CNN on CIFR Assignment:

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.

Import Section

```
#!pip install tensorflow-gpu==2.8.3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from tqdm import tqdm
tqdm.pandas()
import tensorflow as tf
import math
import timeit
from six.moves import cPickle as pickle
import os
import datetime
import platform
from subprocess import check_output
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation,
Flatten,MaxPooling2D
```

```
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import
ModelCheckpoint,EarlyStopping,LearningRateScheduler,ReduceLROnPlateau,
TensorBoard
from tensorflow.keras.utils import plot_model
from keras.preprocessing.image import ImageDataGenerator
```

```
import tensorflow
tensorflow.__version__
```

{"type":"string"}

Lets pull the CIFAR 10 dataset

```
# 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]
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-
python.tar.gz
170498071/170498071 [==============================] - 2s 0us/step
```

Lets Analyse the data

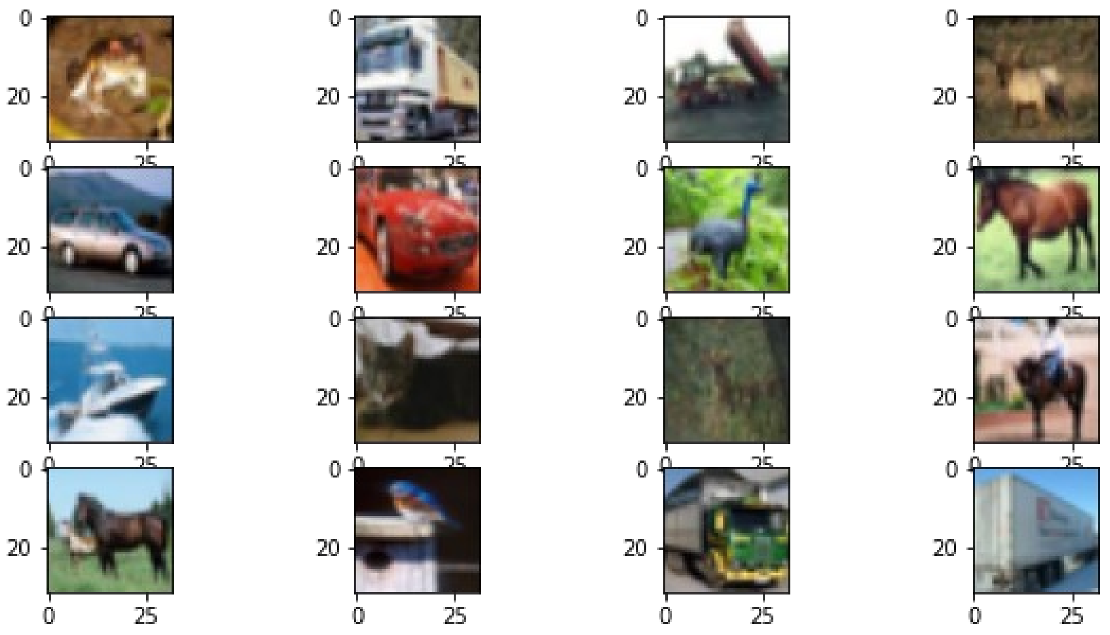
- Shape of each image is 32X32X3 that menas it is a color image

```
X_train[0].shape
```

```
(32, 32, 3)
```

Lets See few images

```
fig = plt.figure(figsize=(10, 5))
rows = 4
columns = 4
for i in range(16):
  image = X_train[i]
  fig.add_subplot(rows, columns, i+1)
  plt.imshow(image)
plt.show()
```



Lets Analyse the classes

```
np.unique(y_train)

array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], dtype=uint8)
```

- We have 10 Classes which are already label encoded

```
num_classes = 10

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)

(50000, 32, 32, 3)
(10000, 32, 32, 3)
(50000, 1)
(10000, 1)
```

Lets convert y_train to one hot encoded vector

```
# 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)
# flatten the label values
y_train, y_test = y_train.flatten(), y_test.flatten()

print(y_train.shape)
print(y_test.shape)

(50000,)
(10000,)
```

We can see that x_train values are in range of 0-255 as input is image lets normalise the image so that nueral network converge faster

```
X_train = X_train/255.0
X_test = X_test/255.0
```

using similar architechture provided in reference

```
tf.keras.backend.clear_session()

# Hyperparameters
batch_size = 64
num_classes = 10
epochs = 50
l = 40
num_filter = 12
compression = 0.5
dropout_rate = 0.2

# Dense Block
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_5_5 = layers.Conv2D(int(num_filter*compression), (5,5), 
use_bias=True ,padding='same')(relu)
        if dropout_rate>0:
            Conv2D_5_5 = layers.Dropout(dropout_rate)(Conv2D_5_5)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_5_5])

        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), 
(5,5), use_bias=False ,padding='same')(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
```

Lets train the model

```
data_generator = ImageDataGenerator(width_shift_range=0.1,
height_shift_range=0.1, horizontal_flip=True)
# prepare training iterator
train_iterator = data_generator.flow(X_train, y_train,
batch_size=batch_size)

num_filter = 12
dropout_rate = 0
l = 12
input = layers.Input(shape=(img_height, img_width, channel,))
# Image Augmentation
#random_flip = layers.RandomFlip("horizontal")(input)
#random_scaling = layers.Rescaling(scale=1./255)(random_flip)
#random_rotation = layers.RandomRotation(0.4)(random_scaling)
#random_zoom = layers.RandomZoom(0.2,0.2)(random_rotation)
#random_contrast = layers.RandomContrast(0.4)(random_zoom)
#random_brightness = layers.RandomBrightness(0.4,value_range=(0,1))
(random_contrast)
First_Conv2D = layers.Conv2D(32, (3,3),
use_bias=False ,padding='same')(input)

First_Block = denseblock(First_Conv2D, num_filter, dropout_rate)
First_Transition = transition(First_Block, 64, dropout_rate)

Second_Block = denseblock(First_Transition, num_filter, dropout_rate)
Second_Transition = transition(Second_Block, 32, dropout_rate)

Third_Block = denseblock(Second_Transition, num_filter, dropout_rate)
Third_Transition = transition(Third_Block, 32, dropout_rate)

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

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, 32)   864
['input_1[0][0]']


 batch_normalization (BatchNorm  (None, 32, 32, 32)  128
['conv2d[0][0]']
 alization)



 activation (Activation)        (None, 32, 32, 32)   0
['batch_normalization[0][0]']


 conv2d_1 (Conv2D)              (None, 32, 32, 6)    4806
['activation[0][0]']


 concatenate (Concatenate)      (None, 32, 32, 38)   0
['conv2d[0][0]',

'conv2d_1[0][0]']


 batch_normalization_1 (BatchNo  (None, 32, 32, 38)  152
['concatenate[0][0]']
 rmalization)



 activation_1 (Activation)      (None, 32, 32, 38)   0
['batch_normalization_1[0][0]']


 conv2d_2 (Conv2D)              (None, 32, 32, 6)    5706
['activation_1[0][0]']


 concatenate_1 (Concatenate)    (None, 32, 32, 44)   0
['concatenate[0][0]',

'conv2d_2[0][0]']
```

```
 batch_normalization_2 (BatchNo  (None, 32, 32, 44)  176
['concatenate_1[0][0]']
 rmalization)



 activation_2 (Activation)      (None, 32, 32, 44)   0
['batch_normalization_2[0][0]']


 conv2d_3 (Conv2D)              (None, 32, 32, 6)    6606
['activation_2[0][0]']


 concatenate_2 (Concatenate)    (None, 32, 32, 50)   0
['concatenate_1[0][0]',

'conv2d_3[0][0]']


 batch_normalization_3 (BatchNo  (None, 32, 32, 50)  200
['concatenate_2[0][0]']
 rmalization)



 activation_3 (Activation)      (None, 32, 32, 50)   0
['batch_normalization_3[0][0]']


 conv2d_4 (Conv2D)              (None, 32, 32, 6)    7506
['activation_3[0][0]']


 concatenate_3 (Concatenate)    (None, 32, 32, 56)   0
['concatenate_2[0][0]',

'conv2d_4[0][0]']


 batch_normalization_4 (BatchNo  (None, 32, 32, 56)  224
['concatenate_3[0][0]']
 rmalization)



 activation_4 (Activation)      (None, 32, 32, 56)   0
```

```
['batch_normalization_4[0][0]']


 conv2d_5 (Conv2D)              (None, 32, 32, 6)    8406
['activation_4[0][0]']


 concatenate_4 (Concatenate)    (None, 32, 32, 62)   0
['concatenate_3[0][0]',

'conv2d_5[0][0]']


 batch_normalization_5 (BatchNo  (None, 32, 32, 62)  248
['concatenate_4[0][0]']
 rmalization)



 activation_5 (Activation)      (None, 32, 32, 62)   0
['batch_normalization_5[0][0]']


 conv2d_6 (Conv2D)              (None, 32, 32, 6)    9306
['activation_5[0][0]']


 concatenate_5 (Concatenate)    (None, 32, 32, 68)   0
['concatenate_4[0][0]',

'conv2d_6[0][0]']


 batch_normalization_6 (BatchNo  (None, 32, 32, 68)  272
['concatenate_5[0][0]']
 rmalization)



 activation_6 (Activation)      (None, 32, 32, 68)   0
['batch_normalization_6[0][0]']


 conv2d_7 (Conv2D)              (None, 32, 32, 6)    10206
['activation_6[0][0]']


 concatenate_6 (Concatenate)    (None, 32, 32, 74)   0
```

```
['concatenate_5[0][0]',

'conv2d_7[0][0]']


 batch_normalization_7 (BatchNo  (None, 32, 32, 74)  296
['concatenate_6[0][0]']
 rmalization)



 activation_7 (Activation)      (None, 32, 32, 74)   0
['batch_normalization_7[0][0]']


 conv2d_8 (Conv2D)              (None, 32, 32, 6)    11106
['activation_7[0][0]']


 concatenate_7 (Concatenate)    (None, 32, 32, 80)   0
['concatenate_6[0][0]',

'conv2d_8[0][0]']


 batch_normalization_8 (BatchNo  (None, 32, 32, 80)  320
['concatenate_7[0][0]']
 rmalization)



 activation_8 (Activation)      (None, 32, 32, 80)   0
['batch_normalization_8[0][0]']


 conv2d_9 (Conv2D)              (None, 32, 32, 6)    12006
['activation_8[0][0]']


 concatenate_8 (Concatenate)    (None, 32, 32, 86)   0
['concatenate_7[0][0]',

'conv2d_9[0][0]']


 batch_normalization_9 (BatchNo  (None, 32, 32, 86)  344
['concatenate_8[0][0]']
 rmalization)
 activation_9 (Activation)      (None, 32, 32, 86)   0
['batch_normalization_9[0][0]']


 conv2d_10 (Conv2D)             (None, 32, 32, 6)    12906
['activation_9[0][0]']


 concatenate_9 (Concatenate)    (None, 32, 32, 92)   0
['concatenate_8[0][0]',

'conv2d_10[0][0]']


 batch_normalization_10 (BatchN  (None, 32, 32, 92)  368
['concatenate_9[0][0]']
 ormalization)



 activation_10 (Activation)     (None, 32, 32, 92)   0
['batch_normalization_10[0][0]']


 conv2d_11 (Conv2D)             (None, 32, 32, 6)    13806
['activation_10[0][0]']


 concatenate_10 (Concatenate)   (None, 32, 32, 98)   0
['concatenate_9[0][0]',

'conv2d_11[0][0]']


 batch_normalization_11 (BatchN  (None, 32, 32, 98)  392
['concatenate_10[0][0]']
 ormalization)



 activation_11 (Activation)     (None, 32, 32, 98)   0
['batch_normalization_11[0][0]']


 conv2d_12 (Conv2D)             (None, 32, 32, 6)    14706
```

```
['activation_11[0][0]']


 concatenate_11 (Concatenate)   (None, 32, 32, 104)  0
['concatenate_10[0][0]',

'conv2d_12[0][0]']


 batch_normalization_12 (BatchN  (None, 32, 32, 104)  416
['concatenate_11[0][0]']
 ormalization)



 activation_12 (Activation)     (None, 32, 32, 104)  0
['batch_normalization_12[0][0]']


 conv2d_13 (Conv2D)             (None, 32, 32, 32)   83200
['activation_12[0][0]']


 average_pooling2d (AveragePool  (None, 16, 16, 32)  0
['conv2d_13[0][0]']
 ing2D)



 batch_normalization_13 (BatchN  (None, 16, 16, 32)  128
['average_pooling2d[0][0]']
 ormalization)



 activation_13 (Activation)     (None, 16, 16, 32)   0
['batch_normalization_13[0][0]']


 conv2d_14 (Conv2D)             (None, 16, 16, 6)    4806
['activation_13[0][0]']


 concatenate_12 (Concatenate)   (None, 16, 16, 38)   0
['average_pooling2d[0][0]',

'conv2d_14[0][0]']
```

```
 batch_normalization_14 (BatchN  (None, 16, 16, 38)  152
['concatenate_12[0][0]']
 ormalization)



 activation_14 (Activation)     (None, 16, 16, 38)   0
['batch_normalization_14[0][0]']


 conv2d_15 (Conv2D)             (None, 16, 16, 6)    5706
['activation_14[0][0]']


 concatenate_13 (Concatenate)   (None, 16, 16, 44)   0
['concatenate_12[0][0]',

'conv2d_15[0][0]']


 batch_normalization_15 (BatchN  (None, 16, 16, 44)  176
['concatenate_13[0][0]']
 ormalization)



 activation_15 (Activation)     (None, 16, 16, 44)   0
['batch_normalization_15[0][0]']


 conv2d_16 (Conv2D)             (None, 16, 16, 6)    6606
['activation_15[0][0]']


 concatenate_14 (Concatenate)   (None, 16, 16, 50)   0
['concatenate_13[0][0]',

'conv2d_16[0][0]']


 batch_normalization_16 (BatchN  (None, 16, 16, 50)  200
['concatenate_14[0][0]']
 ormalization)



 activation_16 (Activation)     (None, 16, 16, 50)   0
```

```
['batch_normalization_16[0][0]']


 conv2d_17 (Conv2D)             (None, 16, 16, 6)    7506
['activation_16[0][0]']


 concatenate_15 (Concatenate)   (None, 16, 16, 56)   0
['concatenate_14[0][0]',

'conv2d_17[0][0]']


 batch_normalization_17 (BatchN  (None, 16, 16, 56)  224
['concatenate_15[0][0]']
 ormalization)



 activation_17 (Activation)     (None, 16, 16, 56)   0
['batch_normalization_17[0][0]']


 conv2d_18 (Conv2D)             (None, 16, 16, 6)    8406
['activation_17[0][0]']


 concatenate_16 (Concatenate)   (None, 16, 16, 62)   0
['concatenate_15[0][0]',

'conv2d_18[0][0]']


 batch_normalization_18 (BatchN  (None, 16, 16, 62)  248
['concatenate_16[0][0]']
 ormalization)



 activation_18 (Activation)     (None, 16, 16, 62)   0
['batch_normalization_18[0][0]']


 conv2d_19 (Conv2D)             (None, 16, 16, 6)    9306
['activation_18[0][0]']


 concatenate_17 (Concatenate)   (None, 16, 16, 68)   0
```

```
['concatenate_16[0][0]',

'conv2d_19[0][0]']


 batch_normalization_19 (BatchN  (None, 16, 16, 68)  272
['concatenate_17[0][0]']
 ormalization)



 activation_19 (Activation)     (None, 16, 16, 68)   0
['batch_normalization_19[0][0]']


 conv2d_20 (Conv2D)             (None, 16, 16, 6)    10206
['activation_19[0][0]']


 concatenate_18 (Concatenate)   (None, 16, 16, 74)   0
['concatenate_17[0][0]',

'conv2d_20[0][0]']


 batch_normalization_20 (BatchN  (None, 16, 16, 74)  296
['concatenate_18[0][0]']
 ormalization)



 activation_20 (Activation)     (None, 16, 16, 74)   0
['batch_normalization_20[0][0]']


 conv2d_21 (Conv2D)             (None, 16, 16, 6)    11106
['activation_20[0][0]']


 concatenate_19 (Concatenate)   (None, 16, 16, 80)   0
['concatenate_18[0][0]',

'conv2d_21[0][0]']


 batch_normalization_21 (BatchN  (None, 16, 16, 80)  320
['concatenate_19[0][0]']
 ormalization)
 activation_21 (Activation)     (None, 16, 16, 80)   0
['batch_normalization_21[0][0]']


 conv2d_22 (Conv2D)             (None, 16, 16, 6)    12006
['activation_21[0][0]']


 concatenate_20 (Concatenate)   (None, 16, 16, 86)   0
['concatenate_19[0][0]',

'conv2d_22[0][0]']


 batch_normalization_22 (BatchN  (None, 16, 16, 86)  344
['concatenate_20[0][0]']
 ormalization)



 activation_22 (Activation)     (None, 16, 16, 86)   0
['batch_normalization_22[0][0]']


 conv2d_23 (Conv2D)             (None, 16, 16, 6)    12906
['activation_22[0][0]']


 concatenate_21 (Concatenate)   (None, 16, 16, 92)   0
['concatenate_20[0][0]',

'conv2d_23[0][0]']


 batch_normalization_23 (BatchN  (None, 16, 16, 92)  368
['concatenate_21[0][0]']
 ormalization)



 activation_23 (Activation)     (None, 16, 16, 92)   0
['batch_normalization_23[0][0]']


 conv2d_24 (Conv2D)             (None, 16, 16, 6)    13806
```

```
['activation_23[0][0]']


 concatenate_22 (Concatenate)   (None, 16, 16, 98)   0
['concatenate_21[0][0]',

'conv2d_24[0][0]']


 batch_normalization_24 (BatchN  (None, 16, 16, 98)  392
['concatenate_22[0][0]']
 ormalization)



 activation_24 (Activation)     (None, 16, 16, 98)   0
['batch_normalization_24[0][0]']


 conv2d_25 (Conv2D)             (None, 16, 16, 6)    14706
['activation_24[0][0]']


 concatenate_23 (Concatenate)   (None, 16, 16, 104)  0
['concatenate_22[0][0]',

'conv2d_25[0][0]']


 batch_normalization_25 (BatchN  (None, 16, 16, 104)  416
['concatenate_23[0][0]']
 ormalization)



 activation_25 (Activation)     (None, 16, 16, 104)  0
['batch_normalization_25[0][0]']


 conv2d_26 (Conv2D)             (None, 16, 16, 16)   41600
['activation_25[0][0]']


 average_pooling2d_1 (AveragePo  (None, 8, 8, 16)    0
['conv2d_26[0][0]']
 oling2D)
 batch_normalization_26 (BatchN  (None, 8, 8, 16)    64
['average_pooling2d_1[0][0]']
 ormalization)



 activation_26 (Activation)     (None, 8, 8, 16)     0
['batch_normalization_26[0][0]']


 conv2d_27 (Conv2D)             (None, 8, 8, 6)      2406
['activation_26[0][0]']


 concatenate_24 (Concatenate)   (None, 8, 8, 22)     0
['average_pooling2d_1[0][0]',

'conv2d_27[0][0]']


 batch_normalization_27 (BatchN  (None, 8, 8, 22)    88
['concatenate_24[0][0]']
 ormalization)



 activation_27 (Activation)     (None, 8, 8, 22)     0
['batch_normalization_27[0][0]']


 conv2d_28 (Conv2D)             (None, 8, 8, 6)      3306
['activation_27[0][0]']


 concatenate_25 (Concatenate)   (None, 8, 8, 28)     0
['concatenate_24[0][0]',

'conv2d_28[0][0]']


 batch_normalization_28 (BatchN  (None, 8, 8, 28)    112
['concatenate_25[0][0]']
 ormalization)



 activation_28 (Activation)     (None, 8, 8, 28)     0
```

```
['batch_normalization_28[0][0]']


 conv2d_29 (Conv2D)             (None, 8, 8, 6)      4206
['activation_28[0][0]']


 concatenate_26 (Concatenate)   (None, 8, 8, 34)     0
['concatenate_25[0][0]',

'conv2d_29[0][0]']


 batch_normalization_29 (BatchN  (None, 8, 8, 34)    136
['concatenate_26[0][0]']
 ormalization)



 activation_29 (Activation)     (None, 8, 8, 34)     0
['batch_normalization_29[0][0]']


 conv2d_30 (Conv2D)             (None, 8, 8, 6)      5106
['activation_29[0][0]']


 concatenate_27 (Concatenate)   (None, 8, 8, 40)     0
['concatenate_26[0][0]',

'conv2d_30[0][0]']


 batch_normalization_30 (BatchN  (None, 8, 8, 40)    160
['concatenate_27[0][0]']
 ormalization)



 activation_30 (Activation)     (None, 8, 8, 40)     0
['batch_normalization_30[0][0]']


 conv2d_31 (Conv2D)             (None, 8, 8, 6)      6006
['activation_30[0][0]']


 concatenate_28 (Concatenate)   (None, 8, 8, 46)     0
```

```
['concatenate_27[0][0]',

'conv2d_31[0][0]']


 batch_normalization_31 (BatchN  (None, 8, 8, 46)    184
['concatenate_28[0][0]']
 ormalization)



 activation_31 (Activation)     (None, 8, 8, 46)     0
['batch_normalization_31[0][0]']


 conv2d_32 (Conv2D)             (None, 8, 8, 6)      6906
['activation_31[0][0]']


 concatenate_29 (Concatenate)   (None, 8, 8, 52)     0
['concatenate_28[0][0]',

'conv2d_32[0][0]']


 batch_normalization_32 (BatchN  (None, 8, 8, 52)    208
['concatenate_29[0][0]']
 ormalization)



 activation_32 (Activation)     (None, 8, 8, 52)     0
['batch_normalization_32[0][0]']


 conv2d_33 (Conv2D)             (None, 8, 8, 6)      7806
['activation_32[0][0]']


 concatenate_30 (Concatenate)   (None, 8, 8, 58)     0
['concatenate_29[0][0]',

'conv2d_33[0][0]']


 batch_normalization_33 (BatchN  (None, 8, 8, 58)    232
['concatenate_30[0][0]']
 ormalization)
 activation_33 (Activation)     (None, 8, 8, 58)     0
['batch_normalization_33[0][0]']


 conv2d_34 (Conv2D)             (None, 8, 8, 6)      8706
['activation_33[0][0]']


 concatenate_31 (Concatenate)   (None, 8, 8, 64)     0
['concatenate_30[0][0]',

'conv2d_34[0][0]']


 batch_normalization_34 (BatchN  (None, 8, 8, 64)    256
['concatenate_31[0][0]']
 ormalization)



 activation_34 (Activation)     (None, 8, 8, 64)     0
['batch_normalization_34[0][0]']


 conv2d_35 (Conv2D)             (None, 8, 8, 6)      9606
['activation_34[0][0]']


 concatenate_32 (Concatenate)   (None, 8, 8, 70)     0
['concatenate_31[0][0]',

'conv2d_35[0][0]']


 batch_normalization_35 (BatchN  (None, 8, 8, 70)    280
['concatenate_32[0][0]']
 ormalization)



 activation_35 (Activation)     (None, 8, 8, 70)     0
['batch_normalization_35[0][0]']


 conv2d_36 (Conv2D)             (None, 8, 8, 6)      10506
```

```
['activation_35[0][0]']


 concatenate_33 (Concatenate)   (None, 8, 8, 76)     0
['concatenate_32[0][0]',

'conv2d_36[0][0]']


 batch_normalization_36 (BatchN  (None, 8, 8, 76)    304
['concatenate_33[0][0]']
 ormalization)



 activation_36 (Activation)     (None, 8, 8, 76)     0
['batch_normalization_36[0][0]']


 conv2d_37 (Conv2D)             (None, 8, 8, 6)      11406
['activation_36[0][0]']


 concatenate_34 (Concatenate)   (None, 8, 8, 82)     0
['concatenate_33[0][0]',

'conv2d_37[0][0]']


 batch_normalization_37 (BatchN  (None, 8, 8, 82)    328
['concatenate_34[0][0]']
 ormalization)



 activation_37 (Activation)     (None, 8, 8, 82)     0
['batch_normalization_37[0][0]']


 conv2d_38 (Conv2D)             (None, 8, 8, 6)      12306
['activation_37[0][0]']


 concatenate_35 (Concatenate)   (None, 8, 8, 88)     0
['concatenate_34[0][0]',

'conv2d_38[0][0]']
```

```
 batch_normalization_38 (BatchN  (None, 8, 8, 88)    352
['concatenate_35[0][0]']
 ormalization)



 activation_38 (Activation)     (None, 8, 8, 88)     0
['batch_normalization_38[0][0]']


 conv2d_39 (Conv2D)             (None, 8, 8, 16)     35200
['activation_38[0][0]']


 average_pooling2d_2 (AveragePo  (None, 4, 4, 16)    0
['conv2d_39[0][0]']
 oling2D)



 batch_normalization_39 (BatchN  (None, 4, 4, 16)    64
['average_pooling2d_2[0][0]']
 ormalization)



 activation_39 (Activation)     (None, 4, 4, 16)     0
['batch_normalization_39[0][0]']


 conv2d_40 (Conv2D)             (None, 4, 4, 6)      2406
['activation_39[0][0]']


 concatenate_36 (Concatenate)   (None, 4, 4, 22)     0
['average_pooling2d_2[0][0]',

'conv2d_40[0][0]']


 batch_normalization_40 (BatchN  (None, 4, 4, 22)    88
['concatenate_36[0][0]']
 ormalization)



 activation_40 (Activation)     (None, 4, 4, 22)     0
```

```
['batch_normalization_40[0][0]']


 conv2d_41 (Conv2D)             (None, 4, 4, 6)      3306
['activation_40[0][0]']


 concatenate_37 (Concatenate)   (None, 4, 4, 28)     0
['concatenate_36[0][0]',

'conv2d_41[0][0]']


 batch_normalization_41 (BatchN  (None, 4, 4, 28)    112
['concatenate_37[0][0]']
 ormalization)



 activation_41 (Activation)     (None, 4, 4, 28)     0
['batch_normalization_41[0][0]']


 conv2d_42 (Conv2D)             (None, 4, 4, 6)      4206
['activation_41[0][0]']


 concatenate_38 (Concatenate)   (None, 4, 4, 34)     0
['concatenate_37[0][0]',

'conv2d_42[0][0]']


 batch_normalization_42 (BatchN  (None, 4, 4, 34)    136
['concatenate_38[0][0]']
 ormalization)



 activation_42 (Activation)     (None, 4, 4, 34)     0
['batch_normalization_42[0][0]']


 conv2d_43 (Conv2D)             (None, 4, 4, 6)      5106
['activation_42[0][0]']


 concatenate_39 (Concatenate)   (None, 4, 4, 40)     0
```

```
['concatenate_38[0][0]',

'conv2d_43[0][0]']


 batch_normalization_43 (BatchN  (None, 4, 4, 40)    160
['concatenate_39[0][0]']
 ormalization)



 activation_43 (Activation)     (None, 4, 4, 40)     0
['batch_normalization_43[0][0]']


 conv2d_44 (Conv2D)             (None, 4, 4, 6)      6006
['activation_43[0][0]']


 concatenate_40 (Concatenate)   (None, 4, 4, 46)     0
['concatenate_39[0][0]',

'conv2d_44[0][0]']


 batch_normalization_44 (BatchN  (None, 4, 4, 46)    184
['concatenate_40[0][0]']
 ormalization)



 activation_44 (Activation)     (None, 4, 4, 46)     0
['batch_normalization_44[0][0]']


 conv2d_45 (Conv2D)             (None, 4, 4, 6)      6906
['activation_44[0][0]']


 concatenate_41 (Concatenate)   (None, 4, 4, 52)     0
['concatenate_40[0][0]',

'conv2d_45[0][0]']


 batch_normalization_45 (BatchN  (None, 4, 4, 52)    208
['concatenate_41[0][0]']
 ormalization)
 activation_45 (Activation)     (None, 4, 4, 52)     0
['batch_normalization_45[0][0]']


 conv2d_46 (Conv2D)             (None, 4, 4, 6)      7806
['activation_45[0][0]']


 concatenate_42 (Concatenate)   (None, 4, 4, 58)     0
['concatenate_41[0][0]',

'conv2d_46[0][0]']


 batch_normalization_46 (BatchN  (None, 4, 4, 58)    232
['concatenate_42[0][0]']
 ormalization)



 activation_46 (Activation)     (None, 4, 4, 58)     0
['batch_normalization_46[0][0]']


 conv2d_47 (Conv2D)             (None, 4, 4, 6)      8706
['activation_46[0][0]']


 concatenate_43 (Concatenate)   (None, 4, 4, 64)     0
['concatenate_42[0][0]',

'conv2d_47[0][0]']


 batch_normalization_47 (BatchN  (None, 4, 4, 64)    256
['concatenate_43[0][0]']
 ormalization)



 activation_47 (Activation)     (None, 4, 4, 64)     0
['batch_normalization_47[0][0]']


 conv2d_48 (Conv2D)             (None, 4, 4, 6)      9606
```

```
['activation_47[0][0]']


 concatenate_44 (Concatenate)   (None, 4, 4, 70)     0
['concatenate_43[0][0]',

'conv2d_48[0][0]']


 batch_normalization_48 (BatchN  (None, 4, 4, 70)    280
['concatenate_44[0][0]']
 ormalization)



 activation_48 (Activation)     (None, 4, 4, 70)     0
['batch_normalization_48[0][0]']


 conv2d_49 (Conv2D)             (None, 4, 4, 6)      10506
['activation_48[0][0]']


 concatenate_45 (Concatenate)   (None, 4, 4, 76)     0
['concatenate_44[0][0]',

'conv2d_49[0][0]']


 batch_normalization_49 (BatchN  (None, 4, 4, 76)    304
['concatenate_45[0][0]']
 ormalization)



 activation_49 (Activation)     (None, 4, 4, 76)     0
['batch_normalization_49[0][0]']


 conv2d_50 (Conv2D)             (None, 4, 4, 6)      11406
['activation_49[0][0]']


 concatenate_46 (Concatenate)   (None, 4, 4, 82)     0
['concatenate_45[0][0]',

'conv2d_50[0][0]']
```

```
 batch_normalization_50 (BatchN  (None, 4, 4, 82)    328
['concatenate_46[0][0]']
 ormalization)



 activation_50 (Activation)     (None, 4, 4, 82)     0
['batch_normalization_50[0][0]']


 conv2d_51 (Conv2D)             (None, 4, 4, 6)      12306
['activation_50[0][0]']


 concatenate_47 (Concatenate)   (None, 4, 4, 88)     0
['concatenate_46[0][0]',

'conv2d_51[0][0]']


 batch_normalization_51 (BatchN  (None, 4, 4, 88)    352
['concatenate_47[0][0]']
 ormalization)



 activation_51 (Activation)     (None, 4, 4, 88)     0
['batch_normalization_51[0][0]']


 average_pooling2d_3 (AveragePo  (None, 2, 2, 88)    0
['activation_51[0][0]']
 oling2D)



 flatten (Flatten)              (None, 352)          0
['average_pooling2d_3[0][0]']


 dense (Dense)                  (None, 10)           3530
['flatten[0][0]']


======================================================================
============================
Total params: 587,562
```

```
Trainable params: 581,322
Non-trainable params: 6,240
______________________________________________________________________
____________________________
```

```
print(len(model.layers))
```

```
211
```

```
filepath="model_save/weights-{epoch:02d}-{val_accuracy:.4f}.hdf5"
checkpoint = ModelCheckpoint(filepath=filepath,
monitor='val_accuracy',  verbose=1, save_best_only=True, mode='auto')
```

```
# Load the TensorBoard notebook extension
%load_ext tensorboard
log_dir = os.path.join("logs",'fits',
datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
tensorboard_callback =
tf.keras.callbacks.TensorBoard(log_dir=log_dir,histogram_freq=1,write_
graph=True)
%reload_ext tensorboard
```

```
# determine Loss function and Optimizer
model.compile(loss='sparse_categorical_crossentropy',
optimizer=Adam(),metrics=['accuracy'])
```

```
reduce_lr = ReduceLROnPlateau(monitor='val_accuracy',
factor=0.1,patience=5, min_lr=0.000001)
```

```
model.fit(train_iterator,batch_size=batch_size,epochs=5,verbose=1,vali
dation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/5
782/782 [==============================] - ETA: 0s - loss: 0.2645 -
accuracy: 0.9068
Epoch 1: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 108ms/step - loss:
0.2645 - accuracy: 0.9068 - val_loss: 0.4605 - val_accuracy: 0.8552 -
lr: 0.0010
Epoch 2/5
782/782 [==============================] - ETA: 0s - loss: 0.2440 -
accuracy: 0.9146
Epoch 2: val_accuracy did not improve from 0.87340
782/782 [==============================] - 85s 108ms/step - loss:
0.2440 - accuracy: 0.9146 - val_loss: 0.6535 - val_accuracy: 0.8000 -
lr: 0.0010
Epoch 3/5
782/782 [==============================] - ETA: 0s - loss: 0.2338 -
accuracy: 0.9174
Epoch 3: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
```

```
0.2338 - accuracy: 0.9174 - val_loss: 0.4759 - val_accuracy: 0.8495 -
lr: 0.0010
Epoch 4/5
782/782 [==============================] - ETA: 0s - loss: 0.2297 -
accuracy: 0.9197
Epoch 4: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
0.2297 - accuracy: 0.9197 - val_loss: 0.6275 - val_accuracy: 0.8142 -
lr: 0.0010
Epoch 5/5
782/782 [==============================] - ETA: 0s - loss: 0.2190 -
accuracy: 0.9226
Epoch 5: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
0.2190 - accuracy: 0.9226 - val_loss: 0.4594 - val_accuracy: 0.8588 -
lr: 0.0010

<keras.callbacks.History at 0x7fefd046bd00>
```

```
model.fit(train_iterator,batch_size=batch_size,epochs=15,verbose=1,val
idation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/15
782/782 [==============================] - ETA: 0s - loss: 0.2113 -
accuracy: 0.9255
Epoch 1: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
0.2113 - accuracy: 0.9255 - val_loss: 0.5140 - val_accuracy: 0.8424 -
lr: 0.0010
Epoch 2/15
782/782 [==============================] - ETA: 0s - loss: 0.2101 -
accuracy: 0.9261
Epoch 2: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 108ms/step - loss:
0.2101 - accuracy: 0.9261 - val_loss: 0.5460 - val_accuracy: 0.8392 -
lr: 0.0010
Epoch 3/15
782/782 [==============================] - ETA: 0s - loss: 0.2022 -
accuracy: 0.9288
Epoch 3: val_accuracy did not improve from 0.87340
782/782 [==============================] - 85s 109ms/step - loss:
0.2022 - accuracy: 0.9288 - val_loss: 0.4144 - val_accuracy: 0.8724 -
lr: 0.0010
Epoch 4/15
782/782 [==============================] - ETA: 0s - loss: 0.1981 -
accuracy: 0.9305
Epoch 4: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 108ms/step - loss:
0.1981 - accuracy: 0.9305 - val_loss: 0.5668 - val_accuracy: 0.8364 -
lr: 0.0010
```

```
Epoch 5/15
782/782 [==============================] - ETA: 0s - loss: 0.1964 -
accuracy: 0.9294
Epoch 5: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
0.1964 - accuracy: 0.9294 - val_loss: 0.4441 - val_accuracy: 0.8627 -
lr: 0.0010
Epoch 6/15
782/782 [==============================] - ETA: 0s - loss: 0.1842 -
accuracy: 0.9354
Epoch 6: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 107ms/step - loss:
0.1842 - accuracy: 0.9354 - val_loss: 0.5729 - val_accuracy: 0.8386 -
lr: 0.0010
Epoch 7/15
782/782 [==============================] - ETA: 0s - loss: 0.1840 -
accuracy: 0.9349
Epoch 7: val_accuracy did not improve from 0.87340
782/782 [==============================] - 84s 108ms/step - loss:
0.1840 - accuracy: 0.9349 - val_loss: 0.4727 - val_accuracy: 0.8580 -
lr: 0.0010
Epoch 8/15
782/782 [==============================] - ETA: 0s - loss: 0.1779 -
accuracy: 0.9361
Epoch 8: val_accuracy did not improve from 0.87340
782/782 [==============================] - 83s 106ms/step - loss:
0.1779 - accuracy: 0.9361 - val_loss: 0.5875 - val_accuracy: 0.8392 -
lr: 0.0010
Epoch 9/15
782/782 [==============================] - ETA: 0s - loss: 0.1319 -
accuracy: 0.9549
Epoch 9: val_accuracy improved from 0.87340 to 0.89210, saving model
to model_save/weights-09-0.8921.hdf5
782/782 [==============================] - 84s 107ms/step - loss:
0.1319 - accuracy: 0.9549 - val_loss: 0.3449 - val_accuracy: 0.8921 -
lr: 1.0000e-04
Epoch 10/15
782/782 [==============================] - ETA: 0s - loss: 0.1096 -
accuracy: 0.9628
Epoch 10: val_accuracy improved from 0.89210 to 0.89370, saving model
to model_save/weights-10-0.8937.hdf5
782/782 [==============================] - 85s 109ms/step - loss:
0.1096 - accuracy: 0.9628 - val_loss: 0.3468 - val_accuracy: 0.8937 -
lr: 1.0000e-04
Epoch 11/15
782/782 [==============================] - ETA: 0s - loss: 0.1026 -
accuracy: 0.9641
Epoch 11: val_accuracy improved from 0.89370 to 0.89480, saving model
to model_save/weights-11-0.8948.hdf5
782/782 [==============================] - 85s 108ms/step - loss:
```

```
0.1026 - accuracy: 0.9641 - val_loss: 0.3530 - val_accuracy: 0.8948 -
lr: 1.0000e-04
Epoch 12/15
782/782 [==============================] - ETA: 0s - loss: 0.0959 -
accuracy: 0.9669
Epoch 12: val_accuracy improved from 0.89480 to 0.89490, saving model
to model_save/weights-12-0.8949.hdf5
782/782 [==============================] - 84s 107ms/step - loss:
0.0959 - accuracy: 0.9669 - val_loss: 0.3550 - val_accuracy: 0.8949 -
lr: 1.0000e-04
Epoch 13/15
782/782 [==============================] - ETA: 0s - loss: 0.0928 -
accuracy: 0.9680
Epoch 13: val_accuracy improved from 0.89490 to 0.89510, saving model
to model_save/weights-13-0.8951.hdf5
782/782 [==============================] - 85s 108ms/step - loss:
0.0928 - accuracy: 0.9680 - val_loss: 0.3664 - val_accuracy: 0.8951 -
lr: 1.0000e-04
Epoch 14/15
782/782 [==============================] - ETA: 0s - loss: 0.0888 -
accuracy: 0.9696
Epoch 14: val_accuracy did not improve from 0.89510
782/782 [==============================] - 83s 106ms/step - loss:
0.0888 - accuracy: 0.9696 - val_loss: 0.3708 - val_accuracy: 0.8927 -
lr: 1.0000e-04
Epoch 15/15
782/782 [==============================] - ETA: 0s - loss: 0.0891 -
accuracy: 0.9691
Epoch 15: val_accuracy improved from 0.89510 to 0.89610, saving model
to model_save/weights-15-0.8961.hdf5
782/782 [==============================] - 86s 109ms/step - loss:
0.0891 - accuracy: 0.9691 - val_loss: 0.3680 - val_accuracy: 0.8961 -
lr: 1.0000e-04
```

```
<keras.callbacks.History at 0x7fef5de97970>
```

```
model.fit(train_iterator,batch_size=batch_size,epochs=15,verbose=1,val
idation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/15
782/782 [==============================] - ETA: 0s - loss: 0.0844 -
accuracy: 0.9709
Epoch 1: val_accuracy did not improve from 0.89610
782/782 [==============================] - 85s 108ms/step - loss:
0.0844 - accuracy: 0.9709 - val_loss: 0.3761 - val_accuracy: 0.8957 -
lr: 1.0000e-04
Epoch 2/15
782/782 [==============================] - ETA: 0s - loss: 0.0838 -
accuracy: 0.9709
Epoch 2: val_accuracy did not improve from 0.89610
```

```
782/782 [==============================] - 83s 106ms/step - loss:
0.0838 - accuracy: 0.9709 - val_loss: 0.3831 - val_accuracy: 0.8939 -
lr: 1.0000e-04
Epoch 3/15
782/782 [==============================] - ETA: 0s - loss: 0.0803 -
accuracy: 0.9721
Epoch 3: val_accuracy did not improve from 0.89610
782/782 [==============================] - 84s 107ms/step - loss:
0.0803 - accuracy: 0.9721 - val_loss: 0.3754 - val_accuracy: 0.8952 -
lr: 1.0000e-04
Epoch 4/15
782/782 [==============================] - ETA: 0s - loss: 0.0796 -
accuracy: 0.9720
Epoch 4: val_accuracy did not improve from 0.89610
782/782 [==============================] - 83s 106ms/step - loss:
0.0796 - accuracy: 0.9720 - val_loss: 0.3859 - val_accuracy: 0.8921 -
lr: 1.0000e-04
Epoch 5/15
782/782 [==============================] - ETA: 0s - loss: 0.0789 -
accuracy: 0.9721
Epoch 5: val_accuracy improved from 0.89610 to 0.89650, saving model
to model_save/weights-05-0.8965.hdf5
782/782 [==============================] - 85s 109ms/step - loss:
0.0789 - accuracy: 0.9721 - val_loss: 0.3807 - val_accuracy: 0.8965 -
lr: 1.0000e-04
Epoch 6/15
782/782 [==============================] - ETA: 0s - loss: 0.0763 -
accuracy: 0.9732
Epoch 6: val_accuracy did not improve from 0.89650
782/782 [==============================] - 84s 107ms/step - loss:
0.0763 - accuracy: 0.9732 - val_loss: 0.3893 - val_accuracy: 0.8940 -
lr: 1.0000e-04
Epoch 7/15
782/782 [==============================] - ETA: 0s - loss: 0.0725 -
accuracy: 0.9747
Epoch 7: val_accuracy did not improve from 0.89650
782/782 [==============================] - 83s 106ms/step - loss:
0.0725 - accuracy: 0.9747 - val_loss: 0.4120 - val_accuracy: 0.8926 -
lr: 1.0000e-04
Epoch 8/15
782/782 [==============================] - ETA: 0s - loss: 0.0715 -
accuracy: 0.9754
Epoch 8: val_accuracy did not improve from 0.89650
782/782 [==============================] - 84s 107ms/step - loss:
0.0715 - accuracy: 0.9754 - val_loss: 0.3988 - val_accuracy: 0.8942 -
lr: 1.0000e-04
Epoch 9/15
782/782 [==============================] - ETA: 0s - loss: 0.0687 -
accuracy: 0.9755
Epoch 9: val_accuracy did not improve from 0.89650
```

```
782/782 [==============================] - 83s 107ms/step - loss:
0.0687 - accuracy: 0.9755 - val_loss: 0.4002 - val_accuracy: 0.8959 -
lr: 1.0000e-04
Epoch 10/15
782/782 [==============================] - ETA: 0s - loss: 0.0691 -
accuracy: 0.9759
Epoch 10: val_accuracy did not improve from 0.89650
782/782 [==============================] - 83s 107ms/step - loss:
0.0691 - accuracy: 0.9759 - val_loss: 0.4074 - val_accuracy: 0.8932 -
lr: 1.0000e-04
Epoch 11/15
782/782 [==============================] - ETA: 0s - loss: 0.0660 -
accuracy: 0.9766
Epoch 11: val_accuracy did not improve from 0.89650
782/782 [==============================] - 86s 111ms/step - loss:
0.0660 - accuracy: 0.9766 - val_loss: 0.3998 - val_accuracy: 0.8952 -
lr: 1.0000e-05
Epoch 12/15
782/782 [==============================] - ETA: 0s - loss: 0.0638 -
accuracy: 0.9774
Epoch 12: val_accuracy did not improve from 0.89650
782/782 [==============================] - 85s 108ms/step - loss:
0.0638 - accuracy: 0.9774 - val_loss: 0.3996 - val_accuracy: 0.8962 -
lr: 1.0000e-05
Epoch 13/15
782/782 [==============================] - ETA: 0s - loss: 0.0631 -
accuracy: 0.9787
Epoch 13: val_accuracy did not improve from 0.89650
782/782 [==============================] - 83s 106ms/step - loss:
0.0631 - accuracy: 0.9787 - val_loss: 0.3979 - val_accuracy: 0.8964 -
lr: 1.0000e-05
Epoch 14/15
782/782 [==============================] - ETA: 0s - loss: 0.0637 -
accuracy: 0.9779
Epoch 14: val_accuracy improved from 0.89650 to 0.89670, saving model
to model_save/weights-14-0.8967.hdf5
782/782 [==============================] - 84s 108ms/step - loss:
0.0637 - accuracy: 0.9779 - val_loss: 0.3978 - val_accuracy: 0.8967 -
lr: 1.0000e-05
Epoch 15/15
782/782 [==============================] - ETA: 0s - loss: 0.0627 -
accuracy: 0.9784
Epoch 15: val_accuracy improved from 0.89670 to 0.89750, saving model
to model_save/weights-15-0.8975.hdf5
782/782 [==============================] - 85s 108ms/step - loss:
0.0627 - accuracy: 0.9784 - val_loss: 0.3961 - val_accuracy: 0.8975 -
lr: 1.0000e-05

<keras.callbacks.History at 0x7fef5deb02b0>
```

```
data_generator = ImageDataGenerator(width_shift_range=0.1,
height_shift_range=0.1, rotation_range =
0.2,shear_range=0.1,zoom_range=0.1,horizontal_flip=True)
# prepare training iterator
train_iterator = data_generator.flow(X_train, y_train,
batch_size=batch_size)
```

```
model.fit(train_iterator,batch_size=batch_size,epochs=10,verbose=1,val
idation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/10
782/782 [==============================] - ETA: 0s - loss: 0.0865 -
accuracy: 0.9695
Epoch 1: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
0.0865 - accuracy: 0.9695 - val_loss: 0.4057 - val_accuracy: 0.8952 -
lr: 1.0000e-05
Epoch 2/10
782/782 [==============================] - ETA: 0s - loss: 0.0836 -
accuracy: 0.9704
Epoch 2: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0836 - accuracy: 0.9704 - val_loss: 0.4039 - val_accuracy: 0.8956 -
lr: 1.0000e-05
Epoch 3/10
782/782 [==============================] - ETA: 0s - loss: 0.0841 -
accuracy: 0.9695
Epoch 3: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0841 - accuracy: 0.9695 - val_loss: 0.4051 - val_accuracy: 0.8946 -
lr: 1.0000e-05
Epoch 4/10
782/782 [==============================] - ETA: 0s - loss: 0.0813 -
accuracy: 0.9710
Epoch 4: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0813 - accuracy: 0.9710 - val_loss: 0.4051 - val_accuracy: 0.8951 -
lr: 1.0000e-05
Epoch 5/10
782/782 [==============================] - ETA: 0s - loss: 0.0835 -
accuracy: 0.9701
Epoch 5: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0835 - accuracy: 0.9701 - val_loss: 0.4027 - val_accuracy: 0.8946 -
lr: 1.0000e-05
Epoch 6/10
782/782 [==============================] - ETA: 0s - loss: 0.0818 -
accuracy: 0.9714
Epoch 6: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
```

```
0.0818 - accuracy: 0.9714 - val_loss: 0.4061 - val_accuracy: 0.8945 -
lr: 1.0000e-05
Epoch 7/10
782/782 [==============================] - ETA: 0s - loss: 0.0792 -
accuracy: 0.9724
Epoch 7: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0792 - accuracy: 0.9724 - val_loss: 0.4025 - val_accuracy: 0.8954 -
lr: 1.0000e-05
Epoch 8/10
782/782 [==============================] - ETA: 0s - loss: 0.0845 -
accuracy: 0.9704
Epoch 8: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0845 - accuracy: 0.9704 - val_loss: 0.4034 - val_accuracy: 0.8950 -
lr: 1.0000e-05
Epoch 9/10
782/782 [==============================] - ETA: 0s - loss: 0.0814 -
accuracy: 0.9713
Epoch 9: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
0.0814 - accuracy: 0.9713 - val_loss: 0.4027 - val_accuracy: 0.8949 -
lr: 1.0000e-05
Epoch 10/10
782/782 [==============================] - ETA: 0s - loss: 0.0805 -
accuracy: 0.9721
Epoch 10: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0805 - accuracy: 0.9721 - val_loss: 0.4020 - val_accuracy: 0.8960 -
lr: 1.0000e-05

<keras.callbacks.History at 0x7fefd0458a60>
```

```
model.fit(train_iterator,batch_size=batch_size,epochs=15,verbose=1,val
idation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/15
782/782 [==============================] - ETA: 0s - loss: 0.0834 -
accuracy: 0.9709
Epoch 1: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
0.0834 - accuracy: 0.9709 - val_loss: 0.4030 - val_accuracy: 0.8953 -
lr: 1.0000e-05
Epoch 2/15
782/782 [==============================] - ETA: 0s - loss: 0.0796 -
accuracy: 0.9713
Epoch 2: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0796 - accuracy: 0.9713 - val_loss: 0.4020 - val_accuracy: 0.8952 -
lr: 1.0000e-05
```

```
Epoch 3/15
782/782 [==============================] - ETA: 0s - loss: 0.0795 -
accuracy: 0.9718
Epoch 3: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0795 - accuracy: 0.9718 - val_loss: 0.4025 - val_accuracy: 0.8954 -
lr: 1.0000e-05
Epoch 4/15
782/782 [==============================] - ETA: 0s - loss: 0.0795 -
accuracy: 0.9721
Epoch 4: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0795 - accuracy: 0.9721 - val_loss: 0.4060 - val_accuracy: 0.8948 -
lr: 1.0000e-05
Epoch 5/15
782/782 [==============================] - ETA: 0s - loss: 0.0774 -
accuracy: 0.9731
Epoch 5: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0774 - accuracy: 0.9731 - val_loss: 0.4033 - val_accuracy: 0.8955 -
lr: 1.0000e-05
Epoch 6/15
782/782 [==============================] - ETA: 0s - loss: 0.0784 -
accuracy: 0.9719
Epoch 6: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0784 - accuracy: 0.9719 - val_loss: 0.4007 - val_accuracy: 0.8957 -
lr: 1.0000e-05
Epoch 7/15
782/782 [==============================] - ETA: 0s - loss: 0.0804 -
accuracy: 0.9718
Epoch 7: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0804 - accuracy: 0.9718 - val_loss: 0.4027 - val_accuracy: 0.8963 -
lr: 1.0000e-05
Epoch 8/15
782/782 [==============================] - ETA: 0s - loss: 0.0807 -
accuracy: 0.9714
Epoch 8: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0807 - accuracy: 0.9714 - val_loss: 0.4037 - val_accuracy: 0.8954 -
lr: 1.0000e-05
Epoch 9/15
782/782 [==============================] - ETA: 0s - loss: 0.0779 -
accuracy: 0.9726
Epoch 9: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0779 - accuracy: 0.9726 - val_loss: 0.4006 - val_accuracy: 0.8968 -
lr: 1.0000e-05
Epoch 10/15
```

```
782/782 [==============================] - ETA: 0s - loss: 0.0787 -
accuracy: 0.9719
Epoch 10: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0787 - accuracy: 0.9719 - val_loss: 0.4000 - val_accuracy: 0.8967 -
lr: 1.0000e-05
Epoch 11/15
782/782 [==============================] - ETA: 0s - loss: 0.0780 -
accuracy: 0.9720
Epoch 11: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0780 - accuracy: 0.9720 - val_loss: 0.4011 - val_accuracy: 0.8968 -
lr: 1.0000e-05
Epoch 12/15
782/782 [==============================] - ETA: 0s - loss: 0.0760 -
accuracy: 0.9730
Epoch 12: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0760 - accuracy: 0.9730 - val_loss: 0.4006 - val_accuracy: 0.8968 -
lr: 1.0000e-05
Epoch 13/15
782/782 [==============================] - ETA: 0s - loss: 0.0765 -
accuracy: 0.9732
Epoch 13: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0765 - accuracy: 0.9732 - val_loss: 0.3999 - val_accuracy: 0.8974 -
lr: 1.0000e-05
Epoch 14/15
782/782 [==============================] - ETA: 0s - loss: 0.0771 -
accuracy: 0.9719
Epoch 14: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0771 - accuracy: 0.9719 - val_loss: 0.4000 - val_accuracy: 0.8973 -
lr: 1.0000e-05
Epoch 15/15
782/782 [==============================] - ETA: 0s - loss: 0.0755 -
accuracy: 0.9733
Epoch 15: val_accuracy did not improve from 0.89750
782/782 [==============================] - 90s 115ms/step - loss:
0.0755 - accuracy: 0.9733 - val_loss: 0.4001 - val_accuracy: 0.8965 -
lr: 1.0000e-05

<keras.callbacks.History at 0x7fef5deef430>
```

```
data_generator = ImageDataGenerator(width_shift_range=0.2,
height_shift_range=0.2, rotation_range =
0.6,shear_range=0.1,zoom_range=0.4,horizontal_flip=True)
# prepare training iterator
train_iterator = data_generator.flow(X_train, y_train,
batch_size=batch_size)
```

```
# determine Loss function and Optimizer
model.compile(loss='sparse_categorical_crossentropy',
optimizer=Adam(learning_rate = 0.0001),metrics=['accuracy'])

reduce_lr = ReduceLROnPlateau(monitor='val_accuracy',
factor=0.1,patience=3, min_lr=0.000001)

model.fit(train_iterator,batch_size=batch_size,epochs=15,verbose=1,val
idation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])

Epoch 1/15
782/782 [==============================] - ETA: 0s - loss: 0.0902 -
accuracy: 0.9688
Epoch 1: val_accuracy did not improve from 0.89750
782/782 [==============================] - 123s 111ms/step - loss:
0.0902 - accuracy: 0.9688 - val_loss: 0.4174 - val_accuracy: 0.8957 -
lr: 1.0000e-04
Epoch 2/15
782/782 [==============================] - ETA: 0s - loss: 0.0879 -
accuracy: 0.9690
Epoch 2: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
0.0879 - accuracy: 0.9690 - val_loss: 0.4029 - val_accuracy: 0.8948 -
lr: 1.0000e-04
Epoch 3/15
782/782 [==============================] - ETA: 0s - loss: 0.0860 -
accuracy: 0.9688
Epoch 3: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0860 - accuracy: 0.9688 - val_loss: 0.4063 - val_accuracy: 0.8942 -
lr: 1.0000e-04
Epoch 4/15
782/782 [==============================] - ETA: 0s - loss: 0.0843 -
accuracy: 0.9701
Epoch 4: val_accuracy did not improve from 0.89750
782/782 [==============================] - 86s 110ms/step - loss:
0.0843 - accuracy: 0.9701 - val_loss: 0.4213 - val_accuracy: 0.8921 -
lr: 1.0000e-04
Epoch 5/15
782/782 [==============================] - ETA: 0s - loss: 0.0809 -
accuracy: 0.9711
Epoch 5: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
0.0809 - accuracy: 0.9711 - val_loss: 0.4009 - val_accuracy: 0.8966 -
lr: 1.0000e-05
Epoch 6/15
782/782 [==============================] - ETA: 0s - loss: 0.0785 -
accuracy: 0.9732
Epoch 6: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
```

```
0.0785 - accuracy: 0.9732 - val_loss: 0.4021 - val_accuracy: 0.8962 -
lr: 1.0000e-05
Epoch 7/15
782/782 [==============================] - ETA: 0s - loss: 0.0773 -
accuracy: 0.9728
Epoch 7: val_accuracy did not improve from 0.89750
782/782 [==============================] - 86s 110ms/step - loss:
0.0773 - accuracy: 0.9728 - val_loss: 0.3999 - val_accuracy: 0.8965 -
lr: 1.0000e-05
Epoch 8/15
782/782 [==============================] - ETA: 0s - loss: 0.0794 -
accuracy: 0.9717
Epoch 8: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0794 - accuracy: 0.9717 - val_loss: 0.3976 - val_accuracy: 0.8969 -
lr: 1.0000e-05
Epoch 9/15
782/782 [==============================] - ETA: 0s - loss: 0.0763 -
accuracy: 0.9730
Epoch 9: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 108ms/step - loss:
0.0763 - accuracy: 0.9730 - val_loss: 0.4007 - val_accuracy: 0.8963 -
lr: 1.0000e-05
Epoch 10/15
782/782 [==============================] - ETA: 0s - loss: 0.0761 -
accuracy: 0.9730
Epoch 10: val_accuracy did not improve from 0.89750
782/782 [==============================] - 86s 109ms/step - loss:
0.0761 - accuracy: 0.9730 - val_loss: 0.3970 - val_accuracy: 0.8971 -
lr: 1.0000e-05
Epoch 11/15
782/782 [==============================] - ETA: 0s - loss: 0.0781 -
accuracy: 0.9727
Epoch 11: val_accuracy did not improve from 0.89750
782/782 [==============================] - 87s 111ms/step - loss:
0.0781 - accuracy: 0.9727 - val_loss: 0.4001 - val_accuracy: 0.8968 -
lr: 1.0000e-05
Epoch 12/15
782/782 [==============================] - ETA: 0s - loss: 0.0775 -
accuracy: 0.9731
Epoch 12: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0775 - accuracy: 0.9731 - val_loss: 0.3988 - val_accuracy: 0.8968 -
lr: 1.0000e-05
Epoch 13/15
782/782 [==============================] - ETA: 0s - loss: 0.0767 -
accuracy: 0.9726
Epoch 13: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 108ms/step - loss:
0.0767 - accuracy: 0.9726 - val_loss: 0.3995 - val_accuracy: 0.8971 -
```

```
lr: 1.0000e-05
Epoch 14/15
782/782 [==============================] - ETA: 0s - loss: 0.0742 -
accuracy: 0.9731
Epoch 14: val_accuracy did not improve from 0.89750
782/782 [==============================] - 84s 107ms/step - loss:
0.0742 - accuracy: 0.9731 - val_loss: 0.4019 - val_accuracy: 0.8964 -
lr: 1.0000e-06
Epoch 15/15
782/782 [==============================] - ETA: 0s - loss: 0.0752 -
accuracy: 0.9737
Epoch 15: val_accuracy did not improve from 0.89750
782/782 [==============================] - 86s 110ms/step - loss:
0.0752 - accuracy: 0.9737 - val_loss: 0.3990 - val_accuracy: 0.8966 -
lr: 1.0000e-06

<keras.callbacks.History at 0x7fef76051190>
```

```
data_generator = ImageDataGenerator(width_shift_range=0.3,
height_shift_range=0.3, rotation_range =
0.6,shear_range=0.3,zoom_range=0.4,horizontal_flip=True)
# prepare training iterator
train_iterator = data_generator.flow(X_train, y_train,
batch_size=batch_size)

reduce_lr = ReduceLROnPlateau(monitor='val_accuracy',
factor=0.1,patience=4, min_lr=0.000000001)

model.fit(train_iterator,batch_size=batch_size,epochs=3,verbose=1,vali
dation_data=(X_test, y_test),callbacks =
[checkpoint,reduce_lr,tensorboard_callback])
```

```
Epoch 1/3
782/782 [==============================] - ETA: 0s - loss: 0.7970 -
accuracy: 0.7960
Epoch 1: val_accuracy did not improve from 0.89750
782/782 [==============================] - 86s 110ms/step - loss:
0.7970 - accuracy: 0.7960 - val_loss: 0.5673 - val_accuracy: 0.8686 -
lr: 1.0000e-09
Epoch 2/3
782/782 [==============================] - ETA: 0s - loss: 0.7903 -
accuracy: 0.7992
Epoch 2: val_accuracy did not improve from 0.89750
782/782 [==============================] - 89s 114ms/step - loss:
0.7903 - accuracy: 0.7992 - val_loss: 0.5685 - val_accuracy: 0.8687 -
lr: 1.0000e-09
Epoch 3/3
782/782 [==============================] - ETA: 0s - loss: 0.7877 -
accuracy: 0.7983
Epoch 3: val_accuracy did not improve from 0.89750
782/782 [==============================] - 85s 109ms/step - loss:
```

```
0.7877 - accuracy: 0.7983 - val_loss: 0.5676 - val_accuracy: 0.8685 -
lr: 1.0000e-09
```

```
<keras.callbacks.History at 0x7feeed061910>
```

```
model.load_weights('/content/model_save/weights-15-0.8975.hdf5')
```

```
# Test the model
score = model.evaluate(X_test, y_test, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
313/313 [==============================] - 6s 19ms/step - loss: 0.3961
- accuracy: 0.8975
Test loss: 0.3961257040500641
Test accuracy: 0.8974999785423279
```

```
# Save the trained weights in to .h5 format
model.save_weights("DNST_model.h5")
print("Saved model to disk")
```

```
Saved model to disk
```