Plant Disease Detection

Youssef Nakhla 900201430

Charbel El-Fakhry 900214262

THE AMERICAN UNIVERSITY IN CAIRO

Problem Statement

Egypt relies heavily on agriculture as a key sector supporting its economy and food security, with a large portion of the population engaged in farming activities. The aim of this project is to develop a machine learning algorithm using deep learning architectures that can accurately recognize and classify diseases affecting apple leaves to help support farmers by enabling early detection, reducing crop losses.

Dataset

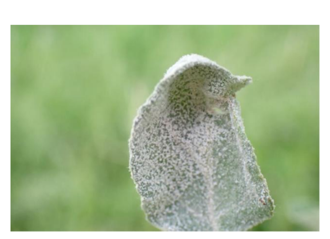
This dataset contains 1530 images of apple tree leaves. Moreover, the data is labeled under 3 different labels: Healthy, Powdery, and Rust. It is already split into training, testing, and validation subsets. The class distributions are:

Subset	Healthy	Powdery	Rust
Training	458	430	434
Testing	50	50	50
Validation	20	20	20

Sample Input

Healthy

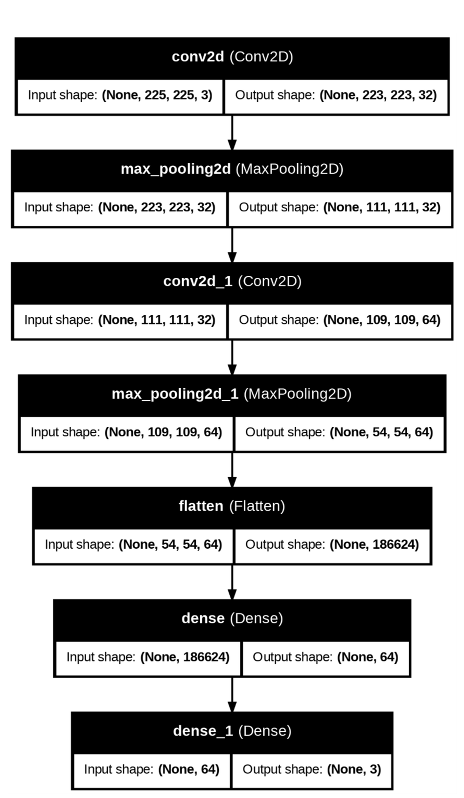
Powdery



Rust

Baseline Model

CNNs are a standard and straightforward choice for image classification tasks, making them easy to implement and interpret as a starting point. With only 1530 images, a basic CNN can perform reasonably well without overfitting, especially with basic preprocessing.



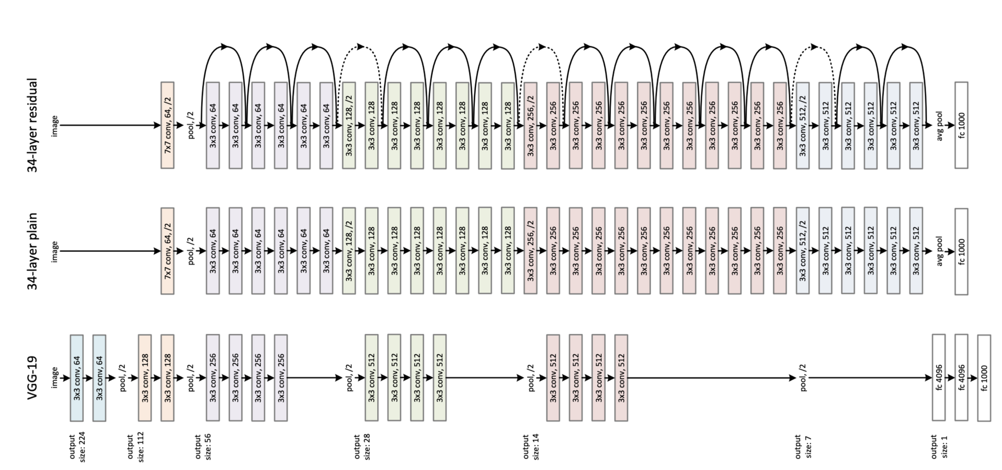
Initial Updates

The Baseline model was deployed, and a more complex GPU powered CNN structure was deployed yet it was quite clear that this model was very simple even though it achieved very promising results, yet we decided to explore more complex architectures. The results of the initial updates can be seen below.

Models

Multiple models were tested including YOLO, CNNs, ResNet-34, RNNs, and more. The best two performing models aside from the CNNs were Inception and ResNet-34, they will be described below:

ResNet-34

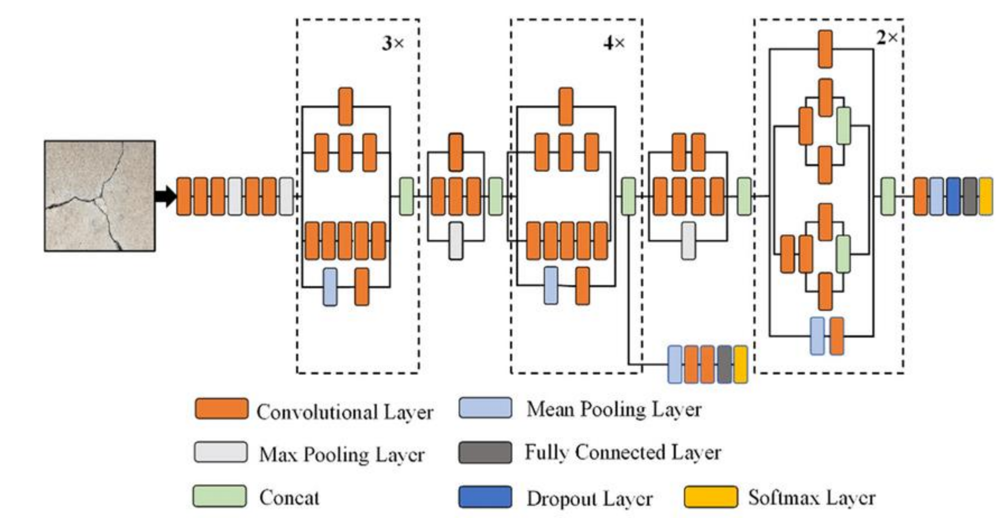


The ResNet-34 model was chosen for its high accuracy in classifying similar dataset images. Its skip connections prevent the vanishing gradients, crucial for multiclass classification.

The base model architecture is:

- 33 convolutional layers
- 1 max pooling layer
- 1 fully connected layer

InceptionV3



The InceptionV3 model was selected for its superior performance in handling diverse datasets and its ability to efficiently capture spatial hierarchies in images through its unique architecture. The model's use of factorized convolutions and inception modules ensures computational efficiency, making it well-suited for multiclass classification tasks.

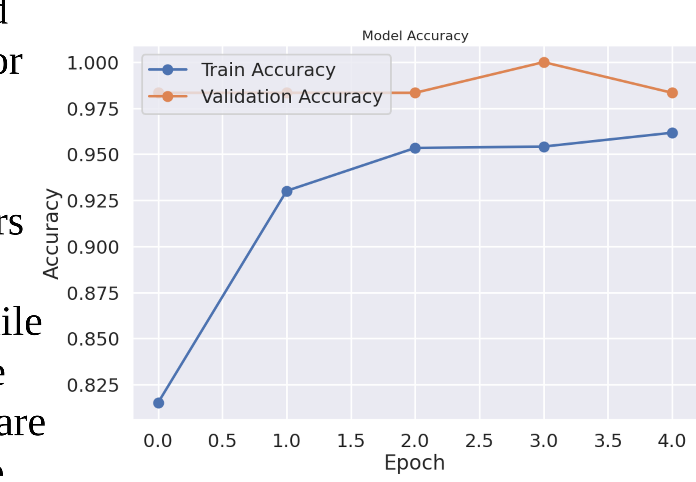
The base model architecture includes a total of 48 layers:

- 11 inception modules
- 2 convolutional layers
- 1 max pooling layer
- 1 average pooling layer

Model Updates

InceptionV3

- Pre-trained InceptionV3 is used as a feature extractor (with the top layer removed).
- Freezing early layers helps retain the learned features while only fine-tuning the later layers (which are more specific to the task at hand).



- Custom classifier with: GlobalAveragePooling2D instead of a fully connected layer for dimensionality reduction. Two dense layers (1024 and 512 units) to increase model capacity.
- Dropout layers (50% and 30%) to prevent overfitting.
- Softmax output layer with 3 neurons for multiclass classification.

ResNet-34

- Down sampling is handled explicitly in the Residual Block with a 1x1 convolution when the input and output dimensions mismatch.
- Custom number of output classes: The model has a flexible number of output classes, based on the dataset.
- Use of Adaptive Average Pooling: This avoids excessive parameters from fully connected layers.

Preprocessing & Hyperparameter Updates

We were relentless to achieve the best results possible, so revisited our preprocessing steps starting form the scaling of the images to data augmentation. The images are rescaled by dividing each pixel value by 255, which normalizes the pixel intensity values to the range [0, 1]. We expanded the size of the training dataset by generating transformed versions of the original images. Furthermore, the following hyperparameters were set:

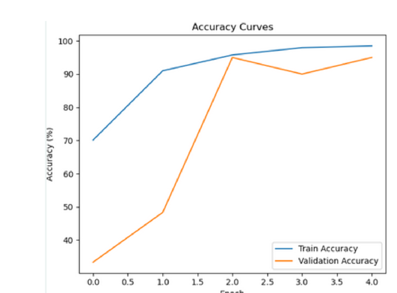
- Dropout Regularization
- Learning rate: 0.0001
- Optimizer: Adam (0.001)

The two models, ResNet-34 and GPU CNN were retrained with these updates and the results can be observed below.

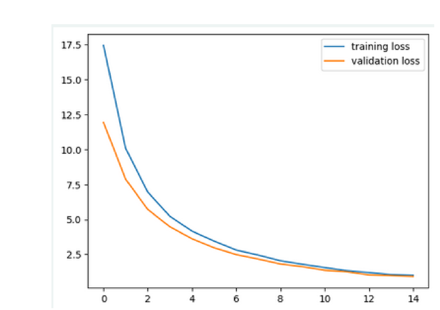
Results

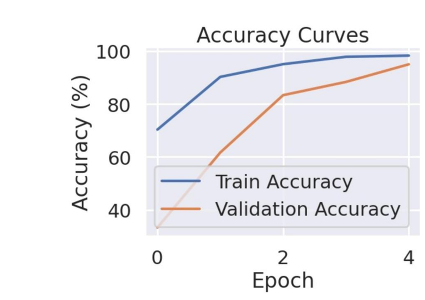
GPU CNN | ResNet-34

Before Updates



After Updates





As can be observed in the graphs in the previous section, the InceptionV3 model was extremely underfitting, yet the VNN and the ResNet-34 were a little bit overfitting, but we were able to improve their performance and reduce the overfitting. Hence, our best performing model was in fact the GPU powered CNN, even though it holds the simplest architecture, it is the most efficient in terms of balancing time complexity and accuracy.

Future Work

- Covering more types of apple leaves diseases
- Expanding the dataset to include more plant types that are crucial to the Egyptian Economy.
- Deploy a real-time application to help farmers test their crops and get accurate analysis to treat the plants.

References

- Baseline Model: https://www.kaggle.com/code/chanchal24/plant-disease-recognition-using-dl/notebook
- Dataset: https://www.kaggle.com/datasets/rashikrahmanpritom/plant-disease-recognition-dataset