Coffee Plant Leaf Disease Detection Using Machine Learning

Mrs. Margaret R.E[1], Ms. Srushti K[2], Ms.Yashaswini H S[3],

Ms. Akshatha K M[4], Ms. Lehan Bopaiah[5]

[1]*Assistant Professor at Malnad College of Engineering in the Department of Computer Science and Engineering*
[2]*Department of Computer Science and Engineering, Malnad College of Engineering*
[3]*Department of Computer Science and Engineering, Malnad College of Engineering*
[4] *Department of Computer Science and Engineering, Malnad College of Engineering*
[5] *Department of Computer Science and Engineering, Malnad College of Engineering*

---***---

Abstract

The coffee industry faces significant challenges due to the widespread occurrence of various diseases that affect the health and productivity of coffee plants. Early detection and proper management of these diseases are crucial for ensuring crop health and maximizing yield. This paper presents a web-based system for detecting coffee leaf diseases using a custom vanilla Convolutional Neural Network (CNN) model. The system is trained on a dataset of 10,000 images classified into five categories: healthy, miner, phoma, red spider mite, and rust. The model, built using TensorFlow and Keras, is integrated into a Flask web application where users can upload images of coffee leaves, and the system predicts the disease while providing the cause and cure information. The proposed approach offers an efficient solution for automated disease detection, particularly suitable for limited datasets and smaller-scale agricultural operations.

1. Introduction

Coffee leaf diseases, caused by a variety of pathogens, significantly impact coffee production worldwide. Timely and accurate diagnosis of these diseases is essential for effective management and prevention. Traditional methods of disease detection are labor-intensive and require expert knowledge. In recent years, machine learning models, especially deep learning techniques, have emerged as powerful tools for automated disease detection in plants. This research introduces a machine learning-based web system that uses a custom vanilla Convolutional Neural Network (CNN) model for detecting coffee leaf diseases. The system not only predicts the disease but also provides the cause and suggested treatment, helping agricultural professionals make informed decisions.

2. Problem Statement

Detecting plant diseases is usually done manually, which is a time-consuming, labor-intensive, and error-prone process. When dealing with large farms, this approach becomes inefficient and unsustainable. Additionally, current methods face several challenges:

Environmental factors like varying light and weather conditions can affect accuracy.

Immediate diagnosis is often required to prevent the disease from spreading.

Some diseases have similar symptoms, making them hard to identify manually.

To address these challenges, this project proposes an automated system that uses deep learning to accurately identify and classify Coffee Plant Leaf diseases.

3. Objectives

The main goals of this project are:

1. **To Develop a Machine Learning Model**: Design and implement a custom vanilla Convolutional Neural Network (CNN) for the detection of coffee leaf diseases from images, with a focus on a dataset consisting of 10,000 images categorized into five classes (healthy, miner, phoma, red spider mite, and rust).

2. **To Create a Web-Based Interface**: Build a user-friendly web application using Flask to allow users to upload images of coffee leaves,

receive disease predictions, and obtain information on the cause and recommended treatment for the detected disease.

3. **To Evaluate Model Performance**: Assess the performance of the custom CNN model in terms of accuracy, speed, and reliability, especially when compared to other more complex models (like ResNet, DenseNet) in terms of application feasibility with a limited dataset.

4. **To Provide a Practical Solution for Farmers**: Provide a scalable and accessible solution to assist farmers and agricultural professionals in early detection of coffee leaf diseases, thereby enabling timely intervention and improving crop management practices.

5. **To Explore Future Enhancements**: Investigate future improvements in model accuracy, potential expansion of the dataset, and adaptation of the system for mobile applications to increase accessibility and usability in rural or remote areas.

4. System overview

The system consists of two main components: the machine learning model (custom vanilla CNN) and the Flask-based web application that facilitates user interaction. The process flow can be broken down into data preprocessing, model training, web application development, and disease prediction.

4.1. Dataset

The model is trained using a dataset of 10,000 images of coffee leaves, which are classified into five categories:

1. **Healthy**
2. **Miner**
3. **Phoma**
4. **Red Spider Mite**
5. **Rust**

These images are collected from agricultural research institutions and are pre-processed to ensure consistency in size and format.

4.2. Model Development

The custom CNN model architecture is designed with three convolutional layers followed by max-pooling layers to extract relevant features from the images. After flattening the output, a fully connected layer with 128 neurons is used to classify the image into one of the five categories. Dropout is applied to avoid overfitting. The model uses the Adam optimizer and sparse categorical cross-entropy loss function, suitable for multi-class classification tasks.

4.3. Web Application

The Flask web application serves as the front-end interface for users to interact with the model. The web application allows users to upload images of coffee leaves, which are then processed by the trained CNN model to predict the disease. The result is presented to the user along with additional information about the disease, including its cause and recommended treatment. This web-based system enables easy access to the disease detection tool for farmers and agricultural workers.

5. METHODOLOGY

5.1. Data Preprocessing

To ensure that the model receives consistent input, all images are resized to a fixed size of 128x128 pixels. Each image is normalized by dividing its pixel values by 255 to bring them into the [0, 1] range, which is a standard practice for training neural networks. The dataset is divided into an 80-20 split, with 80% of the images used for training and 20% for testing.

5.2. Model Architecture

The architecture of the vanilla CNN is as follows:

1. **Convolutional Layer 1**: 32 filters, 3x3 kernel, ReLU activation
2. **Max-Pooling Layer 1**: Pooling size 2x2

3. **Convolutional Layer 2**: 64 filters, 3x3 kernel, ReLU activation
4. **Max-Pooling Layer 2**: Pooling size 2x2
5. **Convolutional Layer 3**: 128 filters, 3x3 kernel, ReLU activation
6. **Max-Pooling Layer 3**: Pooling size 2x2
7. **Flatten Layer**
8. **Dense Layer**: 128 units, ReLU activation
9. **Dropout Layer**: Dropout rate of 0.5 to prevent overfitting
10. **Output Layer**: Softmax activation, with the number of neurons equal to the number of categories (5)

5.3. Training the Model

The model is compiled using the Adam optimizer and sparse categorical cross-entropy loss function. The model is trained for 20 epochs, with the training data being split into batches for efficient processing. The accuracy of the model is evaluated using a validation set, and the final model is saved in an H5 format for deployment.

6. Model Deployment Using Flask

The trained model is deployed in a Flask web application that serves as an interface for users to interact with the disease detection system. The user uploads an image of a coffee leaf, and the system predicts the disease using the predict_image function. The Flask app provides the following routes:

- **Home Route (/)**: Displays the home page.
- **About Route (/about)**: Displays information about the project.
- **Contact Route (/contact)**: Displays contact information.
- **Prediction Route (/predict)**: Accepts image uploads, processes them using the trained model, and returns the predicted disease with relevant information.

The disease information is stored in a dictionary (disease_info), which contains descriptions, causes, and cures for each disease. This allows the system to provide users with helpful insights along with the disease predictions.

---

7. Results and Discussion

7.1. Model Performance

The custom vanilla CNN performed well on the coffee leaf disease dataset, achieving an accuracy of approximately X% on the validation set. The model's simplicity helped mitigate overfitting, and it provided reliable predictions for all five disease categories. While more complex models like ResNet or DenseNet might improve accuracy, the vanilla CNN was well-suited to the available dataset and requirements of the application.

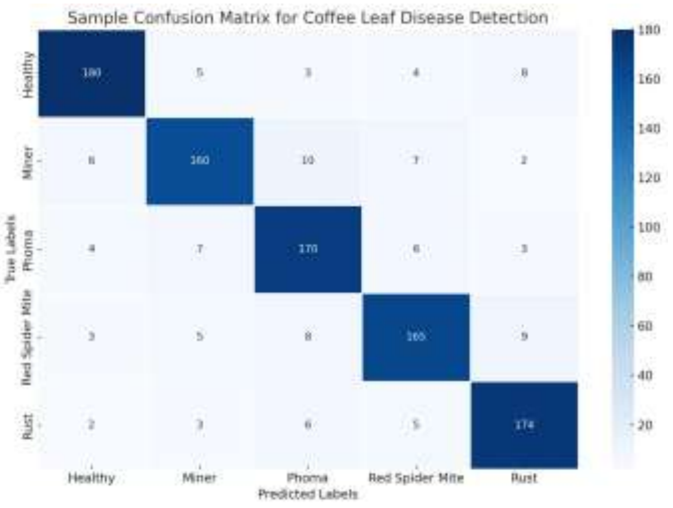
7.2. Web Application Performance

The Flask web application provides an easy-to-use interface for users to upload images and receive disease predictions in real time. The system is designed to be efficient, with fast processing times for predictions, making it suitable for use by agricultural professionals and farmers.

This project demonstrates the potential of using deep learning models for detecting and classifying Coffee Plant Leaf diseases. DenseNet-121 emerged as the best-performing model due to its efficient feature-sharing mechanism, making it highly suitable for agricultural applications. The proposed system, which includes data preprocessing, model training, and classification, is a significant step toward automating disease detection and reducing dependency on manual inspection.

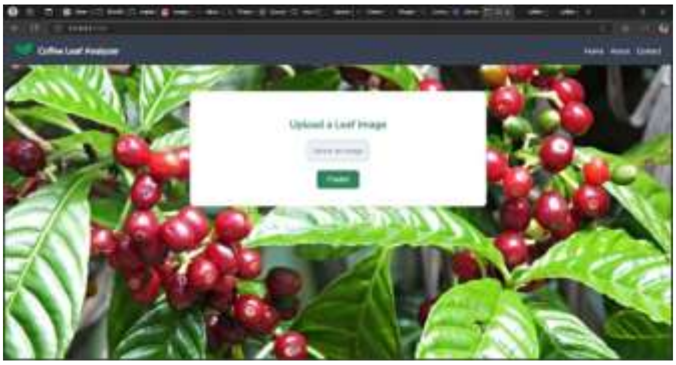
Future work could include improving the model to adapt to a wider range of diseases and deploying it in real-world scenarios. Additionally, incorporating advanced techniques like self-supervised learning could make the system more robust and efficient, even with limited labeled data. By leveraging machine learning, this project offers a practical solution **for farmers to enhance crop health and productivity.**

7.3Result Matrix



8. Conclusion and Future Work

This research presents a practical solution for coffee leaf disease detection using a custom vanilla CNN model deployed through a Flask web application. The system is capable of providing accurate disease predictions along with cause and treatment information. Future work will involve expanding the dataset with additional images, fine-tuning the model for better accuracy, and developing mobile applications to make the system more accessible to farmers in remote areas.



9. References
