1. Please visit this link to access the state-of-art DenseNet code for reference - DenseNet - cifar10 notebook link
2. You need to create a copy of this and "retrain" this model to achieve 90+ test accuracy.
3. You cannot use DropOut layers.
4. You MUST use Image Augmentation Techniques.
5. You cannot use an already trained model as a beginning points, you have to initilize as your own
6. You cannot run the program for more than 300 Epochs, and it should be clear from your log, that you have only used 300 Epochs
7. You cannot use test images for training the model.
8. You cannot change the general architecture of DenseNet (which means you must use Dense Block, Transition and Output blocks as mentioned in the code)
9. You are free to change Convolution types (e.g. from 3x3 normal convolution to Depthwise Separable, etc)
10. You cannot have more than 1 Million parameters in total
11. You are free to move the code from Keras to Tensorflow, Pytorch, MXNET etc.
12. You can use any optimization algorithm you need.
13. You can checkpoint your model and retrain the model from that checkpoint so that no need of training the model from first if you lost at any epoch while training. You can directly load that model and Train from that epoch.

In [1]:
```
# import keras
# from keras.datasets import cifar10
# from keras.models import Model, Sequential
# from keras.layers import Dense, Dropout, Flatten, Input, AveragePooling2D, merge, Activation
# from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
# from keras.layers import Concatenate
# from keras.optimizers import A`dam
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.utils import resample
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from skimage.data import chelsea,astronaut
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
physical_devices = tf.config.experimental.list_physical_devices('GPU')
tf.config.experimental.set_memory_growth(physical_devices[0], True)
```

In [2]:
```
tf.keras.backend.clear_session()
```

In [ ]:
```
```

In [3]:
```
# Hyperparameters


num_classes = 10
num_filter = 12
compression = 0.5
```

In [4]:
```
# Load CIFAR10 Data
(X_train, Y_train), (X_test, Y_test) = tf.keras.datasets.cifar10.load_data()
img_height, img_width, channel = X_train.shape[1],X_train.shape[2],X_train.shape[3]

#X_train = X_train.astype('float32')
#X_test = X_test.astype('float32')
#X_train, X_cv, y_train, y_cv = train_test_split(x_train, Y_train, test_size=0.2, random_state=42)




# convert to one hot encoing
Y_train = tf.keras.utils.to_categorical(Y_train, num_classes)
Y_test = tf.keras.utils.to_categorical(Y_test, num_classes)
#y_cv = tf.keras.utils.to_categorical(y_cv, num_classes)
```

In [5]:
```
X_train.shape,len(Y_train)
```

Out[5]:
```
((50000, 32, 32, 3), 50000)
```

In [6]:
```
X_test.shape
```

Out[6]:
```
(10000, 32, 32, 3)
```

In [7]:
```
img_height, img_width, channel
```

Out[7]: (32, 32, 3)

In [8]:
```
# Dense Block
from tensorflow.keras import regularizers
def denseblock(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    temp = input
    for _ in range(l):
        BatchNorm = layers.BatchNormalization()(temp)
        relu = layers.Activation('relu')(BatchNorm)
        Conv2D_3_3 = layers.Conv2D(int(num_filter*compression), (3,3), use_bias=False ,padding='same',
                                   kernel_initializer=tf.keras.initializers.HeNormal(),
                                   kernel_regularizer=regularizers.L2(0.0001)
                                  )(relu)
        #if dropout_rate>0:
        #    Conv2D_3_3 = layers.Dropout(dropout_rate)(Conv2D_3_3)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_3_3])

        temp = concat

    return temp

## transition Blosck
def transition(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (1,1), use_bias=False ,padding='same',
                                      kernel_initializer=tf.keras.initializers.HeNormal(),
                                       kernel_regularizer=regularizers.L2(0.0001)
                                      )(relu)
    #if dropout_rate>0:
    #    Conv2D_BottleNeck = layers.Dropout(dropout_rate)(Conv2D_BottleNeck)
    avg = layers.AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)
    return avg

#output layer
def output_layer(input):
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    AvgPooling = layers.AveragePooling2D(pool_size=(2,2),)(relu)
    flat = layers.Flatten()(AvgPooling)
    output = layers.Dense(num_classes, activation='softmax')(flat)
    return output
```

In [9]:
```
num_filter = 80
dropout_rate = 0.0
l = 12

from tensorflow.keras.layers import DepthwiseConv2D,SeparableConv2D,Conv2DTranspose,Conv3D,Conv3DTranspose


input = layers.Input(shape=(img_height, img_width, channel,))
First_Conv2D = layers.Conv2D(num_filter, (3,3),strides=(1,1), use_bias=False ,padding='same',activation = 'relu
                             kernel_initializer=tf.keras.initializers.HeNormal(),kernel_regularizer=regularizers
                            )(input)

#First_Conv2D = DepthwiseConv2D(kernel_size = (3,3), use_bias=False ,padding='same',activation = 'relu',
#                               depthwise_initializer=tf.keras.initializers.HeUniform(),
#                               depthwise_regularizer=regularizers.L2())(input)

#First_Conv2D = Conv2DTranspose(kernel_size = (10,10), filters = num_filter,activation = 'relu',
#                               use_bias=False ,padding='same',
#                               kernel_initializer=tf.keras.initializers.HeUniform(),
#                               kernel_regularizer=regularizers.L2())(input)


First_Block = denseblock(First_Conv2D, num_filter, dropout_rate)
First_Transition = transition(First_Block, num_filter, dropout_rate)

Second_Block = denseblock(First_Transition, num_filter, dropout_rate)
Second_Transition = transition(Second_Block, num_filter, dropout_rate)

Third_Block = denseblock(Second_Transition, num_filter, dropout_rate)
Third_Transition = transition(Third_Block, num_filter, dropout_rate)

Last_Block = denseblock(Third_Transition,  num_filter, dropout_rate)
output = output_layer(Last_Block)
```

In [10]:
```
First_Block.shape,Second_Block.shape,Third_Block.shape,Last_Block.shape
```

Out[10]:
```
(TensorShape([None, 32, 32, 560]),
 TensorShape([None, 16, 16, 520]),
 TensorShape([None, 8, 8, 520]),
 TensorShape([None, 4, 4, 520]))
```

In [11]:
```
#https://arxiv.org/pdf/1608.06993.pdf
#from IPython.display import IFrame, YouTubeVideo
#YouTubeVideo(id='-W6y8xnd--U', width=600)
```

In [12]:
```
model = Model(inputs=[input], outputs=[output])
model.summary()
```

```
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 32, 32, 3)]	0	
conv2d (Conv2D)	(None, 32, 32, 80)	2160	input_1[0][0]
batch_normalization (BatchNorma	(None, 32, 32, 80)	320	conv2d[0][0]
activation (Activation)	(None, 32, 32, 80)	0	batch_normalization[0][0]
conv2d_1 (Conv2D)	(None, 32, 32, 40)	28800	activation[0][0]
concatenate (Concatenate)	(None, 32, 32, 120)	0	conv2d[0][0] conv2d_1[0][0]
batch_normalization_1 (BatchNor	(None, 32, 32, 120)	480	concatenate[0][0]
activation_1 (Activation)	(None, 32, 32, 120)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 32, 32, 40)	43200	activation_1[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 160)	0	concatenate[0][0] conv2d_2[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 160)	640	concatenate_1[0][0]
activation_2 (Activation)	(None, 32, 32, 160)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 32, 32, 40)	57600	activation_2[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 200)	0	concatenate_1[0][0] conv2d_3[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 200)	800	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 200)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 40)	72000	activation_3[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 240)	0	concatenate_2[0][0] conv2d_4[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 240)	960	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 240)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 40)	86400	activation_4[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 280)	0	concatenate_3[0][0] conv2d_5[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 280)	1120	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 280)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 40)	100800	activation_5[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 320)	0	concatenate_4[0][0] conv2d_6[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 320)	1280	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 320)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 40)	115200	activation_6[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 360)	0	concatenate_5[0][0] conv2d_7[0][0]

batch_normalization_7 (BatchNor	(None, 32, 32, 360)	1440	concatenate_6[0][0]
activation_7 (Activation)	(None, 32, 32, 360)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 40)	129600	activation_7[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 400)	0	concatenate_6[0][0] conv2d_8[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 400)	1600	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 400)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 40)	144000	activation_8[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 440)	0	concatenate_7[0][0] conv2d_9[0][0]
batch_normalization_9 (BatchNor	(None, 32, 32, 440)	1760	concatenate_8[0][0]
activation_9 (Activation)	(None, 32, 32, 440)	0	batch_normalization_9[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 40)	158400	activation_9[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 480)	0	concatenate_8[0][0] conv2d_10[0][0]
batch_normalization_10 (BatchNo	(None, 32, 32, 480)	1920	concatenate_9[0][0]
activation_10 (Activation)	(None, 32, 32, 480)	0	batch_normalization_10[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 40)	172800	activation_10[0][0]
concatenate_10 (Concatenate)	(None, 32, 32, 520)	0	concatenate_9[0][0] conv2d_11[0][0]
batch_normalization_11 (BatchNo	(None, 32, 32, 520)	2080	concatenate_10[0][0]
activation_11 (Activation)	(None, 32, 32, 520)	0	batch_normalization_11[0][0]
conv2d_12 (Conv2D)	(None, 32, 32, 40)	187200	activation_11[0][0]
concatenate_11 (Concatenate)	(None, 32, 32, 560)	0	concatenate_10[0][0] conv2d_12[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 560)	2240	concatenate_11[0][0]
activation_12 (Activation)	(None, 32, 32, 560)	0	batch_normalization_12[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 40)	22400	activation_12[0][0]
average_pooling2d (AveragePooli	(None, 16, 16, 40)	0	conv2d_13[0][0]
batch_normalization_13 (BatchNo	(None, 16, 16, 40)	160	average_pooling2d[0][0]
activation_13 (Activation)	(None, 16, 16, 40)	0	batch_normalization_13[0][0]
conv2d_14 (Conv2D)	(None, 16, 16, 40)	14400	activation_13[0][0]
concatenate_12 (Concatenate)	(None, 16, 16, 80)	0	average_pooling2d[0][0] conv2d_14[0][0]
batch_normalization_14 (BatchNo	(None, 16, 16, 80)	320	concatenate_12[0][0]
activation_14 (Activation)	(None, 16, 16, 80)	0	batch_normalization_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 40)	28800	activation_14[0][0]
concatenate_13 (Concatenate)	(None, 16, 16, 120)	0	concatenate_12[0][0] conv2d_15[0][0]
batch_normalization_15 (BatchNo	(None, 16, 16, 120)	480	concatenate_13[0][0]
activation_15 (Activation)	(None, 16, 16, 120)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 40)	43200	activation_15[0][0]
concatenate_14 (Concatenate)	(None, 16, 16, 160)	0	concatenate_13[0][0] conv2d_16[0][0]
batch_normalization_16 (BatchNo	(None, 16, 16, 160)	640	concatenate_14[0][0]
activation_16 (Activation)	(None, 16, 16, 160)	0	batch_normalization_16[0][0]

conv2d_17 (Conv2D)	(None, 16, 16, 40)	57600	activation_16[0][0]
concatenate_15 (Concatenate)	(None, 16, 16, 200)	0	concatenate_14[0][0] conv2d_17[0][0]
batch_normalization_17 (BatchNo	(None, 16, 16, 200)	800	concatenate_15[0][0]
activation_17 (Activation)	(None, 16, 16, 200)	0	batch_normalization_17[0][0]
conv2d_18 (Conv2D)	(None, 16, 16, 40)	72000	activation_17[0][0]
concatenate_16 (Concatenate)	(None, 16, 16, 240)	0	concatenate_15[0][0] conv2d_18[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 240)	960	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 240)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 40)	86400	activation_18[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 280)	0	concatenate_16[0][0] conv2d_19[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 280)	1120	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 280)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 40)	100800	activation_19[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 320)	0	concatenate_17[0][0] conv2d_20[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 320)	1280	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 320)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 40)	115200	activation_20[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 360)	0	concatenate_18[0][0] conv2d_21[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 360)	1440	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 360)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 40)	129600	activation_21[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 400)	0	concatenate_19[0][0] conv2d_22[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 400)	1600	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 400)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 40)	144000	activation_22[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 440)	0	concatenate_20[0][0] conv2d_23[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 440)	1760	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 440)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 40)	158400	activation_23[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 480)	0	concatenate_21[0][0] conv2d_24[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 480)	1920	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 480)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 40)	172800	activation_24[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 520)	0	concatenate_22[0][0] conv2d_25[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 520)	2080	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 520)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 40)	20800	activation_25[0][0]

average_pooling2d_1 (AveragePoo	(None, 8, 8, 40)	0	conv2d_26[0][0]
batch_normalization_26 (BatchNo	(None, 8, 8, 40)	160	average_pooling2d_1[0][0]
activation_26 (Activation)	(None, 8, 8, 40)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 8, 8, 40)	14400	activation_26[0][0]
concatenate_24 (Concatenate)	(None, 8, 8, 80)	0	average_pooling2d_1[0][0] conv2d_27[0][0]
batch_normalization_27 (BatchNo	(None, 8, 8, 80)	320	concatenate_24[0][0]
activation_27 (Activation)	(None, 8, 8, 80)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 8, 8, 40)	28800	activation_27[0][0]
concatenate_25 (Concatenate)	(None, 8, 8, 120)	0	concatenate_24[0][0] conv2d_28[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 120)	480	concatenate_25[0][0]
activation_28 (Activation)	(None, 8, 8, 120)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 40)	43200	activation_28[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 160)	0	concatenate_25[0][0] conv2d_29[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 160)	640	concatenate_26[0][0]
activation_29 (Activation)	(None, 8, 8, 160)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 40)	57600	activation_29[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 200)	0	concatenate_26[0][0] conv2d_30[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 200)	800	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 200)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 40)	72000	activation_30[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 240)	0	concatenate_27[0][0] conv2d_31[0][0]
batch_normalization_31 (BatchNo	(None, 8, 8, 240)	960	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 240)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 40)	86400	activation_31[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 280)	0	concatenate_28[0][0] conv2d_32[0][0]
batch_normalization_32 (BatchNo	(None, 8, 8, 280)	1120	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 280)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 40)	100800	activation_32[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 320)	0	concatenate_29[0][0] conv2d_33[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 320)	1280	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 320)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 40)	115200	activation_33[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 360)	0	concatenate_30[0][0] conv2d_34[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 360)	1440	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 360)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 40)	129600	activation_34[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 400)	0	concatenate_31[0][0] conv2d_35[0][0]

batch_normalization_35 (BatchNo	(None, 8, 8, 400)	1600	concatenate_32[0][0]
activation_35 (Activation)	(None, 8, 8, 400)	0	batch_normalization_35[0][0]
conv2d_36 (Conv2D)	(None, 8, 8, 40)	144000	activation_35[0][0]
concatenate_33 (Concatenate)	(None, 8, 8, 440)	0	concatenate_32[0][0] conv2d_36[0][0]
batch_normalization_36 (BatchNo	(None, 8, 8, 440)	1760	concatenate_33[0][0]
activation_36 (Activation)	(None, 8, 8, 440)	0	batch_normalization_36[0][0]
conv2d_37 (Conv2D)	(None, 8, 8, 40)	158400	activation_36[0][0]
concatenate_34 (Concatenate)	(None, 8, 8, 480)	0	concatenate_33[0][0] conv2d_37[0][0]
batch_normalization_37 (BatchNo	(None, 8, 8, 480)	1920	concatenate_34[0][0]
activation_37 (Activation)	(None, 8, 8, 480)	0	batch_normalization_37[0][0]
conv2d_38 (Conv2D)	(None, 8, 8, 40)	172800	activation_37[0][0]
concatenate_35 (Concatenate)	(None, 8, 8, 520)	0	concatenate_34[0][0] conv2d_38[0][0]
batch_normalization_38 (BatchNo	(None, 8, 8, 520)	2080	concatenate_35[0][0]
activation_38 (Activation)	(None, 8, 8, 520)	0	batch_normalization_38[0][0]
conv2d_39 (Conv2D)	(None, 8, 8, 40)	20800	activation_38[0][0]
average_pooling2d_2 (AveragePoo	(None, 4, 4, 40)	0	conv2d_39[0][0]
batch_normalization_39 (BatchNo	(None, 4, 4, 40)	160	average_pooling2d_2[0][0]
activation_39 (Activation)	(None, 4, 4, 40)	0	batch_normalization_39[0][0]
conv2d_40 (Conv2D)	(None, 4, 4, 40)	14400	activation_39[0][0]
concatenate_36 (Concatenate)	(None, 4, 4, 80)	0	average_pooling2d_2[0][0] conv2d_40[0][0]
batch_normalization_40 (BatchNo	(None, 4, 4, 80)	320	concatenate_36[0][0]
activation_40 (Activation)	(None, 4, 4, 80)	0	batch_normalization_40[0][0]
conv2d_41 (Conv2D)	(None, 4, 4, 40)	28800	activation_40[0][0]
concatenate_37 (Concatenate)	(None, 4, 4, 120)	0	concatenate_36[0][0] conv2d_41[0][0]
batch_normalization_41 (BatchNo	(None, 4, 4, 120)	480	concatenate_37[0][0]
activation_41 (Activation)	(None, 4, 4, 120)	0	batch_normalization_41[0][0]
conv2d_42 (Conv2D)	(None, 4, 4, 40)	43200	activation_41[0][0]
concatenate_38 (Concatenate)	(None, 4, 4, 160)	0	concatenate_37[0][0] conv2d_42[0][0]
batch_normalization_42 (BatchNo	(None, 4, 4, 160)	640	concatenate_38[0][0]
activation_42 (Activation)	(None, 4, 4, 160)	0	batch_normalization_42[0][0]
conv2d_43 (Conv2D)	(None, 4, 4, 40)	57600	activation_42[0][0]
concatenate_39 (Concatenate)	(None, 4, 4, 200)	0	concatenate_38[0][0] conv2d_43[0][0]
batch_normalization_43 (BatchNo	(None, 4, 4, 200)	800	concatenate_39[0][0]
activation_43 (Activation)	(None, 4, 4, 200)	0	batch_normalization_43[0][0]
conv2d_44 (Conv2D)	(None, 4, 4, 40)	72000	activation_43[0][0]
concatenate_40 (Concatenate)	(None, 4, 4, 240)	0	concatenate_39[0][0] conv2d_44[0][0]
batch_normalization_44 (BatchNo	(None, 4, 4, 240)	960	concatenate_40[0][0]

```
activation_44 (Activation)      (None, 4, 4, 240)    0           batch_normalization_44[0][0]
__________________________________________________________________________________________________
conv2d_45 (Conv2D)              (None, 4, 4, 40)     86400       activation_44[0][0]
__________________________________________________________________________________________________
concatenate_41 (Concatenate)    (None, 4, 4, 280)    0           concatenate_40[0][0]
                                                                 conv2d_45[0][0]
__________________________________________________________________________________________________
batch_normalization_45 (BatchNo (None, 4, 4, 280)    1120        concatenate_41[0][0]
__________________________________________________________________________________________________
activation_45 (Activation)      (None, 4, 4, 280)    0           batch_normalization_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)              (None, 4, 4, 40)     100800      activation_45[0][0]
__________________________________________________________________________________________________
concatenate_42 (Concatenate)    (None, 4, 4, 320)    0           concatenate_41[0][0]
                                                                 conv2d_46[0][0]
__________________________________________________________________________________________________
batch_normalization_46 (BatchNo (None, 4, 4, 320)    1280        concatenate_42[0][0]
__________________________________________________________________________________________________
activation_46 (Activation)      (None, 4, 4, 320)    0           batch_normalization_46[0][0]
__________________________________________________________________________________________________
conv2d_47 (Conv2D)              (None, 4, 4, 40)     115200      activation_46[0][0]
__________________________________________________________________________________________________
concatenate_43 (Concatenate)    (None, 4, 4, 360)    0           concatenate_42[0][0]
                                                                 conv2d_47[0][0]
__________________________________________________________________________________________________
batch_normalization_47 (BatchNo (None, 4, 4, 360)    1440        concatenate_43[0][0]
__________________________________________________________________________________________________
activation_47 (Activation)      (None, 4, 4, 360)    0           batch_normalization_47[0][0]
__________________________________________________________________________________________________
conv2d_48 (Conv2D)              (None, 4, 4, 40)     129600      activation_47[0][0]
__________________________________________________________________________________________________
concatenate_44 (Concatenate)    (None, 4, 4, 400)    0           concatenate_43[0][0]
                                                                 conv2d_48[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, 4, 4, 400)    1600        concatenate_44[0][0]
__________________________________________________________________________________________________
activation_48 (Activation)      (None, 4, 4, 400)    0           batch_normalization_48[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, 4, 4, 40)     144000      activation_48[0][0]
__________________________________________________________________________________________________
concatenate_45 (Concatenate)    (None, 4, 4, 440)    0           concatenate_44[0][0]
                                                                 conv2d_49[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 4, 4, 440)    1760        concatenate_45[0][0]
__________________________________________________________________________________________________
activation_49 (Activation)      (None, 4, 4, 440)    0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 4, 4, 40)     158400      activation_49[0][0]
__________________________________________________________________________________________________
concatenate_46 (Concatenate)    (None, 4, 4, 480)    0           concatenate_45[0][0]
                                                                 conv2d_50[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, 4, 4, 480)    1920        concatenate_46[0][0]
__________________________________________________________________________________________________
activation_50 (Activation)      (None, 4, 4, 480)    0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 4, 4, 40)     172800      activation_50[0][0]
__________________________________________________________________________________________________
concatenate_47 (Concatenate)    (None, 4, 4, 520)    0           concatenate_46[0][0]
                                                                 conv2d_51[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 4, 4, 520)    2080        concatenate_47[0][0]
__________________________________________________________________________________________________
activation_51 (Activation)      (None, 4, 4, 520)    0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 520)    0           activation_51[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 2080)         0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           20810       flatten[0][0]
==================================================================================================
Total params: 4,812,890
Trainable params: 4,782,730
Non-trainable params: 30,160
__________________________________________________________________________________________________
```

In [13]:
```
print(len(model.layers))
```

```
211
```

In [14]:
```
epochs = 30
batch_size = 128
val_batch_size = 128
steps = len(Y_train)//batch_size
```

```
val_steps = len(Y_test)//val_batch_size
```

In [ ]:

In [15]:
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
      width_shift_range = 0.1,height_shift_range = 0.1,#rescale=1./255.,
    horizontal_flip = True,rotation_range = 10,
    featurewise_center=True,
    featurewise_std_normalization=True,
    zoom_range = 0.2, shear_range = 10,
)
train_datagen.fit(X_train)
```

In [16]:
```
test_datagen = ImageDataGenerator(
      width_shift_range = 0.1,height_shift_range = 0.1,#rescale=1./255.,
    horizontal_flip = True,rotation_range = 10,
    featurewise_center=True,
    featurewise_std_normalization=True,
    zoom_range = 0.2, shear_range = 10,
)

test_datagen.fit(X_test)
```

In [ ]:

In [ ]:

In [ ]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from skimage.data import chelsea,astronaut
import matplotlib.pyplot as plt
import numpy as np

imgs = np.stack([X_train[1111] for i in range(4*4)], axis=0)#np.stack([astronaut() for i in range(4*4)], axis=0

data_gen = ImageDataGenerator(
    width_shift_range = 0.1,height_shift_range = 0.1,#rescale=1./255.,
    fill_mode='nearest',validation_split=0.25,horizontal_flip = True,rotation_range = 90,
    preprocessing_function=lambda x: x[..., np.random.permutation([0, 1, 2])]
)
fig = plt.figure()
plt.subplots_adjust(wspace=.2, hspace=.2)
for index, image in enumerate(next(data_gen.flow(imgs)).astype(int)):
    ax = plt.subplot(4, 4, index + 1)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.imshow(image)
plt.show()
```

In [17]:
```
callbacks = [
    tf.keras.callbacks.ModelCheckpoint('./DenseNet_cifar10.h5', save_weights_only=False,save_best_only=True, \
                                       mode='max', monitor='val_accuracy',verbose=1),
    tf.keras.callbacks.ReduceLROnPlateau(monitor='val_accuracy', patience=2,mode='max',verbose=1),
]
```

In [ ]:

In [18]:
```
opt = tf.keras.optimizers.Adam()#SGD(learning_rate=0.1,momentum=0.9,)
model.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
```

In [ ]:

In [ ]:

In [19]:
```
#model.compile(loss='categorical_crossentropy',
#              optimizer=tf.keras.optimizers.SGD(learning_rate = 0.1,momentum = 0.9,nesterov = True), #Adam(),
#              metrics=['accuracy'])
model.fit(train_datagen.flow(X_train, Y_train,),steps_per_epoch=steps,
          validation_data=test_datagen.flow(X_test, Y_test),validation_steps=val_steps,

          epochs=100,
          callbacks=callbacks,
         )
```

```
Epoch 1/100
781/781 [==============================] - 263s 306ms/step - loss: 1.8543 - accuracy: 0.3461 - val_loss: 1.8558
- val_accuracy: 0.4345
```

```
Epoch 00001: val_accuracy improved from -inf to 0.43450, saving model to .\DenseNet_cifar10.h5
Epoch 2/100
781/781 [==============================] - 229s 293ms/step - loss: 1.4837 - accuracy: 0.4686 - val_loss: 1.4215
- val_accuracy: 0.4928

Epoch 00002: val_accuracy improved from 0.43450 to 0.49279, saving model to .\DenseNet_cifar10.h5
Epoch 3/100
781/781 [==============================] - 229s 293ms/step - loss: 1.2536 - accuracy: 0.5520 - val_loss: 1.3083
- val_accuracy: 0.5240

Epoch 00003: val_accuracy improved from 0.49279 to 0.52404, saving model to .\DenseNet_cifar10.h5
Epoch 4/100
781/781 [==============================] - 229s 293ms/step - loss: 1.0839 - accuracy: 0.6172 - val_loss: 1.1843
- val_accuracy: 0.5964

Epoch 00004: val_accuracy improved from 0.52404 to 0.59635, saving model to .\DenseNet_cifar10.h5
Epoch 5/100
781/781 [==============================] - 229s 293ms/step - loss: 0.9422 - accuracy: 0.6718 - val_loss: 1.0142
- val_accuracy: 0.6478

Epoch 00005: val_accuracy improved from 0.59635 to 0.64784, saving model to .\DenseNet_cifar10.h5
Epoch 6/100
781/781 [==============================] - 229s 293ms/step - loss: 0.8570 - accuracy: 0.7039 - val_loss: 0.9050
- val_accuracy: 0.6841

Epoch 00006: val_accuracy improved from 0.64784 to 0.68409, saving model to .\DenseNet_cifar10.h5
Epoch 7/100
781/781 [==============================] - 229s 293ms/step - loss: 0.7892 - accuracy: 0.7305 - val_loss: 0.8315
- val_accuracy: 0.7222

Epoch 00007: val_accuracy improved from 0.68409 to 0.72216, saving model to .\DenseNet_cifar10.h5
Epoch 8/100
781/781 [==============================] - 229s 293ms/step - loss: 0.7338 - accuracy: 0.7484 - val_loss: 0.7876
- val_accuracy: 0.7336

Epoch 00008: val_accuracy improved from 0.72216 to 0.73357, saving model to .\DenseNet_cifar10.h5
Epoch 9/100
781/781 [==============================] - 229s 293ms/step - loss: 0.6884 - accuracy: 0.7657 - val_loss: 0.8014
- val_accuracy: 0.7296

Epoch 00009: val_accuracy did not improve from 0.73357
Epoch 10/100
781/781 [==============================] - 229s 293ms/step - loss: 0.6426 - accuracy: 0.7846 - val_loss: 0.7702
- val_accuracy: 0.7476

Epoch 00010: val_accuracy improved from 0.73357 to 0.74760, saving model to .\DenseNet_cifar10.h5
Epoch 11/100
781/781 [==============================] - 229s 293ms/step - loss: 0.5976 - accuracy: 0.7974 - val_loss: 0.6578
- val_accuracy: 0.7788

Epoch 00011: val_accuracy improved from 0.74760 to 0.77885, saving model to .\DenseNet_cifar10.h5
Epoch 12/100
781/781 [==============================] - 229s 293ms/step - loss: 0.5759 - accuracy: 0.8046 - val_loss: 0.6731
- val_accuracy: 0.7722

Epoch 00012: val_accuracy did not improve from 0.77885
Epoch 13/100
781/781 [==============================] - 229s 293ms/step - loss: 0.5455 - accuracy: 0.8177 - val_loss: 0.6098
- val_accuracy: 0.7895

Epoch 00013: val_accuracy improved from 0.77885 to 0.78946, saving model to .\DenseNet_cifar10.h5
Epoch 14/100
781/781 [==============================] - 229s 293ms/step - loss: 0.5335 - accuracy: 0.8209 - val_loss: 0.5867
- val_accuracy: 0.7971

Epoch 00014: val_accuracy improved from 0.78946 to 0.79708, saving model to .\DenseNet_cifar10.h5
Epoch 15/100
781/781 [==============================] - 229s 293ms/step - loss: 0.5065 - accuracy: 0.8312 - val_loss: 0.5682
- val_accuracy: 0.8073

Epoch 00015: val_accuracy improved from 0.79708 to 0.80729, saving model to .\DenseNet_cifar10.h5
Epoch 16/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4873 - accuracy: 0.8346 - val_loss: 0.5602
- val_accuracy: 0.8079

Epoch 00016: val_accuracy improved from 0.80729 to 0.80789, saving model to .\DenseNet_cifar10.h5
Epoch 17/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4601 - accuracy: 0.8439 - val_loss: 0.5451
- val_accuracy: 0.8199

Epoch 00017: val_accuracy improved from 0.80789 to 0.81991, saving model to .\DenseNet_cifar10.h5
Epoch 18/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4430 - accuracy: 0.8517 - val_loss: 0.5576
```

```
- val_accuracy: 0.8159

Epoch 00018: val_accuracy did not improve from 0.81991
Epoch 19/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4369 - accuracy: 0.8540 - val_loss: 0.5101
- val_accuracy: 0.8317

Epoch 00019: val_accuracy improved from 0.81991 to 0.83173, saving model to .\DenseNet_cifar10.h5
Epoch 20/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4170 - accuracy: 0.8600 - val_loss: 0.4856
- val_accuracy: 0.8379

Epoch 00020: val_accuracy improved from 0.83173 to 0.83794, saving model to .\DenseNet_cifar10.h5
Epoch 21/100
781/781 [==============================] - 229s 293ms/step - loss: 0.4054 - accuracy: 0.8638 - val_loss: 0.5182
- val_accuracy: 0.8305

Epoch 00021: val_accuracy did not improve from 0.83794
Epoch 22/100
781/781 [==============================] - 229s 293ms/step - loss: 0.3970 - accuracy: 0.8656 - val_loss: 0.4723
- val_accuracy: 0.8377

Epoch 00022: val_accuracy did not improve from 0.83794

Epoch 00022: ReduceLROnPlateau reducing learning rate to 0.00010000000474974513.
Epoch 23/100
781/781 [==============================] - 229s 293ms/step - loss: 0.3190 - accuracy: 0.8948 - val_loss: 0.3742
- val_accuracy: 0.8766

Epoch 00023: val_accuracy improved from 0.83794 to 0.87660, saving model to .\DenseNet_cifar10.h5
Epoch 24/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2849 - accuracy: 0.9060 - val_loss: 0.3608
- val_accuracy: 0.8836

Epoch 00024: val_accuracy improved from 0.87660 to 0.88361, saving model to .\DenseNet_cifar10.h5
Epoch 25/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2755 - accuracy: 0.9090 - val_loss: 0.3631
- val_accuracy: 0.8796

Epoch 00025: val_accuracy did not improve from 0.88361
Epoch 26/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2723 - accuracy: 0.9090 - val_loss: 0.3849
- val_accuracy: 0.8728

Epoch 00026: val_accuracy did not improve from 0.88361

Epoch 00026: ReduceLROnPlateau reducing learning rate to 1.0000000474974514e-05.
Epoch 27/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2683 - accuracy: 0.9098 - val_loss: 0.3379
- val_accuracy: 0.8866

Epoch 00027: val_accuracy improved from 0.88361 to 0.88662, saving model to .\DenseNet_cifar10.h5
Epoch 28/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2633 - accuracy: 0.9117 - val_loss: 0.3514
- val_accuracy: 0.8878

Epoch 00028: val_accuracy improved from 0.88662 to 0.88782, saving model to .\DenseNet_cifar10.h5
Epoch 29/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2597 - accuracy: 0.9126 - val_loss: 0.3581
- val_accuracy: 0.8840

Epoch 00029: val_accuracy did not improve from 0.88782
Epoch 30/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2585 - accuracy: 0.9120 - val_loss: 0.3352
- val_accuracy: 0.8946

Epoch 00030: val_accuracy improved from 0.88782 to 0.89463, saving model to .\DenseNet_cifar10.h5
Epoch 31/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2603 - accuracy: 0.9137 - val_loss: 0.3497
- val_accuracy: 0.8844

Epoch 00031: val_accuracy did not improve from 0.89463
Epoch 32/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2594 - accuracy: 0.9123 - val_loss: 0.3626
- val_accuracy: 0.8808

Epoch 00032: val_accuracy did not improve from 0.89463

Epoch 00032: ReduceLROnPlateau reducing learning rate to 1.0000000656873453e-06.
Epoch 33/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2554 - accuracy: 0.9136 - val_loss: 0.3709
- val_accuracy: 0.8810
```

```
Epoch 00033: val_accuracy did not improve from 0.89463
Epoch 34/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2580 - accuracy: 0.9128 - val_loss: 0.3469
- val_accuracy: 0.8864

Epoch 00034: val_accuracy did not improve from 0.89463

Epoch 00034: ReduceLROnPlateau reducing learning rate to 1.0000001111620805e-07.
Epoch 35/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2499 - accuracy: 0.9159 - val_loss: 0.3448
- val_accuracy: 0.8926

Epoch 00035: val_accuracy did not improve from 0.89463
Epoch 36/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2543 - accuracy: 0.9164 - val_loss: 0.3321
- val_accuracy: 0.8930

Epoch 00036: val_accuracy did not improve from 0.89463

Epoch 00036: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-08.
Epoch 37/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2468 - accuracy: 0.9178 - val_loss: 0.3656
- val_accuracy: 0.8878

Epoch 00037: val_accuracy did not improve from 0.89463
Epoch 38/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2513 - accuracy: 0.9162 - val_loss: 0.3359
- val_accuracy: 0.8924

Epoch 00038: val_accuracy did not improve from 0.89463

Epoch 00038: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-09.
Epoch 39/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2565 - accuracy: 0.9149 - val_loss: 0.3485
- val_accuracy: 0.8836

Epoch 00039: val_accuracy did not improve from 0.89463
Epoch 40/100
781/781 [==============================] - 229s 294ms/step - loss: 0.2572 - accuracy: 0.9140 - val_loss: 0.3481
- val_accuracy: 0.8836

Epoch 00040: val_accuracy did not improve from 0.89463

Epoch 00040: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-10.
Epoch 41/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2513 - accuracy: 0.9141 - val_loss: 0.3471
- val_accuracy: 0.8824

Epoch 00041: val_accuracy did not improve from 0.89463
Epoch 42/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2507 - accuracy: 0.9178 - val_loss: 0.3601
- val_accuracy: 0.8784

Epoch 00042: val_accuracy did not improve from 0.89463

Epoch 00042: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-11.
Epoch 43/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2551 - accuracy: 0.9166 - val_loss: 0.3472
- val_accuracy: 0.8864

Epoch 00043: val_accuracy did not improve from 0.89463
Epoch 44/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2537 - accuracy: 0.9134 - val_loss: 0.3515
- val_accuracy: 0.8852

Epoch 00044: val_accuracy did not improve from 0.89463

Epoch 00044: ReduceLROnPlateau reducing learning rate to 1.000000082740371e-12.
Epoch 45/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2550 - accuracy: 0.9153 - val_loss: 0.3658
- val_accuracy: 0.8790

Epoch 00045: val_accuracy did not improve from 0.89463
Epoch 46/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2548 - accuracy: 0.9159 - val_loss: 0.3830
- val_accuracy: 0.8738

Epoch 00046: val_accuracy did not improve from 0.89463

Epoch 00046: ReduceLROnPlateau reducing learning rate to 1.0000001044244145e-13.
Epoch 47/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2456 - accuracy: 0.9182 - val_loss: 0.3490
- val_accuracy: 0.8808
```

```
Epoch 00047: val_accuracy did not improve from 0.89463
Epoch 48/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2569 - accuracy: 0.9146 - val_loss: 0.3475
- val_accuracy: 0.8870

Epoch 00048: val_accuracy did not improve from 0.89463

Epoch 00048: ReduceLROnPlateau reducing learning rate to 1.0000001179769417e-14.
Epoch 49/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2479 - accuracy: 0.9156 - val_loss: 0.3721
- val_accuracy: 0.8790

Epoch 00049: val_accuracy did not improve from 0.89463
Epoch 50/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2514 - accuracy: 0.9158 - val_loss: 0.3501
- val_accuracy: 0.8824

Epoch 00050: val_accuracy did not improve from 0.89463

Epoch 00050: ReduceLROnPlateau reducing learning rate to 1.0000001518582595e-15.
Epoch 51/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2532 - accuracy: 0.9140 - val_loss: 0.3545
- val_accuracy: 0.8802

Epoch 00051: val_accuracy did not improve from 0.89463
Epoch 52/100
781/781 [==============================] - 230s 294ms/step - loss: 0.2554 - accuracy: 0.9142 - val_loss: 0.3532
- val_accuracy: 0.8796

Epoch 00052: val_accuracy did not improve from 0.89463

Epoch 00052: ReduceLROnPlateau reducing learning rate to 1.0000001095066122e-16.
Epoch 53/100
781/781 [==============================] - 228s 292ms/step - loss: 0.2545 - accuracy: 0.9149 - val_loss: 0.3643
- val_accuracy: 0.8808

Epoch 00053: val_accuracy did not improve from 0.89463
Epoch 54/100
781/781 [==============================] - 228s 292ms/step - loss: 0.2527 - accuracy: 0.9164 - val_loss: 0.3526
- val_accuracy: 0.8862

Epoch 00054: val_accuracy did not improve from 0.89463

Epoch 00054: ReduceLROnPlateau reducing learning rate to 1.0000000830368326e-17.
Epoch 55/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2554 - accuracy: 0.9152 - val_loss: 0.3405
- val_accuracy: 0.8868

Epoch 00055: val_accuracy did not improve from 0.89463
Epoch 56/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2538 - accuracy: 0.9145 - val_loss: 0.3368
- val_accuracy: 0.8896

Epoch 00056: val_accuracy did not improve from 0.89463

Epoch 00056: ReduceLROnPlateau reducing learning rate to 1.0000000664932204e-18.
Epoch 57/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2548 - accuracy: 0.9144 - val_loss: 0.3500
- val_accuracy: 0.8890

Epoch 00057: val_accuracy did not improve from 0.89463
Epoch 58/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2565 - accuracy: 0.9165 - val_loss: 0.3551
- val_accuracy: 0.8838

Epoch 00058: val_accuracy did not improve from 0.89463

Epoch 00058: ReduceLROnPlateau reducing learning rate to 1.000000045813705e-19.
Epoch 59/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2509 - accuracy: 0.9166 - val_loss: 0.3462
- val_accuracy: 0.8888

Epoch 00059: val_accuracy did not improve from 0.89463
Epoch 60/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2581 - accuracy: 0.9150 - val_loss: 0.3616
- val_accuracy: 0.8862

Epoch 00060: val_accuracy did not improve from 0.89463

Epoch 00060: ReduceLROnPlateau reducing learning rate to 1.000000032889008e-20.
Epoch 61/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2522 - accuracy: 0.9163 - val_loss: 0.3485
```

```
- val_accuracy: 0.8850

Epoch 00061: val_accuracy did not improve from 0.89463
Epoch 62/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2505 - accuracy: 0.9164 - val_loss: 0.3417
- val_accuracy: 0.8842

Epoch 00062: val_accuracy did not improve from 0.89463

Epoch 00062: ReduceLROnPlateau reducing learning rate to 1.0000000490448793e-21.
Epoch 63/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2564 - accuracy: 0.9156 - val_loss: 0.3543
- val_accuracy: 0.8880

Epoch 00063: val_accuracy did not improve from 0.89463
Epoch 64/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2540 - accuracy: 0.9141 - val_loss: 0.3333
- val_accuracy: 0.8914

Epoch 00064: val_accuracy did not improve from 0.89463

Epoch 00064: ReduceLROnPlateau reducing learning rate to 1.0000000692397185e-22.
Epoch 65/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2566 - accuracy: 0.9152 - val_loss: 0.3578
- val_accuracy: 0.8802

Epoch 00065: val_accuracy did not improve from 0.89463
Epoch 66/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2561 - accuracy: 0.9138 - val_loss: 0.3514
- val_accuracy: 0.8828

Epoch 00066: val_accuracy did not improve from 0.89463

Epoch 00066: ReduceLROnPlateau reducing learning rate to 1.0000000944832675e-23.
Epoch 67/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2537 - accuracy: 0.9163 - val_loss: 0.3550
- val_accuracy: 0.8820

Epoch 00067: val_accuracy did not improve from 0.89463
Epoch 68/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2568 - accuracy: 0.9136 - val_loss: 0.3537
- val_accuracy: 0.8910

Epoch 00068: val_accuracy did not improve from 0.89463

Epoch 00068: ReduceLROnPlateau reducing learning rate to 1.0000000787060494e-24.
Epoch 69/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2521 - accuracy: 0.9155 - val_loss: 0.3536
- val_accuracy: 0.8818

Epoch 00069: val_accuracy did not improve from 0.89463
Epoch 70/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2566 - accuracy: 0.9161 - val_loss: 0.3380
- val_accuracy: 0.8860

Epoch 00070: val_accuracy did not improve from 0.89463

Epoch 00070: ReduceLROnPlateau reducing learning rate to 1.0000001181490946e-25.
Epoch 71/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2551 - accuracy: 0.9138 - val_loss: 0.3376
- val_accuracy: 0.8856

Epoch 00071: val_accuracy did not improve from 0.89463
Epoch 72/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2572 - accuracy: 0.9159 - val_loss: 0.3566
- val_accuracy: 0.8836

Epoch 00072: val_accuracy did not improve from 0.89463

Epoch 00072: ReduceLROnPlateau reducing learning rate to 1.0000001428009978e-26.
Epoch 73/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2558 - accuracy: 0.9141 - val_loss: 0.3560
- val_accuracy: 0.8844

Epoch 00073: val_accuracy did not improve from 0.89463
Epoch 74/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2584 - accuracy: 0.9145 - val_loss: 0.3551
- val_accuracy: 0.8840

Epoch 00074: val_accuracy did not improve from 0.89463

Epoch 00074: ReduceLROnPlateau reducing learning rate to 1.000000142800998e-27.
Epoch 75/100
```

```
781/781 [==============================] - 229s 293ms/step - loss: 0.2479 - accuracy: 0.9166 - val_loss: 0.3599
- val_accuracy: 0.8824

Epoch 00075: val_accuracy did not improve from 0.89463
Epoch 76/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2565 - accuracy: 0.9150 - val_loss: 0.3590
- val_accuracy: 0.8826

Epoch 00076: val_accuracy did not improve from 0.89463

Epoch 00076: ReduceLROnPlateau reducing learning rate to 1.0000001235416984e-28.
Epoch 77/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2545 - accuracy: 0.9177 - val_loss: 0.3699
- val_accuracy: 0.8836

Epoch 00077: val_accuracy did not improve from 0.89463
Epoch 78/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2564 - accuracy: 0.9144 - val_loss: 0.3579
- val_accuracy: 0.8828

Epoch 00078: val_accuracy did not improve from 0.89463

Epoch 00078: ReduceLROnPlateau reducing learning rate to 1.0000001235416985e-29.
Epoch 79/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2569 - accuracy: 0.9161 - val_loss: 0.3277
- val_accuracy: 0.8922

Epoch 00079: val_accuracy did not improve from 0.89463
Epoch 80/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2570 - accuracy: 0.9148 - val_loss: 0.3532
- val_accuracy: 0.8862

Epoch 00080: val_accuracy did not improve from 0.89463

Epoch 00080: ReduceLROnPlateau reducing learning rate to 1.0000001536343539e-30.
Epoch 81/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2481 - accuracy: 0.9160 - val_loss: 0.3728
- val_accuracy: 0.8790

Epoch 00081: val_accuracy did not improve from 0.89463
Epoch 82/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2521 - accuracy: 0.9166 - val_loss: 0.3572
- val_accuracy: 0.8830

Epoch 00082: val_accuracy did not improve from 0.89463

Epoch 00082: ReduceLROnPlateau reducing learning rate to 1.000000191250173e-31.
Epoch 83/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2602 - accuracy: 0.9114 - val_loss: 0.3463
- val_accuracy: 0.8800

Epoch 00083: val_accuracy did not improve from 0.89463
Epoch 84/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2546 - accuracy: 0.9147 - val_loss: 0.3639
- val_accuracy: 0.8782

Epoch 00084: val_accuracy did not improve from 0.89463

Epoch 00084: ReduceLROnPlateau reducing learning rate to 1.0000002147600601e-32.
Epoch 85/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2542 - accuracy: 0.9144 - val_loss: 0.3540
- val_accuracy: 0.8846

Epoch 00085: val_accuracy did not improve from 0.89463
Epoch 86/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2549 - accuracy: 0.9153 - val_loss: 0.3385
- val_accuracy: 0.8882

Epoch 00086: val_accuracy did not improve from 0.89463

Epoch 00086: ReduceLROnPlateau reducing learning rate to 1.0000002441474188e-33.
Epoch 87/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2507 - accuracy: 0.9155 - val_loss: 0.3434
- val_accuracy: 0.8876

Epoch 00087: val_accuracy did not improve from 0.89463
Epoch 88/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2497 - accuracy: 0.9165 - val_loss: 0.3622
- val_accuracy: 0.8834

Epoch 00088: val_accuracy did not improve from 0.89463

Epoch 00088: ReduceLROnPlateau reducing learning rate to 1.0000002074132203e-34.
```

```
Epoch 89/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2569 - accuracy: 0.9134 - val_loss: 0.3544
- val_accuracy: 0.8884

Epoch 00089: val_accuracy did not improve from 0.89463
Epoch 90/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2535 - accuracy: 0.9152 - val_loss: 0.3569
- val_accuracy: 0.8808

Epoch 00090: val_accuracy did not improve from 0.89463

Epoch 00090: ReduceLROnPlateau reducing learning rate to 1.0000001614954722e-35.
Epoch 91/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2600 - accuracy: 0.9143 - val_loss: 0.3606
- val_accuracy: 0.8858

Epoch 00091: val_accuracy did not improve from 0.89463
Epoch 92/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2486 - accuracy: 0.9169 - val_loss: 0.3525
- val_accuracy: 0.8834

Epoch 00092: val_accuracy did not improve from 0.89463

Epoch 00092: ReduceLROnPlateau reducing learning rate to 1.0000001614954723e-36.
Epoch 93/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2505 - accuracy: 0.9174 - val_loss: 0.3741
- val_accuracy: 0.8770

Epoch 00093: val_accuracy did not improve from 0.89463
Epoch 94/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2536 - accuracy: 0.9151 - val_loss: 0.3524
- val_accuracy: 0.8832

Epoch 00094: val_accuracy did not improve from 0.89463

Epoch 00094: ReduceLROnPlateau reducing learning rate to 1.0000001256222317e-37.
Epoch 95/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2524 - accuracy: 0.9163 - val_loss: 0.3461
- val_accuracy: 0.8908

Epoch 00095: val_accuracy did not improve from 0.89463
Epoch 96/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2597 - accuracy: 0.9137 - val_loss: 0.3563
- val_accuracy: 0.8878

Epoch 00096: val_accuracy did not improve from 0.89463

Epoch 00096: ReduceLROnPlateau reducing learning rate to 1.0000001032014561e-38.
Epoch 97/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2559 - accuracy: 0.9143 - val_loss: 0.3479
- val_accuracy: 0.8868

Epoch 00097: val_accuracy did not improve from 0.89463
Epoch 98/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2534 - accuracy: 0.9148 - val_loss: 0.3655
- val_accuracy: 0.8816

Epoch 00098: val_accuracy did not improve from 0.89463

Epoch 00098: ReduceLROnPlateau reducing learning rate to 1.0000000751754869e-39.
Epoch 99/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2533 - accuracy: 0.9156 - val_loss: 0.3482
- val_accuracy: 0.8820

Epoch 00099: val_accuracy did not improve from 0.89463
Epoch 100/100
781/781 [==============================] - 229s 293ms/step - loss: 0.2558 - accuracy: 0.9151 - val_loss: 0.3455
- val_accuracy: 0.8818

Epoch 00100: val_accuracy did not improve from 0.89463

Epoch 00100: ReduceLROnPlateau reducing learning rate to 1.0000002153053334e-40.
```

Out[19]: <keras.callbacks.History at 0x23f5ecbf100>

In [ ]:

In [ ]:

In [20]:
```
model.save('./MODEL')
```

```
INFO:tensorflow:Assets written to: ./MODEL\assets
```

In [17]:
```
from keras.models import load_model
```

```
m=load_model('./MODEL')
```

In [ ]:

In [18]:
```
callbacks = [
    tf.keras.callbacks.ModelCheckpoint('./DenseNet_cifar10.h5', save_weights_only=False,save_best_only=True, \
                                       mode='max', monitor='val_accuracy',verbose=1),
    tf.keras.callbacks.ReduceLROnPlateau(monitor='val_accuracy', patience=2,mode='max',verbose=1),
]

opt = tf.keras.optimizers.Adam(learning_rate=0.0001)#SGD(learning_rate=0.1,momentum=0.9,)
m.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
m.fit(train_datagen.flow(X_train, Y_train,),steps_per_epoch=steps,
          validation_data=test_datagen.flow(X_test, Y_test),validation_steps=val_steps,

          epochs=100,
          callbacks=callbacks,
         )
```

```
Epoch 1/100
390/390 [==============================] - 138s 297ms/step - loss: 0.2585 - accuracy: 0.9179 - val_loss: 0.4030
- val_accuracy: 0.8706

Epoch 00001: val_accuracy improved from -inf to 0.87059, saving model to .\DenseNet_cifar10.h5
Epoch 2/100
390/390 [==============================] - 118s 304ms/step - loss: 0.2503 - accuracy: 0.9178 - val_loss: 0.3686
- val_accuracy: 0.8826

Epoch 00002: val_accuracy improved from 0.87059 to 0.88261, saving model to .\DenseNet_cifar10.h5
Epoch 3/100
390/390 [==============================] - 119s 305ms/step - loss: 0.2626 - accuracy: 0.9120 - val_loss: 0.3751
- val_accuracy: 0.8814

Epoch 00003: val_accuracy did not improve from 0.88261
Epoch 4/100
390/390 [==============================] - 115s 295ms/step - loss: 0.2528 - accuracy: 0.9135 - val_loss: 0.3635
- val_accuracy: 0.8898

Epoch 00004: val_accuracy improved from 0.88261 to 0.88982, saving model to .\DenseNet_cifar10.h5
Epoch 5/100
390/390 [==============================] - 120s 307ms/step - loss: 0.2461 - accuracy: 0.9170 - val_loss: 0.3431
- val_accuracy: 0.8882

Epoch 00005: val_accuracy did not improve from 0.88982
Epoch 6/100
390/390 [==============================] - 107s 275ms/step - loss: 0.2513 - accuracy: 0.9160 - val_loss: 0.3615
- val_accuracy: 0.8858

Epoch 00006: val_accuracy did not improve from 0.88982

Epoch 00006: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-06.
Epoch 7/100
390/390 [==============================] - 120s 308ms/step - loss: 0.2443 - accuracy: 0.9167 - val_loss: 0.3361
- val_accuracy: 0.8918

Epoch 00007: val_accuracy improved from 0.88982 to 0.89183, saving model to .\DenseNet_cifar10.h5
Epoch 8/100
390/390 [==============================] - 118s 301ms/step - loss: 0.2368 - accuracy: 0.9185 - val_loss: 0.3465
- val_accuracy: 0.8866

Epoch 00008: val_accuracy did not improve from 0.89183
Epoch 9/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2466 - accuracy: 0.9196 - val_loss: 0.3577
- val_accuracy: 0.8822

Epoch 00009: val_accuracy did not improve from 0.89183

Epoch 00009: ReduceLROnPlateau reducing learning rate to 9.999999747378752e-07.
Epoch 10/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2468 - accuracy: 0.9182 - val_loss: 0.3547
- val_accuracy: 0.8866

Epoch 00010: val_accuracy did not improve from 0.89183
Epoch 11/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2450 - accuracy: 0.9196 - val_loss: 0.3443
- val_accuracy: 0.8906

Epoch 00011: val_accuracy did not improve from 0.89183

Epoch 00011: ReduceLROnPlateau reducing learning rate to 9.999999974752428e-08.
Epoch 12/100
```

```
390/390 [==============================] - 114s 293ms/step - loss: 0.2402 - accuracy: 0.9196 - val_loss: 0.3464
- val_accuracy: 0.8858

Epoch 00012: val_accuracy did not improve from 0.89183
Epoch 13/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2368 - accuracy: 0.9203 - val_loss: 0.3509
- val_accuracy: 0.8858

Epoch 00013: val_accuracy did not improve from 0.89183

Epoch 00013: ReduceLROnPlateau reducing learning rate to 1.0000000116860975e-08.
Epoch 14/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2429 - accuracy: 0.9186 - val_loss: 0.3531
- val_accuracy: 0.8898

Epoch 00014: val_accuracy did not improve from 0.89183
Epoch 15/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2458 - accuracy: 0.9192 - val_loss: 0.3638
- val_accuracy: 0.8826

Epoch 00015: val_accuracy did not improve from 0.89183

Epoch 00015: ReduceLROnPlateau reducing learning rate to 9.999999939225292e-10.
Epoch 16/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2320 - accuracy: 0.9247 - val_loss: 0.3391
- val_accuracy: 0.8978

Epoch 00016: val_accuracy improved from 0.89183 to 0.89784, saving model to .\DenseNet_cifar10.h5
Epoch 17/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2443 - accuracy: 0.9175 - val_loss: 0.3751
- val_accuracy: 0.8770

Epoch 00017: val_accuracy did not improve from 0.89784
Epoch 18/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2455 - accuracy: 0.9203 - val_loss: 0.3351
- val_accuracy: 0.8946

Epoch 00018: val_accuracy did not improve from 0.89784

Epoch 00018: ReduceLROnPlateau reducing learning rate to 9.999999717180686e-11.
Epoch 19/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2333 - accuracy: 0.9205 - val_loss: 0.3674
- val_accuracy: 0.8794

Epoch 00019: val_accuracy did not improve from 0.89784
Epoch 20/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2458 - accuracy: 0.9179 - val_loss: 0.3572
- val_accuracy: 0.8830

Epoch 00020: val_accuracy did not improve from 0.89784

Epoch 00020: ReduceLROnPlateau reducing learning rate to 9.99999943962493e-12.
Epoch 21/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2432 - accuracy: 0.9182 - val_loss: 0.3224
- val_accuracy: 0.8974

Epoch 00021: val_accuracy did not improve from 0.89784
Epoch 22/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2461 - accuracy: 0.9174 - val_loss: 0.3715
- val_accuracy: 0.8770

Epoch 00022: val_accuracy did not improve from 0.89784

Epoch 00022: ReduceLROnPlateau reducing learning rate to 9.999999092680235e-13.
Epoch 23/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2432 - accuracy: 0.9194 - val_loss: 0.3583
- val_accuracy: 0.8846

Epoch 00023: val_accuracy did not improve from 0.89784
Epoch 24/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2421 - accuracy: 0.9186 - val_loss: 0.3578
- val_accuracy: 0.8814

Epoch 00024: val_accuracy did not improve from 0.89784

Epoch 00024: ReduceLROnPlateau reducing learning rate to 9.9999988758398e-14.
Epoch 25/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2389 - accuracy: 0.9213 - val_loss: 0.3135
- val_accuracy: 0.8942

Epoch 00025: val_accuracy did not improve from 0.89784
Epoch 26/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2322 - accuracy: 0.9243 - val_loss: 0.3347
```

```
- val_accuracy: 0.8854

Epoch 00026: val_accuracy did not improve from 0.89784

Epoch 00026: ReduceLROnPlateau reducing learning rate to 9.999999146890344e-15.
Epoch 27/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2385 - accuracy: 0.9200 - val_loss: 0.3569
- val_accuracy: 0.8866

Epoch 00027: val_accuracy did not improve from 0.89784
Epoch 28/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2447 - accuracy: 0.9206 - val_loss: 0.3292
- val_accuracy: 0.8958

Epoch 00028: val_accuracy did not improve from 0.89784

Epoch 00028: ReduceLROnPlateau reducing learning rate to 9.999998977483753e-16.
Epoch 29/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2464 - accuracy: 0.9174 - val_loss: 0.3319
- val_accuracy: 0.8874

Epoch 00029: val_accuracy did not improve from 0.89784
Epoch 30/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2456 - accuracy: 0.9183 - val_loss: 0.3302
- val_accuracy: 0.8926

Epoch 00030: val_accuracy did not improve from 0.89784

Epoch 00030: ReduceLROnPlateau reducing learning rate to 9.999998977483754e-17.
Epoch 31/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2320 - accuracy: 0.9230 - val_loss: 0.3608
- val_accuracy: 0.8842

Epoch 00031: val_accuracy did not improve from 0.89784
Epoch 32/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2448 - accuracy: 0.9188 - val_loss: 0.3504
- val_accuracy: 0.8854

Epoch 00032: val_accuracy did not improve from 0.89784

Epoch 00032: ReduceLROnPlateau reducing learning rate to 9.999998845134856e-18.
Epoch 33/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2441 - accuracy: 0.9188 - val_loss: 0.3301
- val_accuracy: 0.8910

Epoch 00033: val_accuracy did not improve from 0.89784
Epoch 34/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2439 - accuracy: 0.9171 - val_loss: 0.3643
- val_accuracy: 0.8874

Epoch 00034: val_accuracy did not improve from 0.89784

Epoch 00034: ReduceLROnPlateau reducing learning rate to 9.999999010570977e-19.
Epoch 35/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2379 - accuracy: 0.9225 - val_loss: 0.3382
- val_accuracy: 0.8922

Epoch 00035: val_accuracy did not improve from 0.89784
Epoch 36/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2442 - accuracy: 0.9173 - val_loss: 0.3850
- val_accuracy: 0.8794

Epoch 00036: val_accuracy did not improve from 0.89784

Epoch 00036: ReduceLROnPlateau reducing learning rate to 9.999999424161285e-20.
Epoch 37/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2344 - accuracy: 0.9202 - val_loss: 0.3574
- val_accuracy: 0.8902

Epoch 00037: val_accuracy did not improve from 0.89784
Epoch 38/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2395 - accuracy: 0.9180 - val_loss: 0.3658
- val_accuracy: 0.8802

Epoch 00038: val_accuracy did not improve from 0.89784

Epoch 00038: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-21.
Epoch 39/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2406 - accuracy: 0.9192 - val_loss: 0.3516
- val_accuracy: 0.8794

Epoch 00039: val_accuracy did not improve from 0.89784
Epoch 40/100
```

```
390/390 [==============================] - 114s 292ms/step - loss: 0.2468 - accuracy: 0.9186 - val_loss: 0.3633
- val_accuracy: 0.8794

Epoch 00040: val_accuracy did not improve from 0.89784

Epoch 00040: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-22.
Epoch 41/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2399 - accuracy: 0.9225 - val_loss: 0.3398
- val_accuracy: 0.8922

Epoch 00041: val_accuracy did not improve from 0.89784
Epoch 42/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2407 - accuracy: 0.9170 - val_loss: 0.3707
- val_accuracy: 0.8806

Epoch 00042: val_accuracy did not improve from 0.89784

Epoch 00042: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-23.
Epoch 43/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2406 - accuracy: 0.9185 - val_loss: 0.3504
- val_accuracy: 0.8890

Epoch 00043: val_accuracy did not improve from 0.89784
Epoch 44/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2412 - accuracy: 0.9212 - val_loss: 0.3354
- val_accuracy: 0.8898

Epoch 00044: val_accuracy did not improve from 0.89784

Epoch 00044: ReduceLROnPlateau reducing learning rate to 9.999999682655227e-24.
Epoch 45/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2403 - accuracy: 0.9196 - val_loss: 0.3697
- val_accuracy: 0.8790

Epoch 00045: val_accuracy did not improve from 0.89784
Epoch 46/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2369 - accuracy: 0.9252 - val_loss: 0.3503
- val_accuracy: 0.8858

Epoch 00046: val_accuracy did not improve from 0.89784

Epoch 00046: ReduceLROnPlateau reducing learning rate to 9.999999998199588e-25.
Epoch 47/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2345 - accuracy: 0.9231 - val_loss: 0.3547
- val_accuracy: 0.8910

Epoch 00047: val_accuracy did not improve from 0.89784
Epoch 48/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2473 - accuracy: 0.9199 - val_loss: 0.3207
- val_accuracy: 0.8942

Epoch 00048: val_accuracy did not improve from 0.89784

Epoch 00048: ReduceLROnPlateau reducing learning rate to 1.0000000195414814e-25.
Epoch 49/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2426 - accuracy: 0.9208 - val_loss: 0.3477
- val_accuracy: 0.8862

Epoch 00049: val_accuracy did not improve from 0.89784
Epoch 50/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2346 - accuracy: 0.9215 - val_loss: 0.3398
- val_accuracy: 0.8954

Epoch 00050: val_accuracy did not improve from 0.89784

Epoch 00050: ReduceLROnPlateau reducing learning rate to 1.0000000195414814e-26.
Epoch 51/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2403 - accuracy: 0.9177 - val_loss: 0.3515
- val_accuracy: 0.8866

Epoch 00051: val_accuracy did not improve from 0.89784
Epoch 52/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2434 - accuracy: 0.9167 - val_loss: 0.3488
- val_accuracy: 0.8878

Epoch 00052: val_accuracy did not improve from 0.89784

Epoch 00052: ReduceLROnPlateau reducing learning rate to 9.999999887266024e-28.
Epoch 53/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2349 - accuracy: 0.9245 - val_loss: 0.3532
- val_accuracy: 0.8902

Epoch 00053: val_accuracy did not improve from 0.89784
```

```
Epoch 54/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2459 - accuracy: 0.9193 - val_loss: 0.3228
- val_accuracy: 0.8978

Epoch 00054: val_accuracy did not improve from 0.89784

Epoch 00054: ReduceLROnPlateau reducing learning rate to 1.0000000272452012e-28.
Epoch 55/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2403 - accuracy: 0.9196 - val_loss: 0.3262
- val_accuracy: 0.8882

Epoch 00055: val_accuracy did not improve from 0.89784
Epoch 56/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2430 - accuracy: 0.9194 - val_loss: 0.3813
- val_accuracy: 0.8790

Epoch 00056: val_accuracy did not improve from 0.89784

Epoch 00056: ReduceLROnPlateau reducing learning rate to 1.0000000031710769e-29.
Epoch 57/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2445 - accuracy: 0.9196 - val_loss: 0.3430
- val_accuracy: 0.8898

Epoch 00057: val_accuracy did not improve from 0.89784
Epoch 58/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2470 - accuracy: 0.9170 - val_loss: 0.3739
- val_accuracy: 0.8790

Epoch 00058: val_accuracy did not improve from 0.89784

Epoch 00058: ReduceLROnPlateau reducing learning rate to 1.0000000031710769e-30.
Epoch 59/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2412 - accuracy: 0.9187 - val_loss: 0.3648
- val_accuracy: 0.8802

Epoch 00059: val_accuracy did not improve from 0.89784
Epoch 60/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2394 - accuracy: 0.9200 - val_loss: 0.3654
- val_accuracy: 0.8882

Epoch 00060: val_accuracy did not improve from 0.89784

Epoch 00060: ReduceLROnPlateau reducing learning rate to 1.000000003171077e-31.
Epoch 61/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2374 - accuracy: 0.9189 - val_loss: 0.3570
- val_accuracy: 0.8886

Epoch 00061: val_accuracy did not improve from 0.89784
Epoch 62/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2451 - accuracy: 0.9176 - val_loss: 0.3542
- val_accuracy: 0.8838

Epoch 00062: val_accuracy did not improve from 0.89784

Epoch 00062: ReduceLROnPlateau reducing learning rate to 9.999999796611899e-33.
Epoch 63/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2448 - accuracy: 0.9193 - val_loss: 0.3498
- val_accuracy: 0.8890

Epoch 00063: val_accuracy did not improve from 0.89784
Epoch 64/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2536 - accuracy: 0.9158 - val_loss: 0.3491
- val_accuracy: 0.8862

Epoch 00064: val_accuracy did not improve from 0.89784

Epoch 00064: ReduceLROnPlateau reducing learning rate to 9.999999502738312e-34.
Epoch 65/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2409 - accuracy: 0.9184 - val_loss: 0.3493
- val_accuracy: 0.8826

Epoch 00065: val_accuracy did not improve from 0.89784
Epoch 66/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2490 - accuracy: 0.9155 - val_loss: 0.3100
- val_accuracy: 0.8934

Epoch 00066: val_accuracy did not improve from 0.89784

Epoch 00066: ReduceLROnPlateau reducing learning rate to 9.999999319067318e-35.
Epoch 67/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2353 - accuracy: 0.9206 - val_loss: 0.3704
- val_accuracy: 0.8846
```

```
Epoch 00067: val_accuracy did not improve from 0.89784
Epoch 68/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2340 - accuracy: 0.9229 - val_loss: 0.3117
- val_accuracy: 0.8974

Epoch 00068: val_accuracy did not improve from 0.89784

Epoch 00068: ReduceLROnPlateau reducing learning rate to 9.999999319067319e-36.
Epoch 69/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2399 - accuracy: 0.9203 - val_loss: 0.3424
- val_accuracy: 0.8922

Epoch 00069: val_accuracy did not improve from 0.89784
Epoch 70/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2424 - accuracy: 0.9186 - val_loss: 0.3215
- val_accuracy: 0.8982

Epoch 00070: val_accuracy improved from 0.89784 to 0.89824, saving model to .\DenseNet_cifar10.h5
Epoch 71/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2382 - accuracy: 0.9208 - val_loss: 0.3300
- val_accuracy: 0.8902

Epoch 00071: val_accuracy did not improve from 0.89824
Epoch 72/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2443 - accuracy: 0.9204 - val_loss: 0.3450
- val_accuracy: 0.8910

Epoch 00072: val_accuracy did not improve from 0.89824

Epoch 00072: ReduceLROnPlateau reducing learning rate to 9.999999462560281e-37.
Epoch 73/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2491 - accuracy: 0.9167 - val_loss: 0.3368
- val_accuracy: 0.8946

Epoch 00073: val_accuracy did not improve from 0.89824
Epoch 74/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2455 - accuracy: 0.9205 - val_loss: 0.3074
- val_accuracy: 0.8966

Epoch 00074: val_accuracy did not improve from 0.89824

Epoch 00074: ReduceLROnPlateau reducing learning rate to 9.99999946256028e-38.
Epoch 75/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2435 - accuracy: 0.9177 - val_loss: 0.3452
- val_accuracy: 0.8914

Epoch 00075: val_accuracy did not improve from 0.89824
Epoch 76/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2446 - accuracy: 0.9198 - val_loss: 0.3196
- val_accuracy: 0.8918

Epoch 00076: val_accuracy did not improve from 0.89824

Epoch 00076: ReduceLROnPlateau reducing learning rate to 9.99999991097579e-39.
Epoch 77/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2392 - accuracy: 0.9216 - val_loss: 0.3527
- val_accuracy: 0.8910

Epoch 00077: val_accuracy did not improve from 0.89824
Epoch 78/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2316 - accuracy: 0.9228 - val_loss: 0.3641
- val_accuracy: 0.8846

Epoch 00078: val_accuracy did not improve from 0.89824

Epoch 00078: ReduceLROnPlateau reducing learning rate to 9.999999350456405e-40.
Epoch 79/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2419 - accuracy: 0.9187 - val_loss: 0.3261
- val_accuracy: 0.8938

Epoch 00079: val_accuracy did not improve from 0.89824
Epoch 80/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2497 - accuracy: 0.9189 - val_loss: 0.3478
- val_accuracy: 0.8826

Epoch 00080: val_accuracy did not improve from 0.89824

Epoch 00080: ReduceLROnPlateau reducing learning rate to 1.0000002153053334e-40.
Epoch 81/100
390/390 [==============================] - 114s 292ms/step - loss: 0.2423 - accuracy: 0.9207 - val_loss: 0.3497
- val_accuracy: 0.8818

Epoch 00081: val_accuracy did not improve from 0.89824
```

```
Epoch 82/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2448 - accuracy: 0.9177 - val_loss: 0.3220
- val_accuracy: 0.9018

Epoch 00082: val_accuracy improved from 0.89824 to 0.90184, saving model to .\DenseNet_cifar10.h5
Epoch 83/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2387 - accuracy: 0.9207 - val_loss: 0.3183
- val_accuracy: 0.8906

Epoch 00083: val_accuracy did not improve from 0.90184
Epoch 84/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2379 - accuracy: 0.9209 - val_loss: 0.3507
- val_accuracy: 0.8826

Epoch 00084: val_accuracy did not improve from 0.90184

Epoch 00084: ReduceLROnPlateau reducing learning rate to 9.99994610111476e-42.
Epoch 85/100
390/390 [==============================] - 115s 294ms/step - loss: 0.2261 - accuracy: 0.9240 - val_loss: 0.3484
- val_accuracy: 0.8838

Epoch 00085: val_accuracy did not improve from 0.90184
Epoch 86/100
390/390 [==============================] - 117s 301ms/step - loss: 0.2461 - accuracy: 0.9166 - val_loss: 0.3558
- val_accuracy: 0.8830

Epoch 00086: val_accuracy did not improve from 0.90184

Epoch 00086: ReduceLROnPlateau reducing learning rate to 9.999665841421895e-43.
Epoch 87/100
390/390 [==============================] - 117s 301ms/step - loss: 0.2472 - accuracy: 0.9162 - val_loss: 0.3413
- val_accuracy: 0.8858

Epoch 00087: val_accuracy did not improve from 0.90184
Epoch 88/100
390/390 [==============================] - 115s 295ms/step - loss: 0.2342 - accuracy: 0.9216 - val_loss: 0.3588
- val_accuracy: 0.8838

Epoch 00088: val_accuracy did not improve from 0.90184

Epoch 00088: ReduceLROnPlateau reducing learning rate to 1.0005271035279195e-43.
Epoch 89/100
390/390 [==============================] - 115s 295ms/step - loss: 0.2365 - accuracy: 0.9211 - val_loss: 0.3379
- val_accuracy: 0.8902

Epoch 00089: val_accuracy did not improve from 0.90184
Epoch 90/100
390/390 [==============================] - 117s 300ms/step - loss: 0.2434 - accuracy: 0.9176 - val_loss: 0.3364
- val_accuracy: 0.8914

Epoch 00090: val_accuracy did not improve from 0.90184

Epoch 00090: ReduceLROnPlateau reducing learning rate to 9.949219096706202e-45.
Epoch 91/100
 30/390 [=>............................] - ETA: 1:45 - loss: 0.2158 - accuracy: 0.9208
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In [18], line 11
      7 opt = tf.keras.optimizers.Adam(learning_rate=0.0001)#SGD(learning_rate=0.1,momentum=0.9,)
      8 m.compile(loss='categorical_crossentropy',
      9               optimizer=opt,
     10               metrics=['accuracy'])
---> 11 m.fit(train_datagen.flow(X_train, Y_train,),steps_per_epoch=steps,
     12           validation_data=test_datagen.flow(X_test, Y_test),validation_steps=val_steps,
     13 
     14           epochs=100,
     15           callbacks=callbacks,
     16          )

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\engine\training.py:1189, in Model.fit(self, x, y, batch_size, epochs, verbose, callbacks, validation_split, validation_data, shuffle, class_weight, sample_weight, initial_epoch, steps_per_epoch, validation_steps, validation_batch_size, validation_freq, max_queue_size, workers, use_multiprocessing)
   1187 logs = tmp_logs  # No error, now safe to assign to logs.
   1188 end_step = step + data_handler.step_increment
-> 1189 callbacks.on_train_batch_end(end_step, logs)
   1190 if self.stop_training:
   1191   break

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:435, in CallbackList.on_train_batch_end(self, batch, logs)
    428 """Calls the `on_train_batch_end` methods of its callbacks.
    429 
```

```
    430 Args:
    431     batch: Integer, index of batch within the current epoch.
    432     logs: Dict. Aggregated metric results up until this batch.
    433 """
    434 if self._should_call_train_batch_hooks:
--> 435   self._call_batch_hook(ModeKeys.TRAIN, 'end', batch, logs=logs)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:295, in CallbackList._call_batch_hook(self, mode, hook, batch, logs)
    293   self._call_batch_begin_hook(mode, batch, logs)
    294 elif hook == 'end':
--> 295   self._call_batch_end_hook(mode, batch, logs)
    296 else:
    297   raise ValueError('Unrecognized hook: {}'.format(hook))

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:315, in CallbackList._call_batch_end_hook(self, mode, batch, logs)
    312   batch_time = time.time() - self._batch_start_time
    313   self._batch_times.append(batch_time)
--> 315 self._call_batch_hook_helper(hook_name, batch, logs)
    317 if len(self._batch_times) >= self._num_batches_for_timing_check:
    318   end_hook_name = hook_name

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:353, in CallbackList._call_batch_hook_helper(self, hook_name, batch, logs)
    351 for callback in self.callbacks:
    352   hook = getattr(callback, hook_name)
--> 353   hook(batch, logs)
    355 if self._check_timing:
    356   if hook_name not in self._hook_times:

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:1028, in ProgbarLogger.on_train_batch_end(self, batch, logs)
   1027 def on_train_batch_end(self, batch, logs=None):
-> 1028   self._batch_update_progbar(batch, logs)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\callbacks.py:1100, in ProgbarLogger._batch_update_progbar(self, batch, logs)
   1096   self.seen += add_seen
   1098 if self.verbose == 1:
   1099   # Only block async when verbose = 1.
-> 1100   logs = tf_utils.sync_to_numpy_or_python_type(logs)
   1101   self.progbar.update(self.seen, list(logs.items()), finalize=False)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\utils\tf_utils.py:516, in sync_to_numpy_or_python_type(tensors)
    513     return x.item() if np.ndim(x) == 0 else x
    514   return t  # Don't turn ragged or sparse tensors to NumPy.
--> 516 return tf.nest.map_structure(_to_single_numpy_or_python_type, tensors)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\tensorflow\python\util\nest.py:869, in map_structure(func, *structure, **kwargs)
    865 flat_structure = (flatten(s, expand_composites) for s in structure)
    866 entries = zip(*flat_structure)
    868 return pack_sequence_as(
--> 869     structure[0], [func(*x) for x in entries],
    870     expand_composites=expand_composites)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\tensorflow\python\util\nest.py:869, in <listcomp>(.0)
    865 flat_structure = (flatten(s, expand_composites) for s in structure)
    866 entries = zip(*flat_structure)
    868 return pack_sequence_as(
--> 869     structure[0], [func(*x) for x in entries],
    870     expand_composites=expand_composites)

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\keras\utils\tf_utils.py:512, in sync_to_numpy_or_python_type.<locals>._to_single_numpy_or_python_type(t)
    510 def _to_single_numpy_or_python_type(t):
    511   if isinstance(t, tf.Tensor):
--> 512     x = t.numpy()
    513     return x.item() if np.ndim(x) == 0 else x
    514   return t

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\tensorflow\python\framework\ops.py:1094, in _EagerTensorBase.numpy(self)
   1071 """Copy of the contents of this Tensor into a NumPy array or scalar.
   1072
   1073 Unlike NumPy arrays, Tensors are immutable, so this method has to copy
   (...)
   1091     NumPy dtype.
   1092 """
   1093 # TODO(slebedev): Consider avoiding a copy for non-CPU or remote tensors.
-> 1094 maybe_arr = self._numpy()  # pylint: disable=protected-access
```

```
   1095 return maybe_arr.copy() if isinstance(maybe_arr, np.ndarray) else maybe_arr

File ~\Anaconda3\envs\tf_GPU\lib\site-packages\tensorflow\python\framework\ops.py:1060, in _EagerTensorBase._nu
mpy(self)
   1058 def _numpy(self):
   1059   try:
-> 1060     return self._numpy_internal()
   1061   except core._NotOkStatusException as e:  # pylint: disable=protected-access
   1062     six.raise_from(core._status_to_exception(e.code, e.message), None)

KeyboardInterrupt:
```

In [ ]:

In [ ]:

In [19]:
```
m.save('./MODEL2')
```

```
INFO:tensorflow:Assets written to: ./MODEL2\assets
```

In [17]:
```
from keras.models import load_model
m1=load_model('./MODEL2')
```

In [19]:
```
callbacks = [
    tf.keras.callbacks.ModelCheckpoint('./DenseNet_cifar10.h5', save_weights_only=False,save_best_only=True, \
                                       mode='max', monitor='val_accuracy',verbose=1),
    tf.keras.callbacks.ReduceLROnPlateau(monitor='val_accuracy', patience=2,mode='max',verbose=1),
]

opt = tf.keras.optimizers.Adam(learning_rate=0.000001)#SGD(learning_rate=0.1,momentum=0.9,)
m1.compile(loss='categorical_crossentropy',
              optimizer=opt,
              metrics=['accuracy'])
m1.fit(train_datagen.flow(X_train, Y_train,),steps_per_epoch=steps,
          validation_data=test_datagen.flow(X_test, Y_test),validation_steps=val_steps,
          epochs=100,
          callbacks=callbacks,
         )
```

```
Epoch 1/100
390/390 [==============================] - 119s 294ms/step - loss: 0.2439 - accuracy: 0.9220 - val_loss: 0.3418
- val_accuracy: 0.8866

Epoch 00001: val_accuracy improved from -inf to 0.88662, saving model to .\DenseNet_cifar10.h5
Epoch 2/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2327 - accuracy: 0.9207 - val_loss: 0.3189
- val_accuracy: 0.8922

Epoch 00002: val_accuracy improved from 0.88662 to 0.89223, saving model to .\DenseNet_cifar10.h5
Epoch 3/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2271 - accuracy: 0.9251 - val_loss: 0.3593
- val_accuracy: 0.8838

Epoch 00003: val_accuracy did not improve from 0.89223
Epoch 4/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2281 - accuracy: 0.9231 - val_loss: 0.3550
- val_accuracy: 0.8854

Epoch 00004: val_accuracy did not improve from 0.89223

Epoch 00004: ReduceLROnPlateau reducing learning rate to 9.999999974752428e-08.
Epoch 5/100
390/390 [==============================] - 122s 312ms/step - loss: 0.2367 - accuracy: 0.9204 - val_loss: 0.3594
- val_accuracy: 0.8790

Epoch 00005: val_accuracy did not improve from 0.89223
Epoch 6/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2450 - accuracy: 0.9192 - val_loss: 0.3813
- val_accuracy: 0.8794

Epoch 00006: val_accuracy did not improve from 0.89223

Epoch 00006: ReduceLROnPlateau reducing learning rate to 1.0000000116860975e-08.
Epoch 7/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2455 - accuracy: 0.9205 - val_loss: 0.3206
- val_accuracy: 0.8926

Epoch 00007: val_accuracy improved from 0.89223 to 0.89263, saving model to .\DenseNet_cifar10.h5
Epoch 8/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2396 - accuracy: 0.9206 - val_loss: 0.3399
- val_accuracy: 0.8874

Epoch 00008: val_accuracy did not improve from 0.89263
```

```
Epoch 9/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2468 - accuracy: 0.9163 - val_loss: 0.3493
- val_accuracy: 0.8862

Epoch 00009: val_accuracy did not improve from 0.89263

Epoch 00009: ReduceLROnPlateau reducing learning rate to 9.999999939225292e-10.
Epoch 10/100
390/390 [==============================] - 126s 323ms/step - loss: 0.2448 - accuracy: 0.9163 - val_loss: 0.3401
- val_accuracy: 0.8810

Epoch 00010: val_accuracy did not improve from 0.89263
Epoch 11/100
390/390 [==============================] - 131s 335ms/step - loss: 0.2432 - accuracy: 0.9211 - val_loss: 0.3562
- val_accuracy: 0.8850

Epoch 00011: val_accuracy did not improve from 0.89263

Epoch 00011: ReduceLROnPlateau reducing learning rate to 9.999999717180686e-11.
Epoch 12/100
390/390 [==============================] - 139s 356ms/step - loss: 0.2408 - accuracy: 0.9193 - val_loss: 0.3598
- val_accuracy: 0.8786

Epoch 00012: val_accuracy did not improve from 0.89263
Epoch 13/100
390/390 [==============================] - 140s 359ms/step - loss: 0.2416 - accuracy: 0.9232 - val_loss: 0.3312
- val_accuracy: 0.8946

Epoch 00013: val_accuracy improved from 0.89263 to 0.89463, saving model to .\DenseNet_cifar10.h5
Epoch 14/100
390/390 [==============================] - 139s 356ms/step - loss: 0.2376 - accuracy: 0.9225 - val_loss: 0.3526
- val_accuracy: 0.8858

Epoch 00014: val_accuracy did not improve from 0.89463
Epoch 15/100
390/390 [==============================] - 139s 358ms/step - loss: 0.2328 - accuracy: 0.9200 - val_loss: 0.3695
- val_accuracy: 0.8822

Epoch 00015: val_accuracy did not improve from 0.89463

Epoch 00015: ReduceLROnPlateau reducing learning rate to 9.99999943962493e-12.
Epoch 16/100
390/390 [==============================] - 139s 356ms/step - loss: 0.2423 - accuracy: 0.9218 - val_loss: 0.3409
- val_accuracy: 0.8870

Epoch 00016: val_accuracy did not improve from 0.89463
Epoch 17/100
390/390 [==============================] - 129s 331ms/step - loss: 0.2353 - accuracy: 0.9216 - val_loss: 0.3393
- val_accuracy: 0.8930

Epoch 00017: val_accuracy did not improve from 0.89463

Epoch 00017: ReduceLROnPlateau reducing learning rate to 9.999999092680235e-13.
Epoch 18/100
390/390 [==============================] - 114s 293ms/step - loss: 0.2314 - accuracy: 0.9231 - val_loss: 0.3431
- val_accuracy: 0.8906

Epoch 00018: val_accuracy did not improve from 0.89463
Epoch 19/100
390/390 [==============================] - 80s 206ms/step - loss: 0.2412 - accuracy: 0.9168 - val_loss: 0.3663
- val_accuracy: 0.8814

Epoch 00019: val_accuracy did not improve from 0.89463

Epoch 00019: ReduceLROnPlateau reducing learning rate to 9.9999988758398e-14.
Epoch 20/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2455 - accuracy: 0.9213 - val_loss: 0.3516
- val_accuracy: 0.8762

Epoch 00020: val_accuracy did not improve from 0.89463
Epoch 21/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2370 - accuracy: 0.9198 - val_loss: 0.3563
- val_accuracy: 0.8866

Epoch 00021: val_accuracy did not improve from 0.89463

Epoch 00021: ReduceLROnPlateau reducing learning rate to 9.999999146890344e-15.
Epoch 22/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2351 - accuracy: 0.9215 - val_loss: 0.3396
- val_accuracy: 0.8942

Epoch 00022: val_accuracy did not improve from 0.89463
Epoch 23/100
```

```
390/390 [==============================] - 81s 207ms/step - loss: 0.2300 - accuracy: 0.9237 - val_loss: 0.3246
- val_accuracy: 0.8950

Epoch 00023: val_accuracy improved from 0.89463 to 0.89503, saving model to .\DenseNet_cifar10.h5
Epoch 24/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2389 - accuracy: 0.9202 - val_loss: 0.3470
- val_accuracy: 0.8854

Epoch 00024: val_accuracy did not improve from 0.89503
Epoch 25/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2335 - accuracy: 0.9216 - val_loss: 0.3149
- val_accuracy: 0.8958

Epoch 00025: val_accuracy improved from 0.89503 to 0.89583, saving model to .\DenseNet_cifar10.h5
Epoch 26/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2405 - accuracy: 0.9185 - val_loss: 0.3545
- val_accuracy: 0.8870

Epoch 00026: val_accuracy did not improve from 0.89583
Epoch 27/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2414 - accuracy: 0.9204 - val_loss: 0.3177
- val_accuracy: 0.8946

Epoch 00027: val_accuracy did not improve from 0.89583

Epoch 00027: ReduceLROnPlateau reducing learning rate to 9.999998977483753e-16.
Epoch 28/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2291 - accuracy: 0.9236 - val_loss: 0.3385
- val_accuracy: 0.8878

Epoch 00028: val_accuracy did not improve from 0.89583
Epoch 29/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2307 - accuracy: 0.9229 - val_loss: 0.3457
- val_accuracy: 0.8938

Epoch 00029: val_accuracy did not improve from 0.89583

Epoch 00029: ReduceLROnPlateau reducing learning rate to 9.999998977483754e-17.
Epoch 30/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2361 - accuracy: 0.9250 - val_loss: 0.3135
- val_accuracy: 0.8974

Epoch 00030: val_accuracy improved from 0.89583 to 0.89744, saving model to .\DenseNet_cifar10.h5
Epoch 31/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2590 - accuracy: 0.9143 - val_loss: 0.3423
- val_accuracy: 0.8898

Epoch 00031: val_accuracy did not improve from 0.89744
Epoch 32/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2483 - accuracy: 0.9201 - val_loss: 0.3110
- val_accuracy: 0.8934

Epoch 00032: val_accuracy did not improve from 0.89744

Epoch 00032: ReduceLROnPlateau reducing learning rate to 9.999998845134856e-18.
Epoch 33/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2383 - accuracy: 0.9192 - val_loss: 0.3355
- val_accuracy: 0.8846

Epoch 00033: val_accuracy did not improve from 0.89744
Epoch 34/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2373 - accuracy: 0.9212 - val_loss: 0.3495
- val_accuracy: 0.8894

Epoch 00034: val_accuracy did not improve from 0.89744

Epoch 00034: ReduceLROnPlateau reducing learning rate to 9.999999010570977e-19.
Epoch 35/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2433 - accuracy: 0.9193 - val_loss: 0.3250
- val_accuracy: 0.8858

Epoch 00035: val_accuracy did not improve from 0.89744
Epoch 36/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2408 - accuracy: 0.9187 - val_loss: 0.3423
- val_accuracy: 0.8950

Epoch 00036: val_accuracy did not improve from 0.89744

Epoch 00036: ReduceLROnPlateau reducing learning rate to 9.999999424161285e-20.
Epoch 37/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2412 - accuracy: 0.9202 - val_loss: 0.3599
- val_accuracy: 0.8766
```

```
Epoch 00037: val_accuracy did not improve from 0.89744
Epoch 38/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2482 - accuracy: 0.9196 - val_loss: 0.3704
- val_accuracy: 0.8830

Epoch 00038: val_accuracy did not improve from 0.89744

Epoch 00038: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-21.
Epoch 39/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2415 - accuracy: 0.9183 - val_loss: 0.3618
- val_accuracy: 0.8818

Epoch 00039: val_accuracy did not improve from 0.89744
Epoch 40/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2438 - accuracy: 0.9187 - val_loss: 0.3381
- val_accuracy: 0.8934

Epoch 00040: val_accuracy did not improve from 0.89744

Epoch 00040: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-22.
Epoch 41/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2320 - accuracy: 0.9232 - val_loss: 0.3426
- val_accuracy: 0.8902

Epoch 00041: val_accuracy did not improve from 0.89744
Epoch 42/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2452 - accuracy: 0.9168 - val_loss: 0.3194
- val_accuracy: 0.8970

Epoch 00042: val_accuracy did not improve from 0.89744

Epoch 00042: ReduceLROnPlateau reducing learning rate to 9.999999682655225e-23.
Epoch 43/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2431 - accuracy: 0.9186 - val_loss: 0.3615
- val_accuracy: 0.8810

Epoch 00043: val_accuracy did not improve from 0.89744
Epoch 44/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2498 - accuracy: 0.9139 - val_loss: 0.3475
- val_accuracy: 0.8918

Epoch 00044: val_accuracy did not improve from 0.89744

Epoch 00044: ReduceLROnPlateau reducing learning rate to 9.999999682655227e-24.
Epoch 45/100
390/390 [==============================] - 82s 211ms/step - loss: 0.2338 - accuracy: 0.9218 - val_loss: 0.3284
- val_accuracy: 0.8914

Epoch 00045: val_accuracy did not improve from 0.89744
Epoch 46/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2390 - accuracy: 0.9198 - val_loss: 0.3461
- val_accuracy: 0.8850

Epoch 00046: val_accuracy did not improve from 0.89744

Epoch 00046: ReduceLROnPlateau reducing learning rate to 9.999999998199588e-25.
Epoch 47/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2517 - accuracy: 0.9158 - val_loss: 0.3464
- val_accuracy: 0.8898

Epoch 00047: val_accuracy did not improve from 0.89744
Epoch 48/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2459 - accuracy: 0.9175 - val_loss: 0.3433
- val_accuracy: 0.8886

Epoch 00048: val_accuracy did not improve from 0.89744

Epoch 00048: ReduceLROnPlateau reducing learning rate to 1.0000000195414814e-25.
Epoch 49/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2427 - accuracy: 0.9203 - val_loss: 0.3609
- val_accuracy: 0.8834

Epoch 00049: val_accuracy did not improve from 0.89744
Epoch 50/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2412 - accuracy: 0.9203 - val_loss: 0.3678
- val_accuracy: 0.8846

Epoch 00050: val_accuracy did not improve from 0.89744

Epoch 00050: ReduceLROnPlateau reducing learning rate to 1.0000000195414814e-26.
Epoch 51/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2479 - accuracy: 0.9189 - val_loss: 0.3719
- val_accuracy: 0.8866
```

```
Epoch 00051: val_accuracy did not improve from 0.89744
Epoch 52/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2300 - accuracy: 0.9232 - val_loss: 0.3176
- val_accuracy: 0.8986

Epoch 00052: val_accuracy improved from 0.89744 to 0.89864, saving model to .\DenseNet_cifar10.h5
Epoch 53/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2333 - accuracy: 0.9238 - val_loss: 0.3634
- val_accuracy: 0.8798

Epoch 00053: val_accuracy did not improve from 0.89864
Epoch 54/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2394 - accuracy: 0.9179 - val_loss: 0.2890
- val_accuracy: 0.9087

Epoch 00054: val_accuracy improved from 0.89864 to 0.90865, saving model to .\DenseNet_cifar10.h5
Epoch 55/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2367 - accuracy: 0.9227 - val_loss: 0.3337
- val_accuracy: 0.8938

Epoch 00055: val_accuracy did not improve from 0.90865
Epoch 56/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2397 - accuracy: 0.9200 - val_loss: 0.3611
- val_accuracy: 0.8842

Epoch 00056: val_accuracy did not improve from 0.90865

Epoch 00056: ReduceLROnPlateau reducing learning rate to 9.999999887266024e-28.
Epoch 57/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2378 - accuracy: 0.9209 - val_loss: 0.3732
- val_accuracy: 0.8830

Epoch 00057: val_accuracy did not improve from 0.90865
Epoch 58/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2258 - accuracy: 0.9233 - val_loss: 0.3313
- val_accuracy: 0.8938

Epoch 00058: val_accuracy did not improve from 0.90865

Epoch 00058: ReduceLROnPlateau reducing learning rate to 1.0000000272452012e-28.
Epoch 59/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2442 - accuracy: 0.9193 - val_loss: 0.3444
- val_accuracy: 0.8878

Epoch 00059: val_accuracy did not improve from 0.90865
Epoch 60/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2518 - accuracy: 0.9162 - val_loss: 0.3626
- val_accuracy: 0.8846

Epoch 00060: val_accuracy did not improve from 0.90865

Epoch 00060: ReduceLROnPlateau reducing learning rate to 1.0000000031710769e-29.
Epoch 61/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2394 - accuracy: 0.9181 - val_loss: 0.2981
- val_accuracy: 0.9022

Epoch 00061: val_accuracy did not improve from 0.90865
Epoch 62/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2402 - accuracy: 0.9214 - val_loss: 0.3361
- val_accuracy: 0.8950

Epoch 00062: val_accuracy did not improve from 0.90865

Epoch 00062: ReduceLROnPlateau reducing learning rate to 1.0000000031710769e-30.
Epoch 63/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2393 - accuracy: 0.9235 - val_loss: 0.3067
- val_accuracy: 0.8982

Epoch 00063: val_accuracy did not improve from 0.90865
Epoch 64/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2356 - accuracy: 0.9200 - val_loss: 0.3328
- val_accuracy: 0.8902

Epoch 00064: val_accuracy did not improve from 0.90865

Epoch 00064: ReduceLROnPlateau reducing learning rate to 1.000000003171077e-31.
Epoch 65/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2407 - accuracy: 0.9187 - val_loss: 0.3645
- val_accuracy: 0.8870

Epoch 00065: val_accuracy did not improve from 0.90865
Epoch 66/100
```

```
390/390 [==============================] - 81s 208ms/step - loss: 0.2487 - accuracy: 0.9183 - val_loss: 0.3375
- val_accuracy: 0.8818

Epoch 00066: val_accuracy did not improve from 0.90865

Epoch 00066: ReduceLROnPlateau reducing learning rate to 9.999999796611899e-33.
Epoch 67/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2405 - accuracy: 0.9204 - val_loss: 0.3454
- val_accuracy: 0.8870

Epoch 00067: val_accuracy did not improve from 0.90865
Epoch 68/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2394 - accuracy: 0.9220 - val_loss: 0.3221
- val_accuracy: 0.9018

Epoch 00068: val_accuracy did not improve from 0.90865

Epoch 00068: ReduceLROnPlateau reducing learning rate to 9.999999502738312e-34.
Epoch 69/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2363 - accuracy: 0.9224 - val_loss: 0.3656
- val_accuracy: 0.8814

Epoch 00069: val_accuracy did not improve from 0.90865
Epoch 70/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2501 - accuracy: 0.9173 - val_loss: 0.3303
- val_accuracy: 0.8950

Epoch 00070: val_accuracy did not improve from 0.90865

Epoch 00070: ReduceLROnPlateau reducing learning rate to 9.999999319067318e-35.
Epoch 71/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2432 - accuracy: 0.9185 - val_loss: 0.3347
- val_accuracy: 0.8886

Epoch 00071: val_accuracy did not improve from 0.90865
Epoch 72/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2448 - accuracy: 0.9179 - val_loss: 0.3451
- val_accuracy: 0.8898

Epoch 00072: val_accuracy did not improve from 0.90865

Epoch 00072: ReduceLROnPlateau reducing learning rate to 9.999999319067319e-36.
Epoch 73/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2436 - accuracy: 0.9178 - val_loss: 0.3530
- val_accuracy: 0.8862

Epoch 00073: val_accuracy did not improve from 0.90865
Epoch 74/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2300 - accuracy: 0.9232 - val_loss: 0.3625
- val_accuracy: 0.8814

Epoch 00074: val_accuracy did not improve from 0.90865

Epoch 00074: ReduceLROnPlateau reducing learning rate to 9.999999462560281e-37.
Epoch 75/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2401 - accuracy: 0.9204 - val_loss: 0.3444
- val_accuracy: 0.8870

Epoch 00075: val_accuracy did not improve from 0.90865
Epoch 76/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2349 - accuracy: 0.9198 - val_loss: 0.3433
- val_accuracy: 0.8922

Epoch 00076: val_accuracy did not improve from 0.90865

Epoch 00076: ReduceLROnPlateau reducing learning rate to 9.99999946256028e-38.
Epoch 77/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2476 - accuracy: 0.9182 - val_loss: 0.3599
- val_accuracy: 0.8922

Epoch 00077: val_accuracy did not improve from 0.90865
Epoch 78/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2437 - accuracy: 0.9169 - val_loss: 0.3297
- val_accuracy: 0.8946

Epoch 00078: val_accuracy did not improve from 0.90865

Epoch 00078: ReduceLROnPlateau reducing learning rate to 9.99999991097579e-39.
Epoch 79/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2364 - accuracy: 0.9208 - val_loss: 0.3430
- val_accuracy: 0.8870

Epoch 00079: val_accuracy did not improve from 0.90865
```

```
Epoch 80/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2375 - accuracy: 0.9217 - val_loss: 0.3365
- val_accuracy: 0.8922

Epoch 00080: val_accuracy did not improve from 0.90865

Epoch 00080: ReduceLROnPlateau reducing learning rate to 9.999999350456405e-40.
Epoch 81/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2390 - accuracy: 0.9190 - val_loss: 0.3353
- val_accuracy: 0.8930

Epoch 00081: val_accuracy did not improve from 0.90865
Epoch 82/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2380 - accuracy: 0.9218 - val_loss: 0.3318
- val_accuracy: 0.8894

Epoch 00082: val_accuracy did not improve from 0.90865

Epoch 00082: ReduceLROnPlateau reducing learning rate to 1.0000002153053334e-40.
Epoch 83/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2538 - accuracy: 0.9140 - val_loss: 0.3476
- val_accuracy: 0.8814

Epoch 00083: val_accuracy did not improve from 0.90865
Epoch 84/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2365 - accuracy: 0.9227 - val_loss: 0.3237
- val_accuracy: 0.8934

Epoch 00084: val_accuracy did not improve from 0.90865

Epoch 00084: ReduceLROnPlateau reducing learning rate to 9.99994610111476e-42.
Epoch 85/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2441 - accuracy: 0.9179 - val_loss: 0.2940
- val_accuracy: 0.9010

Epoch 00085: val_accuracy did not improve from 0.90865
Epoch 86/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2414 - accuracy: 0.9192 - val_loss: 0.3461
- val_accuracy: 0.8842

Epoch 00086: val_accuracy did not improve from 0.90865

Epoch 00086: ReduceLROnPlateau reducing learning rate to 9.999665841421895e-43.
Epoch 87/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2357 - accuracy: 0.9238 - val_loss: 0.3718
- val_accuracy: 0.8754

Epoch 00087: val_accuracy did not improve from 0.90865
Epoch 88/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2377 - accuracy: 0.9223 - val_loss: 0.3527
- val_accuracy: 0.8842

Epoch 00088: val_accuracy did not improve from 0.90865

Epoch 00088: ReduceLROnPlateau reducing learning rate to 1.0005271035279195e-43.
Epoch 89/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2296 - accuracy: 0.9245 - val_loss: 0.3196
- val_accuracy: 0.8910

Epoch 00089: val_accuracy did not improve from 0.90865
Epoch 90/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2493 - accuracy: 0.9173 - val_loss: 0.3201
- val_accuracy: 0.8906

Epoch 00090: val_accuracy did not improve from 0.90865

Epoch 00090: ReduceLROnPlateau reducing learning rate to 9.949219096706202e-45.
Epoch 91/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2417 - accuracy: 0.9186 - val_loss: 0.3437
- val_accuracy: 0.8878

Epoch 00091: val_accuracy did not improve from 0.90865
Epoch 92/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2464 - accuracy: 0.9176 - val_loss: 0.3590
- val_accuracy: 0.8830

Epoch 00092: val_accuracy did not improve from 0.90865

Epoch 00092: ReduceLROnPlateau reducing learning rate to 9.80908925027372e-46.
Epoch 93/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2423 - accuracy: 0.9195 - val_loss: 0.3484
- val_accuracy: 0.8874
```

```
Epoch 00093: val_accuracy did not improve from 0.90865
Epoch 94/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2302 - accuracy: 0.9227 - val_loss: 0.3073
- val_accuracy: 0.9006

Epoch 00094: val_accuracy did not improve from 0.90865

Epoch 00094: ReduceLROnPlateau reducing learning rate to 1.4012984643248171e-46.
Epoch 95/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2333 - accuracy: 0.9227 - val_loss: 0.3456
- val_accuracy: 0.8898

Epoch 00095: val_accuracy did not improve from 0.90865
Epoch 96/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2415 - accuracy: 0.9192 - val_loss: 0.3248
- val_accuracy: 0.8910

Epoch 00096: val_accuracy did not improve from 0.90865
Epoch 97/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2423 - accuracy: 0.9185 - val_loss: 0.3296
- val_accuracy: 0.8902

Epoch 00097: val_accuracy did not improve from 0.90865
Epoch 98/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2443 - accuracy: 0.9181 - val_loss: 0.2993
- val_accuracy: 0.9050

Epoch 00098: val_accuracy did not improve from 0.90865
Epoch 99/100
390/390 [==============================] - 81s 208ms/step - loss: 0.2419 - accuracy: 0.9208 - val_loss: 0.3662
- val_accuracy: 0.8838

Epoch 00099: val_accuracy did not improve from 0.90865
Epoch 100/100
390/390 [==============================] - 81s 207ms/step - loss: 0.2377 - accuracy: 0.9220 - val_loss: 0.3673
- val_accuracy: 0.8866

Epoch 00100: val_accuracy did not improve from 0.90865
```

Out[19]: <keras.callbacks.History at 0x1aefe27e040>

In [ ]: