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.

Let's import Library

In [1]:

```
import warnings
warnings.filterwarnings("ignore")
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,SGD

from tqdm import tqdm
import os
from numpy import expand_dims
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
from tensorflow.keras.callbacks import ModelCheckpoint,LearningRateScheduler,CSVLogger, Callback,ReduceLROnPlatea
u
```

In [2]:

```
import tensorflow as tf

tf.test.gpu_device_name()
```

Out[2]:

```
'/device:GPU:0'
```

In [3]:

```
from google.colab import drive
drive.mount('gdrive',force_remount=True)
```

```
Mounted at gdrive
```

In [4]:

```
# Hyperparameters
batch_size = 128
num_classes = 10
epochs = 250
l = 12
num_filter = 36
compression = 0.5
dropout_rate = 0
```

In [5]:

```
(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]
print(img_height)
print(img_width)
print(channel)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170500096/170498071 [==============================] - 11s 0us/step
170508288/170498071 [==============================] - 11s 0us/step
32
32
3
```

In [6]:

```
print(x_train.shape)
print(x_test.shape)
```

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

In [7]:

```
#convert to one hot encoding
y_train = tf.keras.utils.to_categorical(y_train,num_classes)
y_test = tf.keras.utils.to_categorical(y_test,num_classes)
```

In [8]:

```
print(y_train)
print(y_test)
```

```
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 1.]
 [0. 0. 0. ... 0. 0. 1.]
 ...
 [0. 0. 0. ... 0. 0. 1.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]]
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 [0. 0. 0. ... 0. 1. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 1. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 1. 0. 0.]]
```

In [9]:

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

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

Model with Dense Block, Transition and Output blocks

In [10]:

```
# 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 [11]:

```
num_filter = 36
l = 12
dropout_rate = 0

input = layers.Input(shape=(img_height, img_width, channel,))
First_Conv2D = layers.Conv2D(num_filter,(3,3),activation ='relu',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)
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, 36)        972         input_1[0][0]
_________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 32, 32, 36)     144         conv2d[0][0]
_________________________________________________________________________________________________
activation (Activation)      (None, 32, 32, 36)        0           batch_normalization[0][0]
_________________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 18)        5832        activation[0][0]
_________________________________________________________________________________________________
concatenate (Concatenate)    (None, 32, 32, 54)        0           conv2d[0][0]
                                                                   conv2d_1[0][0]
_________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 32, 32, 54)     216         concatenate[0][0]
_________________________________________________________________________________________________
```

activation_1 (Activation)	(None, 32, 32, 54)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 32, 32, 18)	8748	activation_1[0][0]
concatenate_1 (Concatenate)	(None, 32, 32, 72)	0	concatenate[0][0] conv2d_2[0][0]
batch_normalization_2 (BatchNor	(None, 32, 32, 72)	288	concatenate_1[0][0]
activation_2 (Activation)	(None, 32, 32, 72)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 32, 32, 18)	11664	activation_2[0][0]
concatenate_2 (Concatenate)	(None, 32, 32, 90)	0	concatenate_1[0][0] conv2d_3[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 90)	360	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 90)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 18)	14580	activation_3[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 108)	0	concatenate_2[0][0] conv2d_4[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 108)	432	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 108)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 18)	17496	activation_4[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 126)	0	concatenate_3[0][0] conv2d_5[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 126)	504	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 126)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 18)	20412	activation_5[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 144)	0	concatenate_4[0][0] conv2d_6[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 144)	576	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 144)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 18)	23328	activation_6[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 162)	0	concatenate_5[0][0] conv2d_7[0][0]
batch_normalization_7 (BatchNor	(None, 32, 32, 162)	648	concatenate_6[0][0]
activation_7 (Activation)	(None, 32, 32, 162)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 18)	26244	activation_7[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 180)	0	concatenate_6[0][0] conv2d_8[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 180)	720	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 180)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 18)	29160	activation_8[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 198)	0	concatenate_7[0][0] conv2d_9[0][0]
batch_normalization_9 (BatchNor	(None, 32, 32, 198)	792	concatenate_8[0][0]
activation_9 (Activation)	(None, 32, 32, 198)	0	batch_normalization_9[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 18)	32076	activation_9[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 216)	0	concatenate_8[0][0] conv2d_10[0][0]
batch_normalization_10 (BatchNo	(None, 32, 32, 216)	864	concatenate_9[0][0]
activation_10 (Activation)	(None, 32, 32, 216)	0	batch_normalization_10[0][0]

conv2d_11 (Conv2D)	(None, 32, 32, 18)	34992	activation_10[0][0]
concatenate_10 (Concatenate)	(None, 32, 32, 234)	0	concatenate_9[0][0] conv2d_11[0][0]
batch_normalization_11 (BatchNo	(None, 32, 32, 234)	936	concatenate_10[0][0]
activation_11 (Activation)	(None, 32, 32, 234)	0	batch_normalization_11[0][0]
conv2d_12 (Conv2D)	(None, 32, 32, 18)	37908	activation_11[0][0]
concatenate_11 (Concatenate)	(None, 32, 32, 252)	0	concatenate_10[0][0] conv2d_12[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 252)	1008	concatenate_11[0][0]
activation_12 (Activation)	(None, 32, 32, 252)	0	batch_normalization_12[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 18)	4536	activation_12[0][0]
average_pooling2d (AveragePooli	(None, 16, 16, 18)	0	conv2d_13[0][0]
batch_normalization_13 (BatchNo	(None, 16, 16, 18)	72	average_pooling2d[0][0]
activation_13 (Activation)	(None, 16, 16, 18)	0	batch_normalization_13[0][0]
conv2d_14 (Conv2D)	(None, 16, 16, 18)	2916	activation_13[0][0]
concatenate_12 (Concatenate)	(None, 16, 16, 36)	0	average_pooling2d[0][0] conv2d_14[0][0]
batch_normalization_14 (BatchNo	(None, 16, 16, 36)	144	concatenate_12[0][0]
activation_14 (Activation)	(None, 16, 16, 36)	0	batch_normalization_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 18)	5832	activation_14[0][0]
concatenate_13 (Concatenate)	(None, 16, 16, 54)	0	concatenate_12[0][0] conv2d_15[0][0]
batch_normalization_15 (BatchNo	(None, 16, 16, 54)	216	concatenate_13[0][0]
activation_15 (Activation)	(None, 16, 16, 54)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 18)	8748	activation_15[0][0]
concatenate_14 (Concatenate)	(None, 16, 16, 72)	0	concatenate_13[0][0] conv2d_16[0][0]
batch_normalization_16 (BatchNo	(None, 16, 16, 72)	288	concatenate_14[0][0]
activation_16 (Activation)	(None, 16, 16, 72)	0	batch_normalization_16[0][0]
conv2d_17 (Conv2D)	(None, 16, 16, 18)	11664	activation_16[0][0]
concatenate_15 (Concatenate)	(None, 16, 16, 90)	0	concatenate_14[0][0] conv2d_17[0][0]
batch_normalization_17 (BatchNo	(None, 16, 16, 90)	360	concatenate_15[0][0]
activation_17 (Activation)	(None, 16, 16, 90)	0	batch_normalization_17[0][0]
conv2d_18 (Conv2D)	(None, 16, 16, 18)	14580	activation_17[0][0]
concatenate_16 (Concatenate)	(None, 16, 16, 108)	0	concatenate_15[0][0] conv2d_18[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 108)	432	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 108)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 18)	17496	activation_18[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 126)	0	concatenate_16[0][0] conv2d_19[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 126)	504	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 126)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 18)	20412	activation_19[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 144)	0	concatenate_17[0][0]

			conv2d_20[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 144)	576	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 144)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 18)	23328	activation_20[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 162)	0	concatenate_18[0][0] conv2d_21[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 162)	648	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 162)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 18)	26244	activation_21[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 180)	0	concatenate_19[0][0] conv2d_22[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 180)	720	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 180)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 18)	29160	activation_22[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 198)	0	concatenate_20[0][0] conv2d_23[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 198)	792	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 198)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 18)	32076	activation_23[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 216)	0	concatenate_21[0][0] conv2d_24[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 216)	864	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 216)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 18)	34992	activation_24[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 234)	0	concatenate_22[0][0] conv2d_25[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 234)	936	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 234)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 18)	4212	activation_25[0][0]
average_pooling2d_1 (AveragePoo	(None, 8, 8, 18)	0	conv2d_26[0][0]
batch_normalization_26 (BatchNo	(None, 8, 8, 18)	72	average_pooling2d_1[0][0]
activation_26 (Activation)	(None, 8, 8, 18)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 8, 8, 18)	2916	activation_26[0][0]
concatenate_24 (Concatenate)	(None, 8, 8, 36)	0	average_pooling2d_1[0][0] conv2d_27[0][0]
batch_normalization_27 (BatchNo	(None, 8, 8, 36)	144	concatenate_24[0][0]
activation_27 (Activation)	(None, 8, 8, 36)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 8, 8, 18)	5832	activation_27[0][0]
concatenate_25 (Concatenate)	(None, 8, 8, 54)	0	concatenate_24[0][0] conv2d_28[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 54)	216	concatenate_25[0][0]
activation_28 (Activation)	(None, 8, 8, 54)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 18)	8748	activation_28[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 72)	0	concatenate_25[0][0] conv2d_29[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 72)	288	concatenate_26[0][0]

activation_29 (Activation)	(None, 8, 8, 72)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 18)	11664	activation_29[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 90)	0	concatenate_26[0][0] conv2d_30[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 90)	360	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 90)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 18)	14580	activation_30[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 108)	0	concatenate_27[0][0] conv2d_31[0][0]
batch_normalization_31 (BatchNo	(None, 8, 8, 108)	432	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 108)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 18)	17496	activation_31[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 126)	0	concatenate_28[0][0] conv2d_32[0][0]
batch_normalization_32 (BatchNo	(None, 8, 8, 126)	504	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 126)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 18)	20412	activation_32[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 144)	0	concatenate_29[0][0] conv2d_33[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 144)	576	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 144)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 18)	23328	activation_33[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 162)	0	concatenate_30[0][0] conv2d_34[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 162)	648	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 162)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 18)	26244	activation_34[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 180)	0	concatenate_31[0][0] conv2d_35[0][0]
batch_normalization_35 (BatchNo	(None, 8, 8, 180)	720	concatenate_32[0][0]
activation_35 (Activation)	(None, 8, 8, 180)	0	batch_normalization_35[0][0]
conv2d_36 (Conv2D)	(None, 8, 8, 18)	29160	activation_35[0][0]
concatenate_33 (Concatenate)	(None, 8, 8, 198)	0	concatenate_32[0][0] conv2d_36[0][0]
batch_normalization_36 (BatchNo	(None, 8, 8, 198)	792	concatenate_33[0][0]
activation_36 (Activation)	(None, 8, 8, 198)	0	batch_normalization_36[0][0]
conv2d_37 (Conv2D)	(None, 8, 8, 18)	32076	activation_36[0][0]
concatenate_34 (Concatenate)	(None, 8, 8, 216)	0	concatenate_33[0][0] conv2d_37[0][0]
batch_normalization_37 (BatchNo	(None, 8, 8, 216)	864	concatenate_34[0][0]
activation_37 (Activation)	(None, 8, 8, 216)	0	batch_normalization_37[0][0]
conv2d_38 (Conv2D)	(None, 8, 8, 18)	34992	activation_37[0][0]
concatenate_35 (Concatenate)	(None, 8, 8, 234)	0	concatenate_34[0][0] conv2d_38[0][0]
batch_normalization_38 (BatchNo	(None, 8, 8, 234)	936	concatenate_35[0][0]
activation_38 (Activation)	(None, 8, 8, 234)	0	batch_normalization_38[0][0]

conv2d_39 (Conv2D)	(None, 8, 8, 18)	4212	activation_38[0][0]
average_pooling2d_2 (AveragePoo	(None, 4, 4, 18)	0	conv2d_39[0][0]
batch_normalization_39 (BatchNo	(None, 4, 4, 18)	72	average_pooling2d_2[0][0]
activation_39 (Activation)	(None, 4, 4, 18)	0	batch_normalization_39[0][0]
conv2d_40 (Conv2D)	(None, 4, 4, 18)	2916	activation_39[0][0]
concatenate_36 (Concatenate)	(None, 4, 4, 36)	0	average_pooling2d_2[0][0] conv2d_40[0][0]
batch_normalization_40 (BatchNo	(None, 4, 4, 36)	144	concatenate_36[0][0]
activation_40 (Activation)	(None, 4, 4, 36)	0	batch_normalization_40[0][0]
conv2d_41 (Conv2D)	(None, 4, 4, 18)	5832	activation_40[0][0]
concatenate_37 (Concatenate)	(None, 4, 4, 54)	0	concatenate_36[0][0] conv2d_41[0][0]
batch_normalization_41 (BatchNo	(None, 4, 4, 54)	216	concatenate_37[0][0]
activation_41 (Activation)	(None, 4, 4, 54)	0	batch_normalization_41[0][0]
conv2d_42 (Conv2D)	(None, 4, 4, 18)	8748	activation_41[0][0]
concatenate_38 (Concatenate)	(None, 4, 4, 72)	0	concatenate_37[0][0] conv2d_42[0][0]
batch_normalization_42 (BatchNo	(None, 4, 4, 72)	288	concatenate_38[0][0]
activation_42 (Activation)	(None, 4, 4, 72)	0	batch_normalization_42[0][0]
conv2d_43 (Conv2D)	(None, 4, 4, 18)	11664	activation_42[0][0]
concatenate_39 (Concatenate)	(None, 4, 4, 90)	0	concatenate_38[0][0] conv2d_43[0][0]
batch_normalization_43 (BatchNo	(None, 4, 4, 90)	360	concatenate_39[0][0]
activation_43 (Activation)	(None, 4, 4, 90)	0	batch_normalization_43[0][0]
conv2d_44 (Conv2D)	(None, 4, 4, 18)	14580	activation_43[0][0]
concatenate_40 (Concatenate)	(None, 4, 4, 108)	0	concatenate_39[0][0] conv2d_44[0][0]
batch_normalization_44 (BatchNo	(None, 4, 4, 108)	432	concatenate_40[0][0]
activation_44 (Activation)	(None, 4, 4, 108)	0	batch_normalization_44[0][0]
conv2d_45 (Conv2D)	(None, 4, 4, 18)	17496	activation_44[0][0]
concatenate_41 (Concatenate)	(None, 4, 4, 126)	0	concatenate_40[0][0] conv2d_45[0][0]
batch_normalization_45 (BatchNo	(None, 4, 4, 126)	504	concatenate_41[0][0]
activation_45 (Activation)	(None, 4, 4, 126)	0	batch_normalization_45[0][0]
conv2d_46 (Conv2D)	(None, 4, 4, 18)	20412	activation_45[0][0]
concatenate_42 (Concatenate)	(None, 4, 4, 144)	0	concatenate_41[0][0] conv2d_46[0][0]
batch_normalization_46 (BatchNo	(None, 4, 4, 144)	576	concatenate_42[0][0]
activation_46 (Activation)	(None, 4, 4, 144)	0	batch_normalization_46[0][0]
conv2d_47 (Conv2D)	(None, 4, 4, 18)	23328	activation_46[0][0]
concatenate_43 (Concatenate)	(None, 4, 4, 162)	0	concatenate_42[0][0] conv2d_47[0][0]
batch_normalization_47 (BatchNo	(None, 4, 4, 162)	648	concatenate_43[0][0]
activation_47 (Activation)	(None, 4, 4, 162)	0	batch_normalization_47[0][0]
conv2d_48 (Conv2D)	(None, 4, 4, 18)	26244	activation_47[0][0]

```
concatenate_44 (Concatenate)    (None, 4, 4, 180)    0           concatenate_43[0][0]
                                                                 conv2d_48[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, 4, 4, 180)    720         concatenate_44[0][0]
__________________________________________________________________________________________________
activation_48 (Activation)      (None, 4, 4, 180)    0           batch_normalization_48[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, 4, 4, 18)     29160       activation_48[0][0]
__________________________________________________________________________________________________
concatenate_45 (Concatenate)    (None, 4, 4, 198)    0           concatenate_44[0][0]
                                                                 conv2d_49[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 4, 4, 198)    792         concatenate_45[0][0]
__________________________________________________________________________________________________
activation_49 (Activation)      (None, 4, 4, 198)    0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 4, 4, 18)     32076       activation_49[0][0]
__________________________________________________________________________________________________
concatenate_46 (Concatenate)    (None, 4, 4, 216)    0           concatenate_45[0][0]
                                                                 conv2d_50[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, 4, 4, 216)    864         concatenate_46[0][0]
__________________________________________________________________________________________________
activation_50 (Activation)      (None, 4, 4, 216)    0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 4, 4, 18)     34992       activation_50[0][0]
__________________________________________________________________________________________________
concatenate_47 (Concatenate)    (None, 4, 4, 234)    0           concatenate_46[0][0]
                                                                 conv2d_51[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 4, 4, 234)    936         concatenate_47[0][0]
__________________________________________________________________________________________________
activation_51 (Activation)      (None, 4, 4, 234)    0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 234)    0           activation_51[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 936)          0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           9370        flatten[0][0]
==================================================================================================
Total params: 995,230
Trainable params: 981,658
Non-trainable params: 13,572
__________________________________________________________________________________________________
```

Image Augmentation

In [12]:

```
datagen = ImageDataGenerator(rotation_range=15,horizontal_flip=True,width_shift_range=0.1,height_shift_range=0.1,
zoom_range=0.2,shear_range=15)

datagen.fit(x_train)
```

In [13]:

```
checkpoint = ModelCheckpoint('gdrive/My Drive/cnnoncifar/cifar10_model_save/model-{epoch:03d}-{accuracy:03f}-{val
_accuracy:03f}.h5',
                             monitor='val_accuracy')

csvlog = CSVLogger('gdrive/My Drive/cnnoncifar/csvlog.h5', append = True)
```

In [14]:

```
model.compile(loss= "categorical_crossentropy",optimizer=SGD(0.1,momentum = 0.7),metrics=['accuracy'])
```

In [ ]:

```
path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size,
                    epochs = 30, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '30epochs.h5'))
```

```
Epoch 1/30
390/390 [==============================] - 251s 545ms/step - loss: 1.8326 - accuracy: 0.3547 - val_l
oss: 1.8010 - val_accuracy: 0.3592
Epoch 2/30
390/390 [==============================] - 207s 530ms/step - loss: 1.4308 - accuracy: 0.4819 - val_l
oss: 1.3470 - val_accuracy: 0.5154
Epoch 3/30
390/390 [==============================] - 206s 526ms/step - loss: 1.2324 - accuracy: 0.5565 - val_l
oss: 1.4475 - val_accuracy: 0.5375
```

```
Epoch 4/30
390/390 [==============================] - 206s 527ms/step - loss: 1.0704 - accuracy: 0.6174 - val_l
oss: 1.0906 - val_accuracy: 0.6205
Epoch 5/30
390/390 [==============================] - 206s 527ms/step - loss: 0.9534 - accuracy: 0.6629 - val_l
oss: 1.1539 - val_accuracy: 0.6232
Epoch 6/30
390/390 [==============================] - 206s 527ms/step - loss: 0.8611 - accuracy: 0.6962 - val_l
oss: 1.1443 - val_accuracy: 0.6244
Epoch 7/30
390/390 [==============================] - 206s 527ms/step - loss: 0.7931 - accuracy: 0.7219 - val_l
oss: 0.8913 - val_accuracy: 0.6984
Epoch 8/30
390/390 [==============================] - 206s 527ms/step - loss: 0.7388 - accuracy: 0.7381 - val_l
oss: 0.9972 - val_accuracy: 0.6862
Epoch 9/30
390/390 [==============================] - 206s 527ms/step - loss: 0.6878 - accuracy: 0.7569 - val_l
oss: 1.3510 - val_accuracy: 0.6267
Epoch 10/30
390/390 [==============================] - 205s 525ms/step - loss: 0.6525 - accuracy: 0.7713 - val_l
oss: 1.0587 - val_accuracy: 0.6746
Epoch 11/30
390/390 [==============================] - 211s 540ms/step - loss: 0.6180 - accuracy: 0.7843 - val_l
oss: 0.7362 - val_accuracy: 0.7469
Epoch 12/30
390/390 [==============================] - 205s 525ms/step - loss: 0.5882 - accuracy: 0.7936 - val_l
oss: 0.6735 - val_accuracy: 0.7752
Epoch 13/30
390/390 [==============================] - 205s 524ms/step - loss: 0.5646 - accuracy: 0.8017 - val_l
oss: 0.6184 - val_accuracy: 0.7900
Epoch 14/30
390/390 [==============================] - 205s 525ms/step - loss: 0.5453 - accuracy: 0.8090 - val_l
oss: 0.7838 - val_accuracy: 0.7582
Epoch 15/30
390/390 [==============================] - 205s 524ms/step - loss: 0.5230 - accuracy: 0.8180 - val_l
oss: 0.6884 - val_accuracy: 0.7769
Epoch 16/30
390/390 [==============================] - 205s 525ms/step - loss: 0.5081 - accuracy: 0.8224 - val_l
oss: 0.6058 - val_accuracy: 0.7987
Epoch 17/30
390/390 [==============================] - 205s 525ms/step - loss: 0.4831 - accuracy: 0.8308 - val_l
oss: 0.5384 - val_accuracy: 0.8203
Epoch 18/30
390/390 [==============================] - 206s 527ms/step - loss: 0.4688 - accuracy: 0.8366 - val_l
oss: 0.4834 - val_accuracy: 0.8315
Epoch 19/30
390/390 [==============================] - 206s 526ms/step - loss: 0.4539 - accuracy: 0.8423 - val_l
oss: 0.5028 - val_accuracy: 0.8315
Epoch 20/30
390/390 [==============================] - 206s 527ms/step - loss: 0.4444 - accuracy: 0.8451 - val_l
oss: 0.5023 - val_accuracy: 0.8356
Epoch 21/30
390/390 [==============================] - 207s 530ms/step - loss: 0.4307 - accuracy: 0.8510 - val_l
oss: 0.6200 - val_accuracy: 0.8125
Epoch 22/30
390/390 [==============================] - 207s 530ms/step - loss: 0.4147 - accuracy: 0.8546 - val_l
oss: 0.5028 - val_accuracy: 0.8345
Epoch 23/30
390/390 [==============================] - 207s 530ms/step - loss: 0.4071 - accuracy: 0.8576 - val_l
oss: 0.4947 - val_accuracy: 0.8331
Epoch 24/30
390/390 [==============================] - 204s 523ms/step - loss: 0.3948 - accuracy: 0.8626 - val_l
oss: 0.5093 - val_accuracy: 0.8371
Epoch 25/30
390/390 [==============================] - 204s 523ms/step - loss: 0.3889 - accuracy: 0.8647 - val_l
oss: 0.4296 - val_accuracy: 0.8558
Epoch 26/30
390/390 [==============================] - 204s 523ms/step - loss: 0.3801 - accuracy: 0.8690 - val_l
oss: 0.5406 - val_accuracy: 0.8243
Epoch 27/30
390/390 [==============================] - 204s 523ms/step - loss: 0.3682 - accuracy: 0.8715 - val_l
oss: 0.5834 - val_accuracy: 0.8229
Epoch 28/30
390/390 [==============================] - 211s 539ms/step - loss: 0.3587 - accuracy: 0.8752 - val_l
oss: 0.5762 - val_accuracy: 0.8197
Epoch 29/30
390/390 [==============================] - 205s 526ms/step - loss: 0.3500 - accuracy: 0.8776 - val_l
oss: 0.5836 - val_accuracy: 0.8196
Epoch 30/30
390/390 [==============================] - 207s 530ms/step - loss: 0.3424 - accuracy: 0.8792 - val_l
oss: 0.4844 - val_accuracy: 0.8401
```

In [ ]:

```
model.load_weights('/content/gdrive/MyDrive/cnnoncifar/cifar10_model_save/model-030-0.879160-0.840100.h5')
path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size, in
itial_epoch = 30,
                    epochs = 60, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '60epochs.h5'))
```

```
Epoch 31/60
390/390 [==============================] - 205s 523ms/step - loss: 0.3386 - accuracy: 0.8813 - val_l
oss: 0.5455 - val_accuracy: 0.8327
Epoch 32/60
390/390 [==============================] - 204s 523ms/step - loss: 0.3281 - accuracy: 0.8853 - val_l
oss: 0.4917 - val_accuracy: 0.8461
Epoch 33/60
390/390 [==============================] - 204s 523ms/step - loss: 0.3218 - accuracy: 0.8858 - val_l
oss: 0.4398 - val_accuracy: 0.8573
Epoch 34/60
390/390 [==============================] - 204s 523ms/step - loss: 0.3181 - accuracy: 0.8883 - val_l
oss: 0.5518 - val_accuracy: 0.8365
Epoch 35/60
390/390 [==============================] - 205s 524ms/step - loss: 0.3129 - accuracy: 0.8899 - val_l
oss: 0.5829 - val_accuracy: 0.8247
Epoch 36/60
390/390 [==============================] - 205s 524ms/step - loss: 0.3045 - accuracy: 0.8934 - val_l
oss: 0.3996 - val_accuracy: 0.8678
Epoch 37/60
390/390 [==============================] - 205s 524ms/step - loss: 0.3002 - accuracy: 0.8947 - val_l
oss: 0.6270 - val_accuracy: 0.8200
Epoch 38/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2954 - accuracy: 0.8962 - val_l
oss: 0.4378 - val_accuracy: 0.8638
Epoch 39/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2892 - accuracy: 0.8991 - val_l
oss: 0.4421 - val_accuracy: 0.8563
Epoch 40/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2856 - accuracy: 0.8996 - val_l
oss: 0.4193 - val_accuracy: 0.8676
Epoch 41/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2792 - accuracy: 0.9014 - val_l
oss: 0.4599 - val_accuracy: 0.8582
Epoch 42/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2695 - accuracy: 0.9039 - val_l
oss: 0.4209 - val_accuracy: 0.8711
Epoch 43/60
390/390 [==============================] - 207s 530ms/step - loss: 0.2662 - accuracy: 0.9068 - val_l
oss: 0.4567 - val_accuracy: 0.8629
Epoch 44/60
390/390 [==============================] - 208s 531ms/step - loss: 0.2658 - accuracy: 0.9077 - val_l
oss: 0.3970 - val_accuracy: 0.8753
Epoch 45/60
390/390 [==============================] - 207s 531ms/step - loss: 0.2599 - accuracy: 0.9086 - val_l
oss: 0.4761 - val_accuracy: 0.8521
Epoch 46/60
390/390 [==============================] - 208s 531ms/step - loss: 0.2550 - accuracy: 0.9099 - val_l
oss: 0.4949 - val_accuracy: 0.8509
Epoch 47/60
390/390 [==============================] - 207s 530ms/step - loss: 0.2586 - accuracy: 0.9098 - val_l
oss: 0.5405 - val_accuracy: 0.8490
Epoch 48/60
390/390 [==============================] - 208s 531ms/step - loss: 0.2472 - accuracy: 0.9126 - val_l
oss: 0.4816 - val_accuracy: 0.8555
Epoch 49/60
390/390 [==============================] - 207s 531ms/step - loss: 0.2466 - accuracy: 0.9132 - val_l
oss: 0.4624 - val_accuracy: 0.8631
Epoch 50/60
390/390 [==============================] - 208s 531ms/step - loss: 0.2386 - accuracy: 0.9161 - val_l
oss: 0.4368 - val_accuracy: 0.8678
Epoch 51/60
390/390 [==============================] - 207s 531ms/step - loss: 0.2373 - accuracy: 0.9162 - val_l
oss: 0.4121 - val_accuracy: 0.8686
Epoch 52/60
390/390 [==============================] - 208s 531ms/step - loss: 0.2331 - accuracy: 0.9187 - val_l
oss: 0.4194 - val_accuracy: 0.8688
Epoch 53/60
390/390 [==============================] - 206s 526ms/step - loss: 0.2302 - accuracy: 0.9198 - val_l
oss: 0.4925 - val_accuracy: 0.8592
Epoch 54/60
390/390 [==============================] - 204s 523ms/step - loss: 0.2281 - accuracy: 0.9197 - val_l
oss: 0.4327 - val_accuracy: 0.8694
Epoch 55/60
390/390 [==============================] - 204s 523ms/step - loss: 0.2223 - accuracy: 0.9222 - val_l
```

```
oss: 0.4268 - val_accuracy: 0.8698
Epoch 56/60
390/390 [==============================] - 211s 539ms/step - loss: 0.2243 - accuracy: 0.9210 - val_l
oss: 0.4026 - val_accuracy: 0.8775
Epoch 57/60
390/390 [==============================] - 205s 524ms/step - loss: 0.2190 - accuracy: 0.9222 - val_l
oss: 0.3832 - val_accuracy: 0.8844
Epoch 58/60
390/390 [==============================] - 207s 530ms/step - loss: 0.2100 - accuracy: 0.9254 - val_l
oss: 0.4478 - val_accuracy: 0.8747
Epoch 59/60
375/390 [===========================>..] - ETA: 7s - loss: 0.2123 - accuracy: 0.9245
```

In [ ]:

```
model.load_weights('/content/gdrive/MyDrive/cnnoncifar/cifar10_model_save/model-058-0.925400-0.874700.h5')

path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size, in
itial_epoch = 58,
                    epochs = 100, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '100epochs.h5'))
```

```
Epoch 59/100
390/390 [==============================] - 266s 582ms/step - loss: 0.2071 - accuracy: 0.9270 - val_l
oss: 0.4066 - val_accuracy: 0.8796
Epoch 60/100
390/390 [==============================] - 217s 554ms/step - loss: 0.2085 - accuracy: 0.9263 - val_l
oss: 0.4008 - val_accuracy: 0.8818
Epoch 61/100
390/390 [==============================] - 217s 554ms/step - loss: 0.2085 - accuracy: 0.9263 - val_l
oss: 0.3849 - val_accuracy: 0.8863
Epoch 62/100
390/390 [==============================] - 217s 555ms/step - loss: 0.2000 - accuracy: 0.9287 - val_l
oss: 0.4603 - val_accuracy: 0.8674
Epoch 63/100
390/390 [==============================] - 217s 555ms/step - loss: 0.2026 - accuracy: 0.9285 - val_l
oss: 0.5404 - val_accuracy: 0.8557
Epoch 64/100
390/390 [==============================] - 217s 555ms/step - loss: 0.2030 - accuracy: 0.9277 - val_l
oss: 0.3928 - val_accuracy: 0.8810
Epoch 65/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1926 - accuracy: 0.9313 - val_l
oss: 0.6280 - val_accuracy: 0.8372
Epoch 66/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1964 - accuracy: 0.9295 - val_l
oss: 0.4520 - val_accuracy: 0.8749
Epoch 67/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1919 - accuracy: 0.9320 - val_l
oss: 0.4563 - val_accuracy: 0.8738
Epoch 68/100
390/390 [==============================] - 217s 556ms/step - loss: 0.1900 - accuracy: 0.9335 - val_l
oss: 0.3898 - val_accuracy: 0.8866
Epoch 69/100
390/390 [==============================] - 217s 556ms/step - loss: 0.1830 - accuracy: 0.9345 - val_l
oss: 0.6411 - val_accuracy: 0.8400
Epoch 70/100
390/390 [==============================] - 217s 556ms/step - loss: 0.1829 - accuracy: 0.9349 - val_l
oss: 0.4010 - val_accuracy: 0.8852
Epoch 71/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1823 - accuracy: 0.9343 - val_l
oss: 0.4960 - val_accuracy: 0.8622
Epoch 72/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1764 - accuracy: 0.9376 - val_l
oss: 0.4266 - val_accuracy: 0.8808
Epoch 73/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1781 - accuracy: 0.9377 - val_l
oss: 0.4159 - val_accuracy: 0.8790
Epoch 74/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1762 - accuracy: 0.9378 - val_l
oss: 0.5061 - val_accuracy: 0.8619
Epoch 75/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1778 - accuracy: 0.9375 - val_l
oss: 0.4369 - val_accuracy: 0.8729
Epoch 76/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1716 - accuracy: 0.9403 - val_l
oss: 0.4158 - val_accuracy: 0.8829
Epoch 77/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1652 - accuracy: 0.9409 - val_l
oss: 0.4428 - val_accuracy: 0.8756
Epoch 78/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1666 - accuracy: 0.9410 - val_l
oss: 0.4739 - val_accuracy: 0.8712
```

```
Epoch 79/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1639 - accuracy: 0.9429 - val_l
oss: 0.4994 - val_accuracy: 0.8692
Epoch 80/100
390/390 [==============================] - 217s 556ms/step - loss: 0.1618 - accuracy: 0.9426 - val_l
oss: 0.4694 - val_accuracy: 0.8718
Epoch 81/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1619 - accuracy: 0.9419 - val_l
oss: 0.4469 - val_accuracy: 0.8826
Epoch 82/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1628 - accuracy: 0.9431 - val_l
oss: 0.3872 - val_accuracy: 0.8921
Epoch 83/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1607 - accuracy: 0.9427 - val_l
oss: 0.3881 - val_accuracy: 0.8884
Epoch 84/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1560 - accuracy: 0.9451 - val_l
oss: 0.3622 - val_accuracy: 0.8983
Epoch 85/100
390/390 [==============================] - 217s 555ms/step - loss: 0.1548 - accuracy: 0.9441 - val_l
oss: 0.4590 - val_accuracy: 0.8754
Epoch 86/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1541 - accuracy: 0.9462 - val_l
oss: 0.4220 - val_accuracy: 0.8835
Epoch 87/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1516 - accuracy: 0.9457 - val_l
oss: 0.3835 - val_accuracy: 0.8947
Epoch 88/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1472 - accuracy: 0.9477 - val_l
oss: 0.4406 - val_accuracy: 0.8812
Epoch 89/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1490 - accuracy: 0.9475 - val_l
oss: 0.4865 - val_accuracy: 0.8745
Epoch 90/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1507 - accuracy: 0.9460 - val_l
oss: 0.4925 - val_accuracy: 0.8738
Epoch 91/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1445 - accuracy: 0.9490 - val_l
oss: 0.3943 - val_accuracy: 0.8922
Epoch 92/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1472 - accuracy: 0.9465 - val_l
oss: 0.3601 - val_accuracy: 0.8974
Epoch 93/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1403 - accuracy: 0.9500 - val_l
oss: 0.3710 - val_accuracy: 0.8985
Epoch 94/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1385 - accuracy: 0.9508 - val_l
oss: 0.4378 - val_accuracy: 0.8852
Epoch 95/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1395 - accuracy: 0.9515 - val_l
oss: 0.4358 - val_accuracy: 0.8829
Epoch 96/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1401 - accuracy: 0.9506 - val_l
oss: 0.3969 - val_accuracy: 0.8921
Epoch 97/100
390/390 [==============================] - 222s 569ms/step - loss: 0.1361 - accuracy: 0.9518 - val_l
oss: 0.5035 - val_accuracy: 0.8722
Epoch 98/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1349 - accuracy: 0.9518 - val_l
oss: 0.3865 - val_accuracy: 0.8972
Epoch 99/100
390/390 [==============================] - 216s 553ms/step - loss: 0.1356 - accuracy: 0.9515 - val_l
oss: 0.3974 - val_accuracy: 0.8908
Epoch 100/100
390/390 [==============================] - 216s 554ms/step - loss: 0.1337 - accuracy: 0.9516 - val_l
oss: 0.4682 - val_accuracy: 0.8825
```

In [ ]:

```
model.load_weights('/content/gdrive/MyDrive/cnnoncifar/cifar10_model_save/model-100-0.951640-0.882500.h5')

path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size, in
itial_epoch = 100,
                    epochs = 170, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '170epochs.h5'))
```

```
Epoch 101/170
390/390 [==============================] - 259s 566ms/step - loss: 0.1317 - accuracy: 0.9523 - val_l
oss: 0.3763 - val_accuracy: 0.8964
Epoch 102/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1298 - accuracy: 0.9544 - val_l
oss: 0.4371 - val_accuracy: 0.8885
Epoch 103/170
390/390 [==============================] - 215s 549ms/step - loss: 0.1334 - accuracy: 0.9531 - val_l
oss: 0.4418 - val_accuracy: 0.8836
Epoch 104/170
390/390 [==============================] - 214s 548ms/step - loss: 0.1302 - accuracy: 0.9527 - val_l
oss: 0.4042 - val_accuracy: 0.8973
Epoch 105/170
390/390 [==============================] - 214s 548ms/step - loss: 0.1237 - accuracy: 0.9557 - val_l
oss: 0.4946 - val_accuracy: 0.8764
Epoch 106/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1227 - accuracy: 0.9561 - val_l
oss: 0.4145 - val_accuracy: 0.8940
Epoch 107/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1220 - accuracy: 0.9577 - val_l
oss: 0.3664 - val_accuracy: 0.9007
Epoch 108/170
390/390 [==============================] - 221s 564ms/step - loss: 0.1277 - accuracy: 0.9542 - val_l
oss: 0.3625 - val_accuracy: 0.8996
Epoch 109/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1208 - accuracy: 0.9578 - val_l
oss: 0.3863 - val_accuracy: 0.9003
Epoch 110/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1224 - accuracy: 0.9561 - val_l
oss: 0.4434 - val_accuracy: 0.8861
Epoch 111/170
390/390 [==============================] - 215s 549ms/step - loss: 0.1206 - accuracy: 0.9580 - val_l
oss: 0.4337 - val_accuracy: 0.8891
Epoch 112/170
390/390 [==============================] - 214s 549ms/step - loss: 0.1180 - accuracy: 0.9586 - val_l
oss: 0.3978 - val_accuracy: 0.8922
Epoch 113/170
390/390 [==============================] - 214s 548ms/step - loss: 0.1175 - accuracy: 0.9576 - val_l
oss: 0.3961 - val_accuracy: 0.8952
Epoch 114/170
390/390 [==============================] - 220s 564ms/step - loss: 0.1152 - accuracy: 0.9593 - val_l
oss: 0.4485 - val_accuracy: 0.8886
Epoch 115/170
390/390 [==============================] - 215s 550ms/step - loss: 0.1207 - accuracy: 0.9570 - val_l
oss: 0.4182 - val_accuracy: 0.8960
Epoch 116/170
390/390 [==============================] - 214s 548ms/step - loss: 0.1156 - accuracy: 0.9596 - val_l
oss: 0.4677 - val_accuracy: 0.8847
Epoch 117/170
390/390 [==============================] - 214s 548ms/step - loss: 0.1180 - accuracy: 0.9579 - val_l
oss: 0.5786 - val_accuracy: 0.8594
Epoch 118/170
278/390 [====================>.........] - ETA: 57s - loss: 0.1132 - accuracy: 0.9594
```

In [15]:

```
model.load_weights('/content/gdrive/MyDrive/cnnoncifar/cifar10_model_save/model-117-0.957860-0.859400.h5')
path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size, in
itial_epoch = 117,
                    epochs = 170, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '170epochs.h5'))
```

```
Epoch 118/170
390/390 [==============================] - 261s 571ms/step - loss: 0.1101 - accuracy: 0.9606 - val_l
oss: 0.4266 - val_accuracy: 0.8918
Epoch 119/170
390/390 [==============================] - 217s 556ms/step - loss: 0.1146 - accuracy: 0.9594 - val_l
oss: 0.3557 - val_accuracy: 0.9037
Epoch 120/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1103 - accuracy: 0.9610 - val_l
oss: 0.3594 - val_accuracy: 0.9048
Epoch 121/170
```

```
390/390 [==============================] - 217s 556ms/step - loss: 0.1112 - accuracy: 0.9603 - val_l
oss: 0.4285 - val_accuracy: 0.8923
Epoch 122/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1107 - accuracy: 0.9611 - val_l
oss: 0.3594 - val_accuracy: 0.9033
Epoch 123/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1059 - accuracy: 0.9628 - val_l
oss: 0.3718 - val_accuracy: 0.9029
Epoch 124/170
390/390 [==============================] - 217s 554ms/step - loss: 0.1027 - accuracy: 0.9636 - val_l
oss: 0.3805 - val_accuracy: 0.8987
Epoch 125/170
390/390 [==============================] - 217s 554ms/step - loss: 0.1098 - accuracy: 0.9613 - val_l
oss: 0.4183 - val_accuracy: 0.8956
Epoch 126/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1125 - accuracy: 0.9605 - val_l
oss: 0.3563 - val_accuracy: 0.9035
Epoch 127/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1083 - accuracy: 0.9617 - val_l
oss: 0.3801 - val_accuracy: 0.8988
Epoch 128/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1061 - accuracy: 0.9621 - val_l
oss: 0.3781 - val_accuracy: 0.9013
Epoch 129/170
390/390 [==============================] - 217s 554ms/step - loss: 0.1020 - accuracy: 0.9640 - val_l
oss: 0.4838 - val_accuracy: 0.8833
Epoch 130/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1043 - accuracy: 0.9629 - val_l
oss: 0.4289 - val_accuracy: 0.8951
Epoch 131/170
390/390 [==============================] - 217s 555ms/step - loss: 0.1001 - accuracy: 0.9640 - val_l
oss: 0.3525 - val_accuracy: 0.9086
Epoch 132/170
390/390 [==============================] - 217s 554ms/step - loss: 0.1022 - accuracy: 0.9644 - val_l
oss: 0.4281 - val_accuracy: 0.8933
Epoch 133/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0996 - accuracy: 0.9658 - val_l
oss: 0.4067 - val_accuracy: 0.9008
Epoch 134/170
390/390 [==============================] - 224s 573ms/step - loss: 0.0967 - accuracy: 0.9657 - val_l
oss: 0.3942 - val_accuracy: 0.9020
Epoch 135/170
390/390 [==============================] - 217s 556ms/step - loss: 0.1017 - accuracy: 0.9643 - val_l
oss: 0.3585 - val_accuracy: 0.9074
Epoch 136/170
390/390 [==============================] - 217s 556ms/step - loss: 0.1003 - accuracy: 0.9646 - val_l
oss: 0.4464 - val_accuracy: 0.8886
Epoch 137/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0986 - accuracy: 0.9659 - val_l
oss: 0.4133 - val_accuracy: 0.8965
Epoch 138/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0949 - accuracy: 0.9668 - val_l
oss: 0.5549 - val_accuracy: 0.8706
Epoch 139/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0950 - accuracy: 0.9669 - val_l
oss: 0.4177 - val_accuracy: 0.8975
Epoch 140/170
390/390 [==============================] - 217s 555ms/step - loss: 0.0968 - accuracy: 0.9651 - val_l
oss: 0.4429 - val_accuracy: 0.8914
Epoch 141/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0943 - accuracy: 0.9671 - val_l
oss: 0.4340 - val_accuracy: 0.8943
Epoch 142/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0938 - accuracy: 0.9666 - val_l
oss: 0.4741 - val_accuracy: 0.8903
Epoch 143/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0916 - accuracy: 0.9680 - val_l
oss: 0.3351 - val_accuracy: 0.9140
Epoch 144/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0949 - accuracy: 0.9672 - val_l
oss: 0.4273 - val_accuracy: 0.8961
Epoch 145/170
390/390 [==============================] - 218s 558ms/step - loss: 0.0941 - accuracy: 0.9670 - val_l
oss: 0.3592 - val_accuracy: 0.9056
Epoch 146/170
390/390 [==============================] - 218s 558ms/step - loss: 0.0924 - accuracy: 0.9680 - val_l
oss: 0.3545 - val_accuracy: 0.9068
Epoch 147/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0933 - accuracy: 0.9671 - val_l
oss: 0.4598 - val_accuracy: 0.8914
Epoch 148/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0883 - accuracy: 0.9689 - val_l
oss: 0.3479 - val_accuracy: 0.9121
```

```
Epoch 149/170
390/390 [==============================] - 217s 555ms/step - loss: 0.0893 - accuracy: 0.9687 - val_l
oss: 0.4738 - val_accuracy: 0.8864
Epoch 150/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0890 - accuracy: 0.9691 - val_l
oss: 0.3577 - val_accuracy: 0.9106
Epoch 151/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0874 - accuracy: 0.9696 - val_l
oss: 0.4714 - val_accuracy: 0.8898
Epoch 152/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0889 - accuracy: 0.9686 - val_l
oss: 0.3812 - val_accuracy: 0.9018
Epoch 153/170
390/390 [==============================] - 217s 555ms/step - loss: 0.0860 - accuracy: 0.9695 - val_l
oss: 0.4595 - val_accuracy: 0.8907
Epoch 154/170
390/390 [==============================] - 217s 555ms/step - loss: 0.0863 - accuracy: 0.9695 - val_l
oss: 0.4402 - val_accuracy: 0.8963
Epoch 155/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0834 - accuracy: 0.9700 - val_l
oss: 0.3891 - val_accuracy: 0.9071
Epoch 156/170
390/390 [==============================] - 218s 557ms/step - loss: 0.0829 - accuracy: 0.9714 - val_l
oss: 0.3793 - val_accuracy: 0.9029
Epoch 157/170
390/390 [==============================] - 218s 557ms/step - loss: 0.0855 - accuracy: 0.9696 - val_l
oss: 0.3976 - val_accuracy: 0.9040
Epoch 158/170
390/390 [==============================] - 217s 557ms/step - loss: 0.0826 - accuracy: 0.9707 - val_l
oss: 0.4268 - val_accuracy: 0.8970
Epoch 159/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0874 - accuracy: 0.9686 - val_l
oss: 0.4545 - val_accuracy: 0.8959
Epoch 160/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0835 - accuracy: 0.9706 - val_l
oss: 0.4577 - val_accuracy: 0.8917
Epoch 161/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0851 - accuracy: 0.9704 - val_l
oss: 0.4176 - val_accuracy: 0.8987
Epoch 162/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0805 - accuracy: 0.9710 - val_l
oss: 0.3944 - val_accuracy: 0.9046
Epoch 163/170
390/390 [==============================] - 223s 572ms/step - loss: 0.0834 - accuracy: 0.9708 - val_l
oss: 0.4053 - val_accuracy: 0.9010
Epoch 164/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0802 - accuracy: 0.9720 - val_l
oss: 0.3823 - val_accuracy: 0.9068
Epoch 165/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0828 - accuracy: 0.9714 - val_l
oss: 0.3682 - val_accuracy: 0.9096
Epoch 166/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0784 - accuracy: 0.9724 - val_l
oss: 0.4096 - val_accuracy: 0.8986
Epoch 167/170
390/390 [==============================] - 217s 556ms/step - loss: 0.0765 - accuracy: 0.9730 - val_l
oss: 0.4288 - val_accuracy: 0.9057
Epoch 168/170
390/390 [==============================] - 218s 558ms/step - loss: 0.0769 - accuracy: 0.9732 - val_l
oss: 0.4138 - val_accuracy: 0.9043
Epoch 169/170
390/390 [==============================] - 218s 558ms/step - loss: 0.0809 - accuracy: 0.9721 - val_l
oss: 0.3945 - val_accuracy: 0.9046
Epoch 170/170
390/390 [==============================] - 218s 557ms/step - loss: 0.0766 - accuracy: 0.9732 - val_l
oss: 0.4235 - val_accuracy: 0.8998
```

In [ ]:

```
from tensorflow import keras
keras.backend.set_value(model.optimizer.momentum, 0.7)
keras.backend.set_value(model.optimizer.lr, 0.001)
model.load_weights('/content/gdrive/MyDrive/cnnoncifar/cifar10_model_save/model-170-0.973180-0.899800.h5')
path = 'gdrive/My Drive/cnnoncifar/'
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = x_train.shape[0]/batch_size, in
itial_epoch = 170,
                    epochs = 210, validation_data =(x_test, y_test), callbacks = [checkpoint,csvlog])
model.save_weights(os.path.join(path, '210epochs.h5'))
```

```
Epoch 171/210
390/390 [==============================] - 218s 557ms/step - loss: 0.0734 - accuracy: 0.9740 - val_l
oss: 0.3551 - val_accuracy: 0.9145
Epoch 172/210
```

```
390/390 [==============================] - 217s 556ms/step - loss: 0.0645 - accuracy: 0.9769 - val_l
oss: 0.3490 - val_accuracy: 0.9162
Epoch 173/210
390/390 [==============================] - 218s 558ms/step - loss: 0.0602 - accuracy: 0.9796 - val_l
oss: 0.3467 - val_accuracy: 0.9177
Epoch 174/210
390/390 [==============================] - 218s 557ms/step - loss: 0.0567 - accuracy: 0.9803 - val_l
oss: 0.3451 - val_accuracy: 0.9189
Epoch 175/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0569 - accuracy: 0.9806 - val_l
oss: 0.3409 - val_accuracy: 0.9192
Epoch 176/210
390/390 [==============================] - 217s 554ms/step - loss: 0.0564 - accuracy: 0.9811 - val_l
oss: 0.3397 - val_accuracy: 0.9196
Epoch 177/210
390/390 [==============================] - 217s 555ms/step - loss: 0.0553 - accuracy: 0.9808 - val_l
oss: 0.3392 - val_accuracy: 0.9194
Epoch 178/210
390/390 [==============================] - 217s 555ms/step - loss: 0.0525 - accuracy: 0.9825 - val_l
oss: 0.3391 - val_accuracy: 0.9201
Epoch 179/210
390/390 [==============================] - 223s 571ms/step - loss: 0.0550 - accuracy: 0.9817 - val_l
oss: 0.3377 - val_accuracy: 0.9204
Epoch 180/210
390/390 [==============================] - 217s 555ms/step - loss: 0.0512 - accuracy: 0.9819 - val_l
oss: 0.3373 - val_accuracy: 0.9206
Epoch 181/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0518 - accuracy: 0.9818 - val_l
oss: 0.3373 - val_accuracy: 0.9202
Epoch 182/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0512 - accuracy: 0.9827 - val_l
oss: 0.3353 - val_accuracy: 0.9211
Epoch 183/210
390/390 [==============================] - 217s 555ms/step - loss: 0.0495 - accuracy: 0.9836 - val_l
oss: 0.3370 - val_accuracy: 0.9209
Epoch 184/210
390/390 [==============================] - 223s 571ms/step - loss: 0.0503 - accuracy: 0.9825 - val_l
oss: 0.3371 - val_accuracy: 0.9209
Epoch 185/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0503 - accuracy: 0.9829 - val_l
oss: 0.3345 - val_accuracy: 0.9211
Epoch 186/210
390/390 [==============================] - 217s 557ms/step - loss: 0.0486 - accuracy: 0.9832 - val_l
oss: 0.3347 - val_accuracy: 0.9206
Epoch 187/210
390/390 [==============================] - 217s 557ms/step - loss: 0.0519 - accuracy: 0.9826 - val_l
oss: 0.3348 - val_accuracy: 0.9204
Epoch 188/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0476 - accuracy: 0.9836 - val_l
oss: 0.3330 - val_accuracy: 0.9206
Epoch 189/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0474 - accuracy: 0.9842 - val_l
oss: 0.3341 - val_accuracy: 0.9212
Epoch 190/210
390/390 [==============================] - 217s 556ms/step - loss: 0.0480 - accuracy: 0.9831 - val_l
oss: 0.3328 - val_accuracy: 0.9210
Epoch 191/210
390/390 [==============================] - 218s 557ms/step - loss: 0.0469 - accuracy: 0.9836 - val_l
oss: 0.3343 - val_accuracy: 0.9206
Epoch 192/210
390/390 [==============================] - 218s 558ms/step - loss: 0.0469 - accuracy: 0.9841 - val_l
oss: 0.3345 - val_accuracy: 0.9204
Epoch 193/210
390/390 [==============================] - 218s 557ms/step - loss: 0.0466 - accuracy: 0.9843 - val_l
oss: 0.3326 - val_accuracy: 0.9211
Epoch 194/210
390/390 [==============================] - 218s 558ms/step - loss: 0.0469 - accuracy: 0.9846 - val_l
oss: 0.3338 - val_accuracy: 0.9206
Epoch 195/210
390/390 [==============================] - 218s 558ms/step - loss: 0.0470 - accuracy: 0.9839 - val_l
oss: 0.3331 - val_accuracy: 0.9209
Epoch 196/210
390/390 [==============================] - 218s 558ms/step - loss: 0.0451 - accuracy: 0.9846 - val_l
oss: 0.3338 - val_accuracy: 0.9206
Epoch 197/210
390/390 [==============================] - 218s 559ms/step - loss: 0.0455 - accuracy: 0.9847 - val_l
oss: 0.3339 - val_accuracy: 0.9206
Epoch 198/210
390/390 [==============================] - 218s 559ms/step - loss: 0.0485 - accuracy: 0.9834 - val_l
oss: 0.3337 - val_accuracy: 0.9202
Epoch 199/210
390/390 [==============================] - 218s 559ms/step - loss: 0.0457 - accuracy: 0.9847 - val_l
oss: 0.3339 - val_accuracy: 0.9205
```

```
Epoch 200/210
 34/390 [=>............................] - ETA: 3:06 - loss: 0.0555 - accuracy: 0.9816
```

Observation

We have used cifar10 dataset and did one hot encoding. As per referenece assignment, used dense block,transition block and output_layer.

To convet on cifar10 dataset, we have used 2D Convolutional neural network and top of that, we have applied dense block to create first block,and for first transition, we have applied transition block on top of first block. Similarly, We created second block,second transition and third block, third transition. On top of last block, we have applied output layer and create the model.

We have used image augmentation technique, and fit train data. We have used categorical_crossentropy as loss function and SGD optimizer with 0.7 momentum and trained the model using metrics "accuracy". We have trained by 300 epochs and got test accuracy 90%+.