## Table of Contents

# Project Report

*Topic – Diagnose plant disease solely based on leaf images*

*Name – Sharik Sharif Baig*

## Project Summary

The project focuses to diagnose plant diseases solely based on leaf images. The categories include "healthy", "scab", "rust", and "multiple diseases". Solving this problem is important because diagnosing plant diseases early can save tones of agricultural produce every year. This will benefit not only the general population by reducing hunger, but also the farmers by ensuring they get the harvest they deserve.

Scab - Scab is defined as "any of various plant diseases caused by fungi or bacteria and resulting in crustlike spots on fruit, leaves, or roots. The spots caused by such a disease". The brown marks across the leaf are a sign of these bacterial/fungal infections. Once diagnosed, scab can be treated using chemical or non-chemical methods.

Rust - Rust is defined as "a disease, especially of cereals and other grasses, characterized by rust-colored pustules of spores on the affected leaf blades and sheaths and caused by any of several rust fungi". The yellow spots are a sign of infection by a special type of fungi called "rust fungi". Rust can also be treated with several chemical and non-chemical methods once diagnosed.

In this kernel, I will visualize the data with Matplotlib and Plotly and then demonstrate some important image processing and augmentation techniques using OpenCV. Finally, I will show how different pretrained Keras models, such as DenseNet and EfficientNet, can be used to solve the problem.

## Problem

### Scabs on the leaves

Scab is defined as "any of various plant diseases caused by fungi or bacteria and resulting in crustlike spots on fruit, leaves, or roots. The spots caused by such a disease". The brown marks across the leaf are a sign of these bacterial/fungal infections. Once diagnosed, scab can be treated using chemical or non-chemical methods.

### Rust on the leaves

Rust is defined as "a disease, especially of cereals and other grasses, characterized by rust-colored pustules of spores on the affected leaf blades and sheaths and caused by any of several rust fungi". The yellow spots are a sign of infection by a special type of fungi called "rust fungi". Rust can also be treated with several chemical and non-chemical methods once diagnosed.

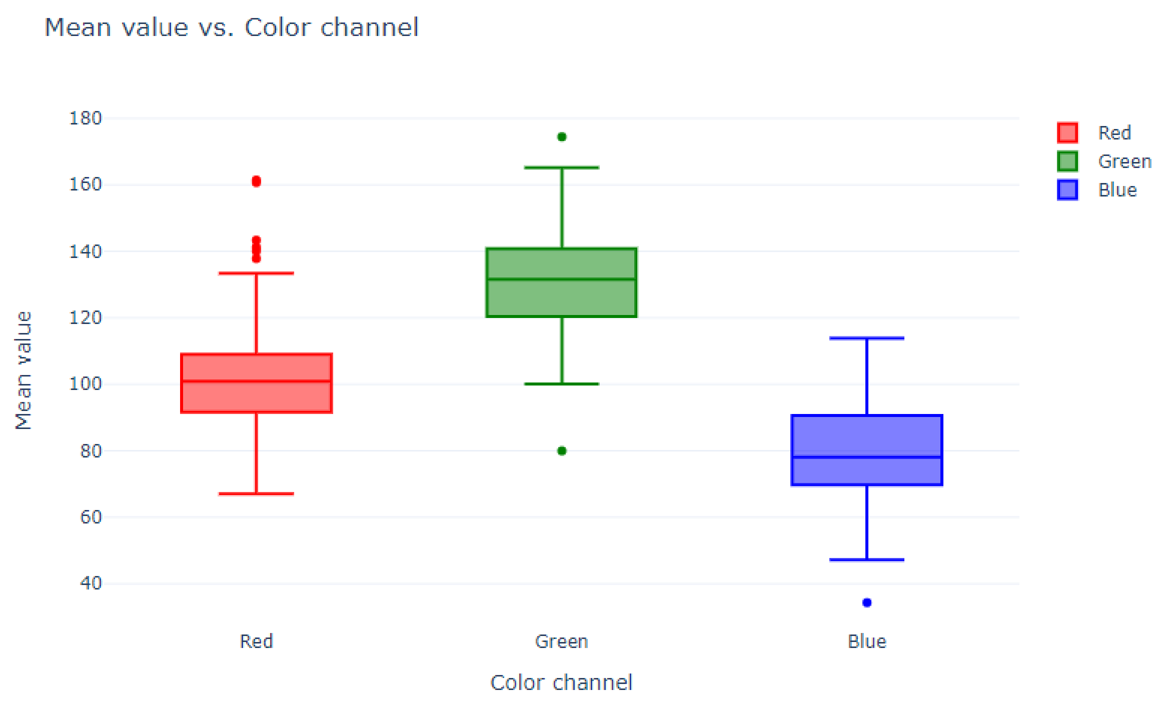
### Multiple diseases on the leaves

If a leaf is identified with more than one type of the problem mentioned above(Scabs or Rust) then it is identified that plant have multiple diseases.

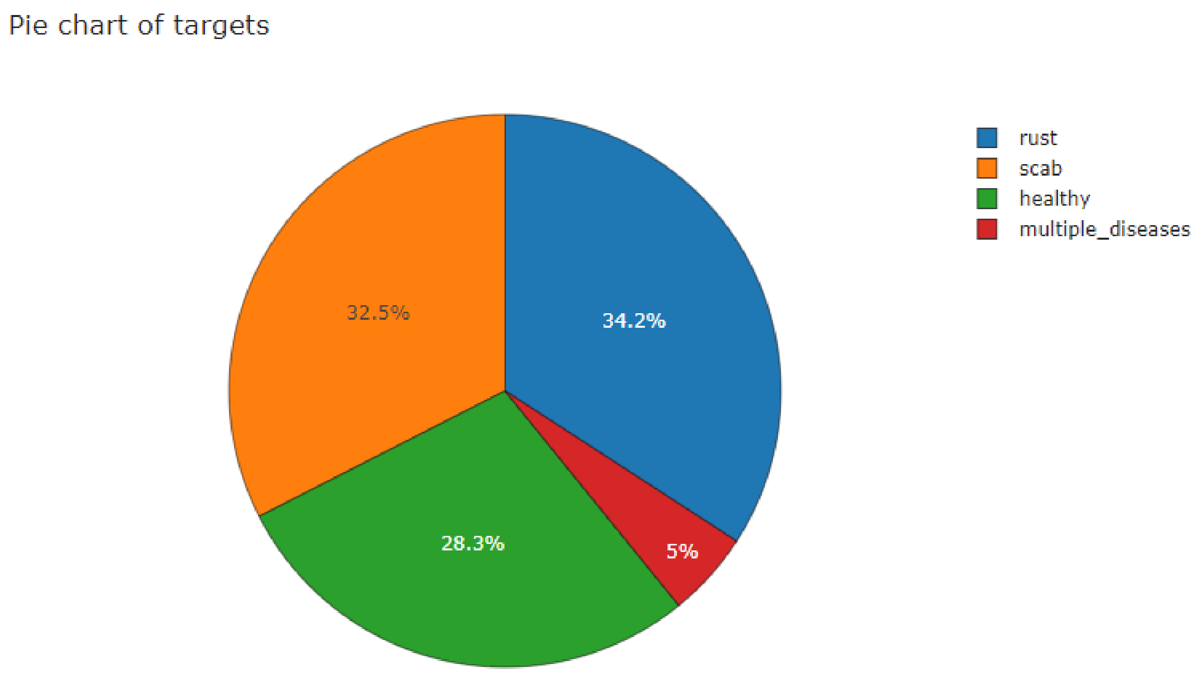
## Techniques used in the project

### Visualizing a sample leaf with the RGB Values.

The green parts of the image have very low blue values, but by contrast, the brown parts have high blue values. This suggests that green (healthy) parts of the image have low blue values, whereas unhealthy parts are more likely to have high blue values. This might suggest that blue channel may be the key to detecting diseases in plant.



## Distribution of the images classified as healthy and unhealthy.



In the pie chart above, we can see that most leaves in the dataset are unhealthy (71.7%). Only 5% of plants have multiple diseases, and "rust" and "scab" occupy approximately one-third of the pie each.

## Image processing and Augmentation

1. Canny Edge Detection.

Canny is a popular edge detection algorithm, and as the name suggests, it detects the edges of objects present in an image. It was developed by John F. Canny in 1986. The algorithm involves several steps.

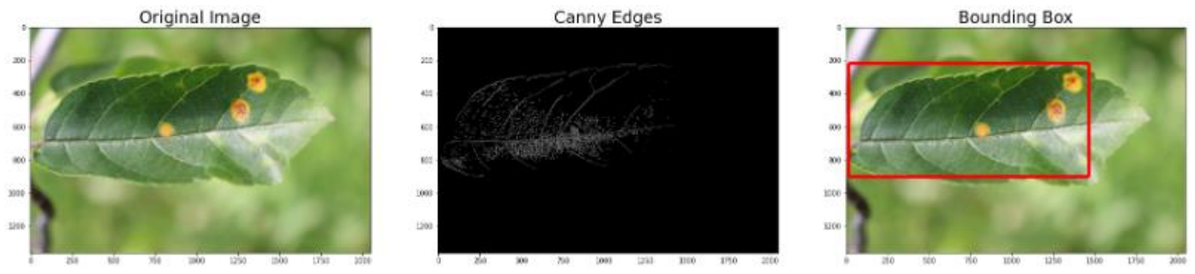
Noise reduction: Since edge detection is susceptible to noise in an image, we remove the noise in the image using a 5x5 Gaussian filter.

Finding Intensity Gradient of the Image: The smoothened image is then filtered with a Sobel kernel in both horizontal and vertical directions to get the first derivative in the horizontal ($G_x$) and vertical ($G_y$) directions. From these two images, one can find the edge gradient and direction for each pixel:

Rounding: The gradient is always perpendicular to edges. So, it is rounded to one of the four angles representing vertical, horizontal and two diagonal directions.

Non-maximum suppression: After getting the gradient magnitude and direction, a full scan of the image is done to remove any unwanted pixels which may not constitute the edge. For this, we check every pixel for being a local maximum in its neighborhood in the direction of the gradient.

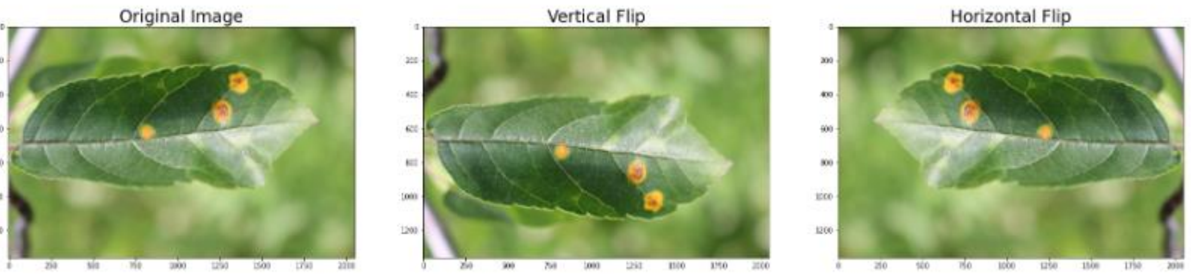
Hysteresis Thresholding: This stage decides which parts are edges and which are not. For this, we need two threshold values, *minVal* and *maxVal*. Any edges with intensity gradient greater than *maxVal* are considered edges and those lesser than *minVal* are considered non-edges, and discarded. Those who lie between these two thresholds are classified edges or non-edges based on their neighborhood. If they are near “sure-edge” pixels, they are considered edges, and otherwise, they are discarded.



The second column of images above contains the Canny edges and the third column contains cropped images. I have taken the Canny edges and used it to predict a bounding box in which the actual leaf is contained. The most extreme edges at the four corners of the image are the vertices of the bounding box. This red box is likely to contain most of if not all of the leaf. These edges and bounding boxes can be used to build more accurate models.

2. Flipping

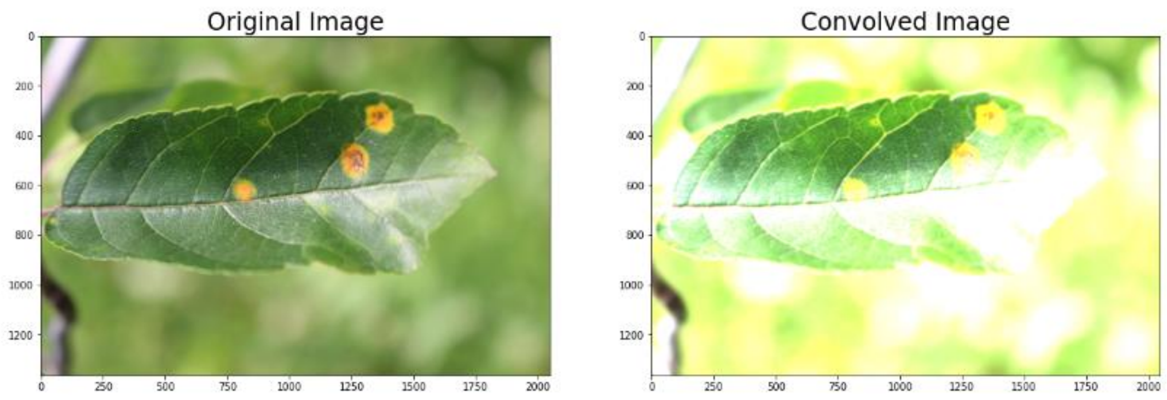
Flipping is a simple transformation that involves index-switching on the image channels. In vertical flipping, the order of rows is exchanged, whereas in vertical flipping, the order of rows is exchanged. Let us assume that $A_{ijk}$ (of size *(m, n, 3)*) is the image we want to flip.



We can see that the images are simply flipped. All major features in the image remain the same, but to a computer algorithm, the flipped images look completely different. These transformations can be used for data augmentation, making models more robust and accurate.

3. Convolution

Convolution is a rather simple algorithm which involves a kernel (a 2D matrix) which moves over the entire image, calculating dot products with each window along the way.



The convolution operator seems to have an apparent "sunshine" effect of the images. This may also serve the purpose of augmenting the data, thus helping to build more robust and accurate models.

4. Blurring

Blurring is simply the addition of noise to the image, resulting in a less-clear image. The noise can be sampled from any distribution of choice, as long as the main content in the image does not become invisible. Only the minor details get obfuscated due to blurring.

The transformation clearly blurs the image by removing detailed, low-level features, while retaining the major, high-level features. This is once again a great way to augment images and train more robust models.

## Modeling

1. MaxPool

Max pooling is very similar to convolution, except it involves finding the maximum value in a window instead of finding the dot product of the window with a kernel. Max pooling does not require a kernel and it is very useful in reducing the dimensionality of convolutional feature maps in CNNs.

2. ReLU

ReLU is an activation function commonly used in neural network architectures. *ReLU(x)* returns 0 for $x < 0$ and $x$ otherwise. This function helps introducenon-linearity in the neural network, thus increasing its capacity ot model the image data.

As mentioned earlier, this function is non-linear and helps increase the modeling capacity of the CNN models.

3. DenseNet

Densely Connected Convolutional Networks (DenseNets), are a popular CNN-based ImageNet used for a variety of applications, inclusing classification, segmentation, localization, etc. Most models before DenseNet relied solely on network depth for representational power. Instead of drawing representational power from extremely deep or wide architectures, DenseNets exploit the potential of the network through feature reuse. This was the main motivation behind the DenseNet architecture.

4. EfficientNet

EfficientNet is another popular (more recent) CNN-based ImageNet model which achieved the SOTA on several image-based tasks in 2019. EfficientNet performs model scaling in an innovative way to achieve excellent accuracy with significantly fewer parameters. It achieves the same if not greater accuracy than ResNet and DenseNet with a mcuh shallower architecture.

5. EfficientNet Noisy Student

EfficientNet NoisyStudent, released in 2020, is based on EfficientNet and uses semi-supervised learning on noisy images to learn rich visual representation. It outperformed EfficientNet on several tasks and is the SOTA at the time of writing (March 2020).

## Conclusions

1. Image processing and augmentation methods such as edge detection, depth estimation, flipping, etc can be used to build models.
2. Several pretrained models like DenseNet and EfficientNet can be used to classify leaf diseases with high accuracy.
3. Ensembling, stacking, and strong validation techniques may lead to more accurate and robust models.

## Contribution

1. By using more than one image processing techniques in the model, I made the model more robust and independent.
2. By adding Canny Image method, I identified that it can be greatly used to identity the problems in the plants to identify multiple diseases.
3. I used multiple factors to identify the leaf and that includes RGB Values, Flipping, and Blurred Images.

4. I used 3 different approaches to solve the problem and those 3 different approaches were independent of each other.
5. DenseNet gave the accuracy of 94.5%, EfficientNet gave the accuracy of 95.6% and the accuracy of EfficientNet NoisyStudent was 89.4%
6. The future approach is to make an Ensemble model combining DenseNet and EfficientNet since they both yielded the highest accuracy out of the three.

## Appendix

Reference - https://www.kaggle.com/code/shellyneira/plant-pathology-2020-eda-models/notebook

Source Code - https://github.com/SharikBaig/Advanced_Machine_Learning/tree/main/Final%20Project

PPT-
https://docs.google.com/presentation/d/1ioHLR3woSeHvhEBCzkKY7wWzOfHuhs2P/edit?usp=sharing&ouid=115103250169495049561&rtpof=true&sd=true