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 Adam
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
```

In [2]:

```
# this part will prevent tensorflow to allocate all the avaliable GPU Memory
# backend
import tensorflow as tf
from tensorflow import keras
```

In [3]:

```
# 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
170500096/170498071 [==============================] - 2s 0us/step
```

In [4]:

```
print(X_train.shape)
print(y_train.shape)
```

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

In [5]:

```
print(X_test.shape)
print(y_test.shape)
```

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

In [6]:

```
num_classes = 10
# 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)
```

In [7]:

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

```
(50000, 10)
(10000, 10)
```

In [8]:

```
X_train.shape
```

Out[8]:

```
(50000, 32, 32, 3)
```

In [9]:

```
import numpy as np
X_train_mean = np.mean(X_train, axis=(0,1,2))
X_train_mean
```

Out[9]:

```
array([125.30691805, 122.95039414, 113.86538318])
```

In [10]:

```
X_train_std = np.std(X_train, axis=(0,1,2))
X_train_std
```

Out[10]:

```
array([62.99321928, 62.08870764, 66.70489964])
```

In [11]:

```
X_train = (X_train - X_train_mean) / X_train_std
X_test = (X_test - X_train_mean) / X_train_std
```

In [12]:

```
#data augmentation
train_datagen=tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255,rotation_range=40,widt
h_shift_range=0.2,height_shift_range=0.2,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255) #we should not
augment test data
```

In [13]:

```
train_generator=train_datagen.flow(X_train,y_train,batch_size=150)
test_generator=test_datagen.flow(X_test,y_test,batch_size=150)
```

callback when val accuracy reaches 90%

In [14]:

```
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if(logs.get('val_accuracy') > 0.90):
            print("\nReached %2.2f%% accuracy, so stopping training!!" %(0.90*100))
            self.model.stop_training = True
```

In [15]:

```
val_acc_callback=myCallback()
```

modelcheckpoint callback

In [16]:

```
filepath='/content/best_model_h5'
```

In [17]:

```
#https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/ModelCheckpoint
best_model_weight=tf.keras.callbacks.ModelCheckpoint(filepath, monitor='val_accuracy', verbose=0, s
ave_best_only=True,save_weights_only=True, mode='auto', save_freq='epoch')
```

checking how many parametrs are there

In [18]:

```
def no_of_connection(l):
    a=(l * (l+ 1))/2 # each layer has connection to its preceding and subsequent layer directly
    return a

def total_parametres(l, no_of_filters,no_of_dense_blocks,no_of_transition_blocks):
    input_layer_param= 3 * 3 * 3 * no_of_filters
    para_dense_block= no_of_dense_blocks * ((no_of_filters * 3 * 3 * no_of_filters *
no_of_connection(l)) + ( 4 * no_of_filters * no_of_connection(l)))
    para_transition_block=no_of_transition_blocks * ( (1 * 1 * no_of_filters * no_of_filters* ((l +
1)) + 4 * no_of_filters * (l+1)))
    output_layer_params=((2 * 2 * no_of_filters * (l+1) * 10) + 10)  + (4 * no_of_filters * (l+1))

    return input_layer_param,para_dense_block,para_transition_block,output_layer_params
```

In [19]:

```
sum(total_parametres(13,17,4,3))
#we want less than 1 miliion parameter so we got here
```

Out[19]:

```
997451.0
```

In [ ]:

```

```

model

In [20]:

```
compression = 1
num_filter = 17
dropout_rate = 0
l = 13
```

In [21]:

```
# Dense Block
def denseblock(input, num_filter = 17, 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='sam
e')(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 = 17, 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')(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 [22]:

```
input = layers.Input(shape=(img_height, img_width, channel,))
First_Conv2D = layers.Conv2D(num_filter, (3,3), use_bias=False ,padding='same')(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 [23]:

```
model = Model(inputs=[input], outputs=[output])
```

In [24]:

```
model.summary()
```

```
Model: "functional_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 32, 32, 3)]  0
__________________________________________________________________________________________________
conv2d (Conv2D)                 (None, 32, 32, 17)   459         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 32, 32, 17)   68          conv2d[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 32, 32, 17)   0           batch_normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 32, 32, 17)   2601        activation[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 32, 32, 34)   0           conv2d[0][0]
                                                                 conv2d_1[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 32, 32, 34)   136         concatenate[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, 32, 32, 34)   0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 32, 32, 17)   5202        activation_1[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 32, 32, 51)   0           concatenate[0][0]
                                                                 conv2d_2[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 32, 32, 51)   204         concatenate_1[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, 32, 32, 51)   0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 32, 32, 17)   7803        activation_2[0][0]
```

concatenate_2 (Concatenate)	(None, 32, 32, 68)	0	concatenate_1[0][0] conv2d_3[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 68)	272	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 68)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 17)	10404	activation_3[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 85)	0	concatenate_2[0][0] conv2d_4[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 85)	340	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 85)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 17)	13005	activation_4[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 102)	0	concatenate_3[0][0] conv2d_5[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 102)	408	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 102)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 17)	15606	activation_5[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 119)	0	concatenate_4[0][0] conv2d_6[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 119)	476	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 119)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 17)	18207	activation_6[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 136)	0	concatenate_5[0][0] conv2d_7[0][0]
batch_normalization_7 (BatchNor	(None, 32, 32, 136)	544	concatenate_6[0][0]
activation_7 (Activation)	(None, 32, 32, 136)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 17)	20808	activation_7[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 153)	0	concatenate_6[0][0] conv2d_8[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 153)	612	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 153)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 17)	23409	activation_8[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 170)	0	concatenate_7[0][0] conv2d_9[0][0]
batch_normalization_9 (BatchNor	(None, 32, 32, 170)	680	concatenate_8[0][0]
activation_9 (Activation)	(None, 32, 32, 170)	0	batch_normalization_9[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 17)	26010	activation_9[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 187)	0	concatenate_8[0][0] conv2d_10[0][0]
batch_normalization_10 (BatchNo	(None, 32, 32, 187)	748	concatenate_9[0][0]
activation_10 (Activation)	(None, 32, 32, 187)	0	batch_normalization_10[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 17)	28611	activation_10[0][0]
concatenate_10 (Concatenate)	(None, 32, 32, 204)	0	concatenate_9[0][0] conv2d_11[0][0]
batch_normalization_11 (BatchNo	(None, 32, 32, 204)	816	concatenate_10[0][0]

activation_11 (Activation)	(None, 32, 32, 204)	0	batch_normalization_11[0][0]
conv2d_12 (Conv2D)	(None, 32, 32, 17)	31212	activation_11[0][0]
concatenate_11 (Concatenate)	(None, 32, 32, 221)	0	concatenate_10[0][0] conv2d_12[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 221)	884	concatenate_11[0][0]
activation_12 (Activation)	(None, 32, 32, 221)	0	batch_normalization_12[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 17)	33813	activation_12[0][0]
concatenate_12 (Concatenate)	(None, 32, 32, 238)	0	concatenate_11[0][0] conv2d_13[0][0]
batch_normalization_13 (BatchNo	(None, 32, 32, 238)	952	concatenate_12[0][0]
activation_13 (Activation)	(None, 32, 32, 238)	0	batch_normalization_13[0][0]
conv2d_14 (Conv2D)	(None, 32, 32, 17)	4046	activation_13[0][0]
average_pooling2d (AveragePooli	(None, 16, 16, 17)	0	conv2d_14[0][0]
batch_normalization_14 (BatchNo	(None, 16, 16, 17)	68	average_pooling2d[0][0]
activation_14 (Activation)	(None, 16, 16, 17)	0	batch_normalization_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 17)	2601	activation_14[0][0]
concatenate_13 (Concatenate)	(None, 16, 16, 34)	0	average_pooling2d[0][0] conv2d_15[0][0]
batch_normalization_15 (BatchNo	(None, 16, 16, 34)	136	concatenate_13[0][0]
activation_15 (Activation)	(None, 16, 16, 34)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 17)	5202	activation_15[0][0]
concatenate_14 (Concatenate)	(None, 16, 16, 51)	0	concatenate_13[0][0] conv2d_16[0][0]
batch_normalization_16 (BatchNo	(None, 16, 16, 51)	204	concatenate_14[0][0]
activation_16 (Activation)	(None, 16, 16, 51)	0	batch_normalization_16[0][0]
conv2d_17 (Conv2D)	(None, 16, 16, 17)	7803	activation_16[0][0]
concatenate_15 (Concatenate)	(None, 16, 16, 68)	0	concatenate_14[0][0] conv2d_17[0][0]
batch_normalization_17 (BatchNo	(None, 16, 16, 68)	272	concatenate_15[0][0]
activation_17 (Activation)	(None, 16, 16, 68)	0	batch_normalization_17[0][0]
conv2d_18 (Conv2D)	(None, 16, 16, 17)	10404	activation_17[0][0]
concatenate_16 (Concatenate)	(None, 16, 16, 85)	0	concatenate_15[0][0] conv2d_18[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 85)	340	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 85)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 17)	13005	activation_18[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 102)	0	concatenate_16[0][0] conv2d_19[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 102)	408	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 102)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 17)	15606	activation_19[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 119)	0	concatenate_17[0][0] conv2d_20[0][0]

			conv2d_20[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 119)	476	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 119)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 17)	18207	activation_20[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 136)	0	concatenate_18[0][0] conv2d_21[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 136)	544	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 136)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 17)	20808	activation_21[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 153)	0	concatenate_19[0][0] conv2d_22[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 153)	612	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 153)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 17)	23409	activation_22[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 170)	0	concatenate_20[0][0] conv2d_23[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 170)	680	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 170)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 17)	26010	activation_23[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 187)	0	concatenate_21[0][0] conv2d_24[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 187)	748	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 187)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 17)	28611	activation_24[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 204)	0	concatenate_22[0][0] conv2d_25[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 204)	816	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 204)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 17)	31212	activation_25[0][0]
concatenate_24 (Concatenate)	(None, 16, 16, 221)	0	concatenate_23[0][0] conv2d_26[0][0]
batch_normalization_26 (BatchNo	(None, 16, 16, 221)	884	concatenate_24[0][0]
activation_26 (Activation)	(None, 16, 16, 221)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 16, 16, 17)	33813	activation_26[0][0]
concatenate_25 (Concatenate)	(None, 16, 16, 238)	0	concatenate_24[0][0] conv2d_27[0][0]
batch_normalization_27 (BatchNo	(None, 16, 16, 238)	952	concatenate_25[0][0]
activation_27 (Activation)	(None, 16, 16, 238)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 16, 16, 17)	4046	activation_27[0][0]
average_pooling2d_1 (AveragePoo	(None, 8, 8, 17)	0	conv2d_28[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 17)	68	average_pooling2d_1[0][0]
activation_28 (Activation)	(None, 8, 8, 17)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 17)	2601	activation_28[0][0]

conv2d_29 (Conv2D)	(None, 8, 8, 17)	2601	activation_28[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 34)	0	average_pooling2d_1[0][0] conv2d_29[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 34)	136	concatenate_26[0][0]
activation_29 (Activation)	(None, 8, 8, 34)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 17)	5202	activation_29[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 51)	0	concatenate_26[0][0] conv2d_30[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 51)	204	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 51)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 17)	7803	activation_30[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 68)	0	concatenate_27[0][0] conv2d_31[0][0]
batch_normalization_31 (BatchNo	(None, 8, 8, 68)	272	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 68)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 17)	10404	activation_31[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 85)	0	concatenate_28[0][0] conv2d_32[0][0]
batch_normalization_32 (BatchNo	(None, 8, 8, 85)	340	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 85)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 17)	13005	activation_32[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 102)	0	concatenate_29[0][0] conv2d_33[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 102)	408	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 102)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 17)	15606	activation_33[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 119)	0	concatenate_30[0][0] conv2d_34[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 119)	476	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 119)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 17)	18207	activation_34[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 136)	0	concatenate_31[0][0] conv2d_35[0][0]
batch_normalization_35 (BatchNo	(None, 8, 8, 136)	544	concatenate_32[0][0]
activation_35 (Activation)	(None, 8, 8, 136)	0	batch_normalization_35[0][0]
conv2d_36 (Conv2D)	(None, 8, 8, 17)	20808	activation_35[0][0]
concatenate_33 (Concatenate)	(None, 8, 8, 153)	0	concatenate_32[0][0] conv2d_36[0][0]
batch_normalization_36 (BatchNo	(None, 8, 8, 153)	612	concatenate_33[0][0]
activation_36 (Activation)	(None, 8, 8, 153)	0	batch_normalization_36[0][0]
conv2d_37 (Conv2D)	(None, 8, 8, 17)	23409	activation_36[0][0]
concatenate_34 (Concatenate)	(None, 8, 8, 170)	0	concatenate_33[0][0] conv2d_37[0][0]

```
batch_normalization_37 (BatchNo (None, 8, 8, 170)    680         concatenate_34[0][0]
__________________________________________________________________________________________________
activation_37 (Activation)      (None, 8, 8, 170)    0           batch_normalization_37[0][0]
__________________________________________________________________________________________________
conv2d_38 (Conv2D)              (None, 8, 8, 17)     26010       activation_37[0][0]
__________________________________________________________________________________________________
concatenate_35 (Concatenate)    (None, 8, 8, 187)    0           concatenate_34[0][0]
                                                                 conv2d_38[0][0]
__________________________________________________________________________________________________
batch_normalization_38 (BatchNo (None, 8, 8, 187)    748         concatenate_35[0][0]
__________________________________________________________________________________________________
activation_38 (Activation)      (None, 8, 8, 187)    0           batch_normalization_38[0][0]
__________________________________________________________________________________________________
conv2d_39 (Conv2D)              (None, 8, 8, 17)     28611       activation_38[0][0]
__________________________________________________________________________________________________
concatenate_36 (Concatenate)    (None, 8, 8, 204)    0           concatenate_35[0][0]
                                                                 conv2d_39[0][0]
__________________________________________________________________________________________________
batch_normalization_39 (BatchNo (None, 8, 8, 204)    816         concatenate_36[0][0]
__________________________________________________________________________________________________
activation_39 (Activation)      (None, 8, 8, 204)    0           batch_normalization_39[0][0]
__________________________________________________________________________________________________
conv2d_40 (Conv2D)              (None, 8, 8, 17)     31212       activation_39[0][0]
__________________________________________________________________________________________________
concatenate_37 (Concatenate)    (None, 8, 8, 221)    0           concatenate_36[0][0]
                                                                 conv2d_40[0][0]
__________________________________________________________________________________________________
batch_normalization_40 (BatchNo (None, 8, 8, 221)    884         concatenate_37[0][0]
__________________________________________________________________________________________________
activation_40 (Activation)      (None, 8, 8, 221)    0           batch_normalization_40[0][0]
__________________________________________________________________________________________________
conv2d_41 (Conv2D)              (None, 8, 8, 17)     33813       activation_40[0][0]
__________________________________________________________________________________________________
concatenate_38 (Concatenate)    (None, 8, 8, 238)    0           concatenate_37[0][0]
                                                                 conv2d_41[0][0]
__________________________________________________________________________________________________
batch_normalization_41 (BatchNo (None, 8, 8, 238)    952         concatenate_38[0][0]
__________________________________________________________________________________________________
activation_41 (Activation)      (None, 8, 8, 238)    0           batch_normalization_41[0][0]
__________________________________________________________________________________________________
conv2d_42 (Conv2D)              (None, 8, 8, 17)     4046        activation_41[0][0]
__________________________________________________________________________________________________
average_pooling2d_2 (AveragePoo (None, 4, 4, 17)     0           conv2d_42[0][0]
__________________________________________________________________________________________________
batch_normalization_42 (BatchNo (None, 4, 4, 17)     68          average_pooling2d_2[0][0]
__________________________________________________________________________________________________
activation_42 (Activation)      (None, 4, 4, 17)     0           batch_normalization_42[0][0]
__________________________________________________________________________________________________
conv2d_43 (Conv2D)              (None, 4, 4, 17)     2601        activation_42[0][0]
__________________________________________________________________________________________________
concatenate_39 (Concatenate)    (None, 4, 4, 34)     0           average_pooling2d_2[0][0]
                                                                 conv2d_43[0][0]
__________________________________________________________________________________________________
batch_normalization_43 (BatchNo (None, 4, 4, 34)     136         concatenate_39[0][0]
__________________________________________________________________________________________________
activation_43 (Activation)      (None, 4, 4, 34)     0           batch_normalization_43[0][0]
__________________________________________________________________________________________________
conv2d_44 (Conv2D)              (None, 4, 4, 17)     5202        activation_43[0][0]
__________________________________________________________________________________________________
concatenate_40 (Concatenate)    (None, 4, 4, 51)     0           concatenate_39[0][0]
                                                                 conv2d_44[0][0]
__________________________________________________________________________________________________
batch_normalization_44 (BatchNo (None, 4, 4, 51)     204         concatenate_40[0][0]
__________________________________________________________________________________________________
activation_44 (Activation)      (None, 4, 4, 51)     0           batch_normalization_44[0][0]
__________________________________________________________________________________________________
conv2d_45 (Conv2D)              (None, 4, 4, 17)     7803        activation_44[0][0]
__________________________________________________________________________________________________
concatenate_41 (Concatenate)    (None, 4, 4, 68)     0           concatenate_40[0][0]
                                                                 conv2d_45[0][0]
__________________________________________________________________________________________________
batch_normalization_45 (BatchNo (None, 4, 4, 68)     272         concatenate_41[0][0]
__________________________________________________________________________________________________
activation_45 (Activation)      (None, 4, 4, 68)     0           batch_normalization_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)              (None, 4, 4, 17)     10404       activation_45[0][0]
__________________________________________________________________________________________________
```

concatenate_42 (Concatenate)	(None, 4, 4, 85)	0	concatenate_41[0][0] conv2d_46[0][0]
batch_normalization_46 (BatchNo	(None, 4, 4, 85)	340	concatenate_42[0][0]
activation_46 (Activation)	(None, 4, 4, 85)	0	batch_normalization_46[0][0]
conv2d_47 (Conv2D)	(None, 4, 4, 17)	13005	activation_46[0][0]
concatenate_43 (Concatenate)	(None, 4, 4, 102)	0	concatenate_42[0][0] conv2d_47[0][0]
batch_normalization_47 (BatchNo	(None, 4, 4, 102)	408	concatenate_43[0][0]
activation_47 (Activation)	(None, 4, 4, 102)	0	batch_normalization_47[0][0]
conv2d_48 (Conv2D)	(None, 4, 4, 17)	15606	activation_47[0][0]
concatenate_44 (Concatenate)	(None, 4, 4, 119)	0	concatenate_43[0][0] conv2d_48[0][0]
batch_normalization_48 (BatchNo	(None, 4, 4, 119)	476	concatenate_44[0][0]
activation_48 (Activation)	(None, 4, 4, 119)	0	batch_normalization_48[0][0]
conv2d_49 (Conv2D)	(None, 4, 4, 17)	18207	activation_48[0][0]
concatenate_45 (Concatenate)	(None, 4, 4, 136)	0	concatenate_44[0][0] conv2d_49[0][0]
batch_normalization_49 (BatchNo	(None, 4, 4, 136)	544	concatenate_45[0][0]
activation_49 (Activation)	(None, 4, 4, 136)	0	batch_normalization_49[0][0]
conv2d_50 (Conv2D)	(None, 4, 4, 17)	20808	activation_49[0][0]
concatenate_46 (Concatenate)	(None, 4, 4, 153)	0	concatenate_45[0][0] conv2d_50[0][0]
batch_normalization_50 (BatchNo	(None, 4, 4, 153)	612	concatenate_46[0][0]
activation_50 (Activation)	(None, 4, 4, 153)	0	batch_normalization_50[0][0]
conv2d_51 (Conv2D)	(None, 4, 4, 17)	23409	activation_50[0][0]
concatenate_47 (Concatenate)	(None, 4, 4, 170)	0	concatenate_46[0][0] conv2d_51[0][0]
batch_normalization_51 (BatchNo	(None, 4, 4, 170)	680	concatenate_47[0][0]
activation_51 (Activation)	(None, 4, 4, 170)	0	batch_normalization_51[0][0]
conv2d_52 (Conv2D)	(None, 4, 4, 17)	26010	activation_51[0][0]
concatenate_48 (Concatenate)	(None, 4, 4, 187)	0	concatenate_47[0][0] conv2d_52[0][0]
batch_normalization_52 (BatchNo	(None, 4, 4, 187)	748	concatenate_48[0][0]
activation_52 (Activation)	(None, 4, 4, 187)	0	batch_normalization_52[0][0]
conv2d_53 (Conv2D)	(None, 4, 4, 17)	28611	activation_52[0][0]
concatenate_49 (Concatenate)	(None, 4, 4, 204)	0	concatenate_48[0][0] conv2d_53[0][0]
batch_normalization_53 (BatchNo	(None, 4, 4, 204)	816	concatenate_49[0][0]
activation_53 (Activation)	(None, 4, 4, 204)	0	batch_normalization_53[0][0]
conv2d_54 (Conv2D)	(None, 4, 4, 17)	31212	activation_53[0][0]
concatenate_50 (Concatenate)	(None, 4, 4, 221)	0	concatenate_49[0][0] conv2d_54[0][0]
batch_normalization_54 (BatchNo	(None, 4, 4, 221)	884	concatenate_50[0][0]

```
activation_54 (Activation)      (None, 4, 4, 221)    0           batch_normalization_54[0][0]
__________________________________________________________________________________________________
conv2d_55 (Conv2D)              (None, 4, 4, 17)     33813       activation_54[0][0]
__________________________________________________________________________________________________
concatenate_51 (Concatenate)    (None, 4, 4, 238)    0           concatenate_50[0][0]
                                                                 conv2d_55[0][0]
__________________________________________________________________________________________________
batch_normalization_55 (BatchNo (None, 4, 4, 238)    952         concatenate_51[0][0]
__________________________________________________________________________________________________
activation_55 (Activation)      (None, 4, 4, 238)    0           batch_normalization_55[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 238)    0           activation_55[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 952)          0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           9530        flatten[0][0]
==================================================================================================
Total params: 997,451
Trainable params: 983,171
Non-trainable params: 14,280
__________________________________________________________________________________________________
```

In [25]:

```
model.compile(optimizer=tf.keras.optimizers.Adam(lr=0.001),loss='categorical_crossentropy',metrics
=['accuracy'])
```

In [26]:

```
model.fit(train_generator,
                    steps_per_epoch=len(X_train)/150,
                    epochs=300,
                    verbose=1,
                    validation_data=test_generator,
                    validation_steps=len(X_test)/150,
                    callbacks=[val_acc_callback,best_model_weight])
```

```
Epoch 1/300
334/333 [==============================] - 54s 160ms/step - loss: 1.7505 - accuracy: 0.3555 - val_
loss: 3.6620 - val_accuracy: 0.1000
Epoch 2/300
334/333 [==============================] - 51s 154ms/step - loss: 1.4090 - accuracy: 0.4871 - val_
loss: 1.6579 - val_accuracy: 0.4158
Epoch 3/300
334/333 [==============================] - 51s 154ms/step - loss: 1.2277 - accuracy: 0.5607 - val_
loss: 1.5663 - val_accuracy: 0.5217
Epoch 4/300
334/333 [==============================] - 51s 154ms/step - loss: 1.1035 - accuracy: 0.6081 - val_
loss: 1.5286 - val_accuracy: 0.5572
Epoch 5/300
334/333 [==============================] - 51s 154ms/step - loss: 1.0081 - accuracy: 0.6391 - val_
loss: 1.1640 - val_accuracy: 0.6104
Epoch 6/300
334/333 [==============================] - 51s 153ms/step - loss: 0.9411 - accuracy: 0.6698 - val_
loss: 1.4103 - val_accuracy: 0.5715
Epoch 7/300
334/333 [==============================] - 51s 154ms/step - loss: 0.8861 - accuracy: 0.6867 - val_
loss: 1.3733 - val_accuracy: 0.6130
Epoch 8/300
334/333 [==============================] - 51s 154ms/step - loss: 0.8360 - accuracy: 0.7069 - val_
loss: 0.8275 - val_accuracy: 0.7148
Epoch 9/300
334/333 [==============================] - 51s 154ms/step - loss: 0.7998 - accuracy: 0.7188 - val_
loss: 0.7996 - val_accuracy: 0.7336
Epoch 10/300
334/333 [==============================] - 51s 152ms/step - loss: 0.7683 - accuracy: 0.7334 - val_
loss: 1.1063 - val_accuracy: 0.6598
Epoch 11/300
334/333 [==============================] - 51s 153ms/step - loss: 0.7342 - accuracy: 0.7431 - val_
loss: 0.9869 - val_accuracy: 0.7004
Epoch 12/300
334/333 [==============================] - 51s 154ms/step - loss: 0.7115 - accuracy: 0.7519 - val_
loss: 0.7422 - val_accuracy: 0.7581
Epoch 13/300
```

```
334/333 [==============================] - 51s 153ms/step - loss: 0.6822 - accuracy: 0.7632 - val_
loss: 1.1851 - val_accuracy: 0.6462
Epoch 14/300
334/333 [==============================] - 51s 154ms/step - loss: 0.6670 - accuracy: 0.7664 - val_
loss: 0.7669 - val_accuracy: 0.7509
Epoch 15/300
334/333 [==============================] - 51s 153ms/step - loss: 0.6434 - accuracy: 0.7743 - val_
loss: 0.7728 - val_accuracy: 0.7503
Epoch 16/300
334/333 [==============================] - 51s 154ms/step - loss: 0.6246 - accuracy: 0.7844 - val_
loss: 0.7229 - val_accuracy: 0.7639
Epoch 17/300
334/333 [==============================] - 51s 154ms/step - loss: 0.6045 - accuracy: 0.7912 - val_
loss: 0.6177 - val_accuracy: 0.7939
Epoch 18/300
334/333 [==============================] - 51s 153ms/step - loss: 0.5900 - accuracy: 0.7967 - val_
loss: 0.8267 - val_accuracy: 0.7412
Epoch 19/300
334/333 [==============================] - 51s 153ms/step - loss: 0.5734 - accuracy: 0.7996 - val_
loss: 0.9488 - val_accuracy: 0.7043
Epoch 20/300
334/333 [==============================] - 51s 152ms/step - loss: 0.5621 - accuracy: 0.8045 - val_
loss: 0.6542 - val_accuracy: 0.7892
Epoch 21/300
334/333 [==============================] - 51s 152ms/step - loss: 0.5489 - accuracy: 0.8085 - val_
loss: 0.6973 - val_accuracy: 0.7699
Epoch 22/300
334/333 [==============================] - 51s 153ms/step - loss: 0.5332 - accuracy: 0.8145 - val_
loss: 0.7134 - val_accuracy: 0.7740
Epoch 23/300
334/333 [==============================] - 51s 153ms/step - loss: 0.5293 - accuracy: 0.8162 - val_
loss: 0.7842 - val_accuracy: 0.7613
Epoch 24/300
334/333 [==============================] - 51s 153ms/step - loss: 0.5050 - accuracy: 0.8250 - val_
loss: 0.5813 - val_accuracy: 0.8102
Epoch 25/300
334/333 [==============================] - 51s 154ms/step - loss: 0.4990 - accuracy: 0.8272 - val_
loss: 0.5424 - val_accuracy: 0.8160
Epoch 26/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4920 - accuracy: 0.8298 - val_
loss: 0.6102 - val_accuracy: 0.8024
Epoch 27/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4847 - accuracy: 0.8314 - val_
loss: 0.5787 - val_accuracy: 0.8105
Epoch 28/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4717 - accuracy: 0.8357 - val_
loss: 0.5465 - val_accuracy: 0.8179
Epoch 29/300
334/333 [==============================] - 52s 154ms/step - loss: 0.4627 - accuracy: 0.8383 - val_
loss: 0.5409 - val_accuracy: 0.8208
Epoch 30/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4524 - accuracy: 0.8412 - val_
loss: 0.6130 - val_accuracy: 0.8023
Epoch 31/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4521 - accuracy: 0.8423 - val_
loss: 0.8937 - val_accuracy: 0.7540
Epoch 32/300
334/333 [==============================] - 51s 154ms/step - loss: 0.4393 - accuracy: 0.8460 - val_
loss: 0.5301 - val_accuracy: 0.8229
Epoch 33/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4341 - accuracy: 0.8478 - val_
loss: 0.6588 - val_accuracy: 0.7985
Epoch 34/300
334/333 [==============================] - 51s 152ms/step - loss: 0.4273 - accuracy: 0.8508 - val_
loss: 0.6132 - val_accuracy: 0.8083
Epoch 35/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4229 - accuracy: 0.8538 - val_
loss: 0.6273 - val_accuracy: 0.8107
Epoch 36/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4120 - accuracy: 0.8555 - val_
loss: 0.5169 - val_accuracy: 0.8348
Epoch 37/300
334/333 [==============================] - 51s 153ms/step - loss: 0.4069 - accuracy: 0.8601 - val_
loss: 0.6033 - val_accuracy: 0.8147
Epoch 38/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3972 - accuracy: 0.8610 - val_
loss: 0.5981 - val_accuracy: 0.8217
Epoch 39/300
```

```
Epoch 39/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3969 - accuracy: 0.8636 - val_
loss: 0.4703 - val_accuracy: 0.8445
Epoch 40/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3922 - accuracy: 0.8627 - val_
loss: 0.5772 - val_accuracy: 0.8241
Epoch 41/300
334/333 [==============================] - 51s 154ms/step - loss: 0.3843 - accuracy: 0.8654 - val_
loss: 0.4637 - val_accuracy: 0.8481
Epoch 42/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3780 - accuracy: 0.8674 - val_
loss: 0.5174 - val_accuracy: 0.8369
Epoch 43/300
334/333 [==============================] - 51s 154ms/step - loss: 0.3738 - accuracy: 0.8699 - val_
loss: 0.4248 - val_accuracy: 0.8574
Epoch 44/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3737 - accuracy: 0.8697 - val_
loss: 0.5075 - val_accuracy: 0.8426
Epoch 45/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3637 - accuracy: 0.8743 - val_
loss: 0.5532 - val_accuracy: 0.8308
Epoch 46/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3578 - accuracy: 0.8742 - val_
loss: 0.4721 - val_accuracy: 0.8460
Epoch 47/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3521 - accuracy: 0.8755 - val_
loss: 0.5202 - val_accuracy: 0.8369
Epoch 48/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3511 - accuracy: 0.8768 - val_
loss: 0.4072 - val_accuracy: 0.8690
Epoch 49/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3432 - accuracy: 0.8785 - val_
loss: 0.4849 - val_accuracy: 0.8496
Epoch 50/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3409 - accuracy: 0.8798 - val_
loss: 0.5081 - val_accuracy: 0.8437
Epoch 51/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3396 - accuracy: 0.8810 - val_
loss: 0.4418 - val_accuracy: 0.8535
Epoch 52/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3342 - accuracy: 0.8830 - val_
loss: 0.4488 - val_accuracy: 0.8549
Epoch 53/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3330 - accuracy: 0.8831 - val_
loss: 0.5777 - val_accuracy: 0.8276
Epoch 54/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3265 - accuracy: 0.8851 - val_
loss: 0.5134 - val_accuracy: 0.8421
Epoch 55/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3232 - accuracy: 0.8880 - val_
loss: 0.6081 - val_accuracy: 0.8226
Epoch 56/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3221 - accuracy: 0.8874 - val_
loss: 0.4668 - val_accuracy: 0.8587
Epoch 57/300
334/333 [==============================] - 51s 152ms/step - loss: 0.3160 - accuracy: 0.8900 - val_
loss: 0.4424 - val_accuracy: 0.8608
Epoch 58/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3136 - accuracy: 0.8919 - val_
loss: 0.4492 - val_accuracy: 0.8617
Epoch 59/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3121 - accuracy: 0.8908 - val_
loss: 0.4586 - val_accuracy: 0.8599
Epoch 60/300
334/333 [==============================] - 51s 153ms/step - loss: 0.3041 - accuracy: 0.8940 - val_
loss: 0.4365 - val_accuracy: 0.8633
Epoch 61/300
334/333 [==============================] - 51s 154ms/step - loss: 0.3069 - accuracy: 0.8928 - val_
loss: 0.4249 - val_accuracy: 0.8699
Epoch 62/300
334/333 [==============================] - 51s 154ms/step - loss: 0.3001 - accuracy: 0.8935 - val_
loss: 0.4152 - val_accuracy: 0.8704
Epoch 63/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2920 - accuracy: 0.8978 - val_
loss: 0.4542 - val_accuracy: 0.8597
Epoch 64/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2953 - accuracy: 0.8970 - val_
```

```
loss: 0.5169 - val_accuracy: 0.8470
Epoch 65/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2946 - accuracy: 0.8975 - val_
loss: 0.4097 - val_accuracy: 0.8766
Epoch 66/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2899 - accuracy: 0.8987 - val_
loss: 0.5117 - val_accuracy: 0.8457
Epoch 67/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2848 - accuracy: 0.9017 - val_
loss: 0.4783 - val_accuracy: 0.8559
Epoch 68/300
334/333 [==============================] - 51s 154ms/step - loss: 0.2827 - accuracy: 0.9013 - val_
loss: 0.3828 - val_accuracy: 0.8789
Epoch 69/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2804 - accuracy: 0.9008 - val_
loss: 0.4253 - val_accuracy: 0.8695
Epoch 70/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2794 - accuracy: 0.9027 - val_
loss: 0.4694 - val_accuracy: 0.8597
Epoch 71/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2782 - accuracy: 0.9016 - val_
loss: 0.4502 - val_accuracy: 0.8631
Epoch 72/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2740 - accuracy: 0.9039 - val_
loss: 0.4099 - val_accuracy: 0.8741
Epoch 73/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2712 - accuracy: 0.9053 - val_
loss: 0.4718 - val_accuracy: 0.8553
Epoch 74/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2675 - accuracy: 0.9062 - val_
loss: 0.4498 - val_accuracy: 0.8657
Epoch 75/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2714 - accuracy: 0.9044 - val_
loss: 0.7495 - val_accuracy: 0.8063
Epoch 76/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2627 - accuracy: 0.9081 - val_
loss: 0.5046 - val_accuracy: 0.8541
Epoch 77/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2661 - accuracy: 0.9061 - val_
loss: 0.4481 - val_accuracy: 0.8630
Epoch 78/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2587 - accuracy: 0.9074 - val_
loss: 0.3905 - val_accuracy: 0.8788
Epoch 79/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2588 - accuracy: 0.9099 - val_
loss: 0.4841 - val_accuracy: 0.8548
Epoch 80/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2553 - accuracy: 0.9098 - val_
loss: 0.4363 - val_accuracy: 0.8737
Epoch 81/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2557 - accuracy: 0.9105 - val_
loss: 0.4626 - val_accuracy: 0.8656
Epoch 82/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2504 - accuracy: 0.9109 - val_
loss: 0.4731 - val_accuracy: 0.8650
Epoch 83/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2495 - accuracy: 0.9109 - val_
loss: 0.4315 - val_accuracy: 0.8718
Epoch 84/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2488 - accuracy: 0.9126 - val_
loss: 0.4556 - val_accuracy: 0.8652
Epoch 85/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2465 - accuracy: 0.9137 - val_
loss: 0.4795 - val_accuracy: 0.8583
Epoch 86/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2454 - accuracy: 0.9134 - val_
loss: 0.5148 - val_accuracy: 0.8521
Epoch 87/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2457 - accuracy: 0.9130 - val_
loss: 0.4733 - val_accuracy: 0.8629
Epoch 88/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2453 - accuracy: 0.9139 - val_
loss: 0.6868 - val_accuracy: 0.8167
Epoch 89/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2385 - accuracy: 0.9162 - val_
loss: 0.3570 - val_accuracy: 0.8882
Epoch 90/300
```

```
334/333 [==============================] - 51s 153ms/step - loss: 0.2371 - accuracy: 0.9168 - val_
loss: 0.3963 - val_accuracy: 0.8837
Epoch 91/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2389 - accuracy: 0.9159 - val_
loss: 0.4143 - val_accuracy: 0.8788
Epoch 92/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2331 - accuracy: 0.9171 - val_
loss: 0.5186 - val_accuracy: 0.8568
Epoch 93/300
334/333 [==============================] - 51s 152ms/step - loss: 0.2305 - accuracy: 0.9188 - val_
loss: 0.3984 - val_accuracy: 0.8861
Epoch 94/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2308 - accuracy: 0.9183 - val_
loss: 0.3791 - val_accuracy: 0.8893
Epoch 95/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2303 - accuracy: 0.9194 - val_
loss: 0.3492 - val_accuracy: 0.8942
Epoch 96/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2251 - accuracy: 0.9196 - val_
loss: 0.4713 - val_accuracy: 0.8666
Epoch 98/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2234 - accuracy: 0.9213 - val_
loss: 0.3811 - val_accuracy: 0.8859
Epoch 99/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2202 - accuracy: 0.9222 - val_
loss: 0.4010 - val_accuracy: 0.8798
Epoch 100/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2171 - accuracy: 0.9237 - val_
loss: 0.4934 - val_accuracy: 0.8638
Epoch 101/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2188 - accuracy: 0.9222 - val_
loss: 0.4383 - val_accuracy: 0.8734
Epoch 102/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2133 - accuracy: 0.9245 - val_
loss: 0.3872 - val_accuracy: 0.8861
Epoch 103/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2165 - accuracy: 0.9240 - val_
loss: 0.4106 - val_accuracy: 0.8843
Epoch 104/300
334/333 [==============================] - 51s 154ms/step - loss: 0.2111 - accuracy: 0.9254 - val_
loss: 0.3657 - val_accuracy: 0.8962
Epoch 105/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2155 - accuracy: 0.9236 - val_
loss: 0.5285 - val_accuracy: 0.8573
Epoch 106/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2069 - accuracy: 0.9262 - val_
loss: 0.4839 - val_accuracy: 0.8693
Epoch 107/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2123 - accuracy: 0.9234 - val_
loss: 0.4152 - val_accuracy: 0.8846
Epoch 108/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2107 - accuracy: 0.9256 - val_
loss: 0.4532 - val_accuracy: 0.8730
Epoch 109/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2065 - accuracy: 0.9272 - val_
loss: 0.3973 - val_accuracy: 0.8873
Epoch 110/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2076 - accuracy: 0.9274 - val_
loss: 0.3897 - val_accuracy: 0.8890
Epoch 111/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2019 - accuracy: 0.9294 - val_
loss: 0.3692 - val_accuracy: 0.8867
Epoch 112/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2065 - accuracy: 0.9276 - val_
loss: 0.4388 - val_accuracy: 0.8743
Epoch 113/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2020 - accuracy: 0.9277 - val_
loss: 0.5312 - val_accuracy: 0.8607
Epoch 114/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2020 - accuracy: 0.9297 - val_
loss: 0.3969 - val_accuracy: 0.8852
Epoch 115/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2041 - accuracy: 0.9285 - val_
loss: 0.3832 - val_accuracy: 0.8885
Epoch 116/300
334/333 [==============================] - 51s 153ms/step - loss: 0.2013 - accuracy: 0.9294 - val_
loss: 0.4033 - val_accuracy: 0.8837
```

```
Epoch 117/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1964 - accuracy: 0.9303 - val_
loss: 0.3878 - val_accuracy: 0.8903
Epoch 118/300
334/333 [==============================] - 51s 154ms/step - loss: 0.1989 - accuracy: 0.9298 - val_
loss: 0.3567 - val_accuracy: 0.8985
Epoch 119/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1945 - accuracy: 0.9304 - val_
loss: 0.3867 - val_accuracy: 0.8904
Epoch 120/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1927 - accuracy: 0.9309 - val_
loss: 0.4144 - val_accuracy: 0.8859
Epoch 121/300
334/333 [==============================] - 51s 154ms/step - loss: 0.1934 - accuracy: 0.9322 - val_
loss: 0.4060 - val_accuracy: 0.8871
Epoch 122/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1935 - accuracy: 0.9312 - val_
loss: 0.5669 - val_accuracy: 0.8507
Epoch 123/300
334/333 [==============================] - 51s 154ms/step - loss: 0.1891 - accuracy: 0.9332 - val_
loss: 0.5069 - val_accuracy: 0.8682
Epoch 124/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1906 - accuracy: 0.9330 - val_
loss: 0.4660 - val_accuracy: 0.8715
Epoch 125/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1867 - accuracy: 0.9340 - val_
loss: 0.3751 - val_accuracy: 0.8936
Epoch 126/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1851 - accuracy: 0.9342 - val_
loss: 0.3837 - val_accuracy: 0.8906
Epoch 127/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1837 - accuracy: 0.9343 - val_
loss: 0.4760 - val_accuracy: 0.8742
Epoch 128/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1848 - accuracy: 0.9347 - val_
loss: 0.3649 - val_accuracy: 0.8981
Epoch 129/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1860 - accuracy: 0.9330 - val_
loss: 0.3407 - val_accuracy: 0.8961
Epoch 130/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1810 - accuracy: 0.9358 - val_
loss: 0.3714 - val_accuracy: 0.8949
Epoch 131/300
334/333 [==============================] - 51s 153ms/step - loss: 0.1813 - accuracy: 0.9352 - val_
loss: 0.3691 - val_accuracy: 0.8950
Epoch 132/300
334/333 [==============================] - ETA: 0s - loss: 0.1841 - accuracy: 0.9343
Reached 90.00% accuracy, so stopping training!!
334/333 [==============================] - 51s 154ms/step - loss: 0.1841 - accuracy: 0.9343 - val_
loss: 0.3415 - val_accuracy: 0.9030
```

Out[26]:

```
<tensorflow.python.keras.callbacks.History at 0x7fe6a9b7b048>
```

In [27]:

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

```
67/67 [==============================] - 3s 37ms/step - loss: 0.3415 - accuracy: 0.9030
Test loss: 0.3415147662162781
Test accuracy: 0.902999997138977
```

In [28]:

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

```
Saved model to disk
```

conclusion:

- in densenet each layer is connected to its preceding layer and subsequent layer directly
- densent is parameter efficient as compared other network like resnet , vgg, alexnet
- here i have created function fo checking how maany parametres does it contain so we can manage keep it below 1 million