import matplotlib.pyplot as plt  
%matplotlib inline  
import pandas as pd  
import numpy as np  
import PIL  
from PIL import Image  
from skimage.color import rgb2gray  
from scipy import ndimage as ndi  
import cv2  
import os  
from os import listdir  
from sklearn.utils import shuffle  
  
import keras  
from keras.models import Sequential, load\_model  
from keras.layers import Dense, Activation, Conv2D, MaxPooling2D, Flatten, Dropout, BatchNormalization  
from keras.optimizers import SGD  
from keras import regularizers  
from keras.callbacks import ModelCheckpoint

Using TensorFlow backend.

directory\_root = "../input/plantvillage/PlantVillage"  
print(len(listdir(directory\_root)))

15

image\_list, label\_list = [], []  
try:  
 print("[INFO] Loading images ...")  
 root\_dir = listdir(directory\_root)  
 for directory in root\_dir :  
 # remove .DS\_Store from list  
 if directory == ".DS\_Store" :  
 root\_dir.remove(directory)  
  
 for plant\_folder in root\_dir :  
 plant\_disease\_folder\_list = listdir(f"{directory\_root}/{plant\_folder}")  
   
 for single\_plant\_disease\_image in plant\_disease\_folder\_list :  
 if single\_plant\_disease\_image == ".DS\_Store" :  
 plant\_disease\_folder\_list.remove(single\_plant\_disease\_image)  
  
 for image in plant\_disease\_folder\_list:  
 image\_directory = f"{directory\_root}/{plant\_folder}/{image}"  
 if image\_directory.endswith(".jpg") == True or image\_directory.endswith(".JPG") == True:  
 image\_list.append(image\_directory)  
 label\_list.append(plant\_folder)  
 print("[INFO] Image loading completed")   
except Exception as e:  
 print(f"Error : {e}")

[INFO] Loading images ...  
[INFO] Image loading completed

img\_info = pd.DataFrame({'image\_path':image\_list,'label':label\_list})  
print(img\_info.head())  
print(len(img\_info))

image\_path label  
0 ../input/plantvillage/PlantVillage/Pepper\_\_bel... Pepper\_\_bell\_\_\_Bacterial\_spot  
1 ../input/plantvillage/PlantVillage/Pepper\_\_bel... Pepper\_\_bell\_\_\_Bacterial\_spot  
2 ../input/plantvillage/PlantVillage/Pepper\_\_bel... Pepper\_\_bell\_\_\_Bacterial\_spot  
3 ../input/plantvillage/PlantVillage/Pepper\_\_bel... Pepper\_\_bell\_\_\_Bacterial\_spot  
4 ../input/plantvillage/PlantVillage/Pepper\_\_bel... Pepper\_\_bell\_\_\_Bacterial\_spot  
20636

#new column (empty)  
img\_info["labels\_integer"] = None  
#index of new column  
index\_labels\_integer = img\_info.columns.get\_loc("labels\_integer")  
#index of species column  
index\_species = img\_info.columns.get\_loc("label")  
#to assign numeric labels starting with 0 for the first species  
k = 0   
for i in range(len(img\_info)):  
 if i == 0:  
 img\_info.iloc[i, index\_labels\_integer] = k #here, k == 0  
 if i > 0:  
 if img\_info.iloc[i-1, index\_species] == img\_info.iloc[i, index\_species]:  
 img\_info.iloc[i, index\_labels\_integer] = k  
 else:  
 k += 1  
 img\_info.iloc[i, index\_labels\_integer] = k  
img\_info.tail()

image\_path ... labels\_integer  
20631 ../input/plantvillage/PlantVillage/Tomato\_\_Tom... ... 14  
20632 ../input/plantvillage/PlantVillage/Tomato\_\_Tom... ... 14  
20633 ../input/plantvillage/PlantVillage/Tomato\_\_Tom... ... 14  
20634 ../input/plantvillage/PlantVillage/Tomato\_\_Tom... ... 14  
20635 ../input/plantvillage/PlantVillage/Tomato\_\_Tom... ... 14  
  
[5 rows x 3 columns]

img\_info = shuffle(img\_info)  
list\_vectors = []  
  
for image\_path in img\_info.image\_path:  
 #read as rgb array  
 img = Image.open(image\_path)  
 size = (64, 64)  
 img = img.resize(size, PIL.Image.ANTIALIAS)   
 img\_array = np.array(img)  
 #append image vector to list  
 list\_vectors.append(img\_array)  
   
print(len(list\_vectors))

20636

X = np.stack((list\_vectors))  
Y = img\_info['labels\_integer']  
print(X.shape)

(20636, 64, 64, 3)

X = X/255  
Y\_one\_hot = keras.utils.to\_categorical(Y, num\_classes=15)  
print(Y.shape, Y\_one\_hot.shape)

(20636,) (20636, 15)

np.savez("x\_images\_arrayscnn", X)  
np.savez("y\_numeric\_labelscnn", Y\_one\_hot)

x\_npz = np.load("x\_images\_arrayscnn.npz")  
X = x\_npz['arr\_0']  
  
y\_npz = np.load("y\_numeric\_labelscnn.npz")  
Y\_one\_hot = y\_npz['arr\_0']  
  
print(X.shape)

(20636, 64, 64, 3)

split\_train = 0.8 #train 0.8, validate 0.1, test 0.1  
split\_val = 0.9  
index\_train = int(split\_train\*len(X))  
index\_val = int(split\_val\*len(X))  
  
X\_train = X[:index\_train]  
X\_val = X[index\_train:index\_val]  
X\_test = X[index\_val:]  
  
Y\_train = Y\_one\_hot[:index\_train]  
Y\_val = Y\_one\_hot[index\_train:index\_val]  
Y\_test = Y\_one\_hot[index\_val:]  
  
  
print(X\_train.shape, X\_val.shape, X\_test.shape, Y\_train.shape, Y\_val.shape, Y\_test.shape)

(16508, 64, 64, 3) (2064, 64, 64, 3) (2064, 64, 64, 3) (16508, 15) (2064, 15) (2064, 15)

input\_shape = (X\_train.shape[1], X\_train.shape[2], X\_train.shape[3])   
num\_classes = 15  
  
model = Sequential()  
model.add(Conv2D(32, kernel\_size=(5, 5), strides=(1, 1), input\_shape=input\_shape))  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool\_size=(2, 2), strides=(2, 2)))  
model.add(Conv2D(64, (5, 5)))  
model.add(Activation('relu'))  
model.add(MaxPooling2D(pool\_size=(2, 2)))  
model.add(Dropout(0.7))  
model.add(Flatten())  
model.add(Dense(1000))  
model.add(Activation('relu'))  
model.add(Dropout(0.7))  
model.add(Dense(num\_classes, activation='softmax'))  
  
  
model.compile(optimizer=keras.optimizers.Adam(lr=0.0003, beta\_1=0.9, beta\_2=0.999, epsilon=None, decay=1e-8, amsgrad=False),  
 loss='categorical\_crossentropy',  
 metrics=['accuracy'])

WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/framework/op\_def\_library.py:263: colocate\_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.  
Instructions for updating:  
Colocations handled automatically by placer.  
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/keras/backend/tensorflow\_backend.py:3445: calling dropout (from tensorflow.python.ops.nn\_ops) with keep\_prob is deprecated and will be removed in a future version.  
Instructions for updating:  
Please use `rate` instead of `keep\_prob`. Rate should be set to `rate = 1 - keep\_prob`.

model.summary()

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Layer (type) Output Shape Param #   
=================================================================  
conv2d\_1 (Conv2D) (None, 60, 60, 32) 2432   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
activation\_1 (Activation) (None, 60, 60, 32) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
max\_pooling2d\_1 (MaxPooling2 (None, 30, 30, 32) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
conv2d\_2 (Conv2D) (None, 26, 26, 64) 51264   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
activation\_2 (Activation) (None, 26, 26, 64) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
max\_pooling2d\_2 (MaxPooling2 (None, 13, 13, 64) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_1 (Dropout) (None, 13, 13, 64) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
flatten\_1 (Flatten) (None, 10816) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_1 (Dense) (None, 1000) 10817000   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
activation\_3 (Activation) (None, 1000) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_2 (Dropout) (None, 1000) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_2 (Dense) (None, 15) 15015   
=================================================================  
Total params: 10,885,711  
Trainable params: 10,885,711  
Non-trainable params: 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

best\_model\_file = "disease\_selected\_100.h5"  
best\_model = ModelCheckpoint(best\_model\_file, monitor='val\_loss', verbose=1, save\_best\_only=True)  
  
print('Training model...')  
results = model.fit(X\_train, Y\_train, epochs=50, batch\_size=64, validation\_data=(X\_val, Y\_val), callbacks=[best\_model])  
print('Training finished.')  
  
print('Loading the best model...')  
model = load\_model(best\_model\_file)  
print('Best Model loaded!')

Training model...  
WARNING:tensorflow:From /opt/conda/lib/python3.6/site-packages/tensorflow/python/ops/math\_ops.py:3066: to\_int32 (from tensorflow.python.ops.math\_ops) is deprecated and will be removed in a future version.  
Instructions for updating:  
Use tf.cast instead.  
Train on 16508 samples, validate on 2064 samples  
Epoch 1/50  
16508/16508 [==============================] - 9s 520us/step - loss: 1.8491 - acc: 0.4113 - val\_loss: 1.1749 - val\_acc: 0.6560  
  
Epoch 00001: val\_loss improved from inf to 1.17491, saving model to disease\_selected\_100.h5  
Epoch 2/50  
16508/16508 [==============================] - 4s 219us/step - loss: 1.0967 - acc: 0.6516 - val\_loss: 0.8050 - val\_acc: 0.7379  
  
Epoch 00002: val\_loss improved from 1.17491 to 0.80496, saving model to disease\_selected\_100.h5  
Epoch 3/50  
16508/16508 [==============================] - 4s 222us/step - loss: 0.8368 - acc: 0.7225 - val\_loss: 0.6123 - val\_acc: 0.8101  
  
Epoch 00003: val\_loss improved from 0.80496 to 0.61230, saving model to disease\_selected\_100.h5  
Epoch 4/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.6965 - acc: 0.7700 - val\_loss: 0.6121 - val\_acc: 0.7980  
  
Epoch 00004: val\_loss improved from 0.61230 to 0.61207, saving model to disease\_selected\_100.h5  
Epoch 5/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.6043 - acc: 0.7937 - val\_loss: 0.5079 - val\_acc: 0.8372  
  
Epoch 00005: val\_loss improved from 0.61207 to 0.50793, saving model to disease\_selected\_100.h5  
Epoch 6/50  
16508/16508 [==============================] - 4s 235us/step - loss: 0.5215 - acc: 0.8241 - val\_loss: 0.4796 - val\_acc: 0.8416  
  
Epoch 00006: val\_loss improved from 0.50793 to 0.47962, saving model to disease\_selected\_100.h5  
Epoch 7/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.4768 - acc: 0.8407 - val\_loss: 0.3791 - val\_acc: 0.8784  
  
Epoch 00007: val\_loss improved from 0.47962 to 0.37914, saving model to disease\_selected\_100.h5  
Epoch 8/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.4309 - acc: 0.8548 - val\_loss: 0.3399 - val\_acc: 0.8886  
  
Epoch 00008: val\_loss improved from 0.37914 to 0.33993, saving model to disease\_selected\_100.h5  
Epoch 9/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.3854 - acc: 0.8703 - val\_loss: 0.3341 - val\_acc: 0.8915  
  
Epoch 00009: val\_loss improved from 0.33993 to 0.33409, saving model to disease\_selected\_100.h5  
Epoch 10/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.3660 - acc: 0.8772 - val\_loss: 0.3587 - val\_acc: 0.8803  
  
Epoch 00010: val\_loss did not improve from 0.33409  
Epoch 11/50  
16508/16508 [==============================] - 3s 212us/step - loss: 0.3461 - acc: 0.8814 - val\_loss: 0.3121 - val\_acc: 0.9007  
  
Epoch 00011: val\_loss improved from 0.33409 to 0.31207, saving model to disease\_selected\_100.h5  
Epoch 12/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.3106 - acc: 0.8969 - val\_loss: 0.2962 - val\_acc: 0.9084  
  
Epoch 00012: val\_loss improved from 0.31207 to 0.29622, saving model to disease\_selected\_100.h5  
Epoch 13/50  
16508/16508 [==============================] - 4s 216us/step - loss: 0.2955 - acc: 0.8997 - val\_loss: 0.2876 - val\_acc: 0.9138  
  
Epoch 00013: val\_loss improved from 0.29622 to 0.28756, saving model to disease\_selected\_100.h5  
Epoch 14/50  
16508/16508 [==============================] - 4s 229us/step - loss: 0.2748 - acc: 0.9085 - val\_loss: 0.2794 - val\_acc: 0.9147  
  
Epoch 00014: val\_loss improved from 0.28756 to 0.27941, saving model to disease\_selected\_100.h5  
Epoch 15/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.2653 - acc: 0.9089 - val\_loss: 0.3136 - val\_acc: 0.9012  
  
Epoch 00015: val\_loss did not improve from 0.27941  
Epoch 16/50  
16508/16508 [==============================] - 3s 212us/step - loss: 0.2501 - acc: 0.9145 - val\_loss: 0.2519 - val\_acc: 0.9254  
  
Epoch 00016: val\_loss improved from 0.27941 to 0.25193, saving model to disease\_selected\_100.h5  
Epoch 17/50  
16508/16508 [==============================] - 3s 211us/step - loss: 0.2394 - acc: 0.9182 - val\_loss: 0.2538 - val\_acc: 0.9220  
  
Epoch 00017: val\_loss did not improve from 0.25193  
Epoch 18/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.2218 - acc: 0.9277 - val\_loss: 0.2444 - val\_acc: 0.9225  
  
Epoch 00018: val\_loss improved from 0.25193 to 0.24441, saving model to disease\_selected\_100.h5  
Epoch 19/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.2019 - acc: 0.9311 - val\_loss: 0.2404 - val\_acc: 0.9215  
  
Epoch 00019: val\_loss improved from 0.24441 to 0.24038, saving model to disease\_selected\_100.h5  
Epoch 20/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.2024 - acc: 0.9308 - val\_loss: 0.2486 - val\_acc: 0.9259  
  
Epoch 00020: val\_loss did not improve from 0.24038  
Epoch 21/50  
16508/16508 [==============================] - 4s 215us/step - loss: 0.1839 - acc: 0.9374 - val\_loss: 0.2291 - val\_acc: 0.9249  
  
Epoch 00021: val\_loss improved from 0.24038 to 0.22915, saving model to disease\_selected\_100.h5  
Epoch 22/50  
16508/16508 [==============================] - 4s 230us/step - loss: 0.1948 - acc: 0.9342 - val\_loss: 0.2226 - val\_acc: 0.9331  
  
Epoch 00022: val\_loss improved from 0.22915 to 0.22264, saving model to disease\_selected\_100.h5  
Epoch 23/50  
16508/16508 [==============================] - 4s 221us/step - loss: 0.1822 - acc: 0.9378 - val\_loss: 0.2359 - val\_acc: 0.9312  
  
Epoch 00023: val\_loss did not improve from 0.22264  
Epoch 24/50  
16508/16508 [==============================] - 4s 214us/step - loss: 0.1649 - acc: 0.9445 - val\_loss: 0.2805 - val\_acc: 0.9133  
  
Epoch 00024: val\_loss did not improve from 0.22264  
Epoch 25/50  
16508/16508 [==============================] - 4s 216us/step - loss: 0.1639 - acc: 0.9438 - val\_loss: 0.2157 - val\_acc: 0.9331  
  
Epoch 00025: val\_loss improved from 0.22264 to 0.21568, saving model to disease\_selected\_100.h5  
Epoch 26/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.1601 - acc: 0.9439 - val\_loss: 0.2165 - val\_acc: 0.9375  
  
Epoch 00026: val\_loss did not improve from 0.21568  
Epoch 27/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.1461 - acc: 0.9478 - val\_loss: 0.2168 - val\_acc: 0.9312  
  
Epoch 00027: val\_loss did not improve from 0.21568  
Epoch 28/50  
16508/16508 [==============================] - 4s 212us/step - loss: 0.1531 - acc: 0.9480 - val\_loss: 0.2294 - val\_acc: 0.9336  
  
Epoch 00028: val\_loss did not improve from 0.21568  
Epoch 29/50  
16508/16508 [==============================] - 4s 219us/step - loss: 0.1514 - acc: 0.9492 - val\_loss: 0.2154 - val\_acc: 0.9341  
  
Epoch 00029: val\_loss improved from 0.21568 to 0.21541, saving model to disease\_selected\_100.h5  
Epoch 30/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.1378 - acc: 0.9512 - val\_loss: 0.2201 - val\_acc: 0.9331  
  
Epoch 00030: val\_loss did not improve from 0.21541  
Epoch 31/50  
16508/16508 [==============================] - 4s 229us/step - loss: 0.1366 - acc: 0.9526 - val\_loss: 0.2230 - val\_acc: 0.9360  
  
Epoch 00031: val\_loss did not improve from 0.21541  
Epoch 32/50  
16508/16508 [==============================] - 4s 215us/step - loss: 0.1289 - acc: 0.9544 - val\_loss: 0.2184 - val\_acc: 0.9394  
  
Epoch 00032: val\_loss did not improve from 0.21541  
Epoch 33/50  
16508/16508 [==============================] - 3s 212us/step - loss: 0.1269 - acc: 0.9564 - val\_loss: 0.2196 - val\_acc: 0.9331  
  
Epoch 00033: val\_loss did not improve from 0.21541  
Epoch 34/50  
16508/16508 [==============================] - 4s 215us/step - loss: 0.1232 - acc: 0.9576 - val\_loss: 0.2087 - val\_acc: 0.9394  
  
Epoch 00034: val\_loss improved from 0.21541 to 0.20866, saving model to disease\_selected\_100.h5  
Epoch 35/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.1184 - acc: 0.9594 - val\_loss: 0.2023 - val\_acc: 0.9443  
  
Epoch 00035: val\_loss improved from 0.20866 to 0.20234, saving model to disease\_selected\_100.h5  
Epoch 36/50  
16508/16508 [==============================] - 4s 217us/step - loss: 0.1222 - acc: 0.9587 - val\_loss: 0.2131 - val\_acc: 0.9394  
  
Epoch 00036: val\_loss did not improve from 0.20234  
Epoch 37/50  
16508/16508 [==============================] - 3s 210us/step - loss: 0.1129 - acc: 0.9601 - val\_loss: 0.2132 - val\_acc: 0.9385  
  
Epoch 00037: val\_loss did not improve from 0.20234  
Epoch 38/50  
16508/16508 [==============================] - 4s 214us/step - loss: 0.1090 - acc: 0.9623 - val\_loss: 0.2207 - val\_acc: 0.9404  
  
Epoch 00038: val\_loss did not improve from 0.20234  
Epoch 39/50  
16508/16508 [==============================] - 4s 226us/step - loss: 0.1037 - acc: 0.9677 - val\_loss: 0.1951 - val\_acc: 0.9428  
  
Epoch 00039: val\_loss improved from 0.20234 to 0.19514, saving model to disease\_selected\_100.h5  
Epoch 40/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.1063 - acc: 0.9626 - val\_loss: 0.2098 - val\_acc: 0.9409  
  
Epoch 00040: val\_loss did not improve from 0.19514  
Epoch 41/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.0972 - acc: 0.9667 - val\_loss: 0.2051 - val\_acc: 0.9394  
  
Epoch 00041: val\_loss did not improve from 0.19514  
Epoch 42/50  
16508/16508 [==============================] - 3s 211us/step - loss: 0.0901 - acc: 0.9693 - val\_loss: 0.2041 - val\_acc: 0.9428  
  
Epoch 00042: val\_loss did not improve from 0.19514  
Epoch 43/50  
16508/16508 [==============================] - 3s 209us/step - loss: 0.1054 - acc: 0.9646 - val\_loss: 0.2074 - val\_acc: 0.9428  
  
Epoch 00043: val\_loss did not improve from 0.19514  
Epoch 44/50  
16508/16508 [==============================] - 3s 210us/step - loss: 0.0908 - acc: 0.9695 - val\_loss: 0.2502 - val\_acc: 0.9307  
  
Epoch 00044: val\_loss did not improve from 0.19514  
Epoch 45/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.1015 - acc: 0.9663 - val\_loss: 0.2024 - val\_acc: 0.9419  
  
Epoch 00045: val\_loss did not improve from 0.19514  
Epoch 46/50  
16508/16508 [==============================] - 4s 218us/step - loss: 0.0859 - acc: 0.9709 - val\_loss: 0.1961 - val\_acc: 0.9433  
  
Epoch 00046: val\_loss did not improve from 0.19514  
Epoch 47/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.0830 - acc: 0.9714 - val\_loss: 0.2079 - val\_acc: 0.9453  
  
Epoch 00047: val\_loss did not improve from 0.19514  
Epoch 48/50  
16508/16508 [==============================] - 4s 233us/step - loss: 0.0936 - acc: 0.9680 - val\_loss: 0.2141 - val\_acc: 0.9453  
  
Epoch 00048: val\_loss did not improve from 0.19514  
Epoch 49/50  
16508/16508 [==============================] - 4s 214us/step - loss: 0.0778 - acc: 0.9738 - val\_loss: 0.1998 - val\_acc: 0.9482  
  
Epoch 00049: val\_loss did not improve from 0.19514  
Epoch 50/50  
16508/16508 [==============================] - 4s 213us/step - loss: 0.0899 - acc: 0.9695 - val\_loss: 0.2002 - val\_acc: 0.9453  
  
Epoch 00050: val\_loss did not improve from 0.19514  
Training finished.  
Loading the best model...  
Best Model loaded!

scores = model.evaluate(X\_test, Y\_test)  
print(f"Test Accuracy: {scores[1]\*100}")

2064/2064 [==============================] - 0s 145us/step  
Test Accuracy: 94.04069767441861