December 02, 2021

# Sprint project - Image Analytics

Fall 2021

Abhishek Kumar Bhagat
Panther Id: 002571274

# Introduction:

Apples are one of the most important temperate fruit crops in the world. Foliar (leaf) diseases pose a major threat to the overall productivity and quality of apple orchards. The current process for disease diagnosis in apple orchards is based on manual scouting by humans, which is time-consuming and expensive.

# Objective:

The main objective of the project is to develop machine learning-based models to accurately classify a given leaf image from the test dataset to a particular disease category, and to identify an individual disease from multiple disease symptoms on a single leaf image. The work focuses on, Neural Network such as CNN with image processing methods. CNN algorithm achieved 75 percent accuracy with 30 epochs.

# Libraries Used:

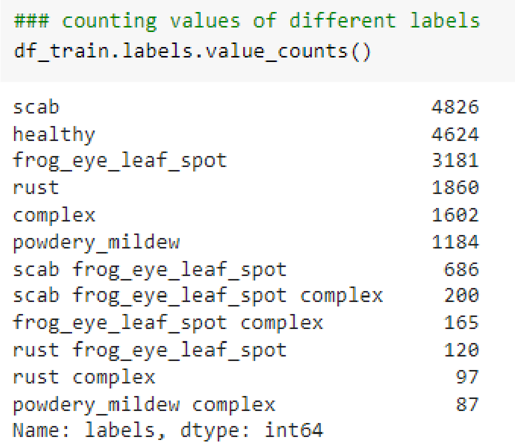
```
##### 1
import numpy as np
import pandas as pd
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
import cv2
import os
import warnings
warnings.filterwarnings('ignore')
import tensorflow as tf
import random
import albumentations as A
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.layers import Dense,Activation,Flatten, Conv2D, MaxPooli
ng2D
from tensorflow.keras.models import Sequential
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
```

# Data Preprocessing

As we deal with this massive amount of data to use for deep learning, we find different ways in which we can enrich this data so we can eventually train, validate, and hyper tune our Convolution Neural Network.

## Dataset:

- train.csv contains information about the image files available in train_images. It contains 18632 rows(images) with 2 columns i.e (image , labels ).
- test.csv The test set images. This competition has a hidden test set: only three images are provided here as samples while the remaining 5,000 images will be available to your notebook once it is submitted.
- https://www.kaggle.com/yashvi/classify-diseases-in-apple-trees-beginner/data

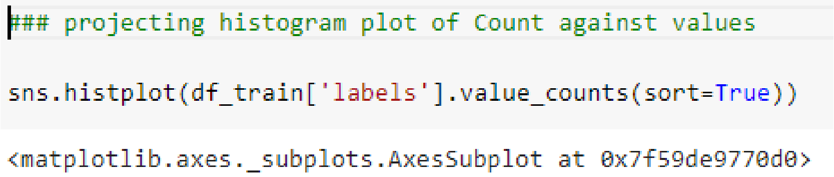


```
### counting values of different labels
df_train.labels.value_counts()
```

```
scab                                4826
healthy                             4624
frog_eye_leaf_spot                  3181
rust                                1860
complex                             1602
powdery_mildew                      1184
scab frog_eye_leaf_spot              686
scab frog_eye_leaf_spot complex      200
frog_eye_leaf_spot complex           165
rust frog_eye_leaf_spot              120
rust complex                          97
powdery_mildew complex                87
Name: labels, dtype: int64
```

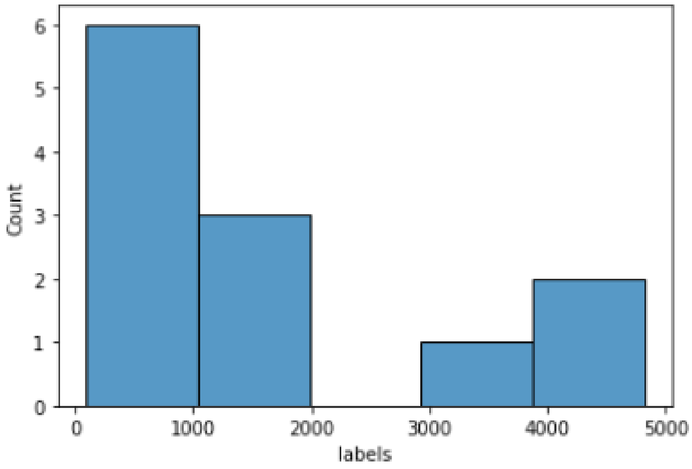
## Batch Visualization of Images:

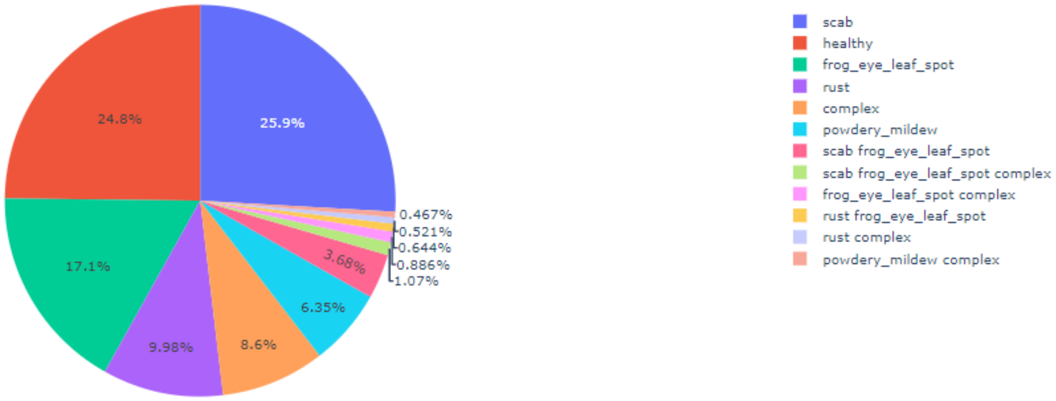
We have analyzed Labels dataset using various plots.



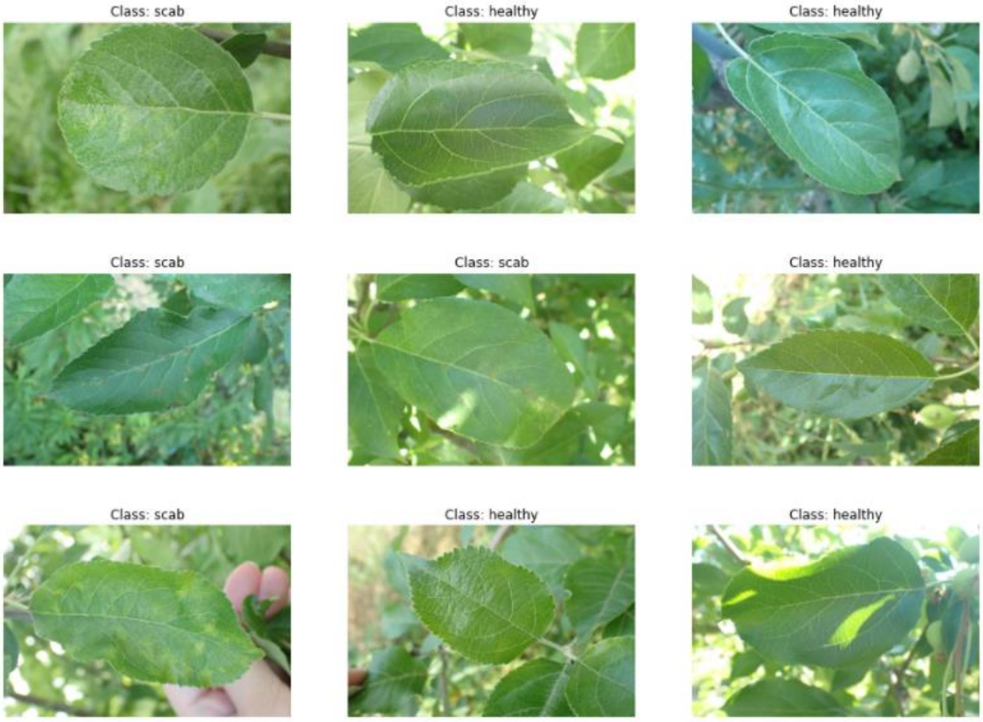
```
### projecting histogram plot of Count against values

sns.histplot(df_train['labels'].value_counts(sort=True))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59de9770d0>
```





## Batch visualization with labels:



# Feature Engineering:

We used Keras TensorFlow to optimizes image dataset. We used ImageDataGenerator and flow_from_dataframe functions to optimize image dataset.

```
HEIGHT = 128
WIDTH=128
SEED = 45
BATCH_SIZE= 64
```

Refer TensorFlow docs for more information here.

# Challenges we faced:

Getting good accuracy was difficult therefore, below are the hyper tunning we performed:
We used
Softmax activation function in Convolution Neural Network.
Increased epoch from 5 to 30 to get better accuracy.

```
learning_rate=0.001
```

**Softmax Activation Function:**
The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1.

# Modeling:

This project deals with Image data and is essentially a classification problem. The goal here is to train models to accurately classify a given leaf image from the test dataset to a particular disease category, and to identify an individual disease from multiple disease symptoms on a single leaf image.

**Convolutional neural network (CNN):**

A convolutional neural network, or CNN, is a deep learning neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification and have also found success in natural language processing for text classification.

A convolutional neural network is a feed-forward neural network, often with up to 20 or 30 layers. The power of a convolutional neural network comes from a special kind of layer called the convolutional layer.

Types of convolutional neural networks

- AlexNet
- VGGNet
- GoogLeNet
- ResNet

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs. They have three main types of layers, which are:

- Convolutional layer
- Pooling layer
- Fully connected (FC) layer

Below is the architecture diagram of Process:

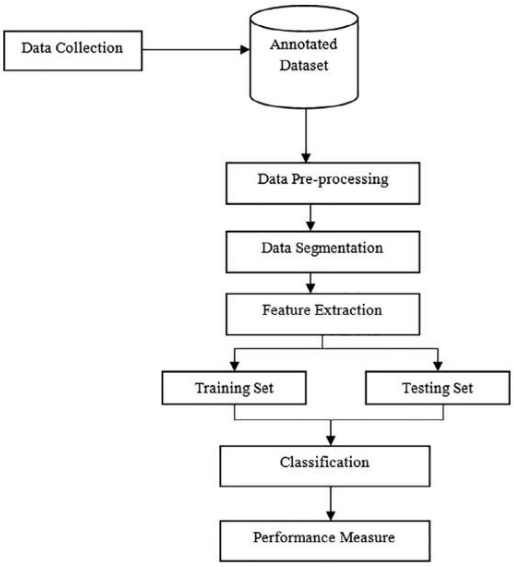
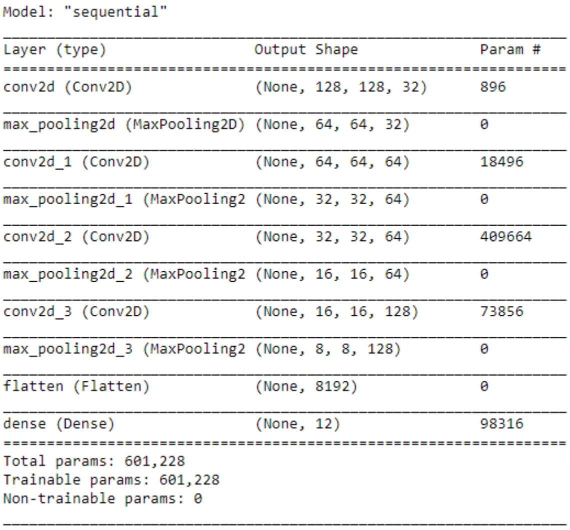


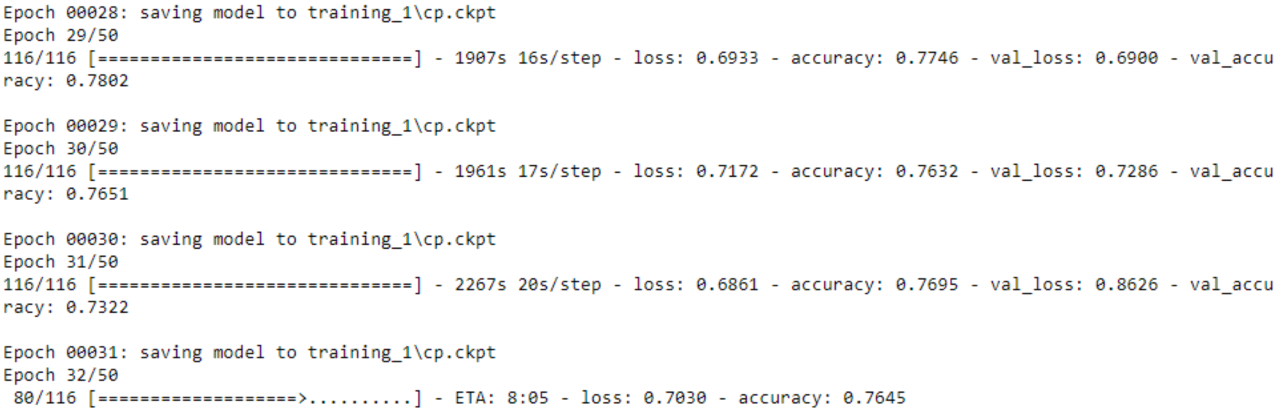
**Fig. 1.** Plant disease classification system.

The training dataset is split in batch of 64 images with 30 epochs. Convolution Neural Network using Sequential model is used to train the model. Below are the details about the model build and train:



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 128, 32)      896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 64, 64, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        409664
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 8, 8, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
dense (Dense)                (None, 12)                98316
=================================================================
Total params: 601,228
Trainable params: 601,228
Non-trainable params: 0
_________________________________________________________________
```

Model Training details:



```
Epoch 00028: saving model to training_1\cp.ckpt
Epoch 29/50
116/116 [==============================] - 1907s 16s/step - loss: 0.6933 - accuracy: 0.7746 - val_loss: 0.6900 - val_accu
racy: 0.7802

Epoch 00029: saving model to training_1\cp.ckpt
Epoch 30/50
116/116 [==============================] - 1961s 17s/step - loss: 0.7172 - accuracy: 0.7632 - val_loss: 0.7286 - val_accu
racy: 0.7651

Epoch 00030: saving model to training_1\cp.ckpt
Epoch 31/50
116/116 [==============================] - 2267s 20s/step - loss: 0.6861 - accuracy: 0.7695 - val_loss: 0.8626 - val_accu
racy: 0.7322

Epoch 00031: saving model to training_1\cp.ckpt
Epoch 32/50
 80/116 [===================>..........] - ETA: 8:05 - loss: 0.7030 - accuracy: 0.7645
```

# Jupyter Notebook:

Classify diseases in apple trees..ipynb

**Presentation deck:**

Week8_Presentation_15Nov.pptx

# References:

https://doi.org/10.1016/j.matpr.2021.07.358

https://www.kaggle.com/yashvi/classify-diseases-in-apple-trees-beginner/data

https://www.ibm.com/cloud/learn/convolutional-neural-networks

https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator

https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

**The End**