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
```

In [3]:
```
# Hyperparameters
batch_size = 128
num_classes = 10
epochs = 10
l = 40
num_filter = 12
compression = 0.5
dropout_rate = 0.2
```

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

# 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)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 6s 0us/step
```

In [5]:
```
X_train.shape
```

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

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

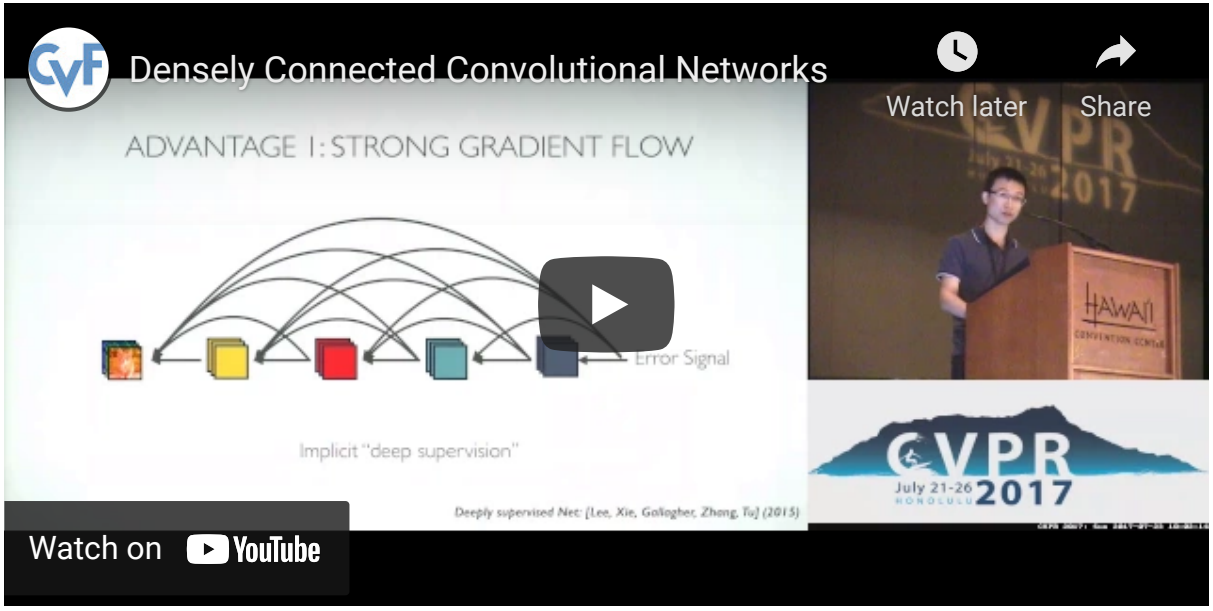
Out[6]:
```
(10000, 32, 32, 3)
```

In [7]:
```
# 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_3_3 = layers.Conv2D(int(num_filter*compression), (3,3), use_bias=False ,padding='same')(relu)
        if dropout_rate>0:
            Conv2D_3_3 = layers.Dropout(dropout_rate)(Conv2D_3_3)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_3_3])

        temp = concat

    return temp

## transition Blosck
def transition(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (1,1), use_bias=False ,padding='same')(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 [8]:
```
num_filter = 12
dropout_rate = 0.2
l = 12
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 [9]:
```
#https://arxiv.org/pdf/1608.06993.pdf
from IPython.display import IFrame, YouTubeVideo
YouTubeVideo(id='-W6y8xnd--U', width=600)
```

Out[9]:



In [10]:
```
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, 12)   324         input_1[0][0]
__________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 32, 32, 12)   48          conv2d[0][0]
__________________________________________________________________________________________________
activation (Activation)         (None, 32, 32, 12)   0           batch_normalization[0][0]
__________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 32, 32, 6)    648         activation[0][0]
__________________________________________________________________________________________________
dropout (Dropout)               (None, 32, 32, 6)    0           conv2d_1[0][0]
__________________________________________________________________________________________________
concatenate (Concatenate)       (None, 32, 32, 18)   0           conv2d[0][0]
                                                                 dropout[0][0]
__________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 32, 32, 18)   72          concatenate[0][0]
__________________________________________________________________________________________________
activation_1 (Activation)       (None, 32, 32, 18)   0           batch_normalization_1[0][0]
__________________________________________________________________________________________________
conv2d_2 (Conv2D)               (None, 32, 32, 6)    972         activation_1[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 32, 32, 6)    0           conv2d_2[0][0]
__________________________________________________________________________________________________
concatenate_1 (Concatenate)     (None, 32, 32, 24)   0           concatenate[0][0]
                                                                 dropout_1[0][0]
__________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 32, 32, 24)   96          concatenate_1[0][0]
__________________________________________________________________________________________________
activation_2 (Activation)       (None, 32, 32, 24)   0           batch_normalization_2[0][0]
__________________________________________________________________________________________________
conv2d_3 (Conv2D)               (None, 32, 32, 6)    1296        activation_2[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 32, 32, 6)    0           conv2d_3[0][0]
__________________________________________________________________________________________________
concatenate_2 (Concatenate)     (None, 32, 32, 30)   0           concatenate_1[0][0]
                                                                 dropout_2[0][0]
__________________________________________________________________________________________________
batch_normalization_3 (BatchNor (None, 32, 32, 30)   120         concatenate_2[0][0]
```

activation_3 (Activation)	(None, 32, 32, 30)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 6)	1620	activation_3[0][0]
dropout_3 (Dropout)	(None, 32, 32, 6)	0	conv2d_4[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 36)	0	concatenate_2[0][0] dropout_3[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 36)	144	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 36)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 6)	1944	activation_4[0][0]
dropout_4 (Dropout)	(None, 32, 32, 6)	0	conv2d_5[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 42)	0	concatenate_3[0][0] dropout_4[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 42)	168	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 42)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 6)	2268	activation_5[0][0]
dropout_5 (Dropout)	(None, 32, 32, 6)	0	conv2d_6[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 48)	0	concatenate_4[0][0] dropout_5[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 48)	192	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 48)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 6)	2592	activation_6[0][0]
dropout_6 (Dropout)	(None, 32, 32, 6)	0	conv2d_7[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 54)	0	concatenate_5[0][0] dropout_6[0][0]
batch_normalization_7 (BatchNor	(None, 32, 32, 54)	216	concatenate_6[0][0]

activation_7 (Activation)	(None, 32, 32, 54)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 6)	2916	activation_7[0][0]
dropout_7 (Dropout)	(None, 32, 32, 6)	0	conv2d_8[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 60)	0	concatenate_6[0][0] dropout_7[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 60)	240	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 60)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 6)	3240	activation_8[0][0]
dropout_8 (Dropout)	(None, 32, 32, 6)	0	conv2d_9[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 66)	0	concatenate_7[0][0] dropout_8[0][0]
batch_normalization_9 (BatchNor	(None, 32, 32, 66)	264	concatenate_8[0][0]
activation_9 (Activation)	(None, 32, 32, 66)	0	batch_normalization_9[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 6)	3564	activation_9[0][0]
dropout_9 (Dropout)	(None, 32, 32, 6)	0	conv2d_10[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 72)	0	concatenate_8[0][0] dropout_9[0][0]
batch_normalization_10 (BatchNo	(None, 32, 32, 72)	288	concatenate_9[0][0]
activation_10 (Activation)	(None, 32, 32, 72)	0	batch_normalization_10[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 6)	3888	activation_10[0][0]
dropout_10 (Dropout)	(None, 32, 32, 6)	0	conv2d_11[0][0]
concatenate_10 (Concatenate)	(None, 32, 32, 78)	0	concatenate_9[0][0] dropout_10[0][0]
batch_normalization_11 (BatchNo	(None, 32, 32, 78)	312	concatenate_10[0][0]
activation_11 (Activation)	(None, 32, 32, 78)	0	batch_normalization_11[0][0]

conv2d_12 (Conv2D)	(None, 32, 32, 6)	4212	activation_11[0][0]
dropout_11 (Dropout)	(None, 32, 32, 6)	0	conv2d_12[0][0]
concatenate_11 (Concatenate)	(None, 32, 32, 84)	0	concatenate_10[0][0] dropout_11[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 84)	336	concatenate_11[0][0]
activation_12 (Activation)	(None, 32, 32, 84)	0	batch_normalization_12[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 6)	504	activation_12[0][0]
dropout_12 (Dropout)	(None, 32, 32, 6)	0	conv2d_13[0][0]
average_pooling2d (AveragePooli	(None, 16, 16, 6)	0	dropout_12[0][0]
batch_normalization_13 (BatchNo	(None, 16, 16, 6)	24	average_pooling2d[0][0]
activation_13 (Activation)	(None, 16, 16, 6)	0	batch_normalization_13[0][0]
conv2d_14 (Conv2D)	(None, 16, 16, 6)	324	activation_13[0][0]
dropout_13 (Dropout)	(None, 16, 16, 6)	0	conv2d_14[0][0]
concatenate_12 (Concatenate)	(None, 16, 16, 12)	0	average_pooling2d[0][0] dropout_13[0][0]
batch_normalization_14 (BatchNo	(None, 16, 16, 12)	48	concatenate_12[0][0]
activation_14 (Activation)	(None, 16, 16, 12)	0	batch_normalization_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 6)	648	activation_14[0][0]
dropout_14 (Dropout)	(None, 16, 16, 6)	0	conv2d_15[0][0]
concatenate_13 (Concatenate)	(None, 16, 16, 18)	0	concatenate_12[0][0] dropout_14[0][0]
batch_normalization_15 (BatchNo	(None, 16, 16, 18)	72	concatenate_13[0][0]
activation_15 (Activation)	(None, 16, 16, 18)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 6)	972	activation_15[0][0]

dropout_15 (Dropout)	(None, 16, 16, 6)	0	conv2d_16[0][0]
concatenate_14 (Concatenate)	(None, 16, 16, 24)	0	concatenate_13[0][0] dropout_15[0][0]
batch_normalization_16 (BatchNo	(None, 16, 16, 24)	96	concatenate_14[0][0]
activation_16 (Activation)	(None, 16, 16, 24)	0	batch_normalization_16[0][0]
conv2d_17 (Conv2D)	(None, 16, 16, 6)	1296	activation_16[0][0]
dropout_16 (Dropout)	(None, 16, 16, 6)	0	conv2d_17[0][0]
concatenate_15 (Concatenate)	(None, 16, 16, 30)	0	concatenate_14[0][0] dropout_16[0][0]
batch_normalization_17 (BatchNo	(None, 16, 16, 30)	120	concatenate_15[0][0]
activation_17 (Activation)	(None, 16, 16, 30)	0	batch_normalization_17[0][0]
conv2d_18 (Conv2D)	(None, 16, 16, 6)	1620	activation_17[0][0]
dropout_17 (Dropout)	(None, 16, 16, 6)	0	conv2d_18[0][0]
concatenate_16 (Concatenate)	(None, 16, 16, 36)	0	concatenate_15[0][0] dropout_17[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 36)	144	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 36)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 6)	1944	activation_18[0][0]
dropout_18 (Dropout)	(None, 16, 16, 6)	0	conv2d_19[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 42)	0	concatenate_16[0][0] dropout_18[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 42)	168	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 42)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 6)	2268	activation_19[0][0]

dropout_19 (Dropout)	(None, 16, 16, 6)	0	conv2d_20[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 48)	0	concatenate_17[0][0] dropout_19[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 48)	192	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 48)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 6)	2592	activation_20[0][0]
dropout_20 (Dropout)	(None, 16, 16, 6)	0	conv2d_21[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 54)	0	concatenate_18[0][0] dropout_20[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 54)	216	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 54)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 6)	2916	activation_21[0][0]
dropout_21 (Dropout)	(None, 16, 16, 6)	0	conv2d_22[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 60)	0	concatenate_19[0][0] dropout_21[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 60)	240	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 60)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 6)	3240	activation_22[0][0]
dropout_22 (Dropout)	(None, 16, 16, 6)	0	conv2d_23[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 66)	0	concatenate_20[0][0] dropout_22[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 66)	264	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 66)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 6)	3564	activation_23[0][0]
dropout_23 (Dropout)	(None, 16, 16, 6)	0	conv2d_24[0][0]

concatenate_22 (Concatenate)	(None, 16, 16, 72)	0	concatenate_21[0][0] dropout_23[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 72)	288	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 72)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 6)	3888	activation_24[0][0]
dropout_24 (Dropout)	(None, 16, 16, 6)	0	conv2d_25[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 78)	0	concatenate_22[0][0] dropout_24[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 78)	312	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 78)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 6)	468	activation_25[0][0]
dropout_25 (Dropout)	(None, 16, 16, 6)	0	conv2d_26[0][0]
average_pooling2d_1 (AveragePoo	(None, 8, 8, 6)	0	dropout_25[0][0]
batch_normalization_26 (BatchNo	(None, 8, 8, 6)	24	average_pooling2d_1[0][0]
activation_26 (Activation)	(None, 8, 8, 6)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 8, 8, 6)	324	activation_26[0][0]
dropout_26 (Dropout)	(None, 8, 8, 6)	0	conv2d_27[0][0]
concatenate_24 (Concatenate)	(None, 8, 8, 12)	0	average_pooling2d_1[0][0] dropout_26[0][0]
batch_normalization_27 (BatchNo	(None, 8, 8, 12)	48	concatenate_24[0][0]
activation_27 (Activation)	(None, 8, 8, 12)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 8, 8, 6)	648	activation_27[0][0]
dropout_27 (Dropout)	(None, 8, 8, 6)	0	conv2d_28[0][0]
concatenate_25 (Concatenate)	(None, 8, 8, 18)	0	concatenate_24[0][0]

			dropout_27[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 18)	72	concatenate_25[0][0]
activation_28 (Activation)	(None, 8, 8, 18)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 6)	972	activation_28[0][0]
dropout_28 (Dropout)	(None, 8, 8, 6)	0	conv2d_29[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 24)	0	concatenate_25[0][0] dropout_28[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 24)	96	concatenate_26[0][0]
activation_29 (Activation)	(None, 8, 8, 24)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 6)	1296	activation_29[0][0]
dropout_29 (Dropout)	(None, 8, 8, 6)	0	conv2d_30[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 30)	0	concatenate_26[0][0] dropout_29[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 30)	120	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 30)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 6)	1620	activation_30[0][0]
dropout_30 (Dropout)	(None, 8, 8, 6)	0	conv2d_31[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 36)	0	concatenate_27[0][0] dropout_30[0][0]
batch_normalization_31 (BatchNo	(None, 8, 8, 36)	144	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 36)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 6)	1944	activation_31[0][0]
dropout_31 (Dropout)	(None, 8, 8, 6)	0	conv2d_32[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 42)	0	concatenate_28[0][0] dropout_31[0][0]

batch_normalization_32 (BatchNo	(None, 8, 8, 42)	168	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 42)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 6)	2268	activation_32[0][0]
dropout_32 (Dropout)	(None, 8, 8, 6)	0	conv2d_33[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 48)	0	concatenate_29[0][0] dropout_32[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 48)	192	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 48)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 6)	2592	activation_33[0][0]
dropout_33 (Dropout)	(None, 8, 8, 6)	0	conv2d_34[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 54)	0	concatenate_30[0][0] dropout_33[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 54)	216	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 54)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 6)	2916	activation_34[0][0]
dropout_34 (Dropout)	(None, 8, 8, 6)	0	conv2d_35[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 60)	0	concatenate_31[0][0] dropout_34[0][0]
batch_normalization_35 (BatchNo	(None, 8, 8, 60)	240	concatenate_32[0][0]
activation_35 (Activation)	(None, 8, 8, 60)	0	batch_normalization_35[0][0]
conv2d_36 (Conv2D)	(None, 8, 8, 6)	3240	activation_35[0][0]
dropout_35 (Dropout)	(None, 8, 8, 6)	0	conv2d_36[0][0]
concatenate_33 (Concatenate)	(None, 8, 8, 66)	0	concatenate_32[0][0] dropout_35[0][0]

batch_normalization_36 (BatchNo	(None, 8, 8, 66)	264	concatenate_33[0][0]
activation_36 (Activation)	(None, 8, 8, 66)	0	batch_normalization_36[0][0]
conv2d_37 (Conv2D)	(None, 8, 8, 6)	3564	activation_36[0][0]
dropout_36 (Dropout)	(None, 8, 8, 6)	0	conv2d_37[0][0]
concatenate_34 (Concatenate)	(None, 8, 8, 72)	0	concatenate_33[0][0] dropout_36[0][0]
batch_normalization_37 (BatchNo	(None, 8, 8, 72)	288	concatenate_34[0][0]
activation_37 (Activation)	(None, 8, 8, 72)	0	batch_normalization_37[0][0]
conv2d_38 (Conv2D)	(None, 8, 8, 6)	3888	activation_37[0][0]
dropout_37 (Dropout)	(None, 8, 8, 6)	0	conv2d_38[0][0]
concatenate_35 (Concatenate)	(None, 8, 8, 78)	0	concatenate_34[0][0] dropout_37[0][0]
batch_normalization_38 (BatchNo	(None, 8, 8, 78)	312	concatenate_35[0][0]
activation_38 (Activation)	(None, 8, 8, 78)	0	batch_normalization_38[0][0]
conv2d_39 (Conv2D)	(None, 8, 8, 6)	468	activation_38[0][0]
dropout_38 (Dropout)	(None, 8, 8, 6)	0	conv2d_39[0][0]
average_pooling2d_2 (AveragePoo	(None, 4, 4, 6)	0	dropout_38[0][0]
batch_normalization_39 (BatchNo	(None, 4, 4, 6)	24	average_pooling2d_2[0][0]
activation_39 (Activation)	(None, 4, 4, 6)	0	batch_normalization_39[0][0]
conv2d_40 (Conv2D)	(None, 4, 4, 6)	324	activation_39[0][0]
dropout_39 (Dropout)	(None, 4, 4, 6)	0	conv2d_40[0][0]
concatenate_36 (Concatenate)	(None, 4, 4, 12)	0	average_pooling2d_2[0][0] dropout_39[0][0]
batch_normalization_40 (BatchNo	(None, 4, 4, 12)	48	concatenate_36[0][0]

activation_40 (Activation)	(None, 4, 4, 12)	0	batch_normalization_40[0][0]
conv2d_41 (Conv2D)	(None, 4, 4, 6)	648	activation_40[0][0]
dropout_40 (Dropout)	(None, 4, 4, 6)	0	conv2d_41[0][0]
concatenate_37 (Concatenate)	(None, 4, 4, 18)	0	concatenate_36[0][0] dropout_40[0][0]
batch_normalization_41 (BatchNo	(None, 4, 4, 18)	72	concatenate_37[0][0]
activation_41 (Activation)	(None, 4, 4, 18)	0	batch_normalization_41[0][0]
conv2d_42 (Conv2D)	(None, 4, 4, 6)	972	activation_41[0][0]
dropout_41 (Dropout)	(None, 4, 4, 6)	0	conv2d_42[0][0]
concatenate_38 (Concatenate)	(None, 4, 4, 24)	0	concatenate_37[0][0] dropout_41[0][0]
batch_normalization_42 (BatchNo	(None, 4, 4, 24)	96	concatenate_38[0][0]
activation_42 (Activation)	(None, 4, 4, 24)	0	batch_normalization_42[0][0]
conv2d_43 (Conv2D)	(None, 4, 4, 6)	1296	activation_42[0][0]
dropout_42 (Dropout)	(None, 4, 4, 6)	0	conv2d_43[0][0]
concatenate_39 (Concatenate)	(None, 4, 4, 30)	0	concatenate_38[0][0] dropout_42[0][0]
batch_normalization_43 (BatchNo	(None, 4, 4, 30)	120	concatenate_39[0][0]
activation_43 (Activation)	(None, 4, 4, 30)	0	batch_normalization_43[0][0]
conv2d_44 (Conv2D)	(None, 4, 4, 6)	1620	activation_43[0][0]
dropout_43 (Dropout)	(None, 4, 4, 6)	0	conv2d_44[0][0]
concatenate_40 (Concatenate)	(None, 4, 4, 36)	0	concatenate_39[0][0] dropout_43[0][0]
batch_normalization_44 (BatchNo	(None, 4, 4, 36)	144	concatenate_40[0][0]
activation_44 (Activation)	(None, 4, 4, 36)	0	batch_normalization_44[0][0]

conv2d_45 (Conv2D)	(None, 4, 4, 6)	1944	activation_44[0][0]
dropout_44 (Dropout)	(None, 4, 4, 6)	0	conv2d_45[0][0]
concatenate_41 (Concatenate)	(None, 4, 4, 42)	0	concatenate_40[0][0] dropout_44[0][0]
batch_normalization_45 (BatchNo	(None, 4, 4, 42)	168	concatenate_41[0][0]
activation_45 (Activation)	(None, 4, 4, 42)	0	batch_normalization_45[0][0]
conv2d_46 (Conv2D)	(None, 4, 4, 6)	2268	activation_45[0][0]
dropout_45 (Dropout)	(None, 4, 4, 6)	0	conv2d_46[0][0]
concatenate_42 (Concatenate)	(None, 4, 4, 48)	0	concatenate_41[0][0] dropout_45[0][0]
batch_normalization_46 (BatchNo	(None, 4, 4, 48)	192	concatenate_42[0][0]
activation_46 (Activation)	(None, 4, 4, 48)	0	batch_normalization_46[0][0]
conv2d_47 (Conv2D)	(None, 4, 4, 6)	2592	activation_46[0][0]
dropout_46 (Dropout)	(None, 4, 4, 6)	0	conv2d_47[0][0]
concatenate_43 (Concatenate)	(None, 4, 4, 54)	0	concatenate_42[0][0] dropout_46[0][0]
batch_normalization_47 (BatchNo	(None, 4, 4, 54)	216	concatenate_43[0][0]
activation_47 (Activation)	(None, 4, 4, 54)	0	batch_normalization_47[0][0]
conv2d_48 (Conv2D)	(None, 4, 4, 6)	2916	activation_47[0][0]
dropout_47 (Dropout)	(None, 4, 4, 6)	0	conv2d_48[0][0]
concatenate_44 (Concatenate)	(None, 4, 4, 60)	0	concatenate_43[0][0] dropout_47[0][0]
batch_normalization_48 (BatchNo	(None, 4, 4, 60)	240	concatenate_44[0][0]
activation_48 (Activation)	(None, 4, 4, 60)	0	batch_normalization_48[0][0]

```
conv2d_49 (Conv2D)              (None, 4, 4, 6)      3240        activation_48[0][0]
__________________________________________________________________________________________________
dropout_48 (Dropout)            (None, 4, 4, 6)      0           conv2d_49[0][0]
__________________________________________________________________________________________________
concatenate_45 (Concatenate)    (None, 4, 4, 66)     0           concatenate_44[0][0]
                                                                 dropout_48[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 4, 4, 66)     264         concatenate_45[0][0]
__________________________________________________________________________________________________
activation_49 (Activation)      (None, 4, 4, 66)     0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 4, 4, 6)      3564        activation_49[0][0]
__________________________________________________________________________________________________
dropout_49 (Dropout)            (None, 4, 4, 6)      0           conv2d_50[0][0]
__________________________________________________________________________________________________
concatenate_46 (Concatenate)    (None, 4, 4, 72)     0           concatenate_45[0][0]
                                                                 dropout_49[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, 4, 4, 72)     288         concatenate_46[0][0]
__________________________________________________________________________________________________
activation_50 (Activation)      (None, 4, 4, 72)     0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 4, 4, 6)      3888        activation_50[0][0]
__________________________________________________________________________________________________
dropout_50 (Dropout)            (None, 4, 4, 6)      0           conv2d_51[0][0]
__________________________________________________________________________________________________
concatenate_47 (Concatenate)    (None, 4, 4, 78)     0           concatenate_46[0][0]
                                                                 dropout_50[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 4, 4, 78)     312         concatenate_47[0][0]
__________________________________________________________________________________________________
activation_51 (Activation)      (None, 4, 4, 78)     0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 78)     0           activation_51[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 312)          0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           3130        flatten[0][0]
==================================================================================================
Total params: 118,918
Trainable params: 114,394
Non-trainable params: 4,524
__________________________________________________________________________________________________
```

In [11]:
```
print(len(model.layers))
```

```
262
```

In [12]:
```
# determine Loss function and Optimizer
model.compile(loss='categorical_crossentropy',
              optimizer=Adam(),
              metrics=['accuracy'])
```

In [ ]:
```
# model.fit(X_train, y_train,
#                     batch_size=batch_size,
#                     epochs=epochs,
#                     verbose=1,
#                     validation_data=(X_test, y_test))
```

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.

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
from tensorflow import keras
import numpy as np
```

In [2]:
```
import tensorflow as tf
```

In [ ]:
```
# Hyperparameters
compression = 1
num_filter = 17
dropout_rate = 0
l = 13
num_classes = 10

# batch_size = 128
# num_classes = 10
# epochs = 10
# l = 40
# num_filter = 12
# compression = 0.5
# dropout_rate = 0.2
```

In [3]:
```
(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 [==============================] - 11s 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]:
```
y_train = tf.keras.utils.to_categorical(y_train, num_classes) #one hot encoding
y_test = tf.keras.utils.to_categorical(y_test, num_classes)  # one hot encoding
```

In [9]:
```
mean_X_tr = np.mean(X_train, axis=(0,1,2))
mean_X_tr
```

Out[9]:
```
array([125.30691805, 122.95039414, 113.86538318])
```

In [10]:
```
std_X_tr = np.std(X_train, axis=(0,1,2))
std_X_tr
```

Out[10]:
```
array([62.99321928, 62.08870764, 66.70489964])
```

In [11]:
```
X_train = (X_train - mean_X_tr) / std_X_tr #normalization
X_test = (X_test - mean_X_tr) / std_X_tr
```

In [12]:
```
#data augmentation
train_datagen=tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255,rotation_range=40,width_shift_range=0.2,
test_datagen=tf.keras.preprocessing.image.ImageDataGenerator(rescale=1./255)
train_generator=train_datagen.flow(X_train,y_train,batch_size=150)
test_generator=test_datagen.flow(X_test,y_test,batch_size=150)
```

In [14]:
```
#https://www.google.com/search?q=callback+function&rlz=1C1RLNS_enIN923IN923&oq=callback&aqs=chrome.1.69i57j0i433l2j0i
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

val_acc_callback=myCallback()
```

In [16]:
```
filepath='/content/best_model_h5'
```

In [17]:
```
#https://www.tensorflow.org/guide/keras/train_and_evaluate
model_best=tf.keras.callbacks.ModelCheckpoint(filepath, monitor='val_accuracy', verbose=0, save_best_only=True,save_w
```

In [18]:
```
#f= no of filter
#d=no of dense block
#t=no of transition blk

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, f,d,t):
  input= 3 * 3 * 3 * f
  dense= d * ((f * 3 * 3 * f *no_of_connection(l)) + ( 4 * f * no_of_connection(l)))
  para=t * ( (1 * 1 * f * f* ((l +1)) + 4 * f * (l+1)))
  out=((2 * 2 * f * (l+1) * 10) + 10) + (4 * f * (l+1))
  return input,dense,para,out
```

In [19]:
```
sum(total_parametres(13,17,4,3))
```

Out[19]:
```
997451.0
```

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='same')(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: "model"
__________________________________________________________________________________________________
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]
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]
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'])
```

```
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/optimizer_v2/optimizer_v2.py:375: UserWarning: The `lr
` argument is deprecated, use `learning_rate` instead.
  "The `lr` argument is deprecated, use `learning_rate` instead.")
```

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,model_best])
```

```
Epoch 1/300
  6/333 [..............................] - ETA: 3:16 - loss: 2.4512 - accuracy: 0.1500WARNING:tensorflow:Callback met
hod `on_train_batch_end` is slow compared to the batch time (batch time: 0.2401s vs `on_train_batch_end` time: 0.2998
s). Check your callbacks.
333/333 [==============================] - 258s 655ms/step - loss: 1.7313 - accuracy: 0.3656 - val_loss: 3.6667 - val
_accuracy: 0.1460
Epoch 2/300
333/333 [==============================] - 213s 638ms/step - loss: 1.3904 - accuracy: 0.4967 - val_loss: 1.4036 - val
_accuracy: 0.5090
Epoch 3/300
333/333 [==============================] - 213s 639ms/step - loss: 1.2227 - accuracy: 0.5606 - val_loss: 1.6734 - val
_accuracy: 0.4848
Epoch 4/300
333/333 [==============================] - 213s 639ms/step - loss: 1.1102 - accuracy: 0.6052 - val_loss: 1.2793 - val
_accuracy: 0.5785
Epoch 5/300
333/333 [==============================] - 213s 638ms/step - loss: 1.0269 - accuracy: 0.6350 - val_loss: 1.1171 - val
_accuracy: 0.6351
Epoch 6/300
333/333 [==============================] - 213s 640ms/step - loss: 0.9524 - accuracy: 0.6634 - val_loss: 1.0752 - val
_accuracy: 0.6388
```

```
Epoch 7/300
333/333 [==============================] - 214s 641ms/step - loss: 0.8998 - accuracy: 0.6850 - val_loss: 1.1919 - val
_accuracy: 0.6350
Epoch 8/300
333/333 [==============================] - 214s 642ms/step - loss: 0.8489 - accuracy: 0.7022 - val_loss: 1.2367 - val
_accuracy: 0.6309
Epoch 9/300
333/333 [==============================] - 214s 641ms/step - loss: 0.8114 - accuracy: 0.7175 - val_loss: 1.1188 - val
_accuracy: 0.6588
Epoch 10/300
333/333 [==============================] - 214s 642ms/step - loss: 0.7785 - accuracy: 0.7302 - val_loss: 1.2969 - val
_accuracy: 0.6364
Epoch 11/300
333/333 [==============================] - 215s 644ms/step - loss: 0.7455 - accuracy: 0.7419 - val_loss: 1.0769 - val
_accuracy: 0.6722
Epoch 12/300
333/333 [==============================] - 214s 642ms/step - loss: 0.7198 - accuracy: 0.7530 - val_loss: 0.9912 - val
_accuracy: 0.7076
Epoch 13/300
333/333 [==============================] - 214s 642ms/step - loss: 0.6958 - accuracy: 0.7572 - val_loss: 0.8645 - val
_accuracy: 0.7248
Epoch 14/300
333/333 [==============================] - 214s 641ms/step - loss: 0.6683 - accuracy: 0.7698 - val_loss: 1.1318 - val
_accuracy: 0.6663
Epoch 15/300
333/333 [==============================] - 214s 642ms/step - loss: 0.6497 - accuracy: 0.7757 - val_loss: 0.8005 - val
_accuracy: 0.7537
Epoch 16/300
333/333 [==============================] - 214s 641ms/step - loss: 0.6316 - accuracy: 0.7797 - val_loss: 0.5678 - val
_accuracy: 0.8012
Epoch 17/300
333/333 [==============================] - 214s 642ms/step - loss: 0.6153 - accuracy: 0.7869 - val_loss: 0.7315 - val
_accuracy: 0.7717
Epoch 18/300
333/333 [==============================] - 214s 642ms/step - loss: 0.5950 - accuracy: 0.7938 - val_loss: 0.6129 - val
_accuracy: 0.7932
Epoch 19/300
333/333 [==============================] - 214s 641ms/step - loss: 0.5780 - accuracy: 0.8009 - val_loss: 0.6090 - val
_accuracy: 0.7938
Epoch 20/300
333/333 [==============================] - 214s 642ms/step - loss: 0.5672 - accuracy: 0.8040 - val_loss: 0.6767 - val
_accuracy: 0.7881
Epoch 21/300
333/333 [==============================] - 214s 642ms/step - loss: 0.5526 - accuracy: 0.8065 - val_loss: 0.6222 - val
_accuracy: 0.7985
```

```
Epoch 22/300
333/333 [==============================] - 214s 642ms/step - loss: 0.5370 - accuracy: 0.8126 - val_loss: 0.8186 - val
_accuracy: 0.7424
Epoch 23/300
333/333 [==============================] - 214s 641ms/step - loss: 0.5273 - accuracy: 0.8167 - val_loss: 0.6413 - val
_accuracy: 0.7991
Epoch 24/300
333/333 [==============================] - 214s 641ms/step - loss: 0.5095 - accuracy: 0.8222 - val_loss: 0.7272 - val
_accuracy: 0.7806
Epoch 25/300
333/333 [==============================] - 214s 642ms/step - loss: 0.5112 - accuracy: 0.8223 - val_loss: 0.5860 - val
_accuracy: 0.8121
Epoch 26/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4969 - accuracy: 0.8271 - val_loss: 1.0409 - val
_accuracy: 0.7038
Epoch 27/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4870 - accuracy: 0.8311 - val_loss: 0.4931 - val
_accuracy: 0.8376
Epoch 28/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4790 - accuracy: 0.8332 - val_loss: 0.5020 - val
_accuracy: 0.8333
Epoch 29/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4672 - accuracy: 0.8380 - val_loss: 0.5794 - val
_accuracy: 0.8070
Epoch 30/300
333/333 [==============================] - 214s 641ms/step - loss: 0.4595 - accuracy: 0.8400 - val_loss: 0.7274 - val
_accuracy: 0.7794
Epoch 31/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4564 - accuracy: 0.8416 - val_loss: 0.5031 - val
_accuracy: 0.8331
Epoch 32/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4446 - accuracy: 0.8450 - val_loss: 0.4157 - val
_accuracy: 0.8600
Epoch 33/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4339 - accuracy: 0.8485 - val_loss: 0.6526 - val
_accuracy: 0.8039
Epoch 34/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4320 - accuracy: 0.8506 - val_loss: 0.5620 - val
_accuracy: 0.8197
Epoch 35/300
333/333 [==============================] - 214s 641ms/step - loss: 0.4211 - accuracy: 0.8530 - val_loss: 0.5452 - val
_accuracy: 0.8198
Epoch 36/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4133 - accuracy: 0.8576 - val_loss: 0.4698 - val
_accuracy: 0.8486
```

```
Epoch 37/300
333/333 [==============================] - 214s 640ms/step - loss: 0.4121 - accuracy: 0.8572 - val_loss: 0.6349 - val
_accuracy: 0.8012
Epoch 38/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4036 - accuracy: 0.8592 - val_loss: 0.5434 - val
_accuracy: 0.8273
Epoch 39/300
333/333 [==============================] - 213s 640ms/step - loss: 0.4042 - accuracy: 0.8592 - val_loss: 0.6283 - val
_accuracy: 0.8123
Epoch 40/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3968 - accuracy: 0.8612 - val_loss: 0.5724 - val
_accuracy: 0.8164
Epoch 41/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3833 - accuracy: 0.8662 - val_loss: 0.5449 - val
_accuracy: 0.8294
Epoch 42/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3756 - accuracy: 0.8674 - val_loss: 0.4837 - val
_accuracy: 0.8423
Epoch 43/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3771 - accuracy: 0.8685 - val_loss: 0.5237 - val
_accuracy: 0.8318
Epoch 44/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3648 - accuracy: 0.8736 - val_loss: 0.5237 - val
_accuracy: 0.8318
Epoch 45/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3665 - accuracy: 0.8738 - val_loss: 0.5043 - val
_accuracy: 0.8440
Epoch 46/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3628 - accuracy: 0.8737 - val_loss: 0.3790 - val
_accuracy: 0.8699
Epoch 47/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3596 - accuracy: 0.8747 - val_loss: 0.5454 - val
_accuracy: 0.8335
Epoch 48/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3488 - accuracy: 0.8784 - val_loss: 0.4337 - val
_accuracy: 0.8577
Epoch 49/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3419 - accuracy: 0.8807 - val_loss: 0.5194 - val
_accuracy: 0.8378
Epoch 50/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3469 - accuracy: 0.8790 - val_loss: 0.5260 - val
_accuracy: 0.8376
Epoch 51/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3357 - accuracy: 0.8830 - val_loss: 0.4212 - val
_accuracy: 0.8615
```

```
Epoch 52/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3367 - accuracy: 0.8814 - val_loss: 0.4122 - val
_accuracy: 0.8685
Epoch 53/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3347 - accuracy: 0.8839 - val_loss: 0.4523 - val
_accuracy: 0.8535
Epoch 54/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3282 - accuracy: 0.8849 - val_loss: 0.5523 - val
_accuracy: 0.8355
Epoch 55/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3243 - accuracy: 0.8851 - val_loss: 0.6252 - val
_accuracy: 0.8100
Epoch 56/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3217 - accuracy: 0.8878 - val_loss: 0.6213 - val
_accuracy: 0.8185
Epoch 57/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3171 - accuracy: 0.8904 - val_loss: 0.4586 - val
_accuracy: 0.8548
Epoch 58/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3197 - accuracy: 0.8889 - val_loss: 0.5771 - val
_accuracy: 0.8303
Epoch 59/300
333/333 [==============================] - 214s 643ms/step - loss: 0.3121 - accuracy: 0.8899 - val_loss: 0.5415 - val
_accuracy: 0.8398
Epoch 60/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3104 - accuracy: 0.8915 - val_loss: 0.4217 - val
_accuracy: 0.8749
Epoch 61/300
333/333 [==============================] - 214s 642ms/step - loss: 0.3040 - accuracy: 0.8935 - val_loss: 0.5649 - val
_accuracy: 0.8440
Epoch 62/300
333/333 [==============================] - 214s 641ms/step - loss: 0.3035 - accuracy: 0.8937 - val_loss: 0.4434 - val
_accuracy: 0.8639
Epoch 63/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2986 - accuracy: 0.8943 - val_loss: 0.7966 - val
_accuracy: 0.7912
Epoch 64/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2954 - accuracy: 0.8961 - val_loss: 0.4774 - val
_accuracy: 0.8631
Epoch 65/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2934 - accuracy: 0.8969 - val_loss: 0.4362 - val
_accuracy: 0.8668
Epoch 66/300
333/333 [==============================] - 213s 638ms/step - loss: 0.2920 - accuracy: 0.8983 - val_loss: 0.4915 - val
_accuracy: 0.8533
```

```
Epoch 67/300
333/333 [==============================] - 213s 638ms/step - loss: 0.2866 - accuracy: 0.8982 - val_loss: 0.4818 - val
_accuracy: 0.8554
Epoch 68/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2812 - accuracy: 0.9013 - val_loss: 0.4000 - val
_accuracy: 0.8752
Epoch 69/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2799 - accuracy: 0.9022 - val_loss: 0.5821 - val
_accuracy: 0.8341
Epoch 70/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2820 - accuracy: 0.9000 - val_loss: 0.4249 - val
_accuracy: 0.8683
Epoch 71/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2747 - accuracy: 0.9041 - val_loss: 0.4803 - val
_accuracy: 0.8553
Epoch 72/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2775 - accuracy: 0.9016 - val_loss: 0.3565 - val
_accuracy: 0.8855
Epoch 73/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2710 - accuracy: 0.9067 - val_loss: 0.3875 - val
_accuracy: 0.8780
Epoch 74/300
333/333 [==============================] - 214s 640ms/step - loss: 0.2706 - accuracy: 0.9034 - val_loss: 0.4517 - val
_accuracy: 0.8679
Epoch 75/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2665 - accuracy: 0.9049 - val_loss: 0.5344 - val
_accuracy: 0.8470
Epoch 76/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2676 - accuracy: 0.9070 - val_loss: 0.4086 - val
_accuracy: 0.8722
Epoch 77/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2595 - accuracy: 0.9082 - val_loss: 0.5532 - val
_accuracy: 0.8453
Epoch 78/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2634 - accuracy: 0.9071 - val_loss: 0.5763 - val
_accuracy: 0.8445
Epoch 79/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2568 - accuracy: 0.9093 - val_loss: 0.3868 - val
_accuracy: 0.8818
Epoch 80/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2566 - accuracy: 0.9101 - val_loss: 0.5036 - val
_accuracy: 0.8498
Epoch 81/300
333/333 [==============================] - 214s 640ms/step - loss: 0.2542 - accuracy: 0.9113 - val_loss: 0.4052 - val
_accuracy: 0.8806
```

```
Epoch 82/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2549 - accuracy: 0.9121 - val_loss: 0.4357 - val
_accuracy: 0.8693
Epoch 83/300
333/333 [==============================] - 214s 640ms/step - loss: 0.2496 - accuracy: 0.9121 - val_loss: 0.4870 - val
_accuracy: 0.8568
Epoch 84/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2490 - accuracy: 0.9128 - val_loss: 0.4894 - val
_accuracy: 0.8583
Epoch 85/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2480 - accuracy: 0.9127 - val_loss: 0.3680 - val
_accuracy: 0.8838
Epoch 86/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2519 - accuracy: 0.9121 - val_loss: 0.3735 - val
_accuracy: 0.8770
Epoch 87/300
333/333 [==============================] - 215s 643ms/step - loss: 0.2475 - accuracy: 0.9137 - val_loss: 0.4352 - val
_accuracy: 0.8695
Epoch 88/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2406 - accuracy: 0.9163 - val_loss: 0.4558 - val
_accuracy: 0.8628
Epoch 89/300
333/333 [==============================] - 213s 640ms/step - loss: 0.2380 - accuracy: 0.9157 - val_loss: 0.4361 - val
_accuracy: 0.8730
Epoch 90/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2374 - accuracy: 0.9160 - val_loss: 0.5117 - val
_accuracy: 0.8591
Epoch 91/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2354 - accuracy: 0.9171 - val_loss: 0.4100 - val
_accuracy: 0.8756
Epoch 92/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2330 - accuracy: 0.9171 - val_loss: 0.3640 - val
_accuracy: 0.8899
Epoch 93/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2372 - accuracy: 0.9175 - val_loss: 0.4236 - val
_accuracy: 0.8769
Epoch 94/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2299 - accuracy: 0.9177 - val_loss: 0.4129 - val
_accuracy: 0.8776
Epoch 95/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2270 - accuracy: 0.9194 - val_loss: 0.3402 - val
_accuracy: 0.8941
Epoch 96/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2223 - accuracy: 0.9208 - val_loss: 0.5140 - val
_accuracy: 0.8585
```

```
Epoch 97/300
333/333 [==============================] - 214s 642ms/step - loss: 0.2310 - accuracy: 0.9180 - val_loss: 0.4450 - val
_accuracy: 0.8741
Epoch 98/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2225 - accuracy: 0.9224 - val_loss: 0.4659 - val
_accuracy: 0.8684
Epoch 99/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2260 - accuracy: 0.9197 - val_loss: 0.3822 - val
_accuracy: 0.8842
Epoch 100/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2212 - accuracy: 0.9215 - val_loss: 0.4771 - val
_accuracy: 0.8695
Epoch 101/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2165 - accuracy: 0.9250 - val_loss: 0.4258 - val
_accuracy: 0.8770
Epoch 102/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2161 - accuracy: 0.9245 - val_loss: 0.3621 - val
_accuracy: 0.8892
Epoch 103/300
333/333 [==============================] - 215s 643ms/step - loss: 0.2178 - accuracy: 0.9247 - val_loss: 0.3896 - val
_accuracy: 0.8864
Epoch 104/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2176 - accuracy: 0.9236 - val_loss: 0.4760 - val
_accuracy: 0.8671
Epoch 105/300
333/333 [==============================] - 215s 643ms/step - loss: 0.2160 - accuracy: 0.9238 - val_loss: 0.4755 - val
_accuracy: 0.8655
Epoch 106/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2128 - accuracy: 0.9267 - val_loss: 0.4772 - val
_accuracy: 0.8627
Epoch 107/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2142 - accuracy: 0.9253 - val_loss: 0.4432 - val
_accuracy: 0.8709
Epoch 108/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2113 - accuracy: 0.9258 - val_loss: 0.5050 - val
_accuracy: 0.8620
Epoch 109/300
333/333 [==============================] - 215s 644ms/step - loss: 0.2119 - accuracy: 0.9252 - val_loss: 0.3654 - val
_accuracy: 0.8897
Epoch 110/300
333/333 [==============================] - 215s 643ms/step - loss: 0.2044 - accuracy: 0.9277 - val_loss: 0.3467 - val
_accuracy: 0.8954
Epoch 111/300
333/333 [==============================] - 214s 643ms/step - loss: 0.2106 - accuracy: 0.9250 - val_loss: 0.3967 - val
_accuracy: 0.8840
```

```
Epoch 112/300
333/333 [==============================] - 215s 643ms/step - loss: 0.2107 - accuracy: 0.9260 - val_loss: 0.3710 - val
_accuracy: 0.8865
Epoch 113/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2013 - accuracy: 0.9291 - val_loss: 0.4584 - val
_accuracy: 0.8725
Epoch 114/300
333/333 [==============================] - 214s 641ms/step - loss: 0.1999 - accuracy: 0.9291 - val_loss: 0.3674 - val
_accuracy: 0.8882
Epoch 115/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2008 - accuracy: 0.9292 - val_loss: 0.4174 - val
_accuracy: 0.8780
Epoch 116/300
333/333 [==============================] - 214s 640ms/step - loss: 0.2021 - accuracy: 0.9287 - val_loss: 0.4005 - val
_accuracy: 0.8843
Epoch 117/300
333/333 [==============================] - 214s 641ms/step - loss: 0.1974 - accuracy: 0.9309 - val_loss: 0.3492 - val
_accuracy: 0.8959
Epoch 118/300
333/333 [==============================] - 214s 641ms/step - loss: 0.2011 - accuracy: 0.9296 - val_loss: 0.3998 - val
_accuracy: 0.8857
Epoch 119/300
333/333 [==============================] - 214s 642ms/step - loss: 0.1944 - accuracy: 0.9320 - val_loss: 0.3739 - val
_accuracy: 0.8916
Epoch 120/300
333/333 [==============================] - 214s 642ms/step - loss: 0.1941 - accuracy: 0.9310 - val_loss: 0.4194 - val
_accuracy: 0.8799
Epoch 121/300
333/333 [==============================] - 214s 640ms/step - loss: 0.1943 - accuracy: 0.9304 - val_loss: 0.4715 - val
_accuracy: 0.8722
Epoch 122/300
333/333 [==============================] - 213s 639ms/step - loss: 0.1906 - accuracy: 0.9321 - val_loss: 0.3354 - val
_accuracy: 0.8994
Epoch 123/300
333/333 [==============================] - 214s 643ms/step - loss: 0.1937 - accuracy: 0.9319 - val_loss: 0.4660 - val
_accuracy: 0.8709
Epoch 124/300
333/333 [==============================] - 215s 643ms/step - loss: 0.1895 - accuracy: 0.9327 - val_loss: 0.3968 - val
_accuracy: 0.8884
Epoch 125/300
333/333 [==============================] - 214s 642ms/step - loss: 0.1891 - accuracy: 0.9329 - val_loss: 0.4497 - val
_accuracy: 0.8799
Epoch 126/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1871 - accuracy: 0.9329 - val_loss: 0.4607 - val
_accuracy: 0.8760
```

```
Epoch 127/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1832 - accuracy: 0.9358 - val_loss: 0.3836 - val
_accuracy: 0.8861
Epoch 128/300
333/333 [==============================] - 213s 639ms/step - loss: 0.1865 - accuracy: 0.9345 - val_loss: 0.4211 - val
_accuracy: 0.8833
Epoch 129/300
333/333 [==============================] - 214s 640ms/step - loss: 0.1828 - accuracy: 0.9349 - val_loss: 0.5003 - val
_accuracy: 0.8645
Epoch 130/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1855 - accuracy: 0.9340 - val_loss: 0.4291 - val
_accuracy: 0.8802
Epoch 131/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1832 - accuracy: 0.9362 - val_loss: 0.4851 - val
_accuracy: 0.8721
Epoch 132/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1849 - accuracy: 0.9347 - val_loss: 0.4820 - val
_accuracy: 0.8736
Epoch 133/300
333/333 [==============================] - 214s 640ms/step - loss: 0.1769 - accuracy: 0.9382 - val_loss: 0.4973 - val
_accuracy: 0.8673
Epoch 134/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1793 - accuracy: 0.9364 - val_loss: 0.4890 - val
_accuracy: 0.8712
Epoch 135/300
333/333 [==============================] - 213s 640ms/step - loss: 0.1803 - accuracy: 0.9354 - val_loss: 0.3394 - val
_accuracy: 0.9018

Reached 90.00% accuracy, so stopping training!!
```

Out[26]:
```
<tensorflow.python.keras.callbacks.History at 0x7fdef697b090>
```

In [27]:
```
# Test the model
score = model.evaluate(test_generator, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
67/67 [==============================] - 12s 178ms/step - loss: 0.3394 - accuracy: 0.9018
Test loss: 0.339419424533844
Test accuracy: 0.9017999768257141
```

Conclusion

- To make the accuracy 90 % let make the epoch exact 300 because as epoch increases accuracy will increase
- And also fix parameter value less than 1000000