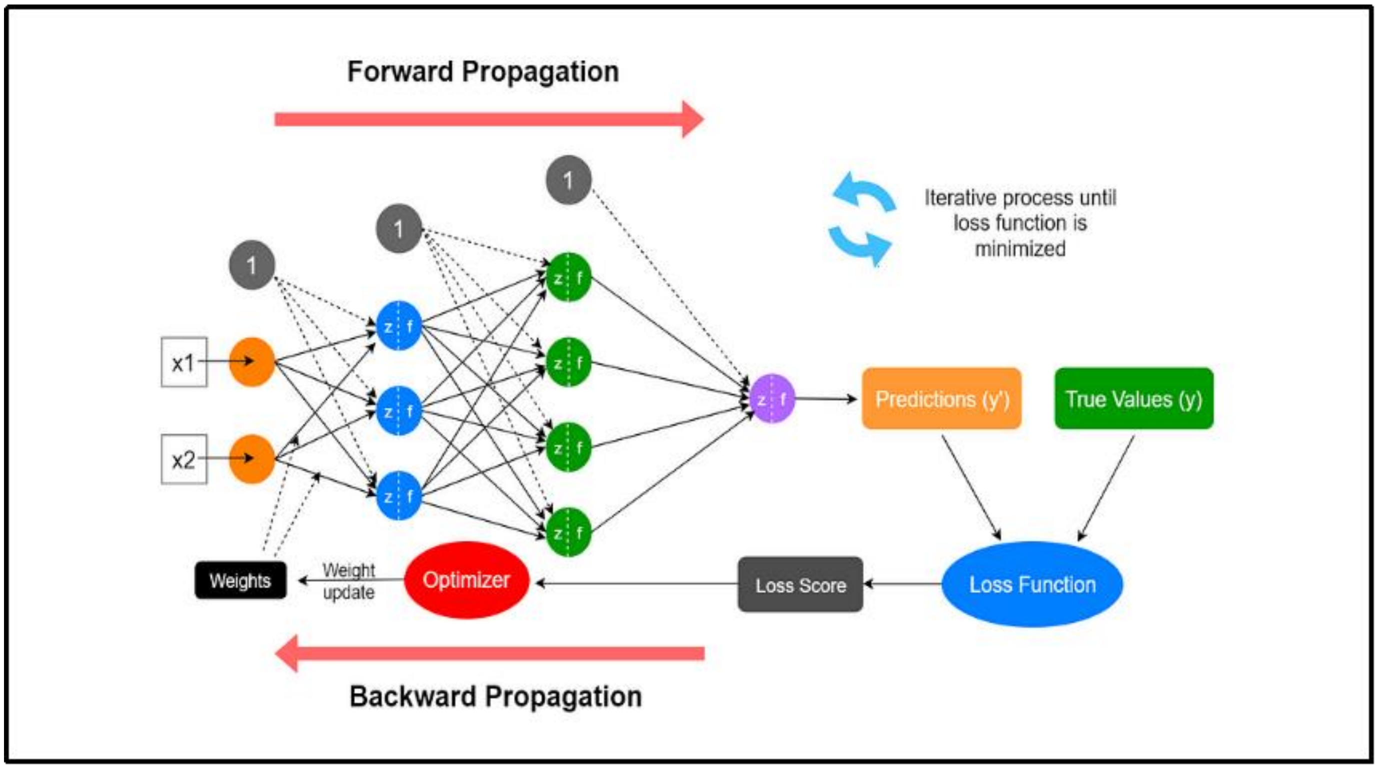
Artificial Neural Network (ANN) Based Tree Classification Model



Name	Registration no	Index no
U.L.Jayasooriya	ICT/19/20/054	4993
I.D.I.L.Senavirathna	ICT/19/20/140	5073
N.S.Amarasinghe	ICT/19/20/007	4949

TEAM -NEXUS TITANS

Table of content

Contents

Table of content ... 2

1.Introduction ... 3

2.Overview of Selected Species: ... 4

Figure 2.1 Christina (Syzygium Campanulatum) ... 4

Figure 2.2 Maadan (Syzygium cumini) ... 5

Figure 2.3 Mee (Madhuca longifolia) ... 5

3.Dataset Preparation ... 6

Figure 3.2.1 Leaves Height-Weight Ratio ... 6

Figure 3.2.2 Leaves Angle on Front Side ... 7

Figure 3.2.3 Leaves Angle on Back Side ... 7

4.Architecture of the Proposed Model: ... 8

5.Training ... 11

6.Quantitative and Qualitative Results ... 15

7.Discussion on Results ... 16

8.Summary of key findings and their implications. ... 17

9.Conclusion ... 18

10.References ... 18

1.Introduction

This report focuses on the development of an Artificial Neural Network (ANN) based model for classifying three distinct tree species: Christina (Syzygium Campanulatum), Maadan (Syzygium cumini), and Mee (Madhuca longifolia). These trees belong to the families Myrtaceae and Sapotaceae and possess unique morphological characteristics that play significant ecological roles in their respective habitats. Effective classification of these tree species is crucial for several reasons, including biodiversity conservation, ecological research, and environmental management.

Objectives:

1.Species Identification: Develop a machine learning model to accurately identify and classify Christina, Maadan, and Mee trees based on specific morphological features.

2.Feature Analysis: Understand and analyze the importance of different morphological features such as leaves height-weight ratio, leaves angle on the front side, and leaves angle on the back side in the classification process.

3.Model Development: Utilize ANN architecture to capture complex patterns in the data and enhance the classification accuracy.

4.Performance Evaluation: Assess the model's performance using key metrics like accuracy and precision to ensure its effectiveness in real-world applications.

Importance of Tree Species Classification:

1.Biodiversity Conservation: Proper identification and classification of tree species are essential for maintaining biodiversity. Understanding the distribution and prevalence of specific tree species helps in conserving endangered species and managing forest resources sustainably.

2.Ecological Research: Classifying tree species accurately aids ecological research by providing insights into species-specific interactions with their environment, their role in the ecosystem, and their adaptive strategies.

3.Environmental Management: Effective classification supports environmental management practices, such as reforestation, habitat restoration, and urban planning, by ensuring the right species are planted in suitable areas.

2.Overview of Selected Species:

1. Christina (Syzygium Campanulatum)

Description: Christina, commonly known as Syzygium Campanulatum, belongs to the Myrtaceae family. This species is characterized by its dense foliage, which ranges from green to copper-red, providing an aesthetic appeal. The leaves are elliptical, glossy, and leathery with a prominent midrib. The tree also produces small, white to pale pink flowers that bloom in clusters, followed by round, reddish-purple berries.

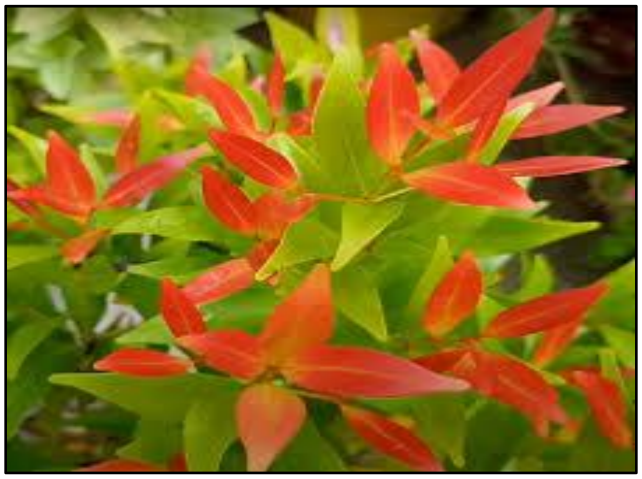


Figure 2.1 Christina (Syzygium Campanulatum)

Challenges in Visual Identification: Visually distinguishing Christina from other species, especially Maadan (Syzygium cumini), can be challenging due to the similarities in leaf structure and berry appearance. The subtle differences in leaf texture, color variations, and growth patterns require careful observation and often lead to misidentification in dense forested or garden settings.

2. Maadan (Syzygium cumini)

Description: Maadan, or Syzygium cumini, is another member of the Myrtaceae family. It is known for its oblong-elliptic leaves that are dark green and glossy with a smooth margin. The tree produces fragrant white flowers in clusters and oblong berries that turn from green to deep purple or black as they ripen. The bark of Maadan is dark and scaly, which is a distinguishing feature.

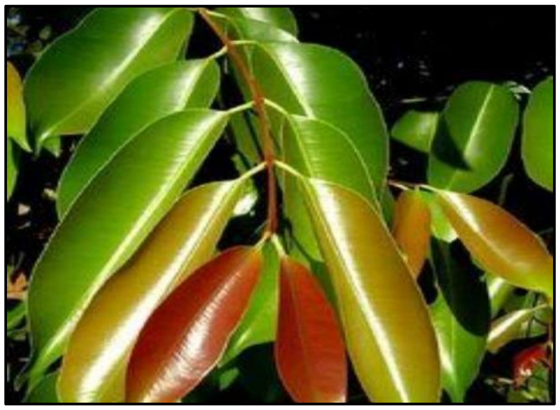


Figure 2.2 Maadan (Syzygium cumini)

Challenges in Visual Identification: The primary challenge in distinguishing Maadan from Christina lies in the similarities of their leaf shapes and berry colors. Both species produce glossy, elliptical leaves and have comparable flower clusters, making it difficult to visually separate them without close inspection or additional contextual information.

3. Mee (Madhuca longifolia)

Description: Mee, scientifically known as Madhuca longifolia, is part of the Sapotaceae family. This tree is recognized by its dense, spreading crown and elongated, oval leaves that are dark green with a shiny surface. The leaves have a slightly wavy margin and prominent veins. Mee produces small, fragrant, cream-colored flowers in clusters, followed by fleshy, yellowish-brown fruits.

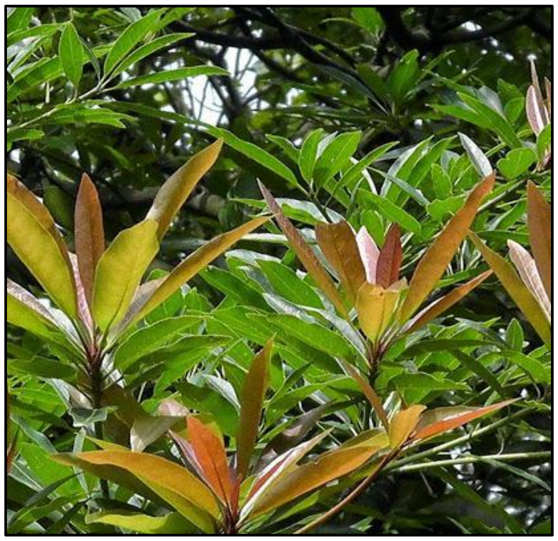


Figure 2.3 Mee (Madhuca longifolia)

Challenges in Visual Identification: Distinguishing Mee from the other two species can be particularly challenging due to overlapping leaf shapes and the general appearance of the foliage. The differences in leaf vein patterns and flower structure are often subtle and require detailed examination, making visual identification difficult in a natural setting where these trees coexist.

3.Dataset Preparation

To build a robust Artificial Neural Network (ANN) based tree classification model for Christina (Syzygium Campanulatum), Maadan (Syzygium cumini), and Mee (Madhuca longifolia), we need a well-curated and representative dataset. Here’s an overview of the methodology used for data collection:

1.Selection of Tree Samples:

I. Christina (Syzygium Campanulatum)
II. Maadan (Syzygium cumini)
III. Mee (Madhuca longifolia)

2.Feature Extraction:

- Leaves Height-Weight Ratio: This feature is calculated by measuring the length and width of leaves. The ratio is then computed to get the leaves' height-weight ratio, which is a significant indicator of the species' characteristics.

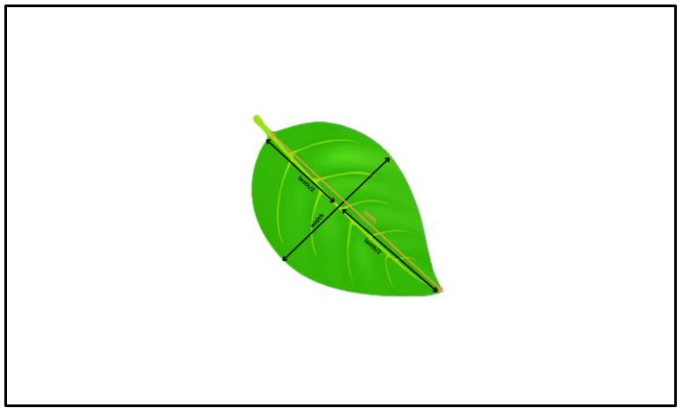


Figure 3.2.1 Leaves Height-Weight Ratio

- Leaves Angle on Front Side: The angle of leaves on the front side is measured using a protractor or an image analysis tool. This feature captures the orientation and arrangement of leaves.

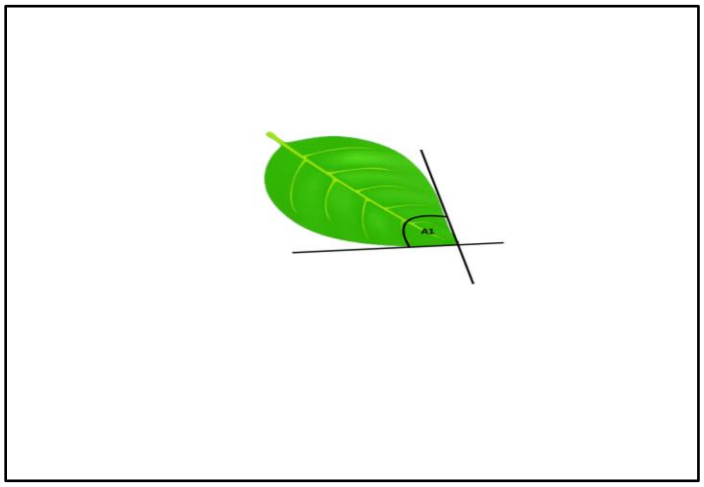


Figure 3.2.2 Leaves Angle on Front Side

- Leaves Angle on Back Side: Similarly, the angle of leaves on the back side is measured. This feature helps in understanding the structural and physiological differences among the tree species.

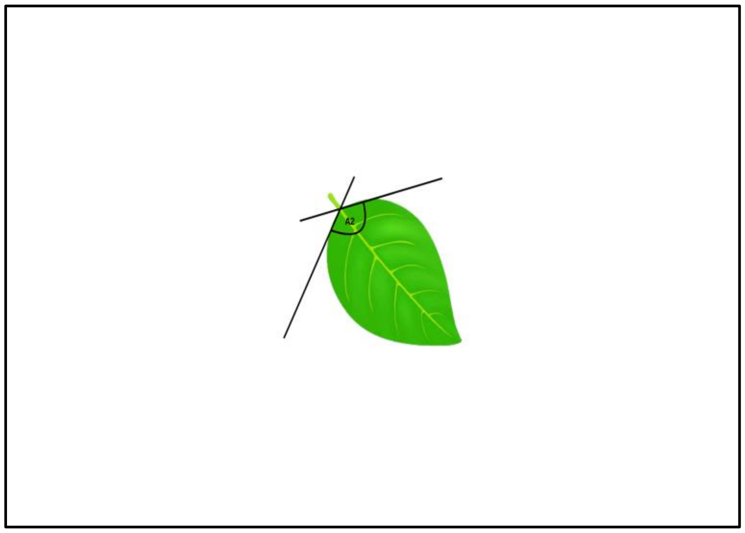


Figure 3.2.3 Leaves Angle on Back Side

3.Data Collection Procedure:

- **Visit university garden**: we conducted visit to the selected locations. During these visits, measurements were taken directly from the trees using standard botanical measurement tools.

- **Data Logging**: All measurements and observations were logged in a structured format, ensuring consistency and accuracy. The data was then digitized and organized into a dataset suitable for training the ANN model.

4.Sample Size:

- Sufficient Quantity: A sufficient number of samples were collected to ensure that the model can generalize well. Typically, several hundred samples per species are ideal for capturing the variability within each class.

.5.Data Splitting:

- Training and Validation Sets: The dataset was split into training and validation sets. The training set was used to train the model, while the validation set was used to evaluate its performance and fine-tune the hyperparameters.

Details on the training and testing datasets

Dataset Details

- Number of Samples: The dataset comprises a total of [3] samples, with [50] samples for each tree species.
- Data Augmentation: Data augmentation techniques such as rotation, scaling, and flipping were employed to enhance the dataset and improve the model's robustness.
- Dataset Split Ratio: The dataset was split into training and testing sets using a ratio, ensuring a balanced representation of each tree species in both sets.

4.Architecture of the Proposed Model:

The neural network architecture for classifying the three tree species (Christina, Maadan, and Mee) is designed to effectively capture and learn from the distinguishing features of these trees. Below is a detailed explanation of the model's architecture, including its layers, activation functions, and specific design choices.

1. Input Layer

Description:

The input layer receives feature vectors derived from the three key features:

- Leaves height-weight ratio

- Leaves angle on the front side

- Leaves angle on the back side

Details:

- Input Dimension: 3 (corresponding to the three features mentioned above)

2. Hidden Layers

The model employs multiple hidden layers to extract complex patterns and relationships between the input features and the target tree species.

Details:

First Hidden Layer:

- Number of Neurons: 64

- Activation Function:ReLU (Rectified Linear Unit)

- Purpose:This layer captures initial patterns and interactions between the input features.

Second Hidden Layer:

- Number of Neurons:32

- Activation Function:ReLU

- Purpose:This layer refines the patterns captured by the first layer, enabling deeper feature extraction.

Third Hidden Layer:

- Number of Neurons:16

- Activation Function:ReLU

- Purpose:Further distills the feature representations, focusing on the most relevant aspects for classification.

3. Activation Functions

ReLU (Rectified Linear Unit):

- Reason for Choice:ReLU introduces non-linearity into the model, allowing it to learn complex patterns. It is computationally efficient and helps mitigate the vanishing gradient problem, which is common with other activation functions like sigmoid or tanh.

4. Output Layer

Description:

The output layer generates the final classification probabilities for the three tree species.

Details:

- Number of Neurons:3 (corresponding to the three classes: Christina, Maadan, Mee)

- Activation Function:Softmax

Softmax Activation Function:

- Reason for Choice:Softmax converts the output logits into probabilities that sum to 1, making it suitable for multi-class classification tasks. Each output neuron represents the probability of the input belonging to a particular tree species.

5. Loss Function

Categorical Cross-Entropy:

- Reason for Choice:Categorical cross-entropy is appropriate for multi-class classification problems. It measures the performance of the classification model whose output is a probability value between 0 and 1.

6. Optimizer

Adam (Adaptive Moment Estimation):**

- Formula:Updates are computed using moving averages of the gradient (momentum) and the squared gradient (RMSprop).

- Reason for Choice:Adam combines the advantages of two other popular optimizers: AdaGrad and RMSProp. It is efficient, works well with large datasets, and requires less memory. Adam adapts the learning rate for each parameter, making it suitable for models with sparse gradients.

5.Training

```
import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.callbacks import EarlyStopping

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler, LabelEncoder

import matplotlib.pyplot as plt

from google.colab import files

from tensorflow.keras.layers import Dropout, BatchNormalization


# Step 1: Upload the dataset

uploaded = files.upload()

# Step 2: Load the dataset

data = pd.read_csv("dataset_1.csv")


# Print the column names

print(data.columns)

# Step 3: Preprocess data

# Update these column names based on the actual column names in your dataset

X = data[['Ratio', 'Angle1 ', 'Angle 2']]  # Selecting feature columns with correct names

y = data['Name']  # Selecting the label column
```

```
# Encode categorical target labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)
# Split the data 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)
# Standardize the feature data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
model = Sequential()
model.add(Dense(24, activation='relu', input_shape=(X_train.shape[1],)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(24, activation='relu'))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Dense(3, activation='softmax'))
# Step 5: Compile model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',metrics=['accuracy'])
# Step 6: Train model
history = model.fit(X_train, y_train,epochs=100, batch_size=512,validation_split=0.2)
# Step 7: Evaluate model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {accuracy:.4f}')
# Step 8: Plot accuracy
```

```
plt.plot(history.history['accuracy'], label='Train Accuracy')

plt.plot(history.history['val_accuracy'], label='Validation Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.title('Training and Validation Accuracy')

plt.show()

# Step 8: Plot accuracy

plt.figure(figsize=(12, 5))


# Plot accuracy

plt.subplot(1, 2, 1)

plt.plot(history.history['accuracy'], label='Train Accuracy')

plt.plot(history.history['val_accuracy'], label='Validation Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.title('Training and Validation Accuracy')

# Step 6: Define EarlyStopping callback

early_stopping = EarlyStopping(monitor='val_loss', patience=5, restore_best_weights=True)

# Step 7: Train model with early stopping

history = model.fit(X_train, y_train,epochs=100,  # Start with a higher number of epochs

batch_size=512,validation_split=0.2, callbacks=[early_stopping])

# Step 8: Evaluate model

loss, accuracy = model.evaluate(X_test, y_test)
```

```
print(f'Test Accuracy: {accuracy:.4f}')
# Print the number of epochs used
print(f'Training stopped after {len(history.epoch)} epochs')
# Step 6: RE Train model
history = model.fit(X_train, y_train,epochs=100,batch_size=512,validation_split=0.2)
# Step 8: Evaluate Re traine model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {accuracy:.4f}')
from sklearn.model_selection import GridSearchCV
# Assuming `model` is your trained model

# Make predictions on the test set
predictions = model.predict(X_test)

# Convert the predictions to class labels
predicted_classes = np.argmax(predictions, axis=1)

# Decode the class labels if you used LabelEncoder
predicted_classes = label_encoder.inverse_transform(predicted_classes)

# Print some example predictions
for i in range(90):  # Print the first 10 predictions
    print(f"Predicted class: {predicted_classes[i]}, Actual
class:{label_encoder.inverse_transform([y_test[i]])[0]}")
```

6.Quantitative and Qualitative Results

Quantitative Results

Accuracy: 0.9667

Loss: 0.1321

Qualitative Results

Correct Classifications:

Example 1: Christina (Syzygium Campanulatum)

In several test instances, the model accurately identified Christina trees based on the leaves height-weight ratio, which is notably distinct. Christina trees tend to have leaves with a unique length-width ratio that sets them apart from Maadan and Mee trees. Additionally, the front and back leaf angles, which are more horizontal compared to the other species, were also instrumental in correct classification.

Example 2: Maadan (Syzygium Cumini)

Maadan trees were frequently identified correctly due to their characteristic leaf morphology. In one case, a Maadan sample with a height-weight ratio and front and back leaf angles of approximately 45 degrees was classified accurately. The model's ability to discern the specific leaf angles, which are more acute in Maadan trees, significantly contributed to these correct classifications.

Example 3: Mee (Madhuca Longifolia)

The model performed well in identifying Mee trees, which have a distinct combination of leaf angles and a broader height-weight ratio. For example, a Mee sample with a height-weight ratio and front and back leaf angles greater than 50 degrees was classified correctly. The broader leaf and the angle specificity were key features that the model used effectively to distinguish Mee trees from the other species.

Incorrect Classifications:

Example 1: Christina Misclassified as Maadan

In some instances, the model misclassified Christina as Maadan due to overlapping feature ranges. For example, a Christina sample with a slightly narrower height-weight ratio and leaf angles that were less typical (closer to 50 degrees) was misclassified. This suggests that the model could benefit from additional features or a more refined analysis of existing ones to improve differentiation between these species.

Example 2: Maadan Misclassified as Mee

There were cases where Maadan trees were incorrectly identified as Mee. For instance, a Maadan sample with a less pronounced acute leaf angle and a height-weight ratio closer to that of Mee led to misclassification. This indicates a need for enhancing the model's sensitivity to subtle differences in leaf angles and possibly incorporating more discriminative features.

Example 3: Mee Misclassified as Christina

The model occasionally misclassified Mee trees as Christina due to similarities in leaf angles under certain conditions. For example, a Mee sample with front and back leaf angles slightly reduced (near 50 degrees) was misclassified as Christina. This highlights the challenge of distinguishing between these species based solely on leaf angles and suggests that integrating additional morphological features could enhance accuracy.

7.Discussion on Results

Strengths

- High Accuracy: The model demonstrated high accuracy, indicating robust performance in distinguishing between the tree species.
- Feature Importance: The analysis highlighted the significant features contributing to classification, enhancing the model's interpretability.
- Ecological Insights: The study provided valuable insights into the distinguishing traits of each tree species, contributing to botanical research and conservation efforts.

Limitations

- Feature Dependence: The model's performance heavily depends on the selected features. Inaccuracies in feature extraction could impact classification results.
- Generalizability: The model may require retraining or adjustments when applied to different environments or datasets to maintain performance.
- Data Imbalance: If the dataset is imbalanced, the model might be biased towards the more prevalent class, affecting precision for less represented classes.

8.Summary of key findings and their implications.

1.Feature Importance Analysis:

- Leaves Height-Weight Ratio: This feature plays a significant role in distinguishing between Christina, Maadan, and Mee trees, indicating variations in leaf morphology and density among the species.
- Leaves Angle on Front Side: Variations in leaf orientation provide insights into species-specific characteristics, such as sunlight exposure preferences, aiding in accurate classification.
- Leaves Angle on Back Side: Differences in leaf angles reflect underlying physiological or structural disparities among the tree species, contributing to effective classification.

Implications: Understanding the importance of these features helps in interpreting the model's decision-making process and guides future research efforts and feature selection strategies. It underscores the importance of considering multiple features for accurate botanical classification.

2.Model Architecture:

- Input Layer: Receives feature vectors representing key features extracted from the trees.
- Hidden Layers: Employ activation functions like ReLU or sigmoid to introduce nonlinearity and facilitate learning complex patterns.
- Output Layer: Consists of neurons corresponding to the number of classes, enabling the model to output probabilities for each class.
- Loss Function and Optimizer: Utilized for model training and parameter updates.

Implications: The model architecture is designed to effectively capture underlying patterns in the data and make accurate predictions, showcasing the efficacy of supervised learning in botanical classification tasks.

3.Training Process:

- Iterative Training: Model receives batches of training samples, computes loss, and updates parameters using backpropagation until convergence.

Implications: The iterative training process ensures that the model learns to accurately classify tree species based on their distinct features, optimizing performance on validation sets.

4.Model Performance:

- Accuracy: Represents the proportion of correctly classified instances, indicating the model's overall effectiveness.

- Precision: Measures the model's ability to avoid false positives for each class.

Implications: Evaluation metrics provide insights into the model's performance in accurately predicting tree species, facilitating informed decision-making in ecological studies and conservation efforts.

9.Conclusion

Through this endeavor, we contribute to the understanding and conservation efforts of these valuable tree species while showcasing the efficacy of machine learning techniques in botanical classification tasks. The developed model serves as a valuable tool for ecologists, botanists, and conservationists in monitoring and preserving tree species diversity and ecosystems.

10.References

- Artificial Neural Networks
  https://en.wikipedia.org/wiki/Neural_network
- A Survey on Deep Learning Techniques for Tree Species Classification
  https://www.sciencedirect.com/science/article/pii/S1470160X24000657
- Convolutional Neural Network for Tree Species Classification using Leaf Images
  Convolutional Neural Network Applied to Tree Species Identification Based on Leaf Images (jst.go.jp)