Report: Crop Disease Classification Using Transfer Learning

Course: CS370 Artificial Intelligence

Submitted To: Dr. Seemab Latif

Class: BSCS 11 A

Project Team:

Name	CMS
Muhammad Sunaam	393223
Hasib Aslam	376610
Syed Aon Raza	375700
Ali Shahzad	369802

Project Link:

Github Link

Checkpoint Link

Contents

Introduction

In recent years, agricultural productivity has encountered significant challenges due to the prevalence of crop diseases, posing substantial threats to global food security. Among the crucial crops sustaining communities worldwide—tomatoes, maize, cassava, and cashew—disease outbreaks have emerged as a formidable adversary, jeopardizing yields and livelihoods.

This report unveils a groundbreaking initiative aimed at mitigating this pressing issue through the implementation of an innovative Crop Disease Classifier. Leveraging cutting-edge technology in machine learning and computer vision, this classifier is designed to revolutionize disease detection and management within these vital crops.

By harnessing the power of advanced algorithms and extensive datasets encompassing various crop diseases, this classifier serves as a robust tool for early detection and precise identification of ailments affecting tomatoes, maize, cassava, and cashew plants. Its accuracy and efficiency offer farmers a proactive approach, enabling timely interventions to curtail the spread of diseases and minimize crop losses.

Problem Statement

The task at hand is to correctly classify diseased crops early on with high accuracy, so that farmers can employ preventive measures to increase yield and growth.

Methodology

Data Gathering

Dataset: Dataset for Crop Pest and Disease Detection, Mendeley Data, V1, doi: 10.17632/bwh3zbpkpv.1

Link: https://data.mendeley.com/datasets/bwh3zbpkpv/1

Our dataset comprises of 25,000 raw images of plant leaves encompassing four major crops of interest—Cashew, Cassava, Maize, and Tomato. The dataset consists of a total of twenty-two classes as follows:

- Cashew has 5 classes: anthracnose, gummosis, healthy, leaf miner, and red rust.

- Cassava has 5 classes: bacterial blight, brown spot, green mite, healthy, and mosaic.
- Maize has 7 classes: fall armyworm, grasshopper, healthy, leaf beetle, leaf blight, leaf spot, and streak virus.
- Tomato has 5 classes: healthy, leaf blight, leaf curl, septoria leaf spot, and verticillium wilt.

Model Architecture

We used a combination of **YOLOv5** and **ResNet50** in the scope of this project.

YOLOv5

YOLOv5 is an advanced object detection algorithm known for its speed and accuracy. It's part of the "You Only Look Once" (YOLO) family of models, utilizing deep learning techniques to detect and classify objects in images in real-time. YOLOv5 improves upon its predecessors by optimizing model architecture and training strategies, achieving impressive performance while being highly efficient in terms of computational resources. This model has gained popularity in computer vision tasks due to its balance between accuracy and speed, making it a go-to choose for object detection applications.

ResNet50

ResNet-50 is a convolutional neural network architecture that's part of the ResNet (Residual Network) family. It's characterized by its depth and use of residual connections, which help address the vanishing gradient problem during training of very deep neural networks. ResNet-50 specifically consists of 50 layers and employs residual blocks to enable the training of deeper networks more effectively. It has been widely used in various computer vision tasks, showcasing strong performance and accuracy in image classification and feature extraction. The last fully connected was changed to have 22 neurons in accordance with our dataset.

Approach

YOLOv5 was employed in our architecture to detect and crop leaf images. In a real-world scenario, images usually contain backgrounds such as a human hand. Practically, images fed to our application will never be perfectly cropped out. As our **ResNet** model is trained on perfectly cropped images of leaves it would fail. So, **YOLOv5** works around this problem by detecting leaves in an image. We then modified its source code to return cropped images instead of creating bounding boxes.

The copped images generated by the **YOLOv5** model are then fed to the **ResNet50** model. The **ResNet50** model classifies the leaf image into the appropriate class and returns the class name and confidence level.

Training Approach

The training process involved fine-tuning all layers of the ResNet-50 model on Google Colab's V100 GPU. Initially, the model was initialized with an initial learning rate of 0.01 and trained using the Stochastic Gradient Descent (SGD) optimizer. The chosen batch size for the initial training was set to 32.

To optimize model performance, a systematic exploration of different batch sizes and learning rates was conducted. Various combinations were tested, evaluating their impact on convergence, training stability, and loss fluctuations. After iterative experimentation and rigorous evaluation, the training process converged most effectively when a final learning rate of 0.0000001 and a final batch size of 128 were adopted.

The adjustment in learning rate was particularly pivotal in stabilizing the training process, mitigating loss fluctuations, and ensuring smoother convergence towards an optimal model state.

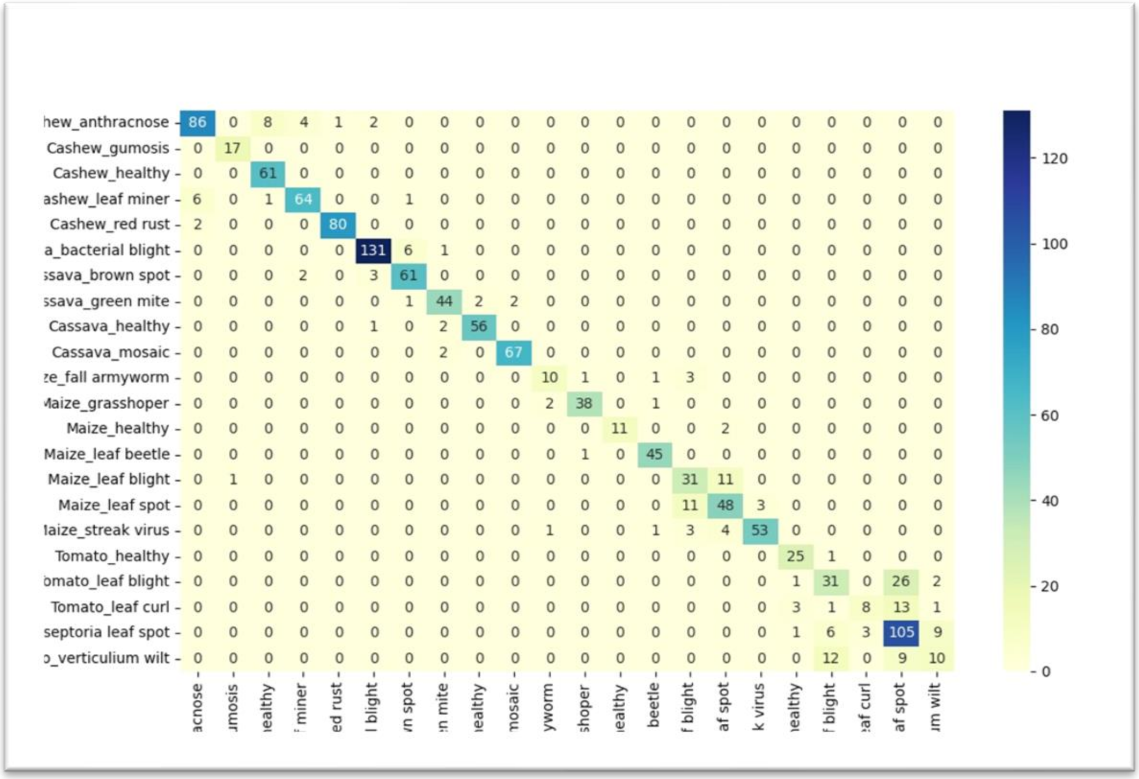
Throughout the training phase, a train-test split ratio of 0.90 was maintained, allocating 90% of the dataset for training and the remaining 10% for validation. This split facilitated robust model validation and rigorous evaluation of model performance on unseen data.

Results and Analysis

The model achieved an accuracy of 87.5% on the test dataset.

Confusion Matrix

A confusion matrix is a table used in machine learning to show the model's performance in classifying data. It summarizes the number of correct and incorrect predictions made by the model compared to the actual outcomes. It helps visualize the model's accuracy, highlighting where it correctly predicts classes and where it tends to make errors, such as false positives and false negatives. This matrix is particularly useful for understanding the model's behavior across different classes in classification tasks.



- **Diagonal Values**: These represent the number of correct classifications for each class. For instance, along the diagonal, higher values indicate accurate predictions. For instance, "Cashew_healthy" has 64 correct predictions, "Cashew_red rust" has 94 correct predictions, and so on.

- **Off-diagonal Values**: These values showcase misclassifications. For example, "Cashew_anthracnose" has 6 instances misclassified as "Cashew_leaf miner," and "Cashew_leaf miner" has 5 instances misclassified as "Cashew_anthracnose."

- **Class Imbalance**: Some classes have more instances than others. For instance, "Cassava_bacterial blight" has a higher number of instances compared to other classes, which might affect the model's ability to distinguish between less frequent classes.

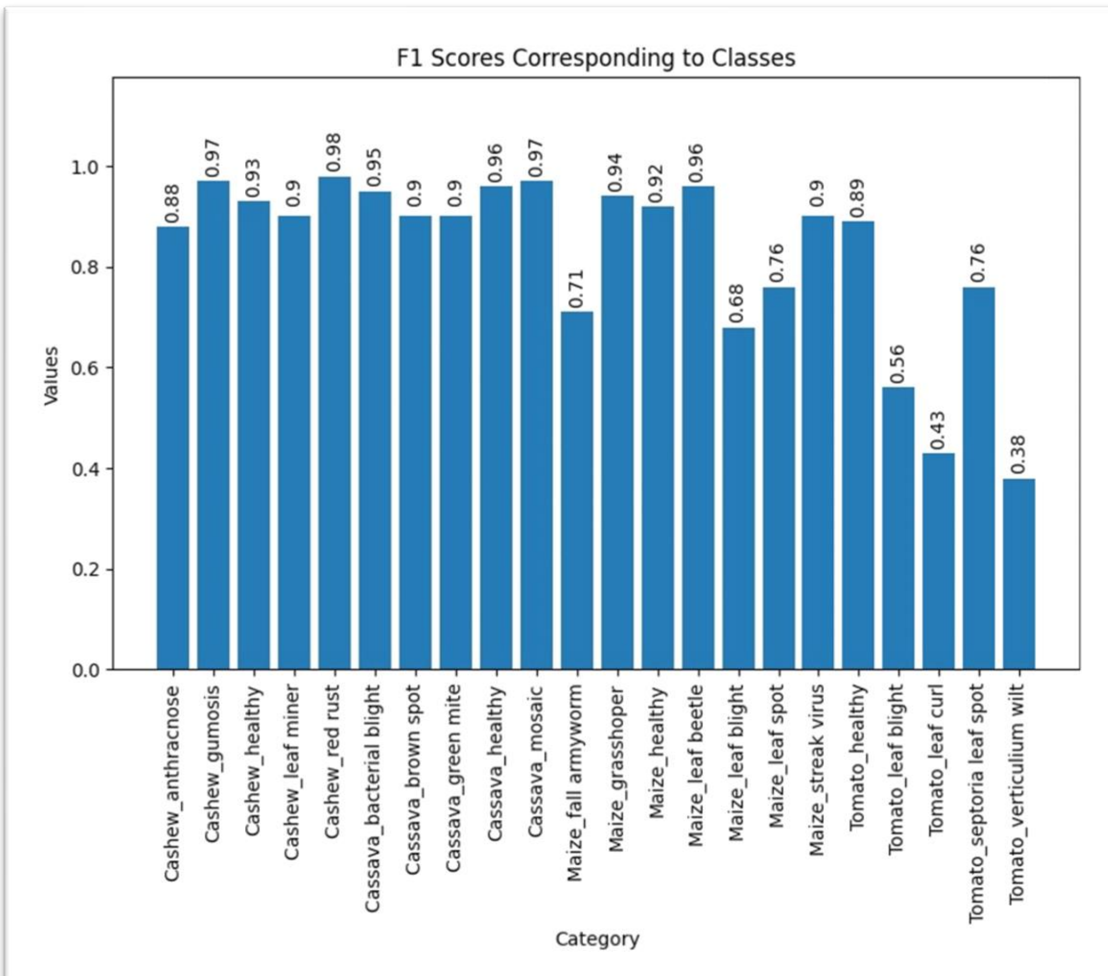
- **Zero Values**: Some classes have zero misclassifications with other classes, indicating the model's robustness in distinguishing them.

- **Patterns**: Patterns within the matrix can indicate specific challenges. For example, confusion might exist between classes that share visual similarities, leading to misclassifications.

F1 Score

The F1 score is a single metric that combines precision and recall into a unified measure of a model's performance in classification tasks. It considers both false positives and false negatives.

It's calculated as the harmonic mean of precision and recall, providing a balanced assessment of a model's accuracy. A high F1 score indicates a model that has both good precision (low false positive rate) and good recall (low false negative rate), making it a useful metric for evaluating classifiers, especially when there's an imbalance between classes in the dataset.



- **High F1 Scores (Above 0.9)**: Classes like "Cashew_red rust," "Cassava_healthy," and "Cassava_mosaic" exhibit exceptional F1 scores, indicating strong performance in both precision and recall. These classes are well-classified by the model.
- **Moderate F1 Scores (Between 0.7 and 0.89)**: Several classes, such as "Cashew_anthracnose," "Cashew_healthy," "Cashew_leaf miner," "Cassava_bacterial blight," "Cassava_brown spot," "Cassava_green mite," "Maize_grasshopper," and "Tomato_healthy," show moderate but reliable F1 scores. The model performs reasonably well in these classes, although there might be room for improvement.
- **Low F1 Scores (Below 0.7)**: Some classes, like "Maize_fall armyworm," "Maize_healthy," "Maize_leaf blight," "Maize_leaf spot," "Tomato_leaf blight," "Tomato_leaf curl," and "Tomato_septoria leaf spot," exhibit lower F1 scores. These classes might pose challenges for the model, indicating potential misclassifications or imbalances in precision and recall.
- **Very Low F1 Score (0.27)**: "Tomato_verticulium wilt" stands out with a notably low F1 score. This class might be particularly challenging for the model, showing significant room for improvement in both precision and recall.

References

1. Smith, A., "Detection and Classification of Plant Diseases: A Review," International Journal of Advanced Research in Computer Engineering & Technology, vol. 5, no. 2, 2016.
2. Reddy, A., et al., "Deep Learning Techniques for Agricultural Disease Detection," Proceedings of the IEEE International Conference on Big Data, 2018.
3. He, K., et al., "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
4. Bochkovskiy, A., et al., "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

5. Redmon, J., et al., "YOLOv3: An Incremental Improvement," arXiv preprint arXiv:1804.02767, 2018.
6. Google Colaboratory: https://colab.research.google.com/
7. Python Software Foundation. Python Language Reference, version 3.8. Available at https://www.python.org/.