Importing Packages:

In [1]:

```
import numpy as np
from keras.preprocessing.image import img_to_array
import matplotlib.pyplot as plt
import pandas as pd
import shutil
import os
from tqdm import tqdm

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from PIL import Image
```

In [2]:

```
import numpy as np
import pickle
import cv2
from os import listdir
import tensorflow as tf
from sklearn.preprocessing import LabelBinarizer
import keras
from keras.models import Sequential
from keras.layers import BatchNormalization

from tensorflow.keras.layers import GlobalAveragePooling2D
from keras.preprocessing.image import ImageDataGenerator
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import optimizers
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
from keras.callbacks import ModelCheckpoint
import matplotlib.pyplot as plt
```

In [3]:

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

For training and test dataset I have clicked pictures of various plant leaves from nearby villages and using these images I will be training a model which will later predict whether a plant is healthy or not by using an image of a leaf as input.

Exploratory Data Ananlysis:

In [ ]:

```
train_wd= os.listdir('/content/gdrive/MyDrive/plant_disease/self_collected_da
train_woutd= os.listdir('/content/gdrive/MyDrive/plant_disease/self_collectec
val_wd= os.listdir('/content/gdrive/MyDrive/plant_disease/self_collected_data
```

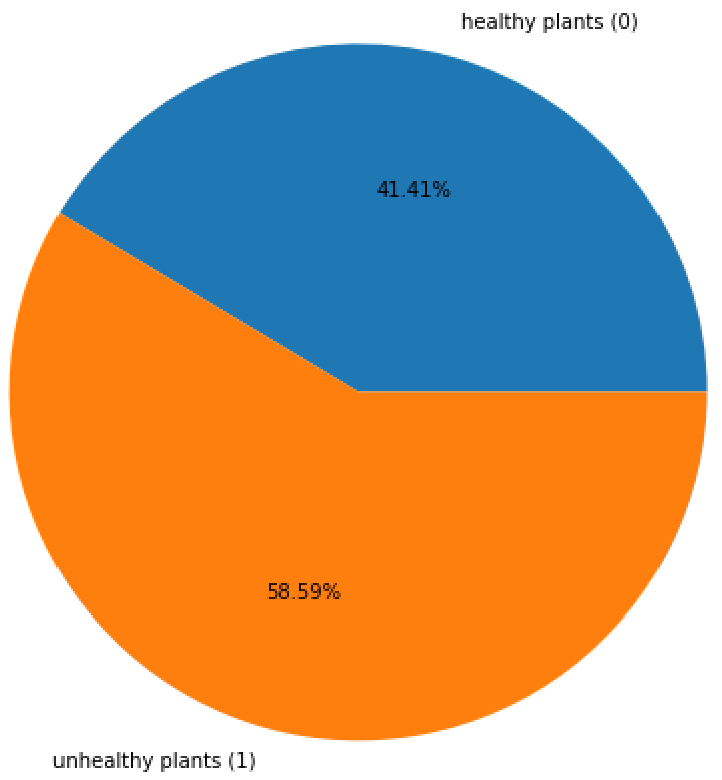
```
val_woutd= os.listdir('/content/gdrive/MyDrive/plant_disease/self_collected_c

train_wd_percent= len(train_wd)/len(train_wd+train_woutd)
train_woutd_percent= len(train_woutd)/len(train_wd+train_woutd)
val_wd_percent= len(val_wd)/len(val_wd+val_woutd)
val_woutd_percent= len(val_woutd)/len(val_wd+ val_woutd)

print("Ratio of healthy and unhealthy plants in training dataset is {}:{}".fo
print("Ratio of healthy and unhealthy plants in validation dataset is {}:{}".
```

```
Ratio of healthy and unhealthy plants in training dataset is 41:58
Ratio of healthy and unhealthy plants in validation dataset is 48:51
```

Pie Chart denoting ratio of healthy and unhleahty images from training dataset:

In [ ]:

```
my_labels=["healthy plants (0)","unhealthy plants (1)"]
plt.rcParams["figure.figsize"] = (8,8)

plt.pie([int(train_woutd_percent*100),
         int(train_wd_percent*100)],
        labels= my_labels,
        autopct='%1.2f%%')
plt.show()

print("Ratio of healthy and unhealthy plants in training dataset is {}:{}".fo
```



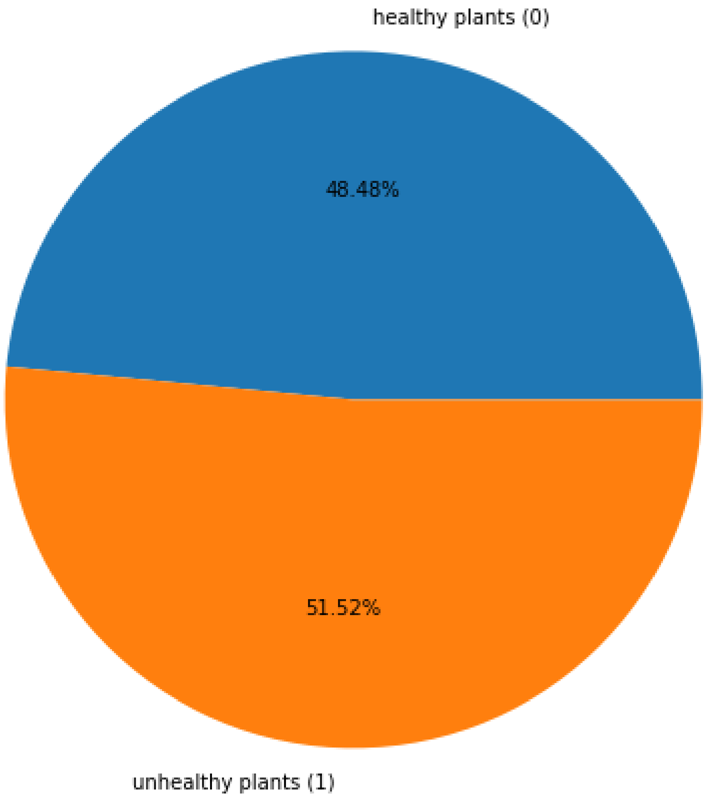

```
Ratio of healthy and unhealthy plants in training dataset is 41:58
```

Pie Chart denoting ratio of healthy and unhleahty images from validation dataset:

In [ ]:

```
my_labels=["healthy plants (0)","unhealthy plants (1)"]
plt.rcParams["figure.figsize"] = (8,8)

plt.pie([int(val_woutd_percent*100),
         int(val_wd_percent*100)],
        labels= my_labels,
```

```
        autopct='%1.2f%%')
plt.show()

print("Ratio of healthy and unhealthy plants in validation dataset is {}:{}".
```




```
Ratio of healthy and unhealthy plants in validation dataset is 48:51
```

Dataset is almost balanced with slight unbalancing in train dataset. Validation dataset seems to be almost balanced with a very slight difference.

Preview of dataset

Lets take a quick look at some of the images from our dataset (healthy and unhealthy plants):

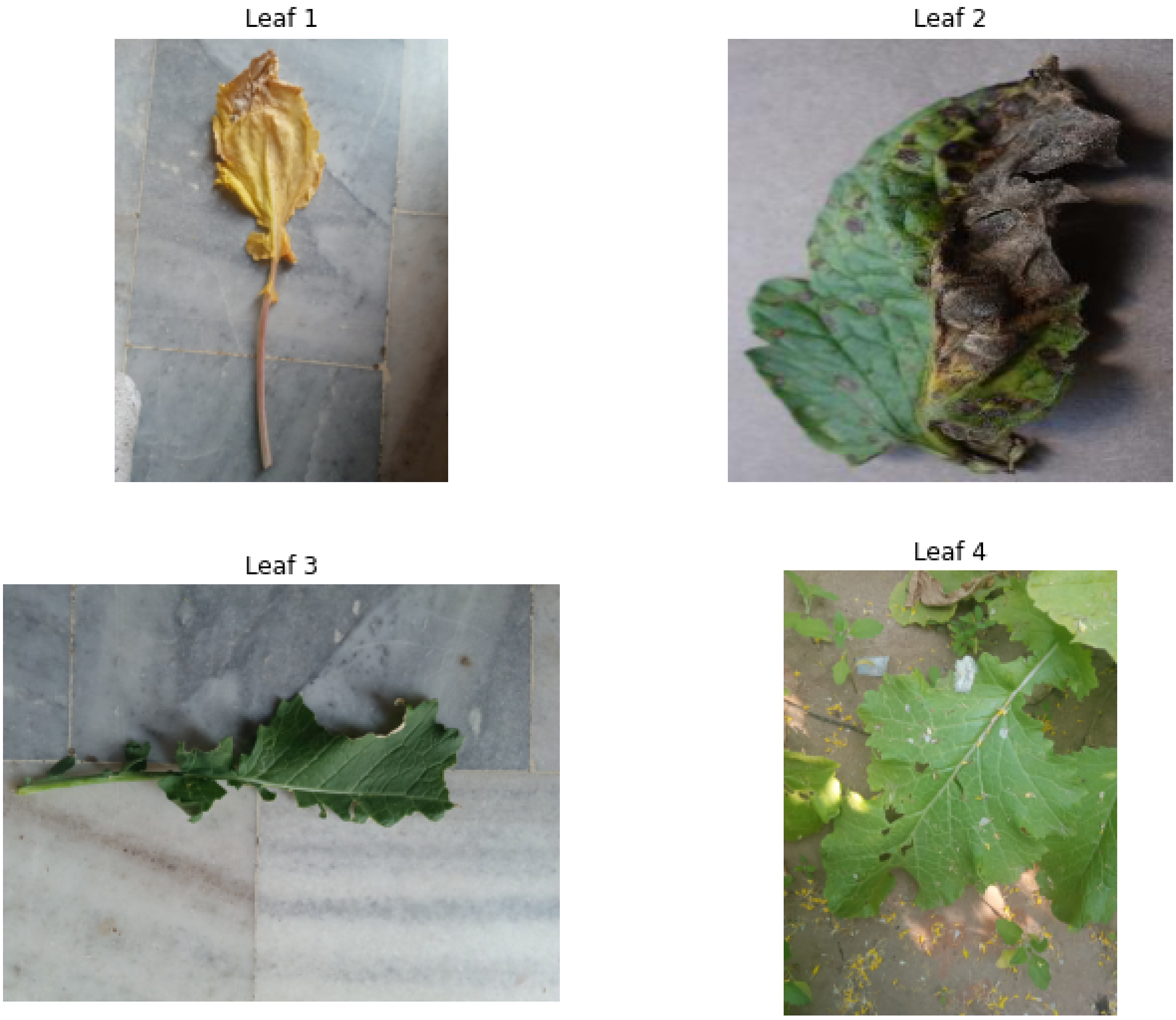
Plants with diseases

In [ ]:

```
# create figure
fig = plt.figure(figsize=(11, 9))

# reading images
Img_1 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_2 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_3 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_4 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data

# Add subplot at the 1st position
fig.add_subplot(2, 2, 1)

# show image
plt.imshow(Img_1)
plt.axis('off')
plt.title("Leaf 1")
```

```
# Add subplot at the 2nd position
fig.add_subplot(2, 2, 2)

# show image
plt.imshow(Img_2)
plt.axis('off')
plt.title("Leaf 2")

# Add subplot at the 3rd position
fig.add_subplot(2, 2, 3)

# show image
plt.imshow(Img_3)
plt.axis('off')
plt.title("Leaf 3")

# Add subplot at the 4th position
fig.add_subplot(2, 2, 4)

# show image
plt.imshow(Img_4)
plt.axis('off')
plt.title("Leaf 4")
```

Out[ ]:
```
Text(0.5, 1.0, 'Leaf 4')
```



As we can see from the above images that plants with disease seems to look a lot different from healthy plants' leaves. Plants suffering from disease tend to have discolored leaves like yellow color and also dry in nature. Apart from that leaves often tend to be disfigures, shrink or have cuts at different places.

Plants without diseases

In [ ]:

```
# create figure
fig = plt.figure(figsize=(11, 9))

# reading images
Img_1 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_2 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_3 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data
Img_4 = Image.open('/content/gdrive/MyDrive/plant_disease/self_collected_data

# Add subplot at the 1st position
fig.add_subplot(2, 2, 1)

# show image
plt.imshow(Img_1)
plt.axis('off')
plt.title("Leaf 1")

# Add subplot at the 2nd position
fig.add_subplot(2, 2, 2)

# show image
plt.imshow(Img_2)
plt.axis('off')
plt.title("Leaf 2")

# Add subplot at the 3rd position
fig.add_subplot(2, 2, 3)

# show image
plt.imshow(Img_3)
plt.axis('off')
plt.title("Leaf 3")

# Add subplot at the 4th position
fig.add_subplot(2, 2, 4)

# show image
plt.imshow(Img_4)
plt.axis('off')
plt.title("Leaf 4")
```

Out[ ]:

```
Text(0.5, 1.0, 'Leaf 4')
```

Leaf 1

Leaf 2

Leaf 3

Leaf 4

Healthy plants look all green in color without any disfigured leaf, without any cuts anywhere in leaf part discoloration or dryness.

Preprocessing:

For preprocessing, I will be using ImageDataGenerator to transform images as per following:

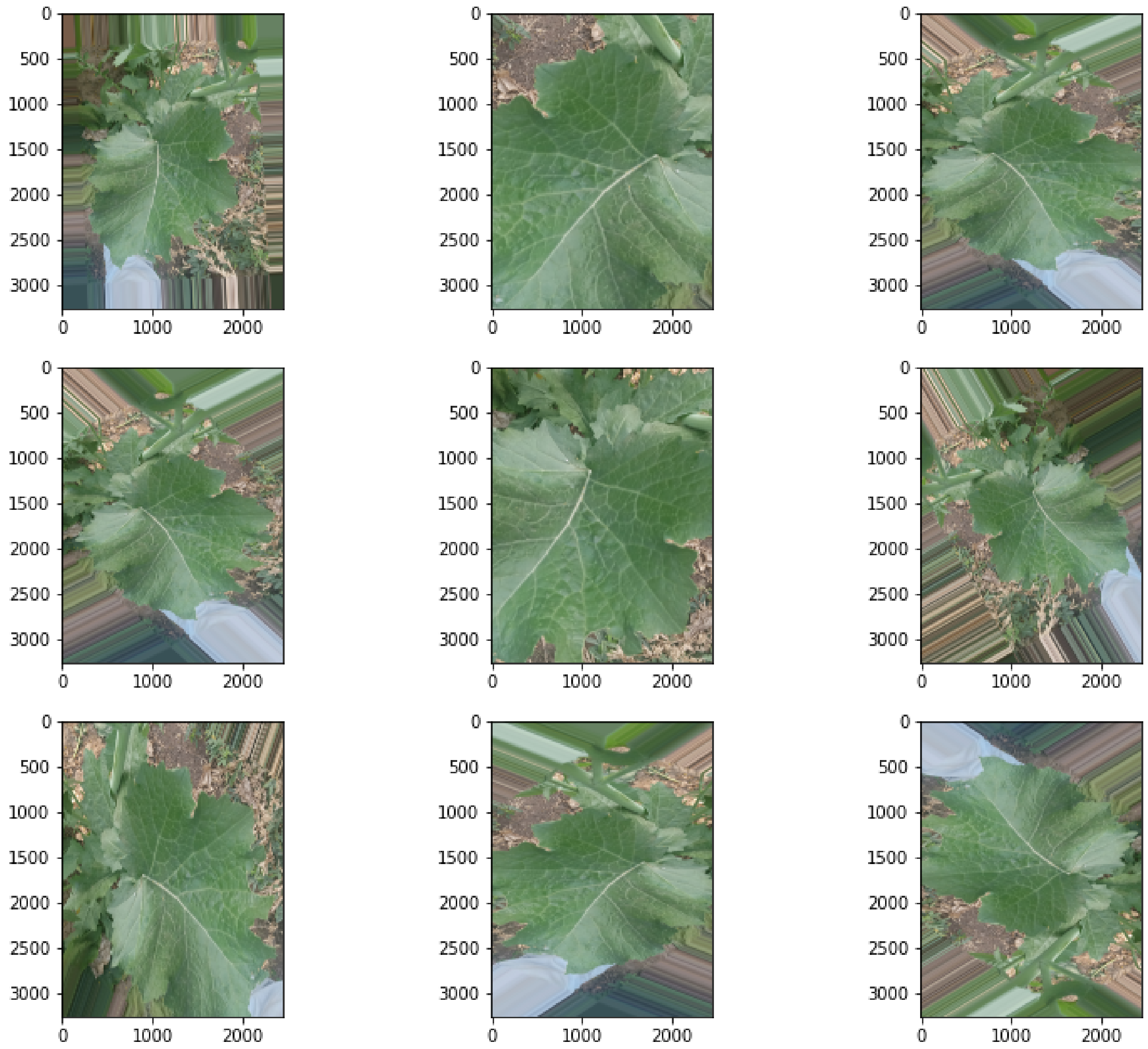
- rotation
- zoom
- vertical_flip
- horizontal_flip
- shear_range

Now, lets check how our ImageDataGenerator will give output image when a single image is input to it:

In [ ]:

```
# load the image
img = load_img('/content/gdrive/MyDrive/plant_disease/self_collected_data/tra

data = img_to_array(img) #covert to arrray

samples = np.expand_dims(data, 0)

datagen = ImageDataGenerator(shear_range=0.25,
                                    rotation_range=90,
                                    zoom_range=0.5,
```

```
                                    horizontal_flip=True,
                                    vertical_flip=True,
                                    validation_split=0.2)
# make iterator
it = datagen.flow(samples, batch_size=1)

fig = plt.figure(figsize=(13, 11))
# generate samples and plot
for i in range(9):
    # define subplot
    fig.add_subplot(330 + 1 + i)
    # generate batch of images
    batch = it.next()
    # convert to unsigned integers for viewing
    image = batch[0].astype('uint8')
    # plot raw pixel data
    plt.imshow(image)
# show the figure
plt.show()
```



Reading images from directory in ImageDataGenerator:

In [ ]:

```
#reading data
data_generator = ImageDataGenerator(validation_split=0.3,
                                    rescale=1/255)

train_generator = data_generator.flow_from_directory(
```

```
    directory=r"/content/gdrive/MyDrive/self_collected/data",
    subset='training',
    target_size=(254, 254),
    color_mode="rgb",
    batch_size= 32,
    class_mode="categorical",
    shuffle=True,
    seed=42
)

valid_generator = data_generator.flow_from_directory(
    directory=r"/content/gdrive/MyDrive/self_collected/data",
    subset='validation',
    target_size=(254, 254),
    color_mode="rgb",
    batch_size= 32,
    class_mode="categorical",
    shuffle=True,
    seed=42
)
```

```
Found 1697 images belonging to 2 classes.
Found 726 images belonging to 2 classes.
```

Since our data is small and we also have little computation power, so to get better performing model we will be using transfer learning using a pre trained weights like 'imagenet'.

Training on 'Plant Village' dataset to get weights for transfer learning:

Converting 'Plant Village' dataset into Binary class Dataset:

'Plant Village' dataset originally contains folders with various plants and particular diseases they are suffering from and healthy plants. So originally 'Plant Village' dataset is multiclass dataset. So I will be changing this dataset into binary class dataset by seperating images of 'healthy' and 'unhealthy' leaves into seperate folders:

In [ ]:

```
import shutil
import os
from tqdm import tqdm

src_path= '/content/PlantVillage/'
src_root = os.listdir('/content/PlantVillage/')
wd=[] #with disease
wtd=[] #without disease

#seperating healthy and unhealthy plants folder names:
for folder in tqdm(src_root):
  full_folder_name= src_path + folder
  for file_name in os.listdir(full_folder_name):
    if full_folder_name[-7:]!='healthy':
      wd.append(full_folder_name)
    else:
      wtd.append(full_folder_name)
```

Out[ ]:

```
"import shutil\nimport os\nfrom tqdm import tqdm\n\nsrc_path= '/content/Plant
Village/'\nsrc_root = os.listdir('/content/PlantVillage/')\nwd=[]\nwtd=[]\n\n
for folder in tqdm(src_root):\n  full_folder_name= src_path + folder\n  for f
ile_name in os.listdir(full_folder_name):\n    if full_folder_name[-7:]!='hea
```

```
lthy':\n        wd.append(full_folder_name)\n    else:\n        wtd.append(full_f
older_name)"
```

In [ ]:
```python
# copy all images of unhealthy leavse into 'wd' i.e with disease folder
cnt=[]
for full_folder_name in list(set(wd)):
  for file_name in tqdm(os.listdir(full_folder_name)):
    full_file_name = full_folder_name + '/' + file_name
    dest= '/content/drive/MyDrive/plant_disease/plant_village/data/wd'
    shutil.copy(full_file_name, dest)
    cnt.append(1)
```

```
100%|██████████| 952/952 [00:11<00:00, 82.68it/s]
100%|██████████| 1771/1771 [00:20<00:00, 87.17it/s]
100%|██████████| 2127/2127 [00:24<00:00, 86.49it/s]
100%|██████████| 1000/1000 [00:11<00:00, 85.52it/s]
100%|██████████| 997/997 [00:11<00:00, 86.16it/s]
100%|██████████| 1404/1404 [00:16<00:00, 85.84it/s]
100%|██████████| 1909/1909 [00:21<00:00, 88.20it/s]
100%|██████████| 1676/1676 [00:19<00:00, 87.70it/s]
100%|██████████| 1000/1000 [00:11<00:00, 85.82it/s]
100%|██████████| 1000/1000 [00:11<00:00, 87.60it/s]
100%|██████████| 3209/3209 [00:36<00:00, 87.04it/s]
100%|██████████| 373/373 [00:04<00:00, 86.56it/s]
```

In [ ]:
```python
len(cnt) #count of images of unhealthy leaves
```

Out[ ]:
```
17418
```

In [ ]:
```python
# copy all images of healthy leavse into 'wtd' i.e without disease folder
cnt=[]
for full_folder_name in list(set(wtd)):
  for file_name in tqdm(os.listdir(full_folder_name)):
    full_file_name = full_folder_name + '/' + file_name
    dest= '/content/drive/MyDrive/plant_disease/plant_village/data/wtd'
    shutil.copy(full_file_name, dest)
    cnt.append(1)
```

```
100%|██████████| 1478/1478 [00:17<00:00, 85.25it/s]
100%|██████████| 152/152 [00:01<00:00, 92.09it/s]
100%|██████████| 1591/1591 [00:18<00:00, 87.76it/s]
```

In [ ]:
```python
len(cnt) #count of images of healthy leaves
```

Out[ ]:
```
3221
```

Reading Plant Village Data:

Now that we have plant village dataset as Binary Class i.e 'healthy' and 'unhealthy' class we can begin reading this dataset.

In [ ]:
```python
images_ds= tf.data.Dataset.list_files("/content/gdrive/MyDrive/plant_disease/
```

In [ ]:
```python
for images in images_ds.take(14):
  print(images)
```

```
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/12703
385-4e74-4e56-b90a-fede2827fbf2___Matt.S_CG 6039.JPG', shape=(), dtype=strin
```

```
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/0982d
864-182e-4d36-959a-dcc67d85d9e6___RS_Erly.B 9481.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/23857
b36-24f3-4745-a877-dedde5c5931c___YLCV_NREC 2520.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/a87b8
e0c-2279-4cf7-85a4-0fba63130902___RS_Late.B 5323.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/56920
6bf-423e-4508-b1b4-2ca6821aceac___RS_Early.B 7554.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/58d5c
bab-162b-4091-8add-21a336398731___Com.G_SpM_FL 8933.JPG', shape=(), dtype=str
ing)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/de583
596-e89f-4a21-be43-a6629318bd6e___NREC_B.Spot 1840.JPG', shape=(), dtype=stri
ng)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/ed1a1
e55-f2b9-46d3-ab88-8742f569e831___GHLB2 Leaf 8758.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/1b589
f92-a658-4e58-96de-db41acc411ce___JR_B.Spot 3239.JPG', shape=(), dtype=strin
g)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wtd/c5f1
eec0-35c2-43ce-8a14-575d86258f73___RS_HL 0337.JPG', shape=(), dtype=string)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wtd/dad1
27e0-8761-47bd-8ea1-21727847a1f4___RS_HL 0416.JPG', shape=(), dtype=string)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wtd/1317
fd49-1819-4065-b3a4-d74f9763c7c4___RS_HL 0252.JPG', shape=(), dtype=string)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/2741b
105-dacf-4ed8-bedf-4fc395f8947c___Keller.St_CG 1943.JPG', shape=(), dtype=str
ing)
tf.Tensor(b'/content/gdrive/MyDrive/plant_disease/plant_village/data/wd/152c3
594-234d-4385-b487-83ea574cd860___Com.G_TgS_FL 1132.JPG', shape=(), dtype=str
ing)
```

In [ ]:

```
image_count= 3000 # number of images(data_size)
```

Due to memory constraints maximum RAM can hold without crashing is 3000 images

In [ ]:

```
data_size= int(image_count) # set size
data= images_ds.take(data_size) # get data
```

In [ ]:

```
# function to get label
def get_label(file_path):
  import os
  return tf.strings.split(file_path, os.path.sep)[-2]
```

In [ ]:

```
#function to get image and label
def process_image(file_path):
  label= get_label(file_path)

  img= tf.io.read_file(file_path)
  img= tf.image.decode_jpeg(img)
  img= tf.image.resize(img, [254,254])
  img= img/255

  return img, label
```

In [ ]:

```
x=[]
y=[]

for image, label in tqdm(data.map(process_image)):
  x.append(image)
  y.append(label)
```

```
100%|██████████| 3000/3000 [09:23<00:00,  5.33it/s]
```

Dump and load pickle file:

In [ ]:

```
import pickle

'''with open('/content/gdrive/MyDrive/plant_disease/plant_village_nondata/pic
  pickle.dump(x, p)
with open('/content/gdrive/MyDrive/plant_disease/plant_village_nondata/pickle
  pickle.dump(y, p)'''
```

In [ ]:

```
with open('/content/gdrive/MyDrive/plant_disease/plant_village_nondata/pickle
  x= pickle.load(p)
with open('/content/gdrive/MyDrive/plant_disease/plant_village_nondata/pickle
  y= pickle.load(p)
```

In [ ]:

```
# converting images to numpy array
x= np.array(x)
y_arr= np.array(y)

y= []
for i  in y_arr:
  i= ((str(i).split(',')[0][12:])[:-1])
  y.append(i)
del y_arr

y= np.asarray(y)
```

In [ ]:

```
#converting strings of y into label (1s and 0s) using LabelBinarizer:
label_binarizer = LabelBinarizer()
y_label = label_binarizer.fit_transform(y)
n_classes = len(label_binarizer.classes_)
```

Train_test_split

In [ ]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y_label, test_size=0.3
x_test, x_val, y_test, y_val = train_test_split(x_test, y_test, test_size=0.3
```

In [ ]:

```
epochs_ = 50
learning_rate = 1e-2
batch_size = 32
width=254
height=254
depth=3
```

Image Augmentation

In [ ]:

```
aug = ImageDataGenerator(
    width_shift_range=0.1,
    height_shift_range=0.1,
    rotation_range=30,
    zoom_range=0.25,
    shear_range=0.25,
    horizontal_flip=True,
    fill_mode="nearest")
```

Making Model

In [ ]:

```
#defining input_shape
input_shape = (height, width, depth)
chanDim = -1

if K.image_data_format() == "channels_first":
    input_shape = (depth, height, width)
    chanDim = 1

#model_activation="tanh"
model_activation="relu"
```

In [ ]:

```
#note for self: set initial dropouts to 0.25
model = Sequential()

inputShape = (height, width, depth)
chanDim = -1

if K.image_data_format() == "channels_first":
    inputShape = (depth, height, width)
    chanDim = 1

model.add(Conv2D(16, (3, 3), padding="same",input_shape=inputShape, name='Cor
model.add(Activation("relu", name='relu_1'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_1'))
model.add(MaxPooling2D(pool_size=(3, 3), name='MaxPool_1'))
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3), padding="same", name='Conv2D_2'))
model.add(Activation("relu", name='relu_2'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_2'))

model.add(Conv2D(64, (3, 3), padding="same", name='Conv2D_3'))
model.add(Activation("relu", name='relu_3'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_3'))
model.add(MaxPooling2D(pool_size=(2, 2), name='MaxPool_2'))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), padding="same", name='Conv2D_4'))
model.add(Activation("relu", name='relu_4'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_4'))

model.add(Conv2D(256, (3, 3), padding="same", name='Conv2D_5'))
model.add(Activation("relu", name='relu_5'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_5'))
model.add(MaxPooling2D(pool_size=(2, 2), name='MaxPool_3'))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation("relu", name='relu_6'))
model.add(BatchNormalization(name='BatchNormalization_6'))
```

```
model.add(Dropout(0.5))

model.add(Dense(1024))
model.add(Activation("relu", name='relu_7'))
model.add(BatchNormalization(name='BatchNormalization_7'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))
```

In [ ]:

```
model.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
Conv2D_1 (Conv2D)            (None, 254, 254, 16)      448       
_________________________________________________________________
relu_1 (Activation)          (None, 254, 254, 16)      0         
_________________________________________________________________
BatchNormalization_1 (BatchN (None, 254, 254, 16)      64        
_________________________________________________________________
MaxPool_1 (MaxPooling2D)     (None, 84, 84, 16)        0         
_________________________________________________________________
dropout_5 (Dropout)          (None, 84, 84, 16)        0         
_________________________________________________________________
Conv2D_2 (Conv2D)            (None, 84, 84, 32)        4640      
_________________________________________________________________
relu_2 (Activation)          (None, 84, 84, 32)        0         
_________________________________________________________________
BatchNormalization_2 (BatchN (None, 84, 84, 32)        128       
_________________________________________________________________
Conv2D_3 (Conv2D)            (None, 84, 84, 64)        18496     
_________________________________________________________________
relu_3 (Activation)          (None, 84, 84, 64)        0         
_________________________________________________________________
BatchNormalization_3 (BatchN (None, 84, 84, 64)        256       
_________________________________________________________________
MaxPool_2 (MaxPooling2D)     (None, 42, 42, 64)        0         
_________________________________________________________________
dropout_6 (Dropout)          (None, 42, 42, 64)        0         
_________________________________________________________________
Conv2D_4 (Conv2D)            (None, 42, 42, 128)       73856     
_________________________________________________________________
relu_4 (Activation)          (None, 42, 42, 128)       0         
_________________________________________________________________
BatchNormalization_4 (BatchN (None, 42, 42, 128)       512       
_________________________________________________________________
Conv2D_5 (Conv2D)            (None, 42, 42, 256)       295168    
_________________________________________________________________
relu_5 (Activation)          (None, 42, 42, 256)       0         
_________________________________________________________________
BatchNormalization_5 (BatchN (None, 42, 42, 256)       1024      
_________________________________________________________________
MaxPool_3 (MaxPooling2D)     (None, 21, 21, 256)       0         
_________________________________________________________________
dropout_7 (Dropout)          (None, 21, 21, 256)       0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 112896)            0         
_________________________________________________________________
dense_3 (Dense)              (None, 512)               57803264  
_________________________________________________________________
relu_6 (Activation)          (None, 512)               0         
_________________________________________________________________
BatchNormalization_6 (BatchN (None, 512)               2048      
_________________________________________________________________
dropout_8 (Dropout)          (None, 512)               0         
_________________________________________________________________
```

```
dense_4 (Dense)              (None, 1024)              525312
_________________________________________________________________
relu_7 (Activation)          (None, 1024)              0
_________________________________________________________________
BatchNormalization_7 (BatchN (None, 1024)              4096
_________________________________________________________________
dropout_9 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_5 (Dense)              (None, 1)                 1025
=================================================================
Total params: 58,730,337
Trainable params: 58,726,273
Non-trainable params: 4,064
_________________________________________________________________
```

In [ ]:

```
from keras.callbacks import ModelCheckpoint

filepath = "/content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1, sav
callbacks_list = [checkpoint]
```

In [ ]:

```
#defining optimizer with learning rate decay with every epochs
opt = tf.keras.optimizers.Adam(learning_rate=learning_rate, decay=learning_ra
model.compile(loss="binary_crossentropy", optimizer=opt,metrics=["accuracy"])
```

In [ ]:

```
history = model.fit(
    aug.flow(x_train, y_train, batch_size=batch_size),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train) // batch_size,
    epochs=epochs_,
    callbacks=callbacks_list,
    verbose=1
    )
```

```
Epoch 1/50
65/65 [==============================] - 36s 518ms/step - loss: 0.1185 - accu
racy: 0.9584 - val_loss: 0.7685 - val_accuracy: 0.8016

Epoch 00001: val_accuracy improved from -inf to 0.80159, saving model to /con
tent/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weights
_aug_18_3.hdf5
Epoch 2/50
65/65 [==============================] - 35s 540ms/step - loss: 0.1411 - accu
racy: 0.9565 - val_loss: 0.6672 - val_accuracy: 0.8238

Epoch 00002: val_accuracy improved from 0.80159 to 0.82381, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 3/50
65/65 [==============================] - 35s 539ms/step - loss: 0.1259 - accu
racy: 0.9613 - val_loss: 0.6055 - val_accuracy: 0.8698

Epoch 00003: val_accuracy improved from 0.82381 to 0.86984, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 4/50
65/65 [==============================] - 33s 511ms/step - loss: 0.0770 - accu
racy: 0.9734 - val_loss: 0.2140 - val_accuracy: 0.9587

Epoch 00004: val_accuracy improved from 0.86984 to 0.95873, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 5/50
65/65 [==============================] - 35s 536ms/step - loss: 0.1079 - accu
```

```
racy: 0.9632 - val_loss: 0.7646 - val_accuracy: 0.7905

Epoch 00005: val_accuracy did not improve from 0.95873
Epoch 6/50
65/65 [==============================] - 33s 509ms/step - loss: 0.3791 - accu
racy: 0.8738 - val_loss: 1.0060 - val_accuracy: 0.7730

Epoch 00006: val_accuracy did not improve from 0.95873
Epoch 7/50
65/65 [==============================] - 33s 510ms/step - loss: 0.3216 - accu
racy: 0.8617 - val_loss: 0.3595 - val_accuracy: 0.8746

Epoch 00007: val_accuracy did not improve from 0.95873
Epoch 8/50
65/65 [==============================] - 33s 509ms/step - loss: 0.2350 - accu
racy: 0.8994 - val_loss: 0.4858 - val_accuracy: 0.8556

Epoch 00008: val_accuracy did not improve from 0.95873
Epoch 9/50
65/65 [==============================] - 33s 512ms/step - loss: 0.2054 - accu
racy: 0.9144 - val_loss: 0.2654 - val_accuracy: 0.9111

Epoch 00009: val_accuracy did not improve from 0.95873
Epoch 10/50
65/65 [==============================] - 33s 507ms/step - loss: 0.1807 - accu
racy: 0.9221 - val_loss: 0.1482 - val_accuracy: 0.9349

Epoch 00010: val_accuracy did not improve from 0.95873
Epoch 11/50
65/65 [==============================] - 33s 505ms/step - loss: 0.1666 - accu
racy: 0.9289 - val_loss: 0.1046 - val_accuracy: 0.9587

Epoch 00011: val_accuracy did not improve from 0.95873
Epoch 12/50
65/65 [==============================] - 33s 508ms/step - loss: 0.1450 - accu
racy: 0.9400 - val_loss: 0.1805 - val_accuracy: 0.9238

Epoch 00012: val_accuracy did not improve from 0.95873
Epoch 13/50
65/65 [==============================] - 33s 506ms/step - loss: 0.1374 - accu
racy: 0.9473 - val_loss: 0.3911 - val_accuracy: 0.8444

Epoch 00013: val_accuracy did not improve from 0.95873
Epoch 14/50
65/65 [==============================] - 33s 507ms/step - loss: 0.1359 - accu
racy: 0.9386 - val_loss: 0.3580 - val_accuracy: 0.8571

Epoch 00014: val_accuracy did not improve from 0.95873
Epoch 15/50
65/65 [==============================] - 33s 510ms/step - loss: 0.1309 - accu
racy: 0.9492 - val_loss: 0.1334 - val_accuracy: 0.9397

Epoch 00015: val_accuracy did not improve from 0.95873
Epoch 16/50
65/65 [==============================] - 33s 510ms/step - loss: 0.1252 - accu
racy: 0.9507 - val_loss: 0.0982 - val_accuracy: 0.9619

Epoch 00016: val_accuracy improved from 0.95873 to 0.96190, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 17/50
65/65 [==============================] - 35s 539ms/step - loss: 0.1008 - accu
racy: 0.9560 - val_loss: 0.1106 - val_accuracy: 0.9651

Epoch 00017: val_accuracy improved from 0.96190 to 0.96508, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 18/50
65/65 [==============================] - 35s 535ms/step - loss: 0.1227 - accu
```

```
racy: 0.9531 - val_loss: 0.0915 - val_accuracy: 0.9714

Epoch 00018: val_accuracy improved from 0.96508 to 0.97143, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 19/50
65/65 [==============================] - 35s 536ms/step - loss: 0.0999 - accu
racy: 0.9599 - val_loss: 0.1530 - val_accuracy: 0.9413

Epoch 00019: val_accuracy did not improve from 0.97143
Epoch 20/50
65/65 [==============================] - 33s 506ms/step - loss: 0.1076 - accu
racy: 0.9565 - val_loss: 0.0905 - val_accuracy: 0.9698

Epoch 00020: val_accuracy did not improve from 0.97143
Epoch 21/50
65/65 [==============================] - 33s 503ms/step - loss: 0.0789 - accu
racy: 0.9657 - val_loss: 0.0946 - val_accuracy: 0.9651

Epoch 00021: val_accuracy did not improve from 0.97143
Epoch 22/50
65/65 [==============================] - 33s 503ms/step - loss: 0.0810 - accu
racy: 0.9695 - val_loss: 0.1197 - val_accuracy: 0.9746

Epoch 00022: val_accuracy improved from 0.97143 to 0.97460, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 23/50
65/65 [==============================] - 35s 533ms/step - loss: 0.0949 - accu
racy: 0.9628 - val_loss: 0.1498 - val_accuracy: 0.9651

Epoch 00023: val_accuracy did not improve from 0.97460
Epoch 24/50
65/65 [==============================] - 33s 502ms/step - loss: 0.0959 - accu
racy: 0.9613 - val_loss: 0.2126 - val_accuracy: 0.9175

Epoch 00024: val_accuracy did not improve from 0.97460
Epoch 25/50
65/65 [==============================] - 33s 503ms/step - loss: 0.0892 - accu
racy: 0.9666 - val_loss: 0.0895 - val_accuracy: 0.9714

Epoch 00025: val_accuracy did not improve from 0.97460
Epoch 26/50
65/65 [==============================] - 33s 502ms/step - loss: 0.0910 - accu
racy: 0.9647 - val_loss: 0.1972 - val_accuracy: 0.9206

Epoch 00026: val_accuracy did not improve from 0.97460
Epoch 27/50
65/65 [==============================] - 33s 503ms/step - loss: 0.0819 - accu
racy: 0.9652 - val_loss: 0.3819 - val_accuracy: 0.8667

Epoch 00027: val_accuracy did not improve from 0.97460
Epoch 28/50
65/65 [==============================] - 33s 503ms/step - loss: 0.0735 - accu
racy: 0.9724 - val_loss: 0.4951 - val_accuracy: 0.9444

Epoch 00028: val_accuracy did not improve from 0.97460
Epoch 29/50
65/65 [==============================] - 33s 502ms/step - loss: 0.0825 - accu
racy: 0.9681 - val_loss: 0.6628 - val_accuracy: 0.9524

Epoch 00029: val_accuracy did not improve from 0.97460
Epoch 30/50
65/65 [==============================] - 33s 502ms/step - loss: 0.0749 - accu
racy: 0.9739 - val_loss: 1.1722 - val_accuracy: 0.8524

Epoch 00030: val_accuracy did not improve from 0.97460
Epoch 31/50
65/65 [==============================] - 33s 501ms/step - loss: 0.0730 - accu
```

```
racy: 0.9729 - val_loss: 0.8733 - val_accuracy: 0.8857

Epoch 00031: val_accuracy did not improve from 0.97460
Epoch 32/50
65/65 [==============================] - 33s 507ms/step - loss: 0.0646 - accu
racy: 0.9763 - val_loss: 0.7038 - val_accuracy: 0.9270

Epoch 00032: val_accuracy did not improve from 0.97460
Epoch 33/50
65/65 [==============================] - 33s 507ms/step - loss: 0.0628 - accu
racy: 0.9773 - val_loss: 0.8078 - val_accuracy: 0.9794

Epoch 00033: val_accuracy improved from 0.97460 to 0.97937, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 34/50
65/65 [==============================] - 35s 536ms/step - loss: 0.1354 - accu
racy: 0.9468 - val_loss: 2.1306 - val_accuracy: 0.6651

Epoch 00034: val_accuracy did not improve from 0.97937
Epoch 35/50
65/65 [==============================] - 33s 512ms/step - loss: 0.1800 - accu
racy: 0.9260 - val_loss: 0.2308 - val_accuracy: 0.9238

Epoch 00035: val_accuracy did not improve from 0.97937
Epoch 36/50
65/65 [==============================] - 33s 507ms/step - loss: 0.1371 - accu
racy: 0.9420 - val_loss: 0.1276 - val_accuracy: 0.9492

Epoch 00036: val_accuracy did not improve from 0.97937
Epoch 37/50
65/65 [==============================] - 33s 510ms/step - loss: 0.1188 - accu
racy: 0.9497 - val_loss: 0.1018 - val_accuracy: 0.9635

Epoch 00037: val_accuracy did not improve from 0.97937
Epoch 38/50
65/65 [==============================] - 33s 507ms/step - loss: 0.1155 - accu
racy: 0.9531 - val_loss: 0.1714 - val_accuracy: 0.9302

Epoch 00038: val_accuracy did not improve from 0.97937
Epoch 39/50
65/65 [==============================] - 33s 508ms/step - loss: 0.0778 - accu
racy: 0.9739 - val_loss: 0.0954 - val_accuracy: 0.9667

Epoch 00039: val_accuracy did not improve from 0.97937
Epoch 40/50
65/65 [==============================] - 33s 509ms/step - loss: 0.0787 - accu
racy: 0.9715 - val_loss: 0.0703 - val_accuracy: 0.9841

Epoch 00040: val_accuracy improved from 0.97937 to 0.98413, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 41/50
65/65 [==============================] - 35s 535ms/step - loss: 0.0853 - accu
racy: 0.9671 - val_loss: 0.0554 - val_accuracy: 0.9857

Epoch 00041: val_accuracy improved from 0.98413 to 0.98571, saving model to /
content/gdrive/MyDrive/plant_disease/plant_village_nondata/plant_village_weig
hts_aug_18_3.hdf5
Epoch 42/50
65/65 [==============================] - 33s 509ms/step - loss: 0.0846 - accu
racy: 0.9657 - val_loss: 0.0719 - val_accuracy: 0.9778

Epoch 00042: val_accuracy did not improve from 0.98571
Epoch 43/50
65/65 [==============================] - 33s 507ms/step - loss: 0.0736 - accu
racy: 0.9729 - val_loss: 0.0709 - val_accuracy: 0.9810

Epoch 00043: val_accuracy did not improve from 0.98571
```

```
Epoch 44/50
65/65 [==============================] - 34s 515ms/step - loss: 0.0610 - accu
racy: 0.9734 - val_loss: 1.3283 - val_accuracy: 0.9746

Epoch 00044: val_accuracy did not improve from 0.98571
Epoch 45/50
65/65 [==============================] - 33s 511ms/step - loss: 0.0656 - accu
racy: 0.9778 - val_loss: 0.0521 - val_accuracy: 0.9825

Epoch 00045: val_accuracy did not improve from 0.98571
Epoch 46/50
65/65 [==============================] - 33s 510ms/step - loss: 0.0693 - accu
racy: 0.9724 - val_loss: 0.0595 - val_accuracy: 0.9810

Epoch 00046: val_accuracy did not improve from 0.98571
Epoch 47/50
65/65 [==============================] - 33s 510ms/step - loss: 0.0618 - accu
racy: 0.9773 - val_loss: 0.1113 - val_accuracy: 0.9603

Epoch 00047: val_accuracy did not improve from 0.98571
Epoch 48/50
65/65 [==============================] - 33s 512ms/step - loss: 0.0674 - accu
racy: 0.9729 - val_loss: 0.0583 - val_accuracy: 0.9762

Epoch 00048: val_accuracy did not improve from 0.98571
Epoch 49/50
65/65 [==============================] - 33s 510ms/step - loss: 0.0594 - accu
racy: 0.9811 - val_loss: 0.3098 - val_accuracy: 0.8905

Epoch 00049: val_accuracy did not improve from 0.98571
Epoch 50/50
65/65 [==============================] - 33s 505ms/step - loss: 0.0673 - accu
racy: 0.9792 - val_loss: 0.0602 - val_accuracy: 0.9825

Epoch 00050: val_accuracy did not improve from 0.98571
```

Now that the model has been trained with 94+ percent accuracy I will save these weights and use later for Transfer Learning on my self collected dataset.

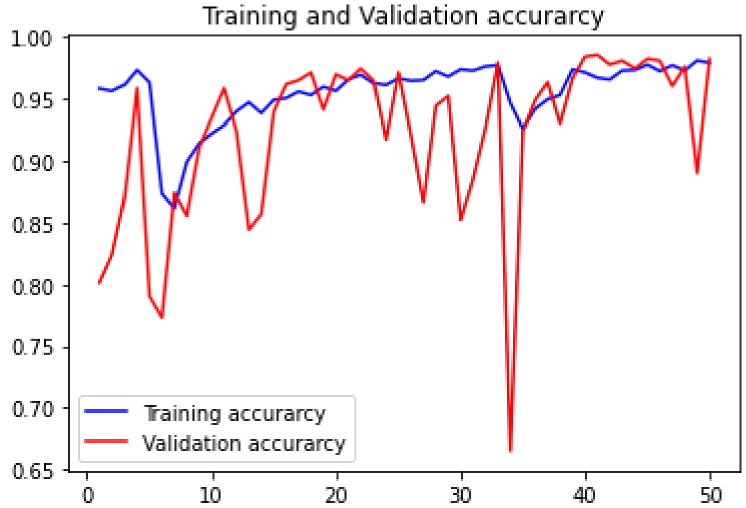
In [ ]:

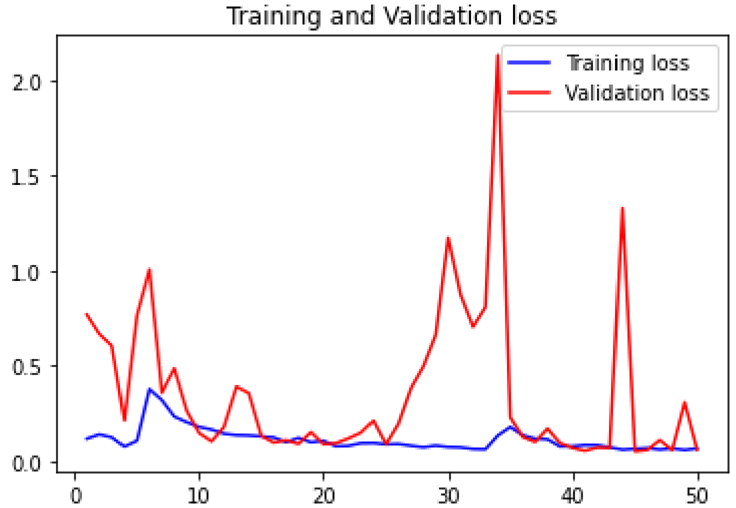
```
model.save_weights('/content/gdrive/MyDrive/Plant_disease/aug_20_4.h5')
```

Evaluating the performance of model:

In [ ]:

```
validation_accuracy = history.history['val_accuracy']
train_accuracy = history.history['accuracy']
train_loss = history.history['loss']
validation_loss = history.history['val_loss']
epochs = range(1, len(train_accuracy) + 1)

#Train and validation accuracy
plt.plot(epochs, train_accuracy, 'b', label='Training accurarcy')
plt.plot(epochs, validation_accuracy, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, train_loss, 'b', label='Training loss')
plt.plot(epochs, validation_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```





In [ ]:

```
x, y = x_test, y_test
x.shape, y.shape

from sklearn.metrics import confusion_matrix, f1_score, precision_score, reca
from keras.utils.np_utils import to_categorical
import tensorflow as tf

scores = model.evaluate(x_test, y_test)

y_pred=[]

for pred in ((model.predict(x_test))): #custom loop with threshold as 0.5
  if pred >= 0.5:
    y_pred.append(1)
  else:
    y_pred.append(0)

y_test_new= []
for y in y_test:
  y= np.array_str(y)
  y= int(y[1])
  y_test_new.append(y)

print("CONFUSION MATRIX:")
print( confusion_matrix(y_test_new , y_pred))
print("=======================")

print(f"ACCURACY for test dataset is:       {scores[1]}")
print("F1_SCORE for test datset is         {}".format(f1_score(y_test_new , y
```

```
print("PRECISION_SCORE for test datset is  {}".format(precision_score(y_test_
print("RECALL_SCORE for test datset is     {}".format(recall_score(y_test_new
```

```
20/20 [==============================] - 1s 59ms/step - loss: 1.4532 - accura
cy: 0.8778
CONFUSION MATRIX:
[[464  72]
 [  5  89]]
======================
ACCURACY for test dataset is:       0.8777777552604675
F1_SCORE for test datset is         0.6980392156862745
PRECISION_SCORE for test datset is  0.5527950310559007
RECALL_SCORE for test datset is     0.9468085106382979
```

Trainig Model on self collected data:

In [ ]:

```
images_ds= tf.data.Dataset.list_files("/content/gdrive/MyDrive/Plant_disease/
```

In [ ]:

```
for images in images_ds.take(3):
  print(images)
```

```
tf.Tensor(b'/content/gdrive/MyDrive/Plant_disease/self_collected/data/without
_disease/IMG_20200112_180808.jpg', shape=(), dtype=string)
tf.Tensor(b'/content/gdrive/MyDrive/Plant_disease/self_collected/data/with_di
sease/withd035 copy.jpg', shape=(), dtype=string)
tf.Tensor(b'/content/gdrive/MyDrive/Plant_disease/self_collected/data/without
_disease/wd251 copy.jpg', shape=(), dtype=string)
```

In [ ]:

```
image_count=len(images_ds) #get number of images(data_size)
image_count
```

Out[ ]: 2423

In [ ]:

```
train_size= int(image_count) # set train size

data= images_ds.take(train_size) # get train data
```

In [ ]:

```
# function to get label
def get_label(file_path):
  import os
  return tf.strings.split(file_path, os.path.sep)[-2]
```

In [ ]:

```
#function to get image and label
def process_image(file_path):
  label= get_label(file_path)

  img= tf.io.read_file(file_path)
  img= tf.image.decode_jpeg(img)
  img= tf.image.resize(img, [254,254])
  img= img/255

  return img, label
```

In [ ]:

```
x=[]
y=[]
```

```
for image, label in tqdm(data.map(process_image)):
  x.append(image)
  y.append(label)
```

```
100%|██████████| 2423/2423 [09:10<00:00,  4.40it/s]
```

In [ ]:

```
x= np.array(x)
y_arr= np.array(y)
```

In [ ]:

```
y= []
for i  in y_arr:
  i= ((str(i).split(',')[0][12:])[:-1])
  y.append(i)
del y_arr
```

In [ ]:

```
y= np.asarray(y)
```

In [ ]:

```
#converting strings of y into label (1s and 0s):
label_binarizer = LabelBinarizer()
y_label = label_binarizer.fit_transform(y)
n_classes = len(label_binarizer.classes_)
```

Saving data to pickle:

In [ ]:

```
import pickle
with open('/content/gdrive/MyDrive/Plant_disease/pickle/x.pkl','wb') as p:
  pickle.dump(x, p)
with open('/content/gdrive/MyDrive/Plant_disease/pickle/y_label.pkl','wb') as
  pickle.dump(y_label, p)
```

In [ ]:

```
with open('/content/gdrive/MyDrive/Plant_disease/pickle/x.pkl','rb') as p:
  x= pickle.load(p)
with open('/content/gdrive/MyDrive/Plant_disease/pickle/y_label.pkl','rb') as
  y_label= pickle.load(p)
```

Train_test_split

In [ ]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y_label, test_size=0.3
x_test, x_val, y_test, y_val = train_test_split(x_test, y_test, test_size=0.3
```

In [ ]:

```
print(x_train.shape)
print(x_val.shape)
print(x_test.shape)
print(y_train.shape)
print(y_val.shape)
print(y_test.shape)
```

```
(1696, 254, 254, 3)
(219, 254, 254, 3)
(508, 254, 254, 3)
(1696, 1)
(219, 1)
(508, 1)
```

In [4]:

```
learning_rate = 1e-2
batch_size = 32
epochs_ = 300
width=254
height=254
depth=3
```

ImageAugmentation using ImageDataGenerator

In [6]:

```
aug = ImageDataGenerator(
    width_shift_range=0.1,
    height_shift_range=0.1,
    rotation_range=30,
    zoom_range=0.25,
    shear_range=0.25,
    horizontal_flip=True,
    fill_mode="nearest")
```

Defining model:

In [7]:

```
#defining input_shape
input_shape = (height, width, depth)
chanDim = -1

if K.image_data_format() == "channels_first":
    input_shape = (depth, height, width)
    chanDim = 1

model_activation="relu"
```

In [8]:

```
#note for self: set initial dropouts to 0.25
model = Sequential()

inputShape = (height, width, depth)
chanDim = -1

if K.image_data_format() == "channels_first":
    inputShape = (depth, height, width)
    chanDim = 1

model.add(Conv2D(16, (3, 3), padding="same",input_shape=inputShape, name='Cor
model.add(Activation("relu", name='relu_1'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_1'))
model.add(MaxPooling2D(pool_size=(3, 3), name='MaxPool_1'))
model.add(Dropout(0.25))

model.add(Conv2D(32, (3, 3), padding="same", name='Conv2D_2'))
model.add(Activation("relu", name='relu_2'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_2'))

model.add(Conv2D(64, (3, 3), padding="same", name='Conv2D_3'))
model.add(Activation("relu", name='relu_3'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_3'))
model.add(MaxPooling2D(pool_size=(2, 2), name='MaxPool_2'))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3), padding="same", name='Conv2D_4'))
model.add(Activation("relu", name='relu_4'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_4'))
```

```
model.add(Conv2D(256, (3, 3), padding="same", name='Conv2D_5'))
model.add(Activation("relu", name='relu_5'))
model.add(BatchNormalization(axis=chanDim, name='BatchNormalization_5'))
model.add(MaxPooling2D(pool_size=(2, 2), name='MaxPool_3'))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation("relu", name='relu_6'))
model.add(BatchNormalization(name='BatchNormalization_6'))
model.add(Dropout(0.5))

model.add(Dense(1024))
model.add(Activation("relu", name='relu_7'))
model.add(BatchNormalization(name='BatchNormalization_7'))
model.add(Dropout(0.5))

model.add(Dense(1, activation='sigmoid'))
```

Load weights from model previosly trained on plant village dataset:

As we have trained a model previously on 'Plant Village' dataset, the weights obtained after that training will be helpfull for classifiying our dataset well.

In [ ]:

```
model.load_weights("/content/gdrive/MyDrive/Plant_disease/aug_20_4.h5")
```

Now that the weights have been loaded, I will be freezing all the layers except last 5 layers of the model on which our collected data will train on:

In [9]:

```
# unfreezing only last 5 layers
for layer in model.layers[:-5]:
  layer.trainable = False
```

In [10]:

```
for l in model.layers:
    print(l.name, l.trainable)
```

```
Conv2D_1 False
relu_1 False
BatchNormalization_1 False
MaxPool_1 False
dropout False
Conv2D_2 False
relu_2 False
BatchNormalization_2 False
Conv2D_3 False
relu_3 False
BatchNormalization_3 False
MaxPool_2 False
dropout_1 False
Conv2D_4 False
relu_4 False
BatchNormalization_4 False
Conv2D_5 False
relu_5 False
BatchNormalization_5 False
MaxPool_3 False
dropout_2 False
flatten False
dense False
relu_6 False
BatchNormalization_6 False
```

```
dropout_3 False
dense_1 True
relu_7 True
BatchNormalization_7 True
dropout_4 True
dense_2 True
```

As we can see above that only last 5 layers are trainable now. Rest all layers have ben freezed.

In [ ]:

```
from keras.callbacks import ModelCheckpoint

filepath = "/content/gdrive/MyDrive/Plant_disease/aug_20_1.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1, sav
callbacks_list = [checkpoint]
```

In [11]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
Conv2D_1 (Conv2D)            (None, 254, 254, 16)      448
_________________________________________________________________
relu_1 (Activation)          (None, 254, 254, 16)      0
_________________________________________________________________
BatchNormalization_1 (BatchN (None, 254, 254, 16)      64
_________________________________________________________________
MaxPool_1 (MaxPooling2D)     (None, 84, 84, 16)        0
_________________________________________________________________
dropout (Dropout)            (None, 84, 84, 16)        0
_________________________________________________________________
Conv2D_2 (Conv2D)            (None, 84, 84, 32)        4640
_________________________________________________________________
relu_2 (Activation)          (None, 84, 84, 32)        0
_________________________________________________________________
BatchNormalization_2 (BatchN (None, 84, 84, 32)        128
_________________________________________________________________
Conv2D_3 (Conv2D)            (None, 84, 84, 64)        18496
_________________________________________________________________
relu_3 (Activation)          (None, 84, 84, 64)        0
_________________________________________________________________
BatchNormalization_3 (BatchN (None, 84, 84, 64)        256
_________________________________________________________________
MaxPool_2 (MaxPooling2D)     (None, 42, 42, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 42, 42, 64)        0
_________________________________________________________________
Conv2D_4 (Conv2D)            (None, 42, 42, 128)       73856
_________________________________________________________________
relu_4 (Activation)          (None, 42, 42, 128)       0
_________________________________________________________________
BatchNormalization_4 (BatchN (None, 42, 42, 128)       512
_________________________________________________________________
Conv2D_5 (Conv2D)            (None, 42, 42, 256)       295168
_________________________________________________________________
relu_5 (Activation)          (None, 42, 42, 256)       0
_________________________________________________________________
BatchNormalization_5 (BatchN (None, 42, 42, 256)       1024
_________________________________________________________________
MaxPool_3 (MaxPooling2D)     (None, 21, 21, 256)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 21, 21, 256)       0
_________________________________________________________________
flatten (Flatten)            (None, 112896)            0
_________________________________________________________________
dense (Dense)                (None, 512)               57803264
_________________________________________________________________
```

```
relu_6 (Activation)          (None, 512)               0
_________________________________________________________________
BatchNormalization_6 (BatchN (None, 512)               2048
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              525312
_________________________________________________________________
relu_7 (Activation)          (None, 1024)              0
_________________________________________________________________
BatchNormalization_7 (BatchN (None, 1024)              4096
_________________________________________________________________
dropout_4 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 1025
=================================================================
Total params: 58,730,337
Trainable params: 528,385
Non-trainable params: 58,201,952
_________________________________________________________________
```

In [ ]:

```
# defining optimizer
opt = tf.keras.optimizers.Adam(learning_rate=learning_rate, decay=learning_ra
model.compile(loss="binary_crossentropy", optimizer=opt,metrics=["accuracy"])
```

In [ ]:

```
history = model.fit(
    aug.flow(x_train, y_train, batch_size= batch_size),
    validation_data=(x_val, y_val),
    steps_per_epoch=len(x_train) // batch_size,
    callbacks= callbacks_list,
    epochs= epochs_,
    verbose=1
    )
```

```
Epoch 1/300
53/53 [==============================] - 27s 498ms/step - loss: 0.7604 - accu
racy: 0.7358 - val_loss: 0.6558 - val_accuracy: 0.7854

Epoch 00001: val_accuracy improved from -inf to 0.78539, saving model to /con
tent/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 2/300
53/53 [==============================] - 28s 535ms/step - loss: 0.5435 - accu
racy: 0.7871 - val_loss: 0.8256 - val_accuracy: 0.7808

Epoch 00002: val_accuracy did not improve from 0.78539
Epoch 3/300
53/53 [==============================] - 26s 497ms/step - loss: 0.5831 - accu
racy: 0.7759 - val_loss: 0.6078 - val_accuracy: 0.7854

Epoch 00003: val_accuracy did not improve from 0.78539
Epoch 4/300
53/53 [==============================] - 26s 489ms/step - loss: 0.5014 - accu
racy: 0.8078 - val_loss: 0.4754 - val_accuracy: 0.8493

Epoch 00004: val_accuracy improved from 0.78539 to 0.84932, saving model to /
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 5/300
53/53 [==============================] - 28s 530ms/step - loss: 0.5104 - accu
racy: 0.7983 - val_loss: 0.3797 - val_accuracy: 0.8128

Epoch 00005: val_accuracy did not improve from 0.84932
Epoch 6/300
53/53 [==============================] - 26s 487ms/step - loss: 0.5079 - accu
racy: 0.8019 - val_loss: 0.3856 - val_accuracy: 0.8584

Epoch 00006: val_accuracy improved from 0.84932 to 0.85845, saving model to /
```

```
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 7/300
53/53 [==============================] - 28s 531ms/step - loss: 0.4511 - accu
racy: 0.8066 - val_loss: 0.6857 - val_accuracy: 0.7169

Epoch 00007: val_accuracy did not improve from 0.85845
Epoch 8/300
53/53 [==============================] - 26s 486ms/step - loss: 0.4699 - accu
racy: 0.8213 - val_loss: 0.5013 - val_accuracy: 0.7808

Epoch 00008: val_accuracy did not improve from 0.85845
Epoch 9/300
53/53 [==============================] - 25s 479ms/step - loss: 0.4429 - accu
racy: 0.8231 - val_loss: 0.5935 - val_accuracy: 0.7443

Epoch 00009: val_accuracy did not improve from 0.85845
Epoch 10/300
53/53 [==============================] - 26s 481ms/step - loss: 0.4941 - accu
racy: 0.8066 - val_loss: 0.5691 - val_accuracy: 0.7306

Epoch 00010: val_accuracy did not improve from 0.85845
Epoch 11/300
53/53 [==============================] - 25s 479ms/step - loss: 0.4363 - accu
racy: 0.8202 - val_loss: 0.4930 - val_accuracy: 0.7900

Epoch 00011: val_accuracy did not improve from 0.85845
Epoch 12/300
53/53 [==============================] - 26s 483ms/step - loss: 0.4588 - accu
racy: 0.8196 - val_loss: 0.4387 - val_accuracy: 0.8128

Epoch 00012: val_accuracy did not improve from 0.85845
Epoch 13/300
53/53 [==============================] - 25s 479ms/step - loss: 0.4509 - accu
racy: 0.8154 - val_loss: 0.4043 - val_accuracy: 0.8356

Epoch 00013: val_accuracy did not improve from 0.85845
Epoch 14/300
53/53 [==============================] - 26s 485ms/step - loss: 0.4337 - accu
racy: 0.8172 - val_loss: 0.4072 - val_accuracy: 0.8630

Epoch 00014: val_accuracy improved from 0.85845 to 0.86301, saving model to /
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 15/300
53/53 [==============================] - 27s 516ms/step - loss: 0.4398 - accu
racy: 0.8284 - val_loss: 0.3272 - val_accuracy: 0.8630

Epoch 00015: val_accuracy did not improve from 0.86301
Epoch 16/300
53/53 [==============================] - 26s 483ms/step - loss: 0.4151 - accu
racy: 0.8272 - val_loss: 0.3473 - val_accuracy: 0.8402

Epoch 00016: val_accuracy did not improve from 0.86301
Epoch 17/300
53/53 [==============================] - 26s 483ms/step - loss: 0.4173 - accu
racy: 0.8225 - val_loss: 0.4213 - val_accuracy: 0.7945

Epoch 00017: val_accuracy did not improve from 0.86301
Epoch 18/300
53/53 [==============================] - 26s 481ms/step - loss: 0.3836 - accu
racy: 0.8267 - val_loss: 0.3546 - val_accuracy: 0.8676

Epoch 00018: val_accuracy improved from 0.86301 to 0.86758, saving model to /
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 19/300
53/53 [==============================] - 28s 521ms/step - loss: 0.3845 - accu
racy: 0.8355 - val_loss: 0.3774 - val_accuracy: 0.8174

Epoch 00019: val_accuracy did not improve from 0.86758
Epoch 20/300
```

```
53/53 [==============================] - 25s 480ms/step - loss: 0.3680 - accu
racy: 0.8461 - val_loss: 0.3150 - val_accuracy: 0.8676

Epoch 00020: val_accuracy did not improve from 0.86758
Epoch 21/300
53/53 [==============================] - 26s 484ms/step - loss: 0.3670 - accu
racy: 0.8502 - val_loss: 0.4015 - val_accuracy: 0.8128

Epoch 00021: val_accuracy did not improve from 0.86758
Epoch 22/300
53/53 [==============================] - 25s 468ms/step - loss: 0.3408 - accu
racy: 0.8550 - val_loss: 0.3752 - val_accuracy: 0.8128

Epoch 00022: val_accuracy did not improve from 0.86758
Epoch 23/300
53/53 [==============================] - 25s 477ms/step - loss: 0.3591 - accu
racy: 0.8432 - val_loss: 0.3668 - val_accuracy: 0.8082

Epoch 00023: val_accuracy did not improve from 0.86758
Epoch 24/300
53/53 [==============================] - 25s 467ms/step - loss: 0.3651 - accu
racy: 0.8467 - val_loss: 0.3646 - val_accuracy: 0.8356

Epoch 00024: val_accuracy did not improve from 0.86758
Epoch 25/300
53/53 [==============================] - 25s 476ms/step - loss: 0.3304 - accu
racy: 0.8697 - val_loss: 0.3944 - val_accuracy: 0.8265

Epoch 00025: val_accuracy did not improve from 0.86758
Epoch 26/300
53/53 [==============================] - 25s 473ms/step - loss: 0.3190 - accu
racy: 0.8603 - val_loss: 0.4033 - val_accuracy: 0.8265

Epoch 00026: val_accuracy did not improve from 0.86758
Epoch 27/300
53/53 [==============================] - 25s 479ms/step - loss: 0.3567 - accu
racy: 0.8485 - val_loss: 0.4195 - val_accuracy: 0.7808

Epoch 00027: val_accuracy did not improve from 0.86758
Epoch 28/300
53/53 [==============================] - 25s 468ms/step - loss: 0.3617 - accu
racy: 0.8538 - val_loss: 0.4790 - val_accuracy: 0.7808

Epoch 00028: val_accuracy did not improve from 0.86758
Epoch 29/300
53/53 [==============================] - 25s 480ms/step - loss: 0.3395 - accu
racy: 0.8520 - val_loss: 0.4143 - val_accuracy: 0.8402

Epoch 00029: val_accuracy did not improve from 0.86758
Epoch 30/300
53/53 [==============================] - 25s 471ms/step - loss: 0.3366 - accu
racy: 0.8508 - val_loss: 0.3698 - val_accuracy: 0.8584

Epoch 00030: val_accuracy did not improve from 0.86758
Epoch 31/300
53/53 [==============================] - 26s 483ms/step - loss: 0.3426 - accu
racy: 0.8591 - val_loss: 0.3629 - val_accuracy: 0.8447

Epoch 00031: val_accuracy did not improve from 0.86758
Epoch 32/300
53/53 [==============================] - 25s 477ms/step - loss: 0.3291 - accu
racy: 0.8591 - val_loss: 0.4151 - val_accuracy: 0.8082

Epoch 00032: val_accuracy did not improve from 0.86758
Epoch 33/300
53/53 [==============================] - 25s 467ms/step - loss: 0.3133 - accu
racy: 0.8667 - val_loss: 0.3093 - val_accuracy: 0.8767

Epoch 00033: val_accuracy improved from 0.86758 to 0.87671, saving model to /
```

```
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 34/300
53/53 [==============================] - 27s 506ms/step - loss: 0.3400 - accu
racy: 0.8491 - val_loss: 0.3271 - val_accuracy: 0.8721

Epoch 00034: val_accuracy did not improve from 0.87671
Epoch 35/300
53/53 [==============================] - 25s 466ms/step - loss: 0.3371 - accu
racy: 0.8550 - val_loss: 0.4928 - val_accuracy: 0.7808

Epoch 00035: val_accuracy did not improve from 0.87671
Epoch 36/300
53/53 [==============================] - 25s 470ms/step - loss: 0.3361 - accu
racy: 0.8691 - val_loss: 0.4055 - val_accuracy: 0.7945

Epoch 00036: val_accuracy did not improve from 0.87671
Epoch 37/300
53/53 [==============================] - 25s 466ms/step - loss: 0.3388 - accu
racy: 0.8608 - val_loss: 0.3902 - val_accuracy: 0.8128

Epoch 00037: val_accuracy did not improve from 0.87671
Epoch 38/300
53/53 [==============================] - 25s 475ms/step - loss: 0.3068 - accu
racy: 0.8620 - val_loss: 0.3068 - val_accuracy: 0.8630

Epoch 00038: val_accuracy did not improve from 0.87671
Epoch 39/300
53/53 [==============================] - 25s 472ms/step - loss: 0.3095 - accu
racy: 0.8650 - val_loss: 0.3941 - val_accuracy: 0.8265

Epoch 00039: val_accuracy did not improve from 0.87671
Epoch 40/300
53/53 [==============================] - 26s 481ms/step - loss: 0.3200 - accu
racy: 0.8608 - val_loss: 0.4861 - val_accuracy: 0.7489

Epoch 00040: val_accuracy did not improve from 0.87671
Epoch 41/300
53/53 [==============================] - 25s 466ms/step - loss: 0.3492 - accu
racy: 0.8449 - val_loss: 0.2998 - val_accuracy: 0.8676

Epoch 00041: val_accuracy did not improve from 0.87671
Epoch 42/300
53/53 [==============================] - 25s 475ms/step - loss: 0.3437 - accu
racy: 0.8644 - val_loss: 0.3392 - val_accuracy: 0.8311

Epoch 00042: val_accuracy did not improve from 0.87671
Epoch 43/300
53/53 [==============================] - 25s 467ms/step - loss: 0.3153 - accu
racy: 0.8726 - val_loss: 0.4536 - val_accuracy: 0.7854

Epoch 00043: val_accuracy did not improve from 0.87671
Epoch 44/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2913 - accu
racy: 0.8756 - val_loss: 0.3335 - val_accuracy: 0.8539

Epoch 00044: val_accuracy did not improve from 0.87671
Epoch 45/300
53/53 [==============================] - 26s 484ms/step - loss: 0.3043 - accu
racy: 0.8768 - val_loss: 0.4528 - val_accuracy: 0.7854

Epoch 00045: val_accuracy did not improve from 0.87671
Epoch 46/300
53/53 [==============================] - 26s 489ms/step - loss: 0.3234 - accu
racy: 0.8715 - val_loss: 0.4942 - val_accuracy: 0.7352

Epoch 00046: val_accuracy did not improve from 0.87671
Epoch 47/300
53/53 [==============================] - 26s 480ms/step - loss: 0.3264 - accu
racy: 0.8608 - val_loss: 0.3154 - val_accuracy: 0.8402
```

```
Epoch 00047: val_accuracy did not improve from 0.87671
Epoch 48/300
53/53 [==============================] - 25s 471ms/step - loss: 0.3056 - accu
racy: 0.8662 - val_loss: 0.2171 - val_accuracy: 0.9178

Epoch 00048: val_accuracy improved from 0.87671 to 0.91781, saving model to /
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 49/300
53/53 [==============================] - 27s 514ms/step - loss: 0.3250 - accu
racy: 0.8644 - val_loss: 0.3127 - val_accuracy: 0.8767

Epoch 00049: val_accuracy did not improve from 0.91781
Epoch 50/300
53/53 [==============================] - 25s 474ms/step - loss: 0.3183 - accu
racy: 0.8620 - val_loss: 0.3074 - val_accuracy: 0.8630

Epoch 00050: val_accuracy did not improve from 0.91781
Epoch 51/300
53/53 [==============================] - 25s 475ms/step - loss: 0.3198 - accu
racy: 0.8662 - val_loss: 0.2487 - val_accuracy: 0.8995

Epoch 00051: val_accuracy did not improve from 0.91781
Epoch 52/300
53/53 [==============================] - 26s 482ms/step - loss: 0.3153 - accu
racy: 0.8703 - val_loss: 0.3249 - val_accuracy: 0.8447

Epoch 00052: val_accuracy did not improve from 0.91781
Epoch 53/300
53/53 [==============================] - 25s 466ms/step - loss: 0.3036 - accu
racy: 0.8809 - val_loss: 0.3550 - val_accuracy: 0.8219

Epoch 00053: val_accuracy did not improve from 0.91781
Epoch 54/300
53/53 [==============================] - 25s 473ms/step - loss: 0.3019 - accu
racy: 0.8697 - val_loss: 0.3378 - val_accuracy: 0.8447

Epoch 00054: val_accuracy did not improve from 0.91781
Epoch 55/300
53/53 [==============================] - 25s 467ms/step - loss: 0.3132 - accu
racy: 0.8632 - val_loss: 0.4078 - val_accuracy: 0.7900

Epoch 00055: val_accuracy did not improve from 0.91781
Epoch 56/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2979 - accu
racy: 0.8779 - val_loss: 0.4717 - val_accuracy: 0.7397

Epoch 00056: val_accuracy did not improve from 0.91781
Epoch 57/300
53/53 [==============================] - 25s 480ms/step - loss: 0.3303 - accu
racy: 0.8579 - val_loss: 0.3125 - val_accuracy: 0.8539

Epoch 00057: val_accuracy did not improve from 0.91781
Epoch 58/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2854 - accu
racy: 0.8815 - val_loss: 0.3962 - val_accuracy: 0.7991

Epoch 00058: val_accuracy did not improve from 0.91781
Epoch 59/300
53/53 [==============================] - 24s 460ms/step - loss: 0.2795 - accu
racy: 0.8821 - val_loss: 0.3754 - val_accuracy: 0.8402

Epoch 00059: val_accuracy did not improve from 0.91781
Epoch 60/300
53/53 [==============================] - 25s 465ms/step - loss: 0.3024 - accu
racy: 0.8715 - val_loss: 0.3455 - val_accuracy: 0.8082

Epoch 00060: val_accuracy did not improve from 0.91781
Epoch 61/300
```

```
53/53 [==============================] - 25s 464ms/step - loss: 0.2938 - accu
racy: 0.8797 - val_loss: 0.5841 - val_accuracy: 0.6484

Epoch 00061: val_accuracy did not improve from 0.91781
Epoch 62/300
53/53 [==============================] - 25s 466ms/step - loss: 0.3034 - accu
racy: 0.8768 - val_loss: 0.5230 - val_accuracy: 0.7032

Epoch 00062: val_accuracy did not improve from 0.91781
Epoch 63/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2928 - accu
racy: 0.8756 - val_loss: 0.3592 - val_accuracy: 0.8265

Epoch 00063: val_accuracy did not improve from 0.91781
Epoch 64/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2901 - accu
racy: 0.8697 - val_loss: 0.6846 - val_accuracy: 0.6986

Epoch 00064: val_accuracy did not improve from 0.91781
Epoch 65/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2891 - accu
racy: 0.8721 - val_loss: 0.3183 - val_accuracy: 0.8721

Epoch 00065: val_accuracy did not improve from 0.91781
Epoch 66/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2974 - accu
racy: 0.8862 - val_loss: 0.7295 - val_accuracy: 0.6347

Epoch 00066: val_accuracy did not improve from 0.91781
Epoch 67/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2936 - accu
racy: 0.8750 - val_loss: 0.3662 - val_accuracy: 0.8311

Epoch 00067: val_accuracy did not improve from 0.91781
Epoch 68/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2963 - accu
racy: 0.8638 - val_loss: 0.3245 - val_accuracy: 0.8630

Epoch 00068: val_accuracy did not improve from 0.91781
Epoch 69/300
53/53 [==============================] - 25s 467ms/step - loss: 0.3028 - accu
racy: 0.8721 - val_loss: 0.3662 - val_accuracy: 0.8174

Epoch 00069: val_accuracy did not improve from 0.91781
Epoch 70/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2782 - accu
racy: 0.8756 - val_loss: 0.5506 - val_accuracy: 0.6895

Epoch 00070: val_accuracy did not improve from 0.91781
Epoch 71/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2833 - accu
racy: 0.8862 - val_loss: 0.5050 - val_accuracy: 0.7352

Epoch 00071: val_accuracy did not improve from 0.91781
Epoch 72/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2830 - accu
racy: 0.8756 - val_loss: 0.2905 - val_accuracy: 0.8630

Epoch 00072: val_accuracy did not improve from 0.91781
Epoch 73/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2663 - accu
racy: 0.8950 - val_loss: 0.3810 - val_accuracy: 0.8174

Epoch 00073: val_accuracy did not improve from 0.91781
Epoch 74/300
53/53 [==============================] - 25s 468ms/step - loss: 0.3021 - accu
racy: 0.8815 - val_loss: 0.3218 - val_accuracy: 0.8447

Epoch 00074: val_accuracy did not improve from 0.91781
```

```
Epoch 75/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2771 - accu
racy: 0.8815 - val_loss: 0.6737 - val_accuracy: 0.6530

Epoch 00075: val_accuracy did not improve from 0.91781
Epoch 76/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2800 - accu
racy: 0.8856 - val_loss: 0.6239 - val_accuracy: 0.6530

Epoch 00076: val_accuracy did not improve from 0.91781
Epoch 77/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2785 - accu
racy: 0.8809 - val_loss: 0.5301 - val_accuracy: 0.6849

Epoch 00077: val_accuracy did not improve from 0.91781
Epoch 78/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2880 - accu
racy: 0.8809 - val_loss: 0.3621 - val_accuracy: 0.8219

Epoch 00078: val_accuracy did not improve from 0.91781
Epoch 79/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2981 - accu
racy: 0.8785 - val_loss: 0.3261 - val_accuracy: 0.8311

Epoch 00079: val_accuracy did not improve from 0.91781
Epoch 80/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2858 - accu
racy: 0.8862 - val_loss: 0.2814 - val_accuracy: 0.9224

Epoch 00080: val_accuracy improved from 0.91781 to 0.92237, saving model to /
content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5
Epoch 81/300
53/53 [==============================] - 27s 505ms/step - loss: 0.2807 - accu
racy: 0.8791 - val_loss: 0.4087 - val_accuracy: 0.7854

Epoch 00081: val_accuracy did not improve from 0.92237
Epoch 82/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2791 - accu
racy: 0.8892 - val_loss: 0.4820 - val_accuracy: 0.8037

Epoch 00082: val_accuracy did not improve from 0.92237
Epoch 83/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2771 - accu
racy: 0.8833 - val_loss: 0.3238 - val_accuracy: 0.8356

Epoch 00083: val_accuracy did not improve from 0.92237
Epoch 84/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2841 - accu
racy: 0.8915 - val_loss: 0.4974 - val_accuracy: 0.7397

Epoch 00084: val_accuracy did not improve from 0.92237
Epoch 85/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2962 - accu
racy: 0.8803 - val_loss: 0.4737 - val_accuracy: 0.7352

Epoch 00085: val_accuracy did not improve from 0.92237
Epoch 86/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2715 - accu
racy: 0.8821 - val_loss: 0.3620 - val_accuracy: 0.8493

Epoch 00086: val_accuracy did not improve from 0.92237
Epoch 87/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2820 - accu
racy: 0.8815 - val_loss: 0.4064 - val_accuracy: 0.7900

Epoch 00087: val_accuracy did not improve from 0.92237
Epoch 88/300
53/53 [==============================] - 25s 465ms/step - loss: 0.3084 - accu
racy: 0.8685 - val_loss: 0.4226 - val_accuracy: 0.7900
```

```
Epoch 00088: val_accuracy did not improve from 0.92237
Epoch 89/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2837 - accu
racy: 0.8779 - val_loss: 0.3675 - val_accuracy: 0.8402

Epoch 00089: val_accuracy did not improve from 0.92237
Epoch 90/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2677 - accu
racy: 0.8909 - val_loss: 0.5678 - val_accuracy: 0.7215

Epoch 00090: val_accuracy did not improve from 0.92237
Epoch 91/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2581 - accu
racy: 0.8974 - val_loss: 0.5760 - val_accuracy: 0.7078

Epoch 00091: val_accuracy did not improve from 0.92237
Epoch 92/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2601 - accu
racy: 0.8897 - val_loss: 0.3721 - val_accuracy: 0.8082

Epoch 00092: val_accuracy did not improve from 0.92237
Epoch 93/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2893 - accu
racy: 0.8897 - val_loss: 0.2800 - val_accuracy: 0.8858

Epoch 00093: val_accuracy did not improve from 0.92237
Epoch 94/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2855 - accu
racy: 0.8779 - val_loss: 0.3243 - val_accuracy: 0.8493

Epoch 00094: val_accuracy did not improve from 0.92237
Epoch 95/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2730 - accu
racy: 0.8903 - val_loss: 0.2947 - val_accuracy: 0.8630

Epoch 00095: val_accuracy did not improve from 0.92237
Epoch 96/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2815 - accu
racy: 0.8850 - val_loss: 0.4364 - val_accuracy: 0.7671

Epoch 00096: val_accuracy did not improve from 0.92237
Epoch 97/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2768 - accu
racy: 0.8874 - val_loss: 0.3101 - val_accuracy: 0.8813

Epoch 00097: val_accuracy did not improve from 0.92237
Epoch 98/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2801 - accu
racy: 0.8886 - val_loss: 0.5564 - val_accuracy: 0.6986

Epoch 00098: val_accuracy did not improve from 0.92237
Epoch 99/300
53/53 [==============================] - 26s 483ms/step - loss: 0.2525 - accu
racy: 0.8974 - val_loss: 0.4565 - val_accuracy: 0.7671

Epoch 00099: val_accuracy did not improve from 0.92237
Epoch 100/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2612 - accu
racy: 0.8886 - val_loss: 0.3669 - val_accuracy: 0.8311

Epoch 00100: val_accuracy did not improve from 0.92237
Epoch 101/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2581 - accu
racy: 0.8868 - val_loss: 0.6498 - val_accuracy: 0.6941

Epoch 00101: val_accuracy did not improve from 0.92237
Epoch 102/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2848 - accu
```

```
racy: 0.8862 - val_loss: 0.5386 - val_accuracy: 0.7078

Epoch 00102: val_accuracy did not improve from 0.92237
Epoch 103/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2771 - accu
racy: 0.8850 - val_loss: 0.8173 - val_accuracy: 0.6256

Epoch 00103: val_accuracy did not improve from 0.92237
Epoch 104/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2723 - accu
racy: 0.8980 - val_loss: 0.4681 - val_accuracy: 0.7626

Epoch 00104: val_accuracy did not improve from 0.92237
Epoch 105/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2836 - accu
racy: 0.8903 - val_loss: 0.2430 - val_accuracy: 0.9087

Epoch 00105: val_accuracy did not improve from 0.92237
Epoch 106/300
53/53 [==============================] - 24s 459ms/step - loss: 0.2623 - accu
racy: 0.8927 - val_loss: 0.3895 - val_accuracy: 0.8082

Epoch 00106: val_accuracy did not improve from 0.92237
Epoch 107/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2652 - accu
racy: 0.8856 - val_loss: 0.4670 - val_accuracy: 0.7352

Epoch 00107: val_accuracy did not improve from 0.92237
Epoch 108/300
53/53 [==============================] - 24s 458ms/step - loss: 0.2654 - accu
racy: 0.8921 - val_loss: 0.6170 - val_accuracy: 0.6941

Epoch 00108: val_accuracy did not improve from 0.92237
Epoch 109/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2720 - accu
racy: 0.8844 - val_loss: 0.2706 - val_accuracy: 0.8858

Epoch 00109: val_accuracy did not improve from 0.92237
Epoch 110/300
53/53 [==============================] - 24s 456ms/step - loss: 0.2611 - accu
racy: 0.8939 - val_loss: 0.3283 - val_accuracy: 0.8493

Epoch 00110: val_accuracy did not improve from 0.92237
Epoch 111/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2300 - accu
racy: 0.9068 - val_loss: 0.4599 - val_accuracy: 0.7854

Epoch 00111: val_accuracy did not improve from 0.92237
Epoch 112/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2528 - accu
racy: 0.8921 - val_loss: 0.4782 - val_accuracy: 0.7489

Epoch 00112: val_accuracy did not improve from 0.92237
Epoch 113/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2696 - accu
racy: 0.8892 - val_loss: 0.4700 - val_accuracy: 0.7397

Epoch 00113: val_accuracy did not improve from 0.92237
Epoch 114/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2711 - accu
racy: 0.8909 - val_loss: 0.4744 - val_accuracy: 0.7763

Epoch 00114: val_accuracy did not improve from 0.92237
Epoch 115/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2385 - accu
racy: 0.8921 - val_loss: 0.3975 - val_accuracy: 0.8128

Epoch 00115: val_accuracy did not improve from 0.92237
Epoch 116/300
```

```
53/53 [==============================] - 25s 469ms/step - loss: 0.2617 - accu
racy: 0.8892 - val_loss: 0.4046 - val_accuracy: 0.8082

Epoch 00116: val_accuracy did not improve from 0.92237
Epoch 117/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2436 - accu
racy: 0.8998 - val_loss: 0.4979 - val_accuracy: 0.7443

Epoch 00117: val_accuracy did not improve from 0.92237
Epoch 118/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2539 - accu
racy: 0.8962 - val_loss: 0.5922 - val_accuracy: 0.7078

Epoch 00118: val_accuracy did not improve from 0.92237
Epoch 119/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2500 - accu
racy: 0.8927 - val_loss: 0.5498 - val_accuracy: 0.7123

Epoch 00119: val_accuracy did not improve from 0.92237
Epoch 120/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2677 - accu
racy: 0.8950 - val_loss: 0.4502 - val_accuracy: 0.7580

Epoch 00120: val_accuracy did not improve from 0.92237
Epoch 121/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2479 - accu
racy: 0.8945 - val_loss: 0.3786 - val_accuracy: 0.8174

Epoch 00121: val_accuracy did not improve from 0.92237
Epoch 122/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2689 - accu
racy: 0.8856 - val_loss: 0.4813 - val_accuracy: 0.7626

Epoch 00122: val_accuracy did not improve from 0.92237
Epoch 123/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2248 - accu
racy: 0.9098 - val_loss: 0.4764 - val_accuracy: 0.7534

Epoch 00123: val_accuracy did not improve from 0.92237
Epoch 124/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2669 - accu
racy: 0.8921 - val_loss: 0.4779 - val_accuracy: 0.7626

Epoch 00124: val_accuracy did not improve from 0.92237
Epoch 125/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2435 - accu
racy: 0.8986 - val_loss: 0.4090 - val_accuracy: 0.7945

Epoch 00125: val_accuracy did not improve from 0.92237
Epoch 126/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2376 - accu
racy: 0.8974 - val_loss: 0.4025 - val_accuracy: 0.8174

Epoch 00126: val_accuracy did not improve from 0.92237
Epoch 127/300
53/53 [==============================] - 26s 480ms/step - loss: 0.2412 - accu
racy: 0.8986 - val_loss: 0.4407 - val_accuracy: 0.7854

Epoch 00127: val_accuracy did not improve from 0.92237
Epoch 128/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2279 - accu
racy: 0.9057 - val_loss: 0.3586 - val_accuracy: 0.8447

Epoch 00128: val_accuracy did not improve from 0.92237
Epoch 129/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2862 - accu
racy: 0.8797 - val_loss: 0.3778 - val_accuracy: 0.8311

Epoch 00129: val_accuracy did not improve from 0.92237
```

```
Epoch 130/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2502 - accu
racy: 0.9004 - val_loss: 0.5412 - val_accuracy: 0.7352

Epoch 00130: val_accuracy did not improve from 0.92237
Epoch 131/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2721 - accu
racy: 0.8844 - val_loss: 0.3450 - val_accuracy: 0.8402

Epoch 00131: val_accuracy did not improve from 0.92237
Epoch 132/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2642 - accu
racy: 0.8903 - val_loss: 0.4415 - val_accuracy: 0.7580

Epoch 00132: val_accuracy did not improve from 0.92237
Epoch 133/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2576 - accu
racy: 0.9004 - val_loss: 0.4513 - val_accuracy: 0.7671

Epoch 00133: val_accuracy did not improve from 0.92237
Epoch 134/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2481 - accu
racy: 0.8950 - val_loss: 0.4457 - val_accuracy: 0.7808

Epoch 00134: val_accuracy did not improve from 0.92237
Epoch 135/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2569 - accu
racy: 0.8956 - val_loss: 0.3682 - val_accuracy: 0.8311

Epoch 00135: val_accuracy did not improve from 0.92237
Epoch 136/300
53/53 [==============================] - 26s 486ms/step - loss: 0.2431 - accu
racy: 0.9086 - val_loss: 0.3657 - val_accuracy: 0.8356

Epoch 00136: val_accuracy did not improve from 0.92237
Epoch 137/300
53/53 [==============================] - 26s 483ms/step - loss: 0.2379 - accu
racy: 0.8998 - val_loss: 0.5002 - val_accuracy: 0.7580

Epoch 00137: val_accuracy did not improve from 0.92237
Epoch 138/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2593 - accu
racy: 0.8992 - val_loss: 0.4681 - val_accuracy: 0.7534

Epoch 00138: val_accuracy did not improve from 0.92237
Epoch 139/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2280 - accu
racy: 0.9027 - val_loss: 0.4271 - val_accuracy: 0.7808

Epoch 00139: val_accuracy did not improve from 0.92237
Epoch 140/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2491 - accu
racy: 0.8956 - val_loss: 0.4221 - val_accuracy: 0.7900

Epoch 00140: val_accuracy did not improve from 0.92237
Epoch 141/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2461 - accu
racy: 0.9004 - val_loss: 0.3653 - val_accuracy: 0.8219

Epoch 00141: val_accuracy did not improve from 0.92237
Epoch 142/300
53/53 [==============================] - 26s 480ms/step - loss: 0.2373 - accu
racy: 0.9021 - val_loss: 0.5245 - val_accuracy: 0.7215

Epoch 00142: val_accuracy did not improve from 0.92237
Epoch 143/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2439 - accu
racy: 0.8945 - val_loss: 0.5764 - val_accuracy: 0.6986
```

```
Epoch 00143: val_accuracy did not improve from 0.92237
Epoch 144/300
53/53 [==============================] - 26s 491ms/step - loss: 0.2549 - accu
racy: 0.8956 - val_loss: 0.5093 - val_accuracy: 0.7306

Epoch 00144: val_accuracy did not improve from 0.92237
Epoch 145/300
53/53 [==============================] - 26s 488ms/step - loss: 0.2508 - accu
racy: 0.8927 - val_loss: 0.5888 - val_accuracy: 0.6804

Epoch 00145: val_accuracy did not improve from 0.92237
Epoch 146/300
53/53 [==============================] - 26s 485ms/step - loss: 0.2339 - accu
racy: 0.9092 - val_loss: 0.5242 - val_accuracy: 0.7306

Epoch 00146: val_accuracy did not improve from 0.92237
Epoch 147/300
53/53 [==============================] - 26s 483ms/step - loss: 0.2560 - accu
racy: 0.8844 - val_loss: 0.5094 - val_accuracy: 0.7306

Epoch 00147: val_accuracy did not improve from 0.92237
Epoch 148/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2484 - accu
racy: 0.8915 - val_loss: 0.3015 - val_accuracy: 0.8904

Epoch 00148: val_accuracy did not improve from 0.92237
Epoch 149/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2240 - accu
racy: 0.9133 - val_loss: 0.5347 - val_accuracy: 0.7260

Epoch 00149: val_accuracy did not improve from 0.92237
Epoch 150/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2691 - accu
racy: 0.8921 - val_loss: 0.5017 - val_accuracy: 0.7306

Epoch 00150: val_accuracy did not improve from 0.92237
Epoch 151/300
53/53 [==============================] - 26s 482ms/step - loss: 0.2265 - accu
racy: 0.9039 - val_loss: 0.4402 - val_accuracy: 0.7671

Epoch 00151: val_accuracy did not improve from 0.92237
Epoch 152/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2427 - accu
racy: 0.9057 - val_loss: 0.6179 - val_accuracy: 0.6849

Epoch 00152: val_accuracy did not improve from 0.92237
Epoch 153/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2224 - accu
racy: 0.9133 - val_loss: 0.5036 - val_accuracy: 0.7626

Epoch 00153: val_accuracy did not improve from 0.92237
Epoch 154/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2476 - accu
racy: 0.9033 - val_loss: 0.3935 - val_accuracy: 0.8082

Epoch 00154: val_accuracy did not improve from 0.92237
Epoch 155/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2624 - accu
racy: 0.8921 - val_loss: 0.4217 - val_accuracy: 0.7808

Epoch 00155: val_accuracy did not improve from 0.92237
Epoch 156/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2500 - accu
racy: 0.9057 - val_loss: 0.3737 - val_accuracy: 0.8082

Epoch 00156: val_accuracy did not improve from 0.92237
Epoch 157/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2199 - accu
racy: 0.9116 - val_loss: 0.3191 - val_accuracy: 0.8767
```

```
Epoch 00157: val_accuracy did not improve from 0.92237
Epoch 158/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2658 - accu
racy: 0.8968 - val_loss: 0.6796 - val_accuracy: 0.6895

Epoch 00158: val_accuracy did not improve from 0.92237
Epoch 159/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2738 - accu
racy: 0.8915 - val_loss: 0.4266 - val_accuracy: 0.7808

Epoch 00159: val_accuracy did not improve from 0.92237
Epoch 160/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2450 - accu
racy: 0.9062 - val_loss: 0.5537 - val_accuracy: 0.7078

Epoch 00160: val_accuracy did not improve from 0.92237
Epoch 161/300
53/53 [==============================] - 26s 497ms/step - loss: 0.2232 - accu
racy: 0.9116 - val_loss: 0.5952 - val_accuracy: 0.7123

Epoch 00161: val_accuracy did not improve from 0.92237
Epoch 162/300
53/53 [==============================] - 26s 494ms/step - loss: 0.2533 - accu
racy: 0.8968 - val_loss: 0.4751 - val_accuracy: 0.7991

Epoch 00162: val_accuracy did not improve from 0.92237
Epoch 163/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2541 - accu
racy: 0.8921 - val_loss: 0.3594 - val_accuracy: 0.8037

Epoch 00163: val_accuracy did not improve from 0.92237
Epoch 164/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2538 - accu
racy: 0.8992 - val_loss: 0.3309 - val_accuracy: 0.8356

Epoch 00164: val_accuracy did not improve from 0.92237
Epoch 165/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2296 - accu
racy: 0.9104 - val_loss: 0.3674 - val_accuracy: 0.8037

Epoch 00165: val_accuracy did not improve from 0.92237
Epoch 166/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2345 - accu
racy: 0.9057 - val_loss: 0.4728 - val_accuracy: 0.7717

Epoch 00166: val_accuracy did not improve from 0.92237
Epoch 167/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2436 - accu
racy: 0.9015 - val_loss: 0.4969 - val_accuracy: 0.7489

Epoch 00167: val_accuracy did not improve from 0.92237
Epoch 168/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2331 - accu
racy: 0.8986 - val_loss: 0.4222 - val_accuracy: 0.7945

Epoch 00168: val_accuracy did not improve from 0.92237
Epoch 169/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2377 - accu
racy: 0.9051 - val_loss: 0.5893 - val_accuracy: 0.7078

Epoch 00169: val_accuracy did not improve from 0.92237
Epoch 170/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2357 - accu
racy: 0.9039 - val_loss: 0.5814 - val_accuracy: 0.7078

Epoch 00170: val_accuracy did not improve from 0.92237
Epoch 171/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2196 - accu
```

```
racy: 0.9098 - val_loss: 0.6641 - val_accuracy: 0.6758

Epoch 00171: val_accuracy did not improve from 0.92237
Epoch 172/300
53/53 [==============================] - 25s 481ms/step - loss: 0.2508 - accu
racy: 0.8998 - val_loss: 0.4717 - val_accuracy: 0.7671

Epoch 00172: val_accuracy did not improve from 0.92237
Epoch 173/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2301 - accu
racy: 0.9110 - val_loss: 0.5439 - val_accuracy: 0.7397

Epoch 00173: val_accuracy did not improve from 0.92237
Epoch 174/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2054 - accu
racy: 0.9151 - val_loss: 0.3917 - val_accuracy: 0.8082

Epoch 00174: val_accuracy did not improve from 0.92237
Epoch 175/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2200 - accu
racy: 0.9051 - val_loss: 0.6435 - val_accuracy: 0.7169

Epoch 00175: val_accuracy did not improve from 0.92237
Epoch 176/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2549 - accu
racy: 0.9021 - val_loss: 0.6182 - val_accuracy: 0.6941

Epoch 00176: val_accuracy did not improve from 0.92237
Epoch 177/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2375 - accu
racy: 0.9092 - val_loss: 0.5875 - val_accuracy: 0.7215

Epoch 00177: val_accuracy did not improve from 0.92237
Epoch 178/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2391 - accu
racy: 0.9033 - val_loss: 0.6015 - val_accuracy: 0.6895

Epoch 00178: val_accuracy did not improve from 0.92237
Epoch 179/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2474 - accu
racy: 0.8986 - val_loss: 0.6155 - val_accuracy: 0.6941

Epoch 00179: val_accuracy did not improve from 0.92237
Epoch 180/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2568 - accu
racy: 0.8980 - val_loss: 0.4473 - val_accuracy: 0.7763

Epoch 00180: val_accuracy did not improve from 0.92237
Epoch 181/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2436 - accu
racy: 0.9051 - val_loss: 0.6871 - val_accuracy: 0.6712

Epoch 00181: val_accuracy did not improve from 0.92237
Epoch 182/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2309 - accu
racy: 0.9057 - val_loss: 0.7686 - val_accuracy: 0.6758

Epoch 00182: val_accuracy did not improve from 0.92237
Epoch 183/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2180 - accu
racy: 0.9145 - val_loss: 0.3813 - val_accuracy: 0.8082

Epoch 00183: val_accuracy did not improve from 0.92237
Epoch 184/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2193 - accu
racy: 0.9021 - val_loss: 0.5587 - val_accuracy: 0.7306

Epoch 00184: val_accuracy did not improve from 0.92237
Epoch 185/300
```

```
53/53 [==============================] - 25s 463ms/step - loss: 0.2320 - accu
racy: 0.9139 - val_loss: 0.7363 - val_accuracy: 0.6849

Epoch 00185: val_accuracy did not improve from 0.92237
Epoch 186/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2362 - accu
racy: 0.9092 - val_loss: 0.7220 - val_accuracy: 0.6849

Epoch 00186: val_accuracy did not improve from 0.92237
Epoch 187/300
53/53 [==============================] - 24s 459ms/step - loss: 0.2402 - accu
racy: 0.9051 - val_loss: 0.8684 - val_accuracy: 0.6530

Epoch 00187: val_accuracy did not improve from 0.92237
Epoch 188/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2314 - accu
racy: 0.9027 - val_loss: 0.4480 - val_accuracy: 0.7991

Epoch 00188: val_accuracy did not improve from 0.92237
Epoch 189/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2302 - accu
racy: 0.9015 - val_loss: 0.7262 - val_accuracy: 0.6575

Epoch 00189: val_accuracy did not improve from 0.92237
Epoch 190/300
53/53 [==============================] - 24s 459ms/step - loss: 0.2247 - accu
racy: 0.9127 - val_loss: 0.5803 - val_accuracy: 0.7078

Epoch 00190: val_accuracy did not improve from 0.92237
Epoch 191/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2374 - accu
racy: 0.9009 - val_loss: 0.6379 - val_accuracy: 0.7032

Epoch 00191: val_accuracy did not improve from 0.92237
Epoch 192/300
53/53 [==============================] - 24s 460ms/step - loss: 0.2298 - accu
racy: 0.9104 - val_loss: 0.5414 - val_accuracy: 0.7489

Epoch 00192: val_accuracy did not improve from 0.92237
Epoch 193/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2445 - accu
racy: 0.9004 - val_loss: 0.4490 - val_accuracy: 0.7763

Epoch 00193: val_accuracy did not improve from 0.92237
Epoch 194/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2096 - accu
racy: 0.9145 - val_loss: 0.5603 - val_accuracy: 0.7489

Epoch 00194: val_accuracy did not improve from 0.92237
Epoch 195/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2538 - accu
racy: 0.8980 - val_loss: 0.5546 - val_accuracy: 0.7260

Epoch 00195: val_accuracy did not improve from 0.92237
Epoch 196/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2314 - accu
racy: 0.9098 - val_loss: 0.6443 - val_accuracy: 0.6895

Epoch 00196: val_accuracy did not improve from 0.92237
Epoch 197/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2176 - accu
racy: 0.9157 - val_loss: 0.5552 - val_accuracy: 0.7123

Epoch 00197: val_accuracy did not improve from 0.92237
Epoch 198/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2206 - accu
racy: 0.9045 - val_loss: 0.4158 - val_accuracy: 0.7900

Epoch 00198: val_accuracy did not improve from 0.92237
```

```
Epoch 199/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2578 - accu
racy: 0.8980 - val_loss: 0.5791 - val_accuracy: 0.7260

Epoch 00199: val_accuracy did not improve from 0.92237
Epoch 200/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2151 - accu
racy: 0.9169 - val_loss: 0.6413 - val_accuracy: 0.7078

Epoch 00200: val_accuracy did not improve from 0.92237
Epoch 201/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2175 - accu
racy: 0.9169 - val_loss: 0.3564 - val_accuracy: 0.8356

Epoch 00201: val_accuracy did not improve from 0.92237
Epoch 202/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2404 - accu
racy: 0.9027 - val_loss: 0.4240 - val_accuracy: 0.7854

Epoch 00202: val_accuracy did not improve from 0.92237
Epoch 203/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2252 - accu
racy: 0.9051 - val_loss: 0.5750 - val_accuracy: 0.7260

Epoch 00203: val_accuracy did not improve from 0.92237
Epoch 204/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2364 - accu
racy: 0.9004 - val_loss: 0.4742 - val_accuracy: 0.7443

Epoch 00204: val_accuracy did not improve from 0.92237
Epoch 205/300
53/53 [==============================] - 24s 460ms/step - loss: 0.2231 - accu
racy: 0.9127 - val_loss: 0.5510 - val_accuracy: 0.7169

Epoch 00205: val_accuracy did not improve from 0.92237
Epoch 206/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2166 - accu
racy: 0.9139 - val_loss: 0.3699 - val_accuracy: 0.8128

Epoch 00206: val_accuracy did not improve from 0.92237
Epoch 207/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2376 - accu
racy: 0.9057 - val_loss: 0.5690 - val_accuracy: 0.7260

Epoch 00207: val_accuracy did not improve from 0.92237
Epoch 208/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2229 - accu
racy: 0.9169 - val_loss: 0.8073 - val_accuracy: 0.6804

Epoch 00208: val_accuracy did not improve from 0.92237
Epoch 209/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2219 - accu
racy: 0.9127 - val_loss: 0.5418 - val_accuracy: 0.7215

Epoch 00209: val_accuracy did not improve from 0.92237
Epoch 210/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2097 - accu
racy: 0.9169 - val_loss: 0.6880 - val_accuracy: 0.6895

Epoch 00210: val_accuracy did not improve from 0.92237
Epoch 211/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2107 - accu
racy: 0.9186 - val_loss: 0.4083 - val_accuracy: 0.7854

Epoch 00211: val_accuracy did not improve from 0.92237
Epoch 212/300
53/53 [==============================] - 26s 483ms/step - loss: 0.2242 - accu
racy: 0.9151 - val_loss: 0.4774 - val_accuracy: 0.7534
```

```
Epoch 00212: val_accuracy did not improve from 0.92237
Epoch 213/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2286 - accu
racy: 0.9086 - val_loss: 0.5610 - val_accuracy: 0.7443

Epoch 00213: val_accuracy did not improve from 0.92237
Epoch 214/300
53/53 [==============================] - 25s 477ms/step - loss: 0.2174 - accu
racy: 0.9133 - val_loss: 0.6050 - val_accuracy: 0.7169

Epoch 00214: val_accuracy did not improve from 0.92237
Epoch 215/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2216 - accu
racy: 0.9127 - val_loss: 0.5495 - val_accuracy: 0.7260

Epoch 00215: val_accuracy did not improve from 0.92237
Epoch 216/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2359 - accu
racy: 0.9139 - val_loss: 0.8301 - val_accuracy: 0.6621

Epoch 00216: val_accuracy did not improve from 0.92237
Epoch 217/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2357 - accu
racy: 0.9021 - val_loss: 0.6228 - val_accuracy: 0.7078

Epoch 00217: val_accuracy did not improve from 0.92237
Epoch 218/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2266 - accu
racy: 0.8968 - val_loss: 0.5883 - val_accuracy: 0.7352

Epoch 00218: val_accuracy did not improve from 0.92237
Epoch 219/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2183 - accu
racy: 0.9133 - val_loss: 0.7788 - val_accuracy: 0.6941

Epoch 00219: val_accuracy did not improve from 0.92237
Epoch 220/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2040 - accu
racy: 0.9186 - val_loss: 0.4147 - val_accuracy: 0.7808

Epoch 00220: val_accuracy did not improve from 0.92237
Epoch 221/300
53/53 [==============================] - 25s 463ms/step - loss: 0.2002 - accu
racy: 0.9192 - val_loss: 0.6821 - val_accuracy: 0.6941

Epoch 00221: val_accuracy did not improve from 0.92237
Epoch 222/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2116 - accu
racy: 0.9127 - val_loss: 0.3907 - val_accuracy: 0.7945

Epoch 00222: val_accuracy did not improve from 0.92237
Epoch 223/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2131 - accu
racy: 0.9169 - val_loss: 0.5154 - val_accuracy: 0.7397

Epoch 00223: val_accuracy did not improve from 0.92237
Epoch 224/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2392 - accu
racy: 0.9009 - val_loss: 0.3973 - val_accuracy: 0.8037

Epoch 00224: val_accuracy did not improve from 0.92237
Epoch 225/300
53/53 [==============================] - 24s 458ms/step - loss: 0.2257 - accu
racy: 0.9116 - val_loss: 0.5063 - val_accuracy: 0.7306

Epoch 00225: val_accuracy did not improve from 0.92237
Epoch 226/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2231 - accu
racy: 0.9092 - val_loss: 0.6196 - val_accuracy: 0.7123
```

```
Epoch 00226: val_accuracy did not improve from 0.92237
Epoch 227/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2393 - accu
racy: 0.9021 - val_loss: 0.5623 - val_accuracy: 0.7169

Epoch 00227: val_accuracy did not improve from 0.92237
Epoch 228/300
53/53 [==============================] - 25s 480ms/step - loss: 0.2141 - accu
racy: 0.9157 - val_loss: 0.6996 - val_accuracy: 0.7123

Epoch 00228: val_accuracy did not improve from 0.92237
Epoch 229/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2183 - accu
racy: 0.9127 - val_loss: 0.6633 - val_accuracy: 0.6941

Epoch 00229: val_accuracy did not improve from 0.92237
Epoch 230/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2170 - accu
racy: 0.9104 - val_loss: 0.5336 - val_accuracy: 0.7352

Epoch 00230: val_accuracy did not improve from 0.92237
Epoch 231/300
53/53 [==============================] - 25s 470ms/step - loss: 0.2485 - accu
racy: 0.9080 - val_loss: 0.5651 - val_accuracy: 0.7123

Epoch 00231: val_accuracy did not improve from 0.92237
Epoch 232/300
53/53 [==============================] - 25s 481ms/step - loss: 0.2025 - accu
racy: 0.9104 - val_loss: 0.5875 - val_accuracy: 0.7215

Epoch 00232: val_accuracy did not improve from 0.92237
Epoch 233/300
53/53 [==============================] - 25s 477ms/step - loss: 0.1792 - accu
racy: 0.9281 - val_loss: 0.4416 - val_accuracy: 0.7854

Epoch 00233: val_accuracy did not improve from 0.92237
Epoch 234/300
53/53 [==============================] - 26s 490ms/step - loss: 0.2356 - accu
racy: 0.9045 - val_loss: 0.7359 - val_accuracy: 0.6849

Epoch 00234: val_accuracy did not improve from 0.92237
Epoch 235/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2274 - accu
racy: 0.9062 - val_loss: 0.6037 - val_accuracy: 0.7123

Epoch 00235: val_accuracy did not improve from 0.92237
Epoch 236/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2221 - accu
racy: 0.9133 - val_loss: 0.4638 - val_accuracy: 0.7626

Epoch 00236: val_accuracy did not improve from 0.92237
Epoch 237/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2375 - accu
racy: 0.9086 - val_loss: 0.4477 - val_accuracy: 0.7671

Epoch 00237: val_accuracy did not improve from 0.92237
Epoch 238/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2067 - accu
racy: 0.9192 - val_loss: 0.6358 - val_accuracy: 0.7123

Epoch 00238: val_accuracy did not improve from 0.92237
Epoch 239/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2369 - accu
racy: 0.9086 - val_loss: 0.4881 - val_accuracy: 0.7443

Epoch 00239: val_accuracy did not improve from 0.92237
Epoch 240/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2067 - accu
```

```
racy: 0.9186 - val_loss: 0.7100 - val_accuracy: 0.6849

Epoch 00240: val_accuracy did not improve from 0.92237
Epoch 241/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2280 - accu
racy: 0.9110 - val_loss: 0.6042 - val_accuracy: 0.7169

Epoch 00241: val_accuracy did not improve from 0.92237
Epoch 242/300
53/53 [==============================] - 26s 492ms/step - loss: 0.2052 - accu
racy: 0.9233 - val_loss: 0.6149 - val_accuracy: 0.7215

Epoch 00242: val_accuracy did not improve from 0.92237
Epoch 243/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2025 - accu
racy: 0.9198 - val_loss: 0.5166 - val_accuracy: 0.7443

Epoch 00243: val_accuracy did not improve from 0.92237
Epoch 244/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2276 - accu
racy: 0.9057 - val_loss: 0.6480 - val_accuracy: 0.7215

Epoch 00244: val_accuracy did not improve from 0.92237
Epoch 245/300
53/53 [==============================] - 25s 469ms/step - loss: 0.2308 - accu
racy: 0.9027 - val_loss: 0.5385 - val_accuracy: 0.7123

Epoch 00245: val_accuracy did not improve from 0.92237
Epoch 246/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2249 - accu
racy: 0.9098 - val_loss: 0.4511 - val_accuracy: 0.7489

Epoch 00246: val_accuracy did not improve from 0.92237
Epoch 247/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2236 - accu
racy: 0.9086 - val_loss: 0.7711 - val_accuracy: 0.6849

Epoch 00247: val_accuracy did not improve from 0.92237
Epoch 248/300
53/53 [==============================] - 25s 472ms/step - loss: 0.2050 - accu
racy: 0.9180 - val_loss: 0.4295 - val_accuracy: 0.7854

Epoch 00248: val_accuracy did not improve from 0.92237
Epoch 249/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2026 - accu
racy: 0.9157 - val_loss: 0.5760 - val_accuracy: 0.7352

Epoch 00249: val_accuracy did not improve from 0.92237
Epoch 250/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2233 - accu
racy: 0.9133 - val_loss: 0.3181 - val_accuracy: 0.8447

Epoch 00250: val_accuracy did not improve from 0.92237
Epoch 251/300
53/53 [==============================] - 25s 467ms/step - loss: 0.2026 - accu
racy: 0.9180 - val_loss: 0.5794 - val_accuracy: 0.7123

Epoch 00251: val_accuracy did not improve from 0.92237
Epoch 252/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2377 - accu
racy: 0.9057 - val_loss: 0.7081 - val_accuracy: 0.6895

Epoch 00252: val_accuracy did not improve from 0.92237
Epoch 253/300
53/53 [==============================] - 25s 464ms/step - loss: 0.2102 - accu
racy: 0.9151 - val_loss: 0.5093 - val_accuracy: 0.7443

Epoch 00253: val_accuracy did not improve from 0.92237
Epoch 254/300
```

```
53/53 [==============================] - 25s 464ms/step - loss: 0.2103 - accu
racy: 0.9086 - val_loss: 0.5608 - val_accuracy: 0.7215

Epoch 00254: val_accuracy did not improve from 0.92237
Epoch 255/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2104 - accu
racy: 0.9163 - val_loss: 0.5502 - val_accuracy: 0.7306

Epoch 00255: val_accuracy did not improve from 0.92237
Epoch 256/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2076 - accu
racy: 0.9175 - val_loss: 0.5205 - val_accuracy: 0.7397

Epoch 00256: val_accuracy did not improve from 0.92237
Epoch 257/300
53/53 [==============================] - 26s 486ms/step - loss: 0.2174 - accu
racy: 0.9216 - val_loss: 0.5861 - val_accuracy: 0.7260

Epoch 00257: val_accuracy did not improve from 0.92237
Epoch 258/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2085 - accu
racy: 0.9251 - val_loss: 0.5898 - val_accuracy: 0.7306

Epoch 00258: val_accuracy did not improve from 0.92237
Epoch 259/300
53/53 [==============================] - 25s 478ms/step - loss: 0.1794 - accu
racy: 0.9281 - val_loss: 0.6250 - val_accuracy: 0.7306

Epoch 00259: val_accuracy did not improve from 0.92237
Epoch 260/300
53/53 [==============================] - 25s 479ms/step - loss: 0.2149 - accu
racy: 0.9204 - val_loss: 0.6476 - val_accuracy: 0.7215

Epoch 00260: val_accuracy did not improve from 0.92237
Epoch 261/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2098 - accu
racy: 0.9163 - val_loss: 0.4858 - val_accuracy: 0.7717

Epoch 00261: val_accuracy did not improve from 0.92237
Epoch 262/300
53/53 [==============================] - 24s 462ms/step - loss: 0.2056 - accu
racy: 0.9139 - val_loss: 0.5184 - val_accuracy: 0.7352

Epoch 00262: val_accuracy did not improve from 0.92237
Epoch 263/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2081 - accu
racy: 0.9180 - val_loss: 0.4989 - val_accuracy: 0.7671

Epoch 00263: val_accuracy did not improve from 0.92237
Epoch 264/300
53/53 [==============================] - 25s 464ms/step - loss: 0.1939 - accu
racy: 0.9222 - val_loss: 0.9079 - val_accuracy: 0.6712

Epoch 00264: val_accuracy did not improve from 0.92237
Epoch 265/300
53/53 [==============================] - 25s 474ms/step - loss: 0.2169 - accu
racy: 0.9116 - val_loss: 0.6674 - val_accuracy: 0.7032

Epoch 00265: val_accuracy did not improve from 0.92237
Epoch 266/300
53/53 [==============================] - 25s 463ms/step - loss: 0.1975 - accu
racy: 0.9133 - val_loss: 0.5885 - val_accuracy: 0.7397

Epoch 00266: val_accuracy did not improve from 0.92237
Epoch 267/300
53/53 [==============================] - 25s 466ms/step - loss: 0.1967 - accu
racy: 0.9233 - val_loss: 0.6510 - val_accuracy: 0.7215

Epoch 00267: val_accuracy did not improve from 0.92237
```

```
Epoch 268/300
53/53 [==============================] - 24s 452ms/step - loss: 0.1902 - accu
racy: 0.9245 - val_loss: 0.6759 - val_accuracy: 0.7123

Epoch 00268: val_accuracy did not improve from 0.92237
Epoch 269/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2328 - accu
racy: 0.9080 - val_loss: 0.4322 - val_accuracy: 0.7854

Epoch 00269: val_accuracy did not improve from 0.92237
Epoch 270/300
53/53 [==============================] - 24s 456ms/step - loss: 0.1851 - accu
racy: 0.9298 - val_loss: 0.5048 - val_accuracy: 0.7626

Epoch 00270: val_accuracy did not improve from 0.92237
Epoch 271/300
53/53 [==============================] - 25s 468ms/step - loss: 0.1918 - accu
racy: 0.9192 - val_loss: 0.5478 - val_accuracy: 0.7397

Epoch 00271: val_accuracy did not improve from 0.92237
Epoch 272/300
53/53 [==============================] - 25s 471ms/step - loss: 0.2233 - accu
racy: 0.9074 - val_loss: 0.5589 - val_accuracy: 0.7306

Epoch 00272: val_accuracy did not improve from 0.92237
Epoch 273/300
53/53 [==============================] - 26s 495ms/step - loss: 0.2046 - accu
racy: 0.9298 - val_loss: 0.6570 - val_accuracy: 0.6941

Epoch 00273: val_accuracy did not improve from 0.92237
Epoch 274/300
53/53 [==============================] - 26s 484ms/step - loss: 0.2029 - accu
racy: 0.9239 - val_loss: 0.5865 - val_accuracy: 0.7306

Epoch 00274: val_accuracy did not improve from 0.92237
Epoch 275/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2196 - accu
racy: 0.9110 - val_loss: 0.5397 - val_accuracy: 0.7260

Epoch 00275: val_accuracy did not improve from 0.92237
Epoch 276/300
53/53 [==============================] - 24s 461ms/step - loss: 0.2138 - accu
racy: 0.9157 - val_loss: 0.5232 - val_accuracy: 0.7580

Epoch 00276: val_accuracy did not improve from 0.92237
Epoch 277/300
53/53 [==============================] - 24s 457ms/step - loss: 0.2118 - accu
racy: 0.9133 - val_loss: 0.4881 - val_accuracy: 0.7626

Epoch 00277: val_accuracy did not improve from 0.92237
Epoch 278/300
53/53 [==============================] - 25s 465ms/step - loss: 0.2026 - accu
racy: 0.9127 - val_loss: 0.6472 - val_accuracy: 0.7260

Epoch 00278: val_accuracy did not improve from 0.92237
Epoch 279/300
53/53 [==============================] - 24s 460ms/step - loss: 0.2209 - accu
racy: 0.9127 - val_loss: 0.4108 - val_accuracy: 0.7945

Epoch 00279: val_accuracy did not improve from 0.92237
Epoch 280/300
53/53 [==============================] - 25s 466ms/step - loss: 0.2158 - accu
racy: 0.9175 - val_loss: 0.6603 - val_accuracy: 0.7123

Epoch 00280: val_accuracy did not improve from 0.92237
Epoch 281/300
53/53 [==============================] - 24s 458ms/step - loss: 0.1901 - accu
racy: 0.9210 - val_loss: 0.5114 - val_accuracy: 0.7397
```

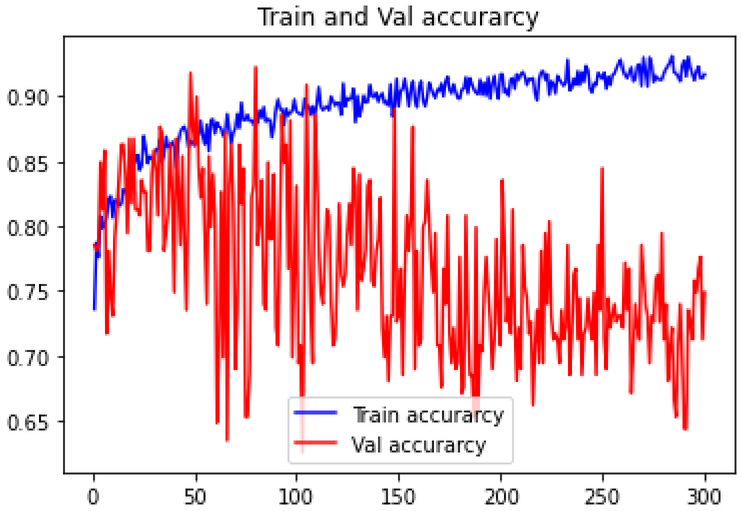
```
Epoch 00281: val_accuracy did not improve from 0.92237
Epoch 282/300
53/53 [==============================] - 24s 452ms/step - loss: 0.2007 - accu
racy: 0.9233 - val_loss: 0.8246 - val_accuracy: 0.6804

Epoch 00282: val_accuracy did not improve from 0.92237
Epoch 283/300
53/53 [==============================] - 24s 460ms/step - loss: 0.2029 - accu
racy: 0.9263 - val_loss: 0.6691 - val_accuracy: 0.7078

Epoch 00283: val_accuracy did not improve from 0.92237
Epoch 284/300
53/53 [==============================] - 26s 482ms/step - loss: 0.1835 - accu
racy: 0.9310 - val_loss: 0.5958 - val_accuracy: 0.7215

Epoch 00284: val_accuracy did not improve from 0.92237
Epoch 285/300
53/53 [==============================] - 25s 477ms/step - loss: 0.1965 - accu
racy: 0.9186 - val_loss: 0.9141 - val_accuracy: 0.6667

Epoch 00285: val_accuracy did not improve from 0.92237
Epoch 286/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2150 - accu
racy: 0.9175 - val_loss: 0.9750 - val_accuracy: 0.6530

Epoch 00286: val_accuracy did not improve from 0.92237
Epoch 287/300
53/53 [==============================] - 25s 462ms/step - loss: 0.2070 - accu
racy: 0.9151 - val_loss: 0.6825 - val_accuracy: 0.7078

Epoch 00287: val_accuracy did not improve from 0.92237
Epoch 288/300
53/53 [==============================] - 26s 481ms/step - loss: 0.2191 - accu
racy: 0.9116 - val_loss: 0.6359 - val_accuracy: 0.7397

Epoch 00288: val_accuracy did not improve from 0.92237
Epoch 289/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2045 - accu
racy: 0.9233 - val_loss: 0.6315 - val_accuracy: 0.6986

Epoch 00289: val_accuracy did not improve from 0.92237
Epoch 290/300
53/53 [==============================] - 25s 480ms/step - loss: 0.1806 - accu
racy: 0.9275 - val_loss: 0.9631 - val_accuracy: 0.6438

Epoch 00290: val_accuracy did not improve from 0.92237
Epoch 291/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2014 - accu
racy: 0.9151 - val_loss: 0.8148 - val_accuracy: 0.6438

Epoch 00291: val_accuracy did not improve from 0.92237
Epoch 292/300
53/53 [==============================] - 25s 477ms/step - loss: 0.1794 - accu
racy: 0.9304 - val_loss: 0.6086 - val_accuracy: 0.7352

Epoch 00292: val_accuracy did not improve from 0.92237
Epoch 293/300
53/53 [==============================] - 25s 470ms/step - loss: 0.1991 - accu
racy: 0.9222 - val_loss: 0.6510 - val_accuracy: 0.7260

Epoch 00293: val_accuracy did not improve from 0.92237
Epoch 294/300
53/53 [==============================] - 25s 475ms/step - loss: 0.2005 - accu
racy: 0.9169 - val_loss: 0.7162 - val_accuracy: 0.7123

Epoch 00294: val_accuracy did not improve from 0.92237
Epoch 295/300
53/53 [==============================] - 25s 473ms/step - loss: 0.2088 - accu
racy: 0.9121 - val_loss: 0.5723 - val_accuracy: 0.7580
```

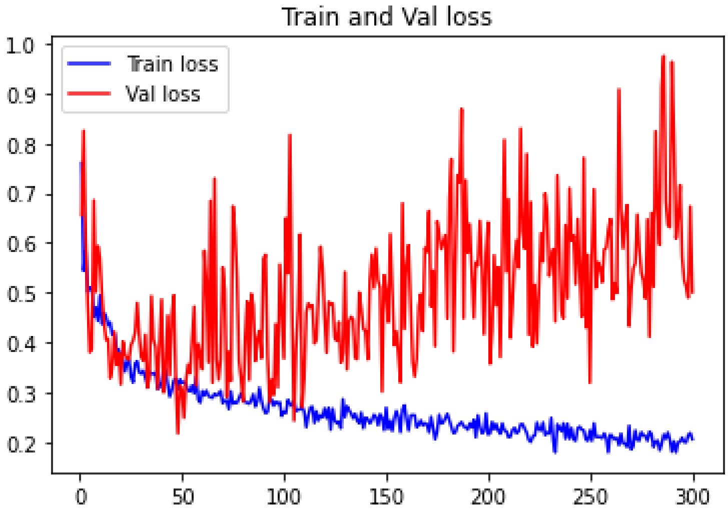
```
Epoch 00295: val_accuracy did not improve from 0.92237
Epoch 296/300
53/53 [==============================] - 25s 476ms/step - loss: 0.2000 - accu
racy: 0.9192 - val_loss: 0.5249 - val_accuracy: 0.7489

Epoch 00296: val_accuracy did not improve from 0.92237
Epoch 297/300
53/53 [==============================] - 25s 471ms/step - loss: 0.1994 - accu
racy: 0.9228 - val_loss: 0.5112 - val_accuracy: 0.7671

Epoch 00297: val_accuracy did not improve from 0.92237
Epoch 298/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2138 - accu
racy: 0.9139 - val_loss: 0.4895 - val_accuracy: 0.7763

Epoch 00298: val_accuracy did not improve from 0.92237
Epoch 299/300
53/53 [==============================] - 25s 468ms/step - loss: 0.2181 - accu
racy: 0.9139 - val_loss: 0.6733 - val_accuracy: 0.7123

Epoch 00299: val_accuracy did not improve from 0.92237
Epoch 300/300
53/53 [==============================] - 25s 478ms/step - loss: 0.2058 - accu
racy: 0.9169 - val_loss: 0.5006 - val_accuracy: 0.7489

Epoch 00300: val_accuracy did not improve from 0.92237
```

Evaluating final trained model:

In [ ]:

```
train_ accuracy = history.history['accuracy']
validation_accuracy = history.history['val_accuracy']
train_loss = history.history['loss']
validation_loss = history.history['val_loss']
epochs = range(1, len(train_accuracy) + 1)

#Train and validation accuracy
plt.plot(epochs, train_accuracy, 'b', label='Training accurarcy')
plt.plot(epochs, validation_accuracy, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, train_loss, 'b', label='Training loss')
plt.plot(epochs, validation_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```





In [ ]:

```
#load model:
path_to_model= "/content/gdrive/MyDrive/Plant_disease/aug_20_2.hdf5"
model = keras.models.load_model(path_to_model)
```

In [ ]:

```
x, y = x_test, y_test
x.shape, y.shape

from sklearn.metrics import confusion_matrix, f1_score, precision_score, reca
from keras.utils.np_utils import to_categorical
import tensorflow as tf

scores = model.evaluate(x_test, y_test)

y_pred=[]

for pred in ((model.predict(x_test))): #custom loop with threshold as 0.1
  if pred >= 0.5:
    y_pred.append(1)
  else:
    y_pred.append(0)

y_test_new= []
for y in y_test:
  y= np.array_str(y)
  y= int(y[1])
  y_test_new.append(y)

print("CONFUSION MATRIX:")
print( confusion_matrix(y_test_new , y_pred))
```

```
print("======================")

print(f"ACCURACY for test dataset is:       {scores[1]}")
print("F1_SCORE for test datset is         {}".format(f1_score(y_test_new , y
print("PRECISION_SCORE for test datset is  {}".format(precision_score(y_test_
print("RECALL_SCORE for test datset is     {}".format(recall_score(y_test_new
```

```
16/16 [==============================] - 1s 64ms/step - loss: 0.3158 - accura
cy: 0.8642
CONFUSION MATRIX:
[[174  26]
 [ 43 265]]
======================
ACCURACY for test dataset is:       0.8641732335090637
F1_SCORE for test datset is         0.8848080133555927
PRECISION_SCORE for test datset is  0.9106529209621993
RECALL_SCORE for test datset is     0.8603896103896104
```

Conclusion:

As we can see that it didn't take much training and results are very great on our data. This is possible because we incorporated 'Transfer Learning' for training on our dataset. The model trained from 'Plant Village' dataset learned greatly to classify healthy and unhealthy plants. Saved Weights from that model helped our second model to train well on self collected data and yields good results.