

Plant Leaf Classification Using Artificial Intelligence

Prepared by: Waleed Alfaifi

Introduction

This project aims to build an intelligent model capable of identifying plant types from leaf images. Machine learning techniques were applied to train a model using a dataset categorized into three classes: Aromatic, Harmful, and Medicinal.

Dataset Description

The dataset used consists of plant leaf images distributed across three folders according to their class. The data was split into 70% for training, 10% for validation, and 20% for testing.

Model Used

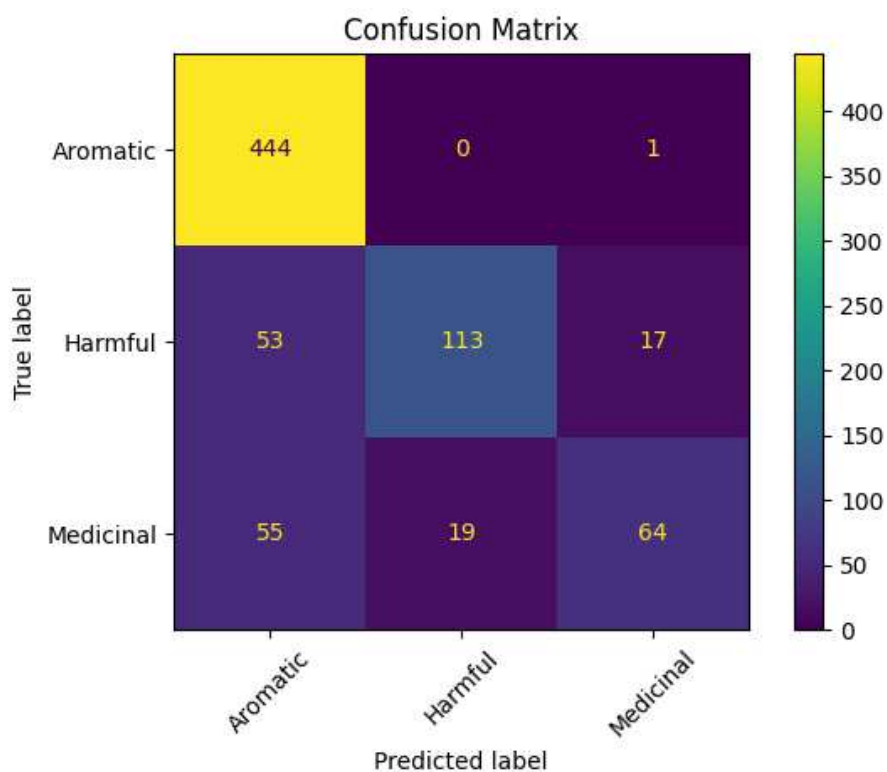
A Random Forest model was used, which is a supervised learning algorithm from the scikit-learn library. Images were converted into numerical arrays using OpenCV by flattening resized images into feature vectors.

Model Results

The model achieved strong performance in classifying the Aromatic class, with some misclassifications observed in the Harmful and Medicinal classes. Overall, the model demonstrated reasonable accuracy given the simplicity of the approach and feature extraction method.

Confusion Matrix

The confusion matrix below illustrates the classification performance across the three plant categories:



Classification Report

The classification report below summarizes precision, recall, and f1-score for each class:

	precision	recall	f1-score	support
Aromatic	0.80	1.00	0.89	445
Harmful	0.86	0.62	0.72	183
Medicinal	0.78	0.46	0.58	138
accuracy			0.81	766
macro avg	0.81	0.69	0.73	766
weighted avg	0.81	0.81	0.79	766

Code Used in This Project

```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
```

```

ConfusionMatrixDisplay
import matplotlib.pyplot as plt

data = []
labels = []
img_size = 64
base_path = 'Plant_Dataset_Split/train'

for class_name in os.listdir(base_path):
    class_path = os.path.join(base_path, class_name)
    if not os.path.isdir(class_path):
        continue
    for img_name in os.listdir(class_path):
        img_path = os.path.join(class_path, img_name)
        try:
            img = cv2.imread(img_path)
            img = cv2.resize(img, (img_size, img_size))
            data.append(img.flatten())
            labels.append(class_name)
        except:
            continue

X = np.array(data)
y = np.array(labels)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

clf = RandomForestClassifier()
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print(classification_report(y_test, y_pred))

cm = confusion_matrix(y_test, y_pred, labels=clf.classes_)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=clf.classes_)
disp.plot(xticks_rotation=45, cmap='viridis')
plt.title("Confusion Matrix")
plt.tight_layout()
plt.savefig("confusion_matrix.png")
plt.show()

```