Leaf Disease Detection System for Autonomous Rover with Drag-and-Drop Testing Tool

Ujwal Waghray (50560587)

Prithvi Chauhan (50496099)

1. Introduction

The goal of the proposed study is to use deep learning and computer vision algorithms to create a Leaf Disease Detection System. Originally intended as part of an autonomous rover, this technology can recognize and categorize six unique types of leaf diseases. Both agricultural professionals and amateurs are expected to benefit from the technology's accurate plant health diagnosis. Originally designed to communicate with an autonomous rover to take and analyze leaf photographs, the technology has evolved into a drag-and-drop testing tool. This modification enhances accessibility while preserving the original concept's high degree of realism.

2. Scope of the Project

State of the Art

Present-day methods for identifying leaf illnesses mostly rely on conventional image processing techniques or basic machine learning, neither of which are trustworthy in practical settings. Recent advances in deep learning have demonstrated increased accuracy in image categorization tasks. This study leverages these advancements by adapting robust pre-trained algorithms specifically for leaf disease detection.

Application Inputs: Drag-and-drop or rover-captured images of both healthy and sick leaves.

Results:

Determining the type of sickness

Confidence in classification

Results: display displays the supplied image together with symptoms of the condition.

3. Overview of Algorithms

The State of the Art

Historically, the detection of leaf diseases has depended on simple machine learning algorithms and classical image processing techniques, which frequently fail to handle real-world complexity like obstructed leaves, overlapping symptoms, or inconsistent image quality. Recent developments in deep learning have shown remarkable robustness and accuracy in picture categorization challenges. Utilizing pre-trained convolutional neural network (CNN) architectures that have been optimized for leaf disease detection, this study takes advantage of these developments.

Algorithms Employed

We trained and assessed four different algorithms in order to identify the best architecture for detecting leaf disease:

AlexNet
AlexNet50 refers to variations of the AlexNet concept in architectures like ResNet50, which blend the legacy principles of AlexNet with more sophisticated techniques like residual learning, even if its primary contribution is revolutionizing computer vision tasks.

ResNet50
A deep residual learning framework that uses residual connections to solve vanishing gradient problems and train very deep networks.
Strengths: High accuracy and reliable feature extraction on big datasets.

EffectiveNetB0
A class of models that use compound scaling to maximize efficiency and accuracy.
Strengths: Its computational efficiency and lightweight design make it appropriate for applications with limited resources.

Inception V3
Renowned for its distinct inception module, which concurrently processes data at several sizes.
Strengths: Effective use of computational resources combined with high accuracy.

MobileNetV2
a CNN design that is lightweight and tailored for embedded and mobile devices.
Strengths: Quick inference and minimal processing overhead.

Selection of Algorithms

Transfer learning was used to train each model on the PlantVillage dataset. The model with the highest validation accuracy was chosen as the application's mainstay after their performance was assessed. This method makes sure that accuracy and computing efficiency were balanced in the finished system.

4. Specifics of Implementation

Self-Coded Features

The project's components were independently coded as follows:

Preprocessing of Data

All photos were resize to 224x224 pixels in order to comply with the CNN models' input specifications. Rotation, zoom, shear, and flipping data augmentation techniques were used to improve the resilience of the model.

Custom layers

The pre-trained models were supplemented with task-specific dense layers to classify the six leaf types. For best results, the learning rate, batch size, and number of epochs were adjusted.

Assessment of the Model

Validation pipelines were put in place to track model loss and correctness throughout training. Selected the top-performing model automatically.

Tool for Drag-and-Drop Testing

Created a graphical user interface (GUI) based on Tkinter that lets users view forecasts and upload photographs. Combined model inference and real-time image preprocessing. The CNN models (AlexNet50, ResNet50, EfficientNetB0, InceptionV3, and MobileNetV2) used in the research were pre-trained weights and architectures from TensorFlow/Keras. These tools guaranteed reliable feature extraction from the dataset and drastically cut down on training time.

5. Needs for Computational Resources and Effort

The computational capabilities :

GPU-equipped personal computers for training models. Scalability is taken into consideration for cloud-based services.

6. Metrics for Evaluation

Classification Accuracy: The proportion of diseases that are accurately identified. In classification tasks, the F1 Score is used to balance recall and precision. InceptionV3 with the highest accuracy score with 80% accuracy score.

Positive Cases: Clearly visible signs of the illness.

Difficult Cases: Low-resolution photos, leaves that are obscured, or symptoms that overlap.

Criteria for Success

If the algorithm and tool correctly segments sick areas, if applicable, and the model achieves an accuracy of at least 80% on validation data, the project is deemed successful.

7. Expectations for the Project

Plant health diagnostics can be made easy and efficient by incorporating the deep learning model into a drag-and-drop application.

Learning Objectives

Expanded knowledge of cloud-based processing systems, transfer learning, and convolutional neural networks (CNNs) in relation to computer vision and robotics.

8. Additional Guidelines

Examine the shortcomings of particular algorithms, paying particular attention to how well they function in situations with different image quality and illness complexity. Test the model against difficult test cases to ensure resilience.

9. Bibliography:

Links:

https://github.com/spMohanty/PlantVillage-Dataset/tree/master/data_distribution_for_SVM/test/33

Citation:

@article{Mohanty_Hughes_Salathé_2016,
title={Using deep learning for image-based plant disease detection},
volume={7},
DOI={10.3389/fpls.2016.01419},
journal={Frontiers in Plant Science},
author={Mohanty, Sharada P. and Hughes, David P. and Salathé, Marcel},
year={2016},
month={Sep}}