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 Dense Layers (also called fully connected layers), or DropOut.
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]:
```
from google.colab import drive
drive.mount('/content/gdrive')
import os
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
```

```
Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee649
1hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3aietf%3awg%3aoauth%3a2.0%3aoob&response_type=code&scope=email%
20https%3a%2f%2fwww.googleapis.com%2fauth%2fdocs.test%20https%3a%2f%2fwww.googleapis.com%2fauth%2fdrive%20https%3a%2
f%2fwww.googleapis.com%2fauth%2fdrive.photos.readonly%20https%3a%2f%2fwww.googleapis.com%2fauth%2fpeopleapi.readonly

Enter your authorization code:
..........
Mounted at /content/gdrive
```

In [2]:
```
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, RMSprop, SGD
from tensorflow.keras.callbacks import ModelCheckpoint
import keras
from keras.datasets import cifar10
from keras.optimizers import Adam, RMSprop, SGD
from keras import regularizers
```

```
Using TensorFlow backend.
```

In [0]:
```
# this part will prevent tensorflow to allocate all the avaliable GPU Memory
# backend
import tensorflow as tf
# Hyperparameters
batch_size = 128
num_classes = 10
epochs = 10
num_filter = 32
compression = 0.5
```

In [4]:
```
# Load CIFAR10 Data
(X_train, y_train), (X_test, y_test) = 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 [==============================] - 11s 0us/step
```

In [0]:
```
# Dense Block
def denseblock(input, num_filter = 32):
    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)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_3_3])
        temp = concat
    return temp

## transition Blosck
def transition(input, num_filter = 32):
    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)
    avg = layers.AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)
    return avg
```

In [0]:
```
num_filter = 32
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)
First_Transition = transition(First_Block, num_filter)

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

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

Last_Block = denseblock(Third_Transition,  num_filter)

BatchNorm = layers.BatchNormalization()(Last_Block)
relu = layers.Activation('relu')(BatchNorm)
AvgPooling = layers.AveragePooling2D(pool_size=(2,2))(relu)
conv_output=layers.Conv2D(10,(2,2))(AvgPooling)
conv_output1=layers.Conv2D(10,(1,1))(conv_output)
output=layers.Flatten()(conv_output1)
output = layers.Activation('softmax')(output)
```

In [58]:
```
model = Model(inputs=[input], outputs=[output])
model.summary()
```

```
Model: "model_8"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_15 (InputLayer)           [(None, 32, 32, 3)]  0
__________________________________________________________________________________________________
conv2d_743 (Conv2D)             (None, 32, 32, 32)   864         input_15[0][0]
__________________________________________________________________________________________________
batch_normalization_728 (BatchN (None, 32, 32, 32)   128         conv2d_743[0][0]
__________________________________________________________________________________________________
activation_745 (Activation)     (None, 32, 32, 32)   0           batch_normalization_728[0][0]
__________________________________________________________________________________________________
conv2d_744 (Conv2D)             (None, 32, 32, 16)   4608        activation_745[0][0]
__________________________________________________________________________________________________
concatenate_672 (Concatenate)   (None, 32, 32, 48)   0           conv2d_743[0][0]
                                                                 conv2d_744[0][0]
__________________________________________________________________________________________________
batch_normalization_729 (BatchN (None, 32, 32, 48)   192         concatenate_672[0][0]
__________________________________________________________________________________________________
activation_746 (Activation)     (None, 32, 32, 48)   0           batch_normalization_729[0][0]
__________________________________________________________________________________________________
conv2d_745 (Conv2D)             (None, 32, 32, 16)   6912        activation_746[0][0]
__________________________________________________________________________________________________
concatenate_673 (Concatenate)   (None, 32, 32, 64)   0           concatenate_672[0][0]
                                                                 conv2d_745[0][0]
__________________________________________________________________________________________________
batch_normalization_730 (BatchN (None, 32, 32, 64)   256         concatenate_673[0][0]
__________________________________________________________________________________________________
activation_747 (Activation)     (None, 32, 32, 64)   0           batch_normalization_730[0][0]
__________________________________________________________________________________________________
conv2d_746 (Conv2D)             (None, 32, 32, 16)   9216        activation_747[0][0]
__________________________________________________________________________________________________
concatenate_674 (Concatenate)   (None, 32, 32, 80)   0           concatenate_673[0][0]
                                                                 conv2d_746[0][0]
__________________________________________________________________________________________________
batch_normalization_731 (BatchN (None, 32, 32, 80)   320         concatenate_674[0][0]
__________________________________________________________________________________________________
activation_748 (Activation)     (None, 32, 32, 80)   0           batch_normalization_731[0][0]
__________________________________________________________________________________________________
conv2d_747 (Conv2D)             (None, 32, 32, 16)   11520       activation_748[0][0]
__________________________________________________________________________________________________
```

concatenate_675 (Concatenate)	(None, 32, 32, 96)	0	concatenate_674[0][0] conv2d_747[0][0]
batch_normalization_732 (BatchN	(None, 32, 32, 96)	384	concatenate_675[0][0]
activation_749 (Activation)	(None, 32, 32, 96)	0	batch_normalization_732[0][0]
conv2d_748 (Conv2D)	(None, 32, 32, 16)	13824	activation_749[0][0]
concatenate_676 (Concatenate)	(None, 32, 32, 112)	0	concatenate_675[0][0] conv2d_748[0][0]
batch_normalization_733 (BatchN	(None, 32, 32, 112)	448	concatenate_676[0][0]
activation_750 (Activation)	(None, 32, 32, 112)	0	batch_normalization_733[0][0]
conv2d_749 (Conv2D)	(None, 32, 32, 16)	16128	activation_750[0][0]
concatenate_677 (Concatenate)	(None, 32, 32, 128)	0	concatenate_676[0][0] conv2d_749[0][0]
batch_normalization_734 (BatchN	(None, 32, 32, 128)	512	concatenate_677[0][0]
activation_751 (Activation)	(None, 32, 32, 128)	0	batch_normalization_734[0][0]
conv2d_750 (Conv2D)	(None, 32, 32, 16)	18432	activation_751[0][0]
concatenate_678 (Concatenate)	(None, 32, 32, 144)	0	concatenate_677[0][0] conv2d_750[0][0]
batch_normalization_735 (BatchN	(None, 32, 32, 144)	576	concatenate_678[0][0]
activation_752 (Activation)	(None, 32, 32, 144)	0	batch_normalization_735[0][0]
conv2d_751 (Conv2D)	(None, 32, 32, 16)	20736	activation_752[0][0]
concatenate_679 (Concatenate)	(None, 32, 32, 160)	0	concatenate_678[0][0] conv2d_751[0][0]
batch_normalization_736 (BatchN	(None, 32, 32, 160)	640	concatenate_679[0][0]
activation_753 (Activation)	(None, 32, 32, 160)	0	batch_normalization_736[0][0]

conv2d_752 (Conv2D)	(None, 32, 32, 16)	23040	activation_753[0][0]
concatenate_680 (Concatenate)	(None, 32, 32, 176)	0	concatenate_679[0][0] conv2d_752[0][0]
batch_normalization_737 (BatchN	(None, 32, 32, 176)	704	concatenate_680[0][0]
activation_754 (Activation)	(None, 32, 32, 176)	0	batch_normalization_737[0][0]
conv2d_753 (Conv2D)	(None, 32, 32, 16)	25344	activation_754[0][0]
concatenate_681 (Concatenate)	(None, 32, 32, 192)	0	concatenate_680[0][0] conv2d_753[0][0]
batch_normalization_738 (BatchN	(None, 32, 32, 192)	768	concatenate_681[0][0]
activation_755 (Activation)	(None, 32, 32, 192)	0	batch_normalization_738[0][0]
conv2d_754 (Conv2D)	(None, 32, 32, 16)	27648	activation_755[0][0]
concatenate_682 (Concatenate)	(None, 32, 32, 208)	0	concatenate_681[0][0] conv2d_754[0][0]
batch_normalization_739 (BatchN	(None, 32, 32, 208)	832	concatenate_682[0][0]
activation_756 (Activation)	(None, 32, 32, 208)	0	batch_normalization_739[0][0]
conv2d_755 (Conv2D)	(None, 32, 32, 16)	29952	activation_756[0][0]
concatenate_683 (Concatenate)	(None, 32, 32, 224)	0	concatenate_682[0][0] conv2d_755[0][0]
batch_normalization_740 (BatchN	(None, 32, 32, 224)	896	concatenate_683[0][0]
activation_757 (Activation)	(None, 32, 32, 224)	0	batch_normalization_740[0][0]
conv2d_756 (Conv2D)	(None, 32, 32, 16)	3584	activation_757[0][0]
average_pooling2d_56 (AveragePo	(None, 16, 16, 16)	0	conv2d_756[0][0]
batch_normalization_741 (BatchN	(None, 16, 16, 16)	64	average_pooling2d_56[0][0]

activation_758 (Activation)	(None, 16, 16, 16)	0	batch_normalization_741[0][0]
conv2d_757 (Conv2D)	(None, 16, 16, 16)	2304	activation_758[0][0]
concatenate_684 (Concatenate)	(None, 16, 16, 32)	0	average_pooling2d_56[0][0] conv2d_757[0][0]
batch_normalization_742 (BatchN	(None, 16, 16, 32)	128	concatenate_684[0][0]
activation_759 (Activation)	(None, 16, 16, 32)	0	batch_normalization_742[0][0]
conv2d_758 (Conv2D)	(None, 16, 16, 16)	4608	activation_759[0][0]
concatenate_685 (Concatenate)	(None, 16, 16, 48)	0	concatenate_684[0][0] conv2d_758[0][0]
batch_normalization_743 (BatchN	(None, 16, 16, 48)	192	concatenate_685[0][0]
activation_760 (Activation)	(None, 16, 16, 48)	0	batch_normalization_743[0][0]
conv2d_759 (Conv2D)	(None, 16, 16, 16)	6912	activation_760[0][0]
concatenate_686 (Concatenate)	(None, 16, 16, 64)	0	concatenate_685[0][0] conv2d_759[0][0]
batch_normalization_744 (BatchN	(None, 16, 16, 64)	256	concatenate_686[0][0]
activation_761 (Activation)	(None, 16, 16, 64)	0	batch_normalization_744[0][0]
conv2d_760 (Conv2D)	(None, 16, 16, 16)	9216	activation_761[0][0]
concatenate_687 (Concatenate)	(None, 16, 16, 80)	0	concatenate_686[0][0] conv2d_760[0][0]
batch_normalization_745 (BatchN	(None, 16, 16, 80)	320	concatenate_687[0][0]
activation_762 (Activation)	(None, 16, 16, 80)	0	batch_normalization_745[0][0]
conv2d_761 (Conv2D)	(None, 16, 16, 16)	11520	activation_762[0][0]
concatenate_688 (Concatenate)	(None, 16, 16, 96)	0	concatenate_687[0][0]

			conv2d_761[0][0]
batch_normalization_746 (BatchN	(None, 16, 16, 96)	384	concatenate_688[0][0]
activation_763 (Activation)	(None, 16, 16, 96)	0	batch_normalization_746[0][0]
conv2d_762 (Conv2D)	(None, 16, 16, 16)	13824	activation_763[0][0]
concatenate_689 (Concatenate)	(None, 16, 16, 112)	0	concatenate_688[0][0] conv2d_762[0][0]
batch_normalization_747 (BatchN	(None, 16, 16, 112)	448	concatenate_689[0][0]
activation_764 (Activation)	(None, 16, 16, 112)	0	batch_normalization_747[0][0]
conv2d_763 (Conv2D)	(None, 16, 16, 16)	16128	activation_764[0][0]
concatenate_690 (Concatenate)	(None, 16, 16, 128)	0	concatenate_689[0][0] conv2d_763[0][0]
batch_normalization_748 (BatchN	(None, 16, 16, 128)	512	concatenate_690[0][0]
activation_765 (Activation)	(None, 16, 16, 128)	0	batch_normalization_748[0][0]
conv2d_764 (Conv2D)	(None, 16, 16, 16)	18432	activation_765[0][0]
concatenate_691 (Concatenate)	(None, 16, 16, 144)	0	concatenate_690[0][0] conv2d_764[0][0]
batch_normalization_749 (BatchN	(None, 16, 16, 144)	576	concatenate_691[0][0]
activation_766 (Activation)	(None, 16, 16, 144)	0	batch_normalization_749[0][0]
conv2d_765 (Conv2D)	(None, 16, 16, 16)	20736	activation_766[0][0]
concatenate_692 (Concatenate)	(None, 16, 16, 160)	0	concatenate_691[0][0] conv2d_765[0][0]
batch_normalization_750 (BatchN	(None, 16, 16, 160)	640	concatenate_692[0][0]
activation_767 (Activation)	(None, 16, 16, 160)	0	batch_normalization_750[0][0]

conv2d_766 (Conv2D)	(None, 16, 16, 16)	23040	activation_767[0][0]
concatenate_693 (Concatenate)	(None, 16, 16, 176)	0	concatenate_692[0][0] conv2d_766[0][0]
batch_normalization_751 (BatchN	(None, 16, 16, 176)	704	concatenate_693[0][0]
activation_768 (Activation)	(None, 16, 16, 176)	0	batch_normalization_751[0][0]
conv2d_767 (Conv2D)	(None, 16, 16, 16)	25344	activation_768[0][0]
concatenate_694 (Concatenate)	(None, 16, 16, 192)	0	concatenate_693[0][0] conv2d_767[0][0]
batch_normalization_752 (BatchN	(None, 16, 16, 192)	768	concatenate_694[0][0]
activation_769 (Activation)	(None, 16, 16, 192)	0	batch_normalization_752[0][0]
conv2d_768 (Conv2D)	(None, 16, 16, 16)	27648	activation_769[0][0]
concatenate_695 (Concatenate)	(None, 16, 16, 208)	0	concatenate_694[0][0] conv2d_768[0][0]
batch_normalization_753 (BatchN	(None, 16, 16, 208)	832	concatenate_695[0][0]
activation_770 (Activation)	(None, 16, 16, 208)	0	batch_normalization_753[0][0]
conv2d_769 (Conv2D)	(None, 16, 16, 16)	3328	activation_770[0][0]
average_pooling2d_57 (AveragePo	(None, 8, 8, 16)	0	conv2d_769[0][0]
batch_normalization_754 (BatchN	(None, 8, 8, 16)	64	average_pooling2d_57[0][0]
activation_771 (Activation)	(None, 8, 8, 16)	0	batch_normalization_754[0][0]
conv2d_770 (Conv2D)	(None, 8, 8, 16)	2304	activation_771[0][0]
concatenate_696 (Concatenate)	(None, 8, 8, 32)	0	average_pooling2d_57[0][0] conv2d_770[0][0]
batch_normalization_755 (BatchN	(None, 8, 8, 32)	128	concatenate_696[0][0]

activation_772 (Activation)	(None, 8, 8, 32)	0	batch_normalization_755[0][0]
conv2d_771 (Conv2D)	(None, 8, 8, 16)	4608	activation_772[0][0]
concatenate_697 (Concatenate)	(None, 8, 8, 48)	0	concatenate_696[0][0] conv2d_771[0][0]
batch_normalization_756 (BatchN	(None, 8, 8, 48)	192	concatenate_697[0][0]
activation_773 (Activation)	(None, 8, 8, 48)	0	batch_normalization_756[0][0]
conv2d_772 (Conv2D)	(None, 8, 8, 16)	6912	activation_773[0][0]
concatenate_698 (Concatenate)	(None, 8, 8, 64)	0	concatenate_697[0][0] conv2d_772[0][0]
batch_normalization_757 (BatchN	(None, 8, 8, 64)	256	concatenate_698[0][0]
activation_774 (Activation)	(None, 8, 8, 64)	0	batch_normalization_757[0][0]
conv2d_773 (Conv2D)	(None, 8, 8, 16)	9216	activation_774[0][0]
concatenate_699 (Concatenate)	(None, 8, 8, 80)	0	concatenate_698[0][0] conv2d_773[0][0]
batch_normalization_758 (BatchN	(None, 8, 8, 80)	320	concatenate_699[0][0]
activation_775 (Activation)	(None, 8, 8, 80)	0	batch_normalization_758[0][0]
conv2d_774 (Conv2D)	(None, 8, 8, 16)	11520	activation_775[0][0]
concatenate_700 (Concatenate)	(None, 8, 8, 96)	0	concatenate_699[0][0] conv2d_774[0][0]
batch_normalization_759 (BatchN	(None, 8, 8, 96)	384	concatenate_700[0][0]
activation_776 (Activation)	(None, 8, 8, 96)	0	batch_normalization_759[0][0]
conv2d_775 (Conv2D)	(None, 8, 8, 16)	13824	activation_776[0][0]
concatenate_701 (Concatenate)	(None, 8, 8, 112)	0	concatenate_700[0][0] conv2d_775[0][0]

batch_normalization_760 (BatchN	(None, 8, 8, 112)	448	concatenate_701[0][0]
activation_777 (Activation)	(None, 8, 8, 112)	0	batch_normalization_760[0][0]
conv2d_776 (Conv2D)	(None, 8, 8, 16)	16128	activation_777[0][0]
concatenate_702 (Concatenate)	(None, 8, 8, 128)	0	concatenate_701[0][0] conv2d_776[0][0]
batch_normalization_761 (BatchN	(None, 8, 8, 128)	512	concatenate_702[0][0]
activation_778 (Activation)	(None, 8, 8, 128)	0	batch_normalization_761[0][0]
conv2d_777 (Conv2D)	(None, 8, 8, 16)	18432	activation_778[0][0]
concatenate_703 (Concatenate)	(None, 8, 8, 144)	0	concatenate_702[0][0] conv2d_777[0][0]
batch_normalization_762 (BatchN	(None, 8, 8, 144)	576	concatenate_703[0][0]
activation_779 (Activation)	(None, 8, 8, 144)	0	batch_normalization_762[0][0]
conv2d_778 (Conv2D)	(None, 8, 8, 16)	20736	activation_779[0][0]
concatenate_704 (Concatenate)	(None, 8, 8, 160)	0	concatenate_703[0][0] conv2d_778[0][0]
batch_normalization_763 (BatchN	(None, 8, 8, 160)	640	concatenate_704[0][0]
activation_780 (Activation)	(None, 8, 8, 160)	0	batch_normalization_763[0][0]
conv2d_779 (Conv2D)	(None, 8, 8, 16)	23040	activation_780[0][0]
concatenate_705 (Concatenate)	(None, 8, 8, 176)	0	concatenate_704[0][0] conv2d_779[0][0]
batch_normalization_764 (BatchN	(None, 8, 8, 176)	704	concatenate_705[0][0]
activation_781 (Activation)	(None, 8, 8, 176)	0	batch_normalization_764[0][0]
conv2d_780 (Conv2D)	(None, 8, 8, 16)	25344	activation_781[0][0]

concatenate_706 (Concatenate)	(None, 8, 8, 192)	0	concatenate_705[0][0] conv2d_780[0][0]
batch_normalization_765 (BatchN	(None, 8, 8, 192)	768	concatenate_706[0][0]
activation_782 (Activation)	(None, 8, 8, 192)	0	batch_normalization_765[0][0]
conv2d_781 (Conv2D)	(None, 8, 8, 16)	27648	activation_782[0][0]
concatenate_707 (Concatenate)	(None, 8, 8, 208)	0	concatenate_706[0][0] conv2d_781[0][0]
batch_normalization_766 (BatchN	(None, 8, 8, 208)	832	concatenate_707[0][0]
activation_783 (Activation)	(None, 8, 8, 208)	0	batch_normalization_766[0][0]
conv2d_782 (Conv2D)	(None, 8, 8, 16)	3328	activation_783[0][0]
average_pooling2d_58 (AveragePo	(None, 4, 4, 16)	0	conv2d_782[0][0]
batch_normalization_767 (BatchN	(None, 4, 4, 16)	64	average_pooling2d_58[0][0]
activation_784 (Activation)	(None, 4, 4, 16)	0	batch_normalization_767[0][0]
conv2d_783 (Conv2D)	(None, 4, 4, 16)	2304	activation_784[0][0]
concatenate_708 (Concatenate)	(None, 4, 4, 32)	0	average_pooling2d_58[0][0] conv2d_783[0][0]
batch_normalization_768 (BatchN	(None, 4, 4, 32)	128	concatenate_708[0][0]
activation_785 (Activation)	(None, 4, 4, 32)	0	batch_normalization_768[0][0]
conv2d_784 (Conv2D)	(None, 4, 4, 16)	4608	activation_785[0][0]
concatenate_709 (Concatenate)	(None, 4, 4, 48)	0	concatenate_708[0][0] conv2d_784[0][0]
batch_normalization_769 (BatchN	(None, 4, 4, 48)	192	concatenate_709[0][0]
activation_786 (Activation)	(None, 4, 4, 48)	0	batch_normalization_769[0][0]

conv2d_785 (Conv2D)	(None, 4, 4, 16)	6912	activation_786[0][0]
concatenate_710 (Concatenate)	(None, 4, 4, 64)	0	concatenate_709[0][0] conv2d_785[0][0]
batch_normalization_770 (BatchN	(None, 4, 4, 64)	256	concatenate_710[0][0]
activation_787 (Activation)	(None, 4, 4, 64)	0	batch_normalization_770[0][0]
conv2d_786 (Conv2D)	(None, 4, 4, 16)	9216	activation_787[0][0]
concatenate_711 (Concatenate)	(None, 4, 4, 80)	0	concatenate_710[0][0] conv2d_786[0][0]
batch_normalization_771 (BatchN	(None, 4, 4, 80)	320	concatenate_711[0][0]
activation_788 (Activation)	(None, 4, 4, 80)	0	batch_normalization_771[0][0]
conv2d_787 (Conv2D)	(None, 4, 4, 16)	11520	activation_788[0][0]
concatenate_712 (Concatenate)	(None, 4, 4, 96)	0	concatenate_711[0][0] conv2d_787[0][0]
batch_normalization_772 (BatchN	(None, 4, 4, 96)	384	concatenate_712[0][0]
activation_789 (Activation)	(None, 4, 4, 96)	0	batch_normalization_772[0][0]
conv2d_788 (Conv2D)	(None, 4, 4, 16)	13824	activation_789[0][0]
concatenate_713 (Concatenate)	(None, 4, 4, 112)	0	concatenate_712[0][0] conv2d_788[0][0]
batch_normalization_773 (BatchN	(None, 4, 4, 112)	448	concatenate_713[0][0]
activation_790 (Activation)	(None, 4, 4, 112)	0	batch_normalization_773[0][0]
conv2d_789 (Conv2D)	(None, 4, 4, 16)	16128	activation_790[0][0]
concatenate_714 (Concatenate)	(None, 4, 4, 128)	0	concatenate_713[0][0] conv2d_789[0][0]

batch_normalization_774 (BatchN	(None, 4, 4, 128)	512	concatenate_714[0][0]
activation_791 (Activation)	(None, 4, 4, 128)	0	batch_normalization_774[0][0]
conv2d_790 (Conv2D)	(None, 4, 4, 16)	18432	activation_791[0][0]
concatenate_715 (Concatenate)	(None, 4, 4, 144)	0	concatenate_714[0][0] conv2d_790[0][0]
batch_normalization_775 (BatchN	(None, 4, 4, 144)	576	concatenate_715[0][0]
activation_792 (Activation)	(None, 4, 4, 144)	0	batch_normalization_775[0][0]
conv2d_791 (Conv2D)	(None, 4, 4, 16)	20736	activation_792[0][0]
concatenate_716 (Concatenate)	(None, 4, 4, 160)	0	concatenate_715[0][0] conv2d_791[0][0]
batch_normalization_776 (BatchN	(None, 4, 4, 160)	640	concatenate_716[0][0]
activation_793 (Activation)	(None, 4, 4, 160)	0	batch_normalization_776[0][0]
conv2d_792 (Conv2D)	(None, 4, 4, 16)	23040	activation_793[0][0]
concatenate_717 (Concatenate)	(None, 4, 4, 176)	0	concatenate_716[0][0] conv2d_792[0][0]
batch_normalization_777 (BatchN	(None, 4, 4, 176)	704	concatenate_717[0][0]
activation_794 (Activation)	(None, 4, 4, 176)	0	batch_normalization_777[0][0]
conv2d_793 (Conv2D)	(None, 4, 4, 16)	25344	activation_794[0][0]
concatenate_718 (Concatenate)	(None, 4, 4, 192)	0	concatenate_717[0][0] conv2d_793[0][0]
batch_normalization_778 (BatchN	(None, 4, 4, 192)	768	concatenate_718[0][0]
activation_795 (Activation)	(None, 4, 4, 192)	0	batch_normalization_778[0][0]
conv2d_794 (Conv2D)	(None, 4, 4, 16)	27648	activation_795[0][0]

```
concatenate_719 (Concatenate)   (None, 4, 4, 208)    0           concatenate_718[0][0]
                                                                 conv2d_794[0][0]
__________________________________________________________________________________________________
batch_normalization_779 (BatchN (None, 4, 4, 208)    832         concatenate_719[0][0]
__________________________________________________________________________________________________
activation_796 (Activation)     (None, 4, 4, 208)    0           batch_normalization_779[0][0]
__________________________________________________________________________________________________
average_pooling2d_59 (AveragePo (None, 2, 2, 208)    0           activation_796[0][0]
__________________________________________________________________________________________________
conv2d_795 (Conv2D)             (None, 1, 1, 10)     8330        average_pooling2d_59[0][0]
__________________________________________________________________________________________________
conv2d_796 (Conv2D)             (None, 1, 1, 10)     110         conv2d_795[0][0]
__________________________________________________________________________________________________
flatten_5 (Flatten)             (None, 10)           0           conv2d_796[0][0]
__________________________________________________________________________________________________
activation_797 (Activation)     (None, 10)           0           flatten_5[0][0]
==================================================================================================
Total params: 790,168
Trainable params: 778,104
Non-trainable params: 12,064
__________________________________________________________________________________________________
```

In [0]:
```
from keras.preprocessing.image import ImageDataGenerator
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 [0]:
```
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
```

In [0]:
```
sgd=SGD(lr=0.01, momentum = 0.9)
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

Model: epochs:300 Test Accuracy:89%

In [62]:
```
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])
model.save_weights(os.path.join(path, 'model_new_30epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 86s 219ms/step - loss: 1.8385 - accuracy: 0.3240 - val_loss: 1.7987 - val_
accuracy: 0.3746
Epoch 2/30
391/390 [==============================] - 85s 218ms/step - loss: 1.5182 - accuracy: 0.4459 - val_loss: 1.5324 - val_
accuracy: 0.4470
Epoch 3/30
391/390 [==============================] - 85s 218ms/step - loss: 1.3841 - accuracy: 0.5002 - val_loss: 1.4850 - val_
accuracy: 0.4671
Epoch 4/30
391/390 [==============================] - 85s 218ms/step - loss: 1.2878 - accuracy: 0.5344 - val_loss: 1.5536 - val_
accuracy: 0.4764
Epoch 5/30
391/390 [==============================] - 85s 218ms/step - loss: 1.2062 - accuracy: 0.5648 - val_loss: 1.3894 - val_
accuracy: 0.5274
Epoch 6/30
391/390 [==============================] - 86s 219ms/step - loss: 1.1471 - accuracy: 0.5881 - val_loss: 1.3093 - val_
accuracy: 0.5445
Epoch 7/30
391/390 [==============================] - 86s 220ms/step - loss: 1.0908 - accuracy: 0.6112 - val_loss: 1.2884 - val_
accuracy: 0.5687
Epoch 8/30
391/390 [==============================] - 86s 220ms/step - loss: 1.0427 - accuracy: 0.6277 - val_loss: 1.1628 - val_
accuracy: 0.6116
Epoch 9/30
391/390 [==============================] - 86s 219ms/step - loss: 1.0093 - accuracy: 0.6402 - val_loss: 1.0381 - val_
accuracy: 0.6353
Epoch 10/30
391/390 [==============================] - 86s 219ms/step - loss: 0.9728 - accuracy: 0.6521 - val_loss: 1.0044 - val_
accuracy: 0.6403
Epoch 11/30
391/390 [==============================] - 86s 220ms/step - loss: 0.9431 - accuracy: 0.6624 - val_loss: 0.9891 - val_
accuracy: 0.6597
Epoch 12/30
391/390 [==============================] - 86s 220ms/step - loss: 0.9145 - accuracy: 0.6766 - val_loss: 0.8881 - val_
accuracy: 0.6825
Epoch 13/30
391/390 [==============================] - 86s 220ms/step - loss: 0.8901 - accuracy: 0.6830 - val_loss: 1.0305 - val_
accuracy: 0.6368
Epoch 14/30
391/390 [==============================] - 86s 219ms/step - loss: 0.8638 - accuracy: 0.6929 - val_loss: 1.0720 - val_
```

```
accuracy: 0.6341
Epoch 15/30
391/390 [==============================] - 85s 219ms/step - loss: 0.8421 - accuracy: 0.7008 - val_loss: 1.1271 - val_
accuracy: 0.6257
Epoch 16/30
391/390 [==============================] - 86s 219ms/step - loss: 0.8280 - accuracy: 0.7070 - val_loss: 0.9711 - val_
accuracy: 0.6641
Epoch 17/30
391/390 [==============================] - 86s 220ms/step - loss: 0.8093 - accuracy: 0.7135 - val_loss: 0.8327 - val_
accuracy: 0.7083
Epoch 18/30
391/390 [==============================] - 86s 219ms/step - loss: 0.7896 - accuracy: 0.7201 - val_loss: 0.8634 - val_
accuracy: 0.7001
Epoch 19/30
391/390 [==============================] - 86s 219ms/step - loss: 0.7752 - accuracy: 0.7266 - val_loss: 1.1867 - val_
accuracy: 0.6183
Epoch 20/30
391/390 [==============================] - 86s 219ms/step - loss: 0.7584 - accuracy: 0.7310 - val_loss: 0.8665 - val_
accuracy: 0.7004
Epoch 21/30
391/390 [==============================] - 86s 220ms/step - loss: 0.7470 - accuracy: 0.7367 - val_loss: 0.8972 - val_
accuracy: 0.6973
Epoch 22/30
391/390 [==============================] - 86s 219ms/step - loss: 0.7341 - accuracy: 0.7423 - val_loss: 0.7879 - val_
accuracy: 0.7351
Epoch 23/30
391/390 [==============================] - 85s 218ms/step - loss: 0.7194 - accuracy: 0.7439 - val_loss: 0.6973 - val_
accuracy: 0.7575
Epoch 24/30
391/390 [==============================] - 85s 218ms/step - loss: 0.7064 - accuracy: 0.7512 - val_loss: 0.8781 - val_
accuracy: 0.7079
Epoch 25/30
391/390 [==============================] - 85s 219ms/step - loss: 0.6942 - accuracy: 0.7557 - val_loss: 0.7396 - val_
accuracy: 0.7510
Epoch 26/30
391/390 [==============================] - 86s 219ms/step - loss: 0.6840 - accuracy: 0.7583 - val_loss: 0.8380 - val_
accuracy: 0.7226
Epoch 27/30
391/390 [==============================] - 86s 220ms/step - loss: 0.6720 - accuracy: 0.7635 - val_loss: 0.7876 - val_
accuracy: 0.7346
Epoch 28/30
391/390 [==============================] - 86s 220ms/step - loss: 0.6624 - accuracy: 0.7676 - val_loss: 0.8091 - val_
```

```
accuracy: 0.7256
Epoch 29/30
391/390 [==============================] - 86s 221ms/step - loss: 0.6503 - accuracy: 0.7729 - val_loss: 0.6954 - val_
accuracy: 0.7610
Epoch 30/30
391/390 [==============================] - 86s 220ms/step - loss: 0.6408 - accuracy: 0.7759 - val_loss: 0.7821 - val_
accuracy: 0.7375
```

In [63]:
```
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])
model.save_weights(os.path.join(path, 'model_new_60epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 85s 218ms/step - loss: 0.6338 - accuracy: 0.7768 - val_loss: 0.7100 - val_
accuracy: 0.7604
Epoch 2/30
391/390 [==============================] - 86s 219ms/step - loss: 0.6247 - accuracy: 0.7812 - val_loss: 0.7236 - val_
accuracy: 0.7575
Epoch 3/30
391/390 [==============================] - 85s 218ms/step - loss: 0.6162 - accuracy: 0.7827 - val_loss: 0.7967 - val_
accuracy: 0.7382
Epoch 4/30
391/390 [==============================] - 85s 218ms/step - loss: 0.6115 - accuracy: 0.7831 - val_loss: 0.6847 - val_
accuracy: 0.7668
Epoch 5/30
391/390 [==============================] - 86s 219ms/step - loss: 0.6013 - accuracy: 0.7921 - val_loss: 0.6477 - val_
accuracy: 0.7801
Epoch 6/30
391/390 [==============================] - 86s 219ms/step - loss: 0.5903 - accuracy: 0.7932 - val_loss: 0.7652 - val_
accuracy: 0.7449
Epoch 7/30
391/390 [==============================] - 86s 219ms/step - loss: 0.5824 - accuracy: 0.7960 - val_loss: 0.5867 - val_
accuracy: 0.7982
Epoch 8/30
391/390 [==============================] - 86s 219ms/step - loss: 0.5756 - accuracy: 0.8001 - val_loss: 0.7672 - val_
accuracy: 0.7432
Epoch 9/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5748 - accuracy: 0.7988 - val_loss: 0.7108 - val_
accuracy: 0.7669
Epoch 10/30
391/390 [==============================] - 86s 219ms/step - loss: 0.5654 - accuracy: 0.8029 - val_loss: 0.6110 - val_
accuracy: 0.7925
Epoch 11/30
391/390 [==============================] - 86s 220ms/step - loss: 0.5587 - accuracy: 0.8038 - val_loss: 0.6078 - val_
accuracy: 0.7959
Epoch 12/30
391/390 [==============================] - 86s 219ms/step - loss: 0.5577 - accuracy: 0.8062 - val_loss: 0.7557 - val_
accuracy: 0.7569
Epoch 13/30
391/390 [==============================] - 85s 219ms/step - loss: 0.5484 - accuracy: 0.8091 - val_loss: 0.7392 - val_
accuracy: 0.7635
Epoch 14/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5414 - accuracy: 0.8118 - val_loss: 0.7972 - val_
```

```
accuracy: 0.7530
Epoch 15/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5350 - accuracy: 0.8126 - val_loss: 0.7559 - val_
accuracy: 0.7614
Epoch 16/30
391/390 [==============================] - 85s 217ms/step - loss: 0.5319 - accuracy: 0.8167 - val_loss: 0.6356 - val_
accuracy: 0.7881
Epoch 17/30
391/390 [==============================] - 85s 219ms/step - loss: 0.5250 - accuracy: 0.8160 - val_loss: 0.6528 - val_
accuracy: 0.7882
Epoch 18/30
391/390 [==============================] - 85s 217ms/step - loss: 0.5235 - accuracy: 0.8167 - val_loss: 0.7403 - val_
accuracy: 0.7530
Epoch 19/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5103 - accuracy: 0.8228 - val_loss: 0.6608 - val_
accuracy: 0.7816
Epoch 20/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5044 - accuracy: 0.8243 - val_loss: 0.7759 - val_
accuracy: 0.7474
Epoch 21/30
391/390 [==============================] - 85s 218ms/step - loss: 0.5026 - accuracy: 0.8258 - val_loss: 0.7846 - val_
accuracy: 0.7537
Epoch 22/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4984 - accuracy: 0.8263 - val_loss: 0.6535 - val_
accuracy: 0.7822
Epoch 23/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4932 - accuracy: 0.8277 - val_loss: 0.5989 - val_
accuracy: 0.8048
Epoch 24/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4904 - accuracy: 0.8279 - val_loss: 0.6264 - val_
accuracy: 0.8003
Epoch 25/30
391/390 [==============================] - 85s 217ms/step - loss: 0.4853 - accuracy: 0.8305 - val_loss: 0.5826 - val_
accuracy: 0.8023
Epoch 26/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4813 - accuracy: 0.8317 - val_loss: 0.5676 - val_
accuracy: 0.8110
Epoch 27/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4767 - accuracy: 0.8325 - val_loss: 0.5995 - val_
accuracy: 0.7962
Epoch 28/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4678 - accuracy: 0.8368 - val_loss: 0.6762 - val_
```

```
accuracy: 0.7814
Epoch 29/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4664 - accuracy: 0.8376 - val_loss: 0.6559 - val_
accuracy: 0.7851
Epoch 30/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4651 - accuracy: 0.8384 - val_loss: 0.5924 - val_
accuracy: 0.8076
```

In [64]:
```
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])
model.save_weights(os.path.join(path, 'model_new_90epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 85s 217ms/step - loss: 0.4601 - accuracy: 0.8406 - val_loss: 0.5665 - val_
accuracy: 0.8139
Epoch 2/30
391/390 [==============================] - 85s 217ms/step - loss: 0.4611 - accuracy: 0.8396 - val_loss: 0.5623 - val_
accuracy: 0.8140
Epoch 3/30
391/390 [==============================] - 85s 217ms/step - loss: 0.4509 - accuracy: 0.8425 - val_loss: 0.6243 - val_
accuracy: 0.7931
Epoch 4/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4533 - accuracy: 0.8435 - val_loss: 0.6515 - val_
accuracy: 0.7912
Epoch 5/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4448 - accuracy: 0.8440 - val_loss: 0.5613 - val_
accuracy: 0.8104
Epoch 6/30
391/390 [==============================] - 86s 220ms/step - loss: 0.4396 - accuracy: 0.8459 - val_loss: 0.6198 - val_
accuracy: 0.8026
Epoch 7/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4383 - accuracy: 0.8471 - val_loss: 0.6001 - val_
accuracy: 0.8074
Epoch 8/30
391/390 [==============================] - 85s 217ms/step - loss: 0.4333 - accuracy: 0.8493 - val_loss: 0.5018 - val_
accuracy: 0.8324
Epoch 9/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4297 - accuracy: 0.8501 - val_loss: 0.5326 - val_
accuracy: 0.8265
Epoch 10/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4308 - accuracy: 0.8500 - val_loss: 0.5224 - val_
accuracy: 0.8261
Epoch 11/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4247 - accuracy: 0.8527 - val_loss: 0.6710 - val_
accuracy: 0.7927
Epoch 12/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4190 - accuracy: 0.8530 - val_loss: 0.5916 - val_
accuracy: 0.8117
Epoch 13/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4165 - accuracy: 0.8556 - val_loss: 0.6275 - val_
accuracy: 0.7996
Epoch 14/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4164 - accuracy: 0.8538 - val_loss: 0.6319 - val_
```

```
accuracy: 0.7998
Epoch 15/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4116 - accuracy: 0.8563 - val_loss: 0.6627 - val_
accuracy: 0.7924
Epoch 16/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4141 - accuracy: 0.8551 - val_loss: 0.5942 - val_
accuracy: 0.8062
Epoch 17/30
391/390 [==============================] - 85s 219ms/step - loss: 0.4093 - accuracy: 0.8582 - val_loss: 0.6255 - val_
accuracy: 0.8014
Epoch 18/30
391/390 [==============================] - 85s 218ms/step - loss: 0.4033 - accuracy: 0.8601 - val_loss: 0.5755 - val_
accuracy: 0.8152
Epoch 19/30
391/390 [==============================] - 86s 219ms/step - loss: 0.4066 - accuracy: 0.8584 - val_loss: 0.4705 - val_
accuracy: 0.8411
Epoch 20/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3985 - accuracy: 0.8588 - val_loss: 0.5003 - val_
accuracy: 0.8320
Epoch 21/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3943 - accuracy: 0.8619 - val_loss: 0.4802 - val_
accuracy: 0.8446
Epoch 22/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3953 - accuracy: 0.8623 - val_loss: 0.5594 - val_
accuracy: 0.8188
Epoch 23/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3933 - accuracy: 0.8622 - val_loss: 0.5043 - val_
accuracy: 0.8343
Epoch 24/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3888 - accuracy: 0.8633 - val_loss: 0.5373 - val_
accuracy: 0.8254
Epoch 25/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3857 - accuracy: 0.8644 - val_loss: 0.6445 - val_
accuracy: 0.8035
Epoch 26/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3846 - accuracy: 0.8644 - val_loss: 0.4975 - val_
accuracy: 0.8416
Epoch 27/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3821 - accuracy: 0.8656 - val_loss: 0.4914 - val_
accuracy: 0.8339
Epoch 28/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3779 - accuracy: 0.8664 - val_loss: 0.4787 - val_
```

```
accuracy: 0.8427
Epoch 29/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3762 - accuracy: 0.8671 - val_loss: 0.5086 - val_
accuracy: 0.8338
Epoch 30/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3763 - accuracy: 0.8679 - val_loss: 0.4758 - val_
accuracy: 0.8421
```

In [65]:
```
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])
model.save_weights(os.path.join(path, 'model_new_120epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3709 - accuracy: 0.8698 - val_loss: 0.4988 - val_
accuracy: 0.8364
Epoch 2/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3689 - accuracy: 0.8708 - val_loss: 0.5581 - val_
accuracy: 0.8205
Epoch 3/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3643 - accuracy: 0.8724 - val_loss: 0.5163 - val_
accuracy: 0.8325
Epoch 4/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3681 - accuracy: 0.8701 - val_loss: 0.5318 - val_
accuracy: 0.8264
Epoch 5/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3605 - accuracy: 0.8748 - val_loss: 0.5757 - val_
accuracy: 0.8181
Epoch 6/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3561 - accuracy: 0.8774 - val_loss: 0.5582 - val_
accuracy: 0.8253
Epoch 7/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3530 - accuracy: 0.8755 - val_loss: 0.4628 - val_
accuracy: 0.8457
Epoch 8/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3521 - accuracy: 0.8769 - val_loss: 0.5477 - val_
accuracy: 0.8254
Epoch 9/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3562 - accuracy: 0.8743 - val_loss: 0.4886 - val_
accuracy: 0.8379
Epoch 10/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3491 - accuracy: 0.8788 - val_loss: 0.6067 - val_
accuracy: 0.8095
Epoch 11/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3467 - accuracy: 0.8788 - val_loss: 0.4536 - val_
accuracy: 0.8481
Epoch 12/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3466 - accuracy: 0.8790 - val_loss: 0.5083 - val_
accuracy: 0.8370
Epoch 13/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3446 - accuracy: 0.8802 - val_loss: 0.4969 - val_
accuracy: 0.8363
Epoch 14/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3453 - accuracy: 0.8790 - val_loss: 0.5665 - val_
```

```
accuracy: 0.8253
Epoch 15/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3395 - accuracy: 0.8821 - val_loss: 0.5026 - val_
accuracy: 0.8397
Epoch 16/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3389 - accuracy: 0.8804 - val_loss: 0.4352 - val_
accuracy: 0.8544
Epoch 17/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3358 - accuracy: 0.8830 - val_loss: 0.4630 - val_
accuracy: 0.8514
Epoch 18/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3362 - accuracy: 0.8814 - val_loss: 0.5628 - val_
accuracy: 0.8256
Epoch 19/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3353 - accuracy: 0.8831 - val_loss: 0.5729 - val_
accuracy: 0.8241
Epoch 20/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3313 - accuracy: 0.8854 - val_loss: 0.5225 - val_
accuracy: 0.8360
Epoch 21/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3308 - accuracy: 0.8825 - val_loss: 0.5445 - val_
accuracy: 0.8297
Epoch 22/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3281 - accuracy: 0.8861 - val_loss: 0.5421 - val_
accuracy: 0.8326
Epoch 23/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3261 - accuracy: 0.8868 - val_loss: 0.5369 - val_
accuracy: 0.8306
Epoch 24/30
391/390 [==============================] - 85s 219ms/step - loss: 0.3245 - accuracy: 0.8853 - val_loss: 0.5940 - val_
accuracy: 0.8201
Epoch 25/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3258 - accuracy: 0.8848 - val_loss: 0.4445 - val_
accuracy: 0.8566
Epoch 26/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3189 - accuracy: 0.8874 - val_loss: 0.5087 - val_
accuracy: 0.8384
Epoch 27/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3208 - accuracy: 0.8883 - val_loss: 0.4585 - val_
accuracy: 0.8496
Epoch 28/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3174 - accuracy: 0.8893 - val_loss: 0.4624 - val_
```

```
accuracy: 0.8530
Epoch 29/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3175 - accuracy: 0.8899 - val_loss: 0.4626 - val_
accuracy: 0.8519
Epoch 30/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3142 - accuracy: 0.8893 - val_loss: 0.4622 - val_
accuracy: 0.8519
```

In [66]:
```
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])
model.save_weights(os.path.join(path, 'model_new_150epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3101 - accuracy: 0.8923 - val_loss: 0.4372 - val_
accuracy: 0.8559
Epoch 2/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3104 - accuracy: 0.8897 - val_loss: 0.4566 - val_
accuracy: 0.8507
Epoch 3/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3067 - accuracy: 0.8931 - val_loss: 0.4852 - val_
accuracy: 0.8471
Epoch 4/30
391/390 [==============================] - 86s 219ms/step - loss: 0.3078 - accuracy: 0.8914 - val_loss: 0.5366 - val_
accuracy: 0.8327
Epoch 5/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3023 - accuracy: 0.8954 - val_loss: 0.5466 - val_
accuracy: 0.8345
Epoch 6/30
391/390 [==============================] - 85s 218ms/step - loss: 0.3022 - accuracy: 0.8939 - val_loss: 0.5084 - val_
accuracy: 0.8406
Epoch 7/30
391/390 [==============================] - 85s 217ms/step - loss: 0.3026 - accuracy: 0.8930 - val_loss: 0.4417 - val_
accuracy: 0.8548
Epoch 8/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2980 - accuracy: 0.8948 - val_loss: 0.5283 - val_
accuracy: 0.8350
Epoch 9/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2988 - accuracy: 0.8951 - val_loss: 0.4844 - val_
accuracy: 0.8471
Epoch 10/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2996 - accuracy: 0.8951 - val_loss: 0.4265 - val_
accuracy: 0.8644
Epoch 11/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2982 - accuracy: 0.8953 - val_loss: 0.4392 - val_
accuracy: 0.8598
Epoch 12/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2961 - accuracy: 0.8961 - val_loss: 0.5158 - val_
accuracy: 0.8366
Epoch 13/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2906 - accuracy: 0.8976 - val_loss: 0.5237 - val_
accuracy: 0.8404
Epoch 14/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2877 - accuracy: 0.8993 - val_loss: 0.4760 - val_
```

```
accuracy: 0.8540
Epoch 15/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2878 - accuracy: 0.8988 - val_loss: 0.5600 - val_
accuracy: 0.8267
Epoch 16/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2901 - accuracy: 0.8973 - val_loss: 0.4110 - val_
accuracy: 0.8660
Epoch 17/30
391/390 [==============================] - 85s 217ms/step - loss: 0.2862 - accuracy: 0.9004 - val_loss: 0.4698 - val_
accuracy: 0.8568
Epoch 18/30
391/390 [==============================] - 85s 217ms/step - loss: 0.2834 - accuracy: 0.8990 - val_loss: 0.4755 - val_
accuracy: 0.8545
Epoch 19/30
391/390 [==============================] - 85s 217ms/step - loss: 0.2861 - accuracy: 0.9007 - val_loss: 0.5179 - val_
accuracy: 0.8377
Epoch 20/30
391/390 [==============================] - 85s 217ms/step - loss: 0.2845 - accuracy: 0.8997 - val_loss: 0.5602 - val_
accuracy: 0.8319
Epoch 21/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2776 - accuracy: 0.9037 - val_loss: 0.5171 - val_
accuracy: 0.8415
Epoch 22/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2841 - accuracy: 0.9001 - val_loss: 0.4847 - val_
accuracy: 0.8510
Epoch 23/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2803 - accuracy: 0.9016 - val_loss: 0.4602 - val_
accuracy: 0.8565
Epoch 24/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2739 - accuracy: 0.9030 - val_loss: 0.5062 - val_
accuracy: 0.8454
Epoch 25/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2740 - accuracy: 0.9025 - val_loss: 0.4562 - val_
accuracy: 0.8573
Epoch 26/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2753 - accuracy: 0.9021 - val_loss: 0.4560 - val_
accuracy: 0.8604
Epoch 27/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2757 - accuracy: 0.9038 - val_loss: 0.4813 - val_
accuracy: 0.8495
Epoch 28/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2731 - accuracy: 0.9042 - val_loss: 0.3907 - val_
```

```
accuracy: 0.8738
Epoch 29/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2697 - accuracy: 0.9049 - val_loss: 0.5149 - val_
accuracy: 0.8468
Epoch 30/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2719 - accuracy: 0.9057 - val_loss: 0.4990 - val_
accuracy: 0.8483
```

In [67]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_150epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new.hdf5'), monitor = 'val_acc')
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)
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])
model.save_weights(os.path.join(path, 'model_180epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 87s 223ms/step - loss: 0.2433 - accuracy: 0.9137 - val_loss: 0.3915 - val_
accuracy: 0.8722
Epoch 2/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2301 - accuracy: 0.9202 - val_loss: 0.3770 - val_
accuracy: 0.8757
Epoch 3/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2310 - accuracy: 0.9195 - val_loss: 0.3831 - val_
accuracy: 0.8751
Epoch 4/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2264 - accuracy: 0.9193 - val_loss: 0.3884 - val_
accuracy: 0.8770
Epoch 5/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2292 - accuracy: 0.9196 - val_loss: 0.3894 - val_
accuracy: 0.8759
Epoch 6/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2257 - accuracy: 0.9216 - val_loss: 0.3884 - val_
accuracy: 0.8767
Epoch 7/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2274 - accuracy: 0.9200 - val_loss: 0.3821 - val_
accuracy: 0.8774
Epoch 8/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2246 - accuracy: 0.9211 - val_loss: 0.3999 - val_
accuracy: 0.8717
Epoch 9/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2226 - accuracy: 0.9220 - val_loss: 0.4015 - val_
accuracy: 0.8729
Epoch 10/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2225 - accuracy: 0.9234 - val_loss: 0.3856 - val_
accuracy: 0.8771
Epoch 11/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2193 - accuracy: 0.9242 - val_loss: 0.4039 - val_
accuracy: 0.8729
Epoch 12/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2181 - accuracy: 0.9228 - val_loss: 0.3908 - val_
accuracy: 0.8766
Epoch 13/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2209 - accuracy: 0.9231 - val_loss: 0.3766 - val_
accuracy: 0.8802
Epoch 14/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2222 - accuracy: 0.9219 - val_loss: 0.3784 - val_
```

```
accuracy: 0.8764
Epoch 15/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2188 - accuracy: 0.9232 - val_loss: 0.3897 - val_
accuracy: 0.8748
Epoch 16/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2197 - accuracy: 0.9236 - val_loss: 0.3816 - val_
accuracy: 0.8769
Epoch 17/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2183 - accuracy: 0.9232 - val_loss: 0.3899 - val_
accuracy: 0.8773
Epoch 18/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2137 - accuracy: 0.9246 - val_loss: 0.3806 - val_
accuracy: 0.8774
Epoch 19/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2224 - accuracy: 0.9219 - val_loss: 0.3809 - val_
accuracy: 0.8792
Epoch 20/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2178 - accuracy: 0.9229 - val_loss: 0.4025 - val_
accuracy: 0.8729
Epoch 21/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2169 - accuracy: 0.9246 - val_loss: 0.3906 - val_
accuracy: 0.8766
Epoch 22/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2160 - accuracy: 0.9238 - val_loss: 0.3914 - val_
accuracy: 0.8773
Epoch 23/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2172 - accuracy: 0.9230 - val_loss: 0.3859 - val_
accuracy: 0.8775
Epoch 24/30
391/390 [==============================] - 86s 219ms/step - loss: 0.2104 - accuracy: 0.9263 - val_loss: 0.3916 - val_
accuracy: 0.8781
Epoch 25/30
391/390 [==============================] - 85s 219ms/step - loss: 0.2119 - accuracy: 0.9254 - val_loss: 0.3897 - val_
accuracy: 0.8770
Epoch 26/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2153 - accuracy: 0.9244 - val_loss: 0.3834 - val_
accuracy: 0.8788
Epoch 27/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2142 - accuracy: 0.9238 - val_loss: 0.3840 - val_
accuracy: 0.8770
Epoch 28/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2163 - accuracy: 0.9252 - val_loss: 0.3755 - val_
```

```
accuracy: 0.8804
Epoch 29/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2141 - accuracy: 0.9248 - val_loss: 0.3883 - val_
accuracy: 0.8767
Epoch 30/30
391/390 [==============================] - 85s 218ms/step - loss: 0.2135 - accuracy: 0.9249 - val_loss: 0.3915 - val_
accuracy: 0.8780
```

In [69]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_180epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.0001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_190epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 86s 220ms/step - loss: 0.2100 - accuracy: 0.9256 - val_loss: 0.3873 - val_
accuracy: 0.8776
Epoch 2/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2112 - accuracy: 0.9251 - val_loss: 0.3847 - val_
accuracy: 0.8796
Epoch 3/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2089 - accuracy: 0.9278 - val_loss: 0.3867 - val_
accuracy: 0.8779
Epoch 4/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2105 - accuracy: 0.9249 - val_loss: 0.3844 - val_
accuracy: 0.8788
Epoch 5/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2105 - accuracy: 0.9258 - val_loss: 0.3804 - val_
accuracy: 0.8801
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2106 - accuracy: 0.9266 - val_loss: 0.3850 - val_
accuracy: 0.8788
Epoch 7/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2071 - accuracy: 0.9275 - val_loss: 0.3820 - val_
accuracy: 0.8803
Epoch 8/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2096 - accuracy: 0.9274 - val_loss: 0.3822 - val_
accuracy: 0.8799
Epoch 9/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2094 - accuracy: 0.9261 - val_loss: 0.3818 - val_
accuracy: 0.8809
Epoch 10/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2066 - accuracy: 0.9264 - val_loss: 0.3791 - val_
accuracy: 0.8810
```

In [7]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_190epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.0001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_200epochs.h5'))
```

```
WARNING:tensorflow:From <ipython-input-7-03751f19b93e>:13: Model.fit_generator (from tensorflow.python.keras.engine.t
raining) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/10
391/390 [==============================] - 90s 231ms/step - loss: 0.2092 - accuracy: 0.9270 - val_loss: 0.3844 - val_
accuracy: 0.8793
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2074 - accuracy: 0.9278 - val_loss: 0.3801 - val_
accuracy: 0.8807
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2092 - accuracy: 0.9267 - val_loss: 0.3844 - val_
accuracy: 0.8792
Epoch 4/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2096 - accuracy: 0.9258 - val_loss: 0.3794 - val_
accuracy: 0.8807
Epoch 5/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2078 - accuracy: 0.9270 - val_loss: 0.3835 - val_
accuracy: 0.8801
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2056 - accuracy: 0.9285 - val_loss: 0.3802 - val_
accuracy: 0.8802
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2095 - accuracy: 0.9266 - val_loss: 0.3840 - val_
accuracy: 0.8784
Epoch 8/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2103 - accuracy: 0.9257 - val_loss: 0.3824 - val_
accuracy: 0.8791
Epoch 9/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2110 - accuracy: 0.9261 - val_loss: 0.3829 - val_
accuracy: 0.8789
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2077 - accuracy: 0.9267 - val_loss: 0.3820 - val_
accuracy: 0.8784
```

In [10]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_200epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.00001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_210epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2082 - accuracy: 0.9273 - val_loss: 0.3813 - val_
accuracy: 0.8790
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2097 - accuracy: 0.9261 - val_loss: 0.3818 - val_
accuracy: 0.8789
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2072 - accuracy: 0.9260 - val_loss: 0.3815 - val_
accuracy: 0.8785
Epoch 4/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2084 - accuracy: 0.9264 - val_loss: 0.3845 - val_
accuracy: 0.8778
Epoch 5/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2066 - accuracy: 0.9269 - val_loss: 0.3808 - val_
accuracy: 0.8794
Epoch 6/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2057 - accuracy: 0.9274 - val_loss: 0.3812 - val_
accuracy: 0.8794
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2081 - accuracy: 0.9272 - val_loss: 0.3805 - val_
accuracy: 0.8794
Epoch 8/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2099 - accuracy: 0.9273 - val_loss: 0.3809 - val_
accuracy: 0.8799
Epoch 9/10
391/390 [==============================] - 85s 217ms/step - loss: 0.2097 - accuracy: 0.9259 - val_loss: 0.3816 - val_
accuracy: 0.8793
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2095 - accuracy: 0.9261 - val_loss: 0.3824 - val_
accuracy: 0.8789
```

In [11]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_210epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.000001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_220epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2068 - accuracy: 0.9271 - val_loss: 0.3794 - val_
accuracy: 0.8804
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2090 - accuracy: 0.9275 - val_loss: 0.3814 - val_
accuracy: 0.8790
Epoch 3/10
391/390 [==============================] - 86s 219ms/step - loss: 0.2078 - accuracy: 0.9266 - val_loss: 0.3826 - val_
accuracy: 0.8792
Epoch 4/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2065 - accuracy: 0.9278 - val_loss: 0.3835 - val_
accuracy: 0.8788
Epoch 5/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2078 - accuracy: 0.9262 - val_loss: 0.3826 - val_
accuracy: 0.8790
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2086 - accuracy: 0.9270 - val_loss: 0.3830 - val_
accuracy: 0.8792
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2079 - accuracy: 0.9275 - val_loss: 0.3825 - val_
accuracy: 0.8792
Epoch 8/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2091 - accuracy: 0.9262 - val_loss: 0.3839 - val_
accuracy: 0.8786
Epoch 9/10
391/390 [==============================] - 86s 220ms/step - loss: 0.2077 - accuracy: 0.9271 - val_loss: 0.3822 - val_
accuracy: 0.8786
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2117 - accuracy: 0.9254 - val_loss: 0.3831 - val_
accuracy: 0.8788
```

In [12]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_220epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.0000001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_230epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 86s 220ms/step - loss: 0.2070 - accuracy: 0.9283 - val_loss: 0.3819 - val_
accuracy: 0.8791
Epoch 2/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2054 - accuracy: 0.9287 - val_loss: 0.3818 - val_
accuracy: 0.8792
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2074 - accuracy: 0.9272 - val_loss: 0.3823 - val_
accuracy: 0.8792
Epoch 4/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2072 - accuracy: 0.9265 - val_loss: 0.3814 - val_
accuracy: 0.8795
Epoch 5/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2095 - accuracy: 0.9271 - val_loss: 0.3829 - val_
accuracy: 0.8786
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2079 - accuracy: 0.9281 - val_loss: 0.3823 - val_
accuracy: 0.8789
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2085 - accuracy: 0.9272 - val_loss: 0.3807 - val_
accuracy: 0.8799
Epoch 8/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2083 - accuracy: 0.9257 - val_loss: 0.3820 - val_
accuracy: 0.8792
Epoch 9/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2116 - accuracy: 0.9263 - val_loss: 0.3815 - val_
accuracy: 0.8792
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2061 - accuracy: 0.9275 - val_loss: 0.3823 - val_
accuracy: 0.8788
```

In [13]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_230epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
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)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_240epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 86s 220ms/step - loss: 0.2113 - accuracy: 0.9245 - val_loss: 0.3870 - val_
accuracy: 0.8771
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2130 - accuracy: 0.9260 - val_loss: 0.4107 - val_
accuracy: 0.8719
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2150 - accuracy: 0.9246 - val_loss: 0.3936 - val_
accuracy: 0.8766
Epoch 4/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2128 - accuracy: 0.9247 - val_loss: 0.3988 - val_
accuracy: 0.8765
Epoch 5/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2096 - accuracy: 0.9269 - val_loss: 0.3807 - val_
accuracy: 0.8807
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2110 - accuracy: 0.9252 - val_loss: 0.3818 - val_
accuracy: 0.8801
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2118 - accuracy: 0.9261 - val_loss: 0.3922 - val_
accuracy: 0.8785
Epoch 8/10
391/390 [==============================] - 85s 219ms/step - loss: 0.2134 - accuracy: 0.9242 - val_loss: 0.3694 - val_
accuracy: 0.8831
Epoch 9/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2099 - accuracy: 0.9258 - val_loss: 0.3919 - val_
accuracy: 0.8776
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.2096 - accuracy: 0.9257 - val_loss: 0.3782 - val_
accuracy: 0.8804
```

In [14]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_240epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
datagen = ImageDataGenerator(rotation_range = 15, horizontal_flip = True, width_shift_range = 0.1, height_shift_range
= 0.1)
datagen.fit(X_train)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_250epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 86s 220ms/step - loss: 0.1612 - accuracy: 0.9447 - val_loss: 0.3827 - val_
accuracy: 0.8804
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1590 - accuracy: 0.9449 - val_loss: 0.3883 - val_
accuracy: 0.8795
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1569 - accuracy: 0.9460 - val_loss: 0.3925 - val_
accuracy: 0.8780
Epoch 4/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1566 - accuracy: 0.9457 - val_loss: 0.3749 - val_
accuracy: 0.8807
Epoch 5/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1561 - accuracy: 0.9455 - val_loss: 0.3712 - val_
accuracy: 0.8816
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1573 - accuracy: 0.9449 - val_loss: 0.3715 - val_
accuracy: 0.8849
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1581 - accuracy: 0.9456 - val_loss: 0.3800 - val_
accuracy: 0.8820
Epoch 8/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1588 - accuracy: 0.9450 - val_loss: 0.3781 - val_
accuracy: 0.8819
Epoch 9/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1545 - accuracy: 0.9466 - val_loss: 0.3744 - val_
accuracy: 0.8832
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1563 - accuracy: 0.9457 - val_loss: 0.3868 - val_
accuracy: 0.8814
```

In [15]:
```
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_260epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1524 - accuracy: 0.9469 - val_loss: 0.3701 - val_
accuracy: 0.8845
Epoch 2/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1543 - accuracy: 0.9462 - val_loss: 0.3691 - val_
accuracy: 0.8848
Epoch 3/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1536 - accuracy: 0.9461 - val_loss: 0.3690 - val_
accuracy: 0.8859
Epoch 4/10
391/390 [==============================] - 85s 217ms/step - loss: 0.1521 - accuracy: 0.9464 - val_loss: 0.3850 - val_
accuracy: 0.8815
Epoch 5/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1531 - accuracy: 0.9476 - val_loss: 0.3802 - val_
accuracy: 0.8843
Epoch 6/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1508 - accuracy: 0.9469 - val_loss: 0.3827 - val_
accuracy: 0.8834
Epoch 7/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1515 - accuracy: 0.9465 - val_loss: 0.3807 - val_
accuracy: 0.8819
Epoch 8/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1526 - accuracy: 0.9466 - val_loss: 0.3919 - val_
accuracy: 0.8818
Epoch 9/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1509 - accuracy: 0.9462 - val_loss: 0.3916 - val_
accuracy: 0.8820
Epoch 10/10
391/390 [==============================] - 85s 218ms/step - loss: 0.1497 - accuracy: 0.9482 - val_loss: 0.3830 - val_
accuracy: 0.8816
```

In [5]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_260epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
datagen = ImageDataGenerator(rotation_range = 15, horizontal_flip = True, width_shift_range = 0.1, height_shift_range
= 0.1)
datagen.fit(X_train)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_270epochs.h5'))
```

```
WARNING:tensorflow:From <ipython-input-5-19c5a1d32407>:13: Model.fit_generator (from tensorflow.python.keras.engine.t
raining) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/10
391/390 [==============================] - 50s 128ms/step - loss: 0.1541 - accuracy: 0.9451 - val_loss: 0.3879 - val_
accuracy: 0.8819
Epoch 2/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1509 - accuracy: 0.9480 - val_loss: 0.3815 - val_
accuracy: 0.8842
Epoch 3/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1506 - accuracy: 0.9481 - val_loss: 0.3821 - val_
accuracy: 0.8831
Epoch 4/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1508 - accuracy: 0.9472 - val_loss: 0.3806 - val_
accuracy: 0.8836
Epoch 5/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1486 - accuracy: 0.9480 - val_loss: 0.3823 - val_
accuracy: 0.8814
Epoch 6/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1492 - accuracy: 0.9473 - val_loss: 0.3940 - val_
accuracy: 0.8799
Epoch 7/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1506 - accuracy: 0.9474 - val_loss: 0.3785 - val_
accuracy: 0.8837
Epoch 8/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1495 - accuracy: 0.9476 - val_loss: 0.3898 - val_
accuracy: 0.8812
Epoch 9/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1498 - accuracy: 0.9478 - val_loss: 0.3766 - val_
accuracy: 0.8850
Epoch 10/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1477 - accuracy: 0.9478 - val_loss: 0.3859 - val_
accuracy: 0.8806
```

In [6]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_270epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
datagen = ImageDataGenerator(rotation_range = 15, horizontal_flip = True, width_shift_range = 0.1)
datagen.fit(X_train)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_280epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1349 - accuracy: 0.9532 - val_loss: 0.3741 - val_
accuracy: 0.8838
Epoch 2/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1328 - accuracy: 0.9543 - val_loss: 0.3691 - val_
accuracy: 0.8856
Epoch 3/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1323 - accuracy: 0.9548 - val_loss: 0.3743 - val_
accuracy: 0.8845
Epoch 4/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1279 - accuracy: 0.9560 - val_loss: 0.3715 - val_
accuracy: 0.8852
Epoch 5/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1316 - accuracy: 0.9532 - val_loss: 0.3816 - val_
accuracy: 0.8836
Epoch 6/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1316 - accuracy: 0.9539 - val_loss: 0.3820 - val_
accuracy: 0.8848
Epoch 7/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1308 - accuracy: 0.9542 - val_loss: 0.3811 - val_
accuracy: 0.8849
Epoch 8/10
391/390 [==============================] - 49s 125ms/step - loss: 0.1291 - accuracy: 0.9554 - val_loss: 0.3827 - val_
accuracy: 0.8859
Epoch 9/10
391/390 [==============================] - 49s 125ms/step - loss: 0.1306 - accuracy: 0.9546 - val_loss: 0.3694 - val_
accuracy: 0.8854
Epoch 10/10
391/390 [==============================] - 48s 123ms/step - loss: 0.1285 - accuracy: 0.9552 - val_loss: 0.3859 - val_
accuracy: 0.8842
```

In [7]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_280epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
datagen = ImageDataGenerator(rotation_range = 15, horizontal_flip = True)
datagen.fit(X_train)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_290epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1197 - accuracy: 0.9582 - val_loss: 0.3701 - val_
accuracy: 0.8882
Epoch 2/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1195 - accuracy: 0.9582 - val_loss: 0.3649 - val_
accuracy: 0.8866
Epoch 3/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1196 - accuracy: 0.9577 - val_loss: 0.3754 - val_
accuracy: 0.8863
Epoch 4/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1162 - accuracy: 0.9602 - val_loss: 0.3650 - val_
accuracy: 0.8889
Epoch 5/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1178 - accuracy: 0.9601 - val_loss: 0.3635 - val_
accuracy: 0.8904
Epoch 6/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1125 - accuracy: 0.9615 - val_loss: 0.3636 - val_
accuracy: 0.8883
Epoch 7/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1140 - accuracy: 0.9605 - val_loss: 0.3668 - val_
accuracy: 0.8876
Epoch 8/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1166 - accuracy: 0.9605 - val_loss: 0.3595 - val_
accuracy: 0.8881
Epoch 9/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1146 - accuracy: 0.9612 - val_loss: 0.3802 - val_
accuracy: 0.8858
Epoch 10/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1117 - accuracy: 0.9627 - val_loss: 0.3603 - val_
accuracy: 0.8890
```

In [11]:
```
import os
import keras
from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import load_model
path = os.path.abspath('gdrive/My Drive/CNN_CIFAR')
path = os.path.join(path, 'Cifar_training')
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.load_weights(os.path.join(path, 'model_new_290epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.0001)
checkpoint = ModelCheckpoint(os.path.join(path, 'cnn_cifar_model_new1.hdf5'), monitor = 'val_acc')
datagen = ImageDataGenerator(rotation_range = 15)
datagen.fit(X_train)
model.fit_generator(datagen.flow(X_train, y_train, batch_size), steps_per_epoch = X_train.shape[0]/batch_size, epochs
= 10, validation_data =(X_test, y_test), callbacks = [checkpoint])
model.save_weights(os.path.join(path, 'model_new_300epochs.h5'))
```

```
Epoch 1/10
391/390 [==============================] - 49s 125ms/step - loss: 0.1119 - accuracy: 0.9613 - val_loss: 0.3632 - val_
accuracy: 0.8881
Epoch 2/10
391/390 [==============================] - 49s 126ms/step - loss: 0.1093 - accuracy: 0.9629 - val_loss: 0.3631 - val_
accuracy: 0.8886
Epoch 3/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1092 - accuracy: 0.9618 - val_loss: 0.3632 - val_
accuracy: 0.8890
Epoch 4/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1093 - accuracy: 0.9621 - val_loss: 0.3627 - val_
accuracy: 0.8889
Epoch 5/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1102 - accuracy: 0.9622 - val_loss: 0.3644 - val_
accuracy: 0.8886
Epoch 6/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1085 - accuracy: 0.9629 - val_loss: 0.3625 - val_
accuracy: 0.8892
Epoch 7/10
391/390 [==============================] - 49s 125ms/step - loss: 0.1077 - accuracy: 0.9626 - val_loss: 0.3634 - val_
accuracy: 0.8883
Epoch 8/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1083 - accuracy: 0.9621 - val_loss: 0.3630 - val_
accuracy: 0.8885
Epoch 9/10
391/390 [==============================] - 49s 124ms/step - loss: 0.1084 - accuracy: 0.9627 - val_loss: 0.3617 - val_
accuracy: 0.8892
Epoch 10/10
391/390 [==============================] - 48s 124ms/step - loss: 0.1059 - accuracy: 0.9628 - val_loss: 0.3620 - val_
accuracy: 0.8894
```

In [12]:
```
from tensorflow.keras.models import load_model
model = load_model('/content/gdrive/My Drive/CNN_CIFAR/Cifar_training/cnn_cifar_model_new1.hdf5')
model.evaluate(X_test, y_test)
```

```
313/313 [==============================] - 4s 12ms/step - loss: 0.3620 - accuracy: 0.8894
```

Out[12]:
```
[0.3620176315307617, 0.8894000053405762]
```