In [0]:

```
from google.colab import drive
drive.mount('/content/gdrive')
import os
path = os.path.abspath('gdrive/My Drive/Inkers')
#os.mkdir(os.path.join(path, '4thIter'))
path = os.path.join(path, '4thIter')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call
drive.mount("/content/gdrive", force_remount=True).
```

In [0]:

```
import keras
from keras.datasets import cifar10
from keras.models import Model, Sequential
from keras.layers import Dense, Dropout, Flatten, Input, AveragePooling2D, merge, Activation, Spati
alDropout2D
from keras.layers import Conv2D, MaxPooling2D, BatchNormalization, SeparableConv2D
from keras.layers import Concatenate
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
from keras import backend as k

# Don't pre-allocate memory; allocate as-needed
config = tf.ConfigProto()
config.gpu_options.allow_growth = True

# Create a session with the above options specified.
k.tensorflow_backend.set_session(tf.Session(config=config))
```

In [0]:

```
# Hyperparameters
batch_size = 128
num_classes = 10
epochs = 250
l = 12
num_filter = 36 #added 24 more filters
compression = 0.5
dropout_rate = 0.2
```

In [0]:

```
# 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 = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
```

In [0]:

```
# Dense Block
# removed the dropout
```

```
def add_denseblock(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    temp = input
    for _ in range(l):
        BatchNorm = BatchNormalization()(temp)
        relu = Activation('relu')(BatchNorm)
        Conv2D_3_3 = Conv2D(int(num_filter*compression), (3,3), use_bias=False ,padding='same')(rel
u)
        #if dropout_rate>0:
         # Conv2D_3_3 = Dropout2D(dropout_rate)(Conv2D_3_3)
        concat = Concatenate(axis=-1)([temp,Conv2D_3_3])

        temp = concat

    return temp
```

In [0]:

```
def add_transition(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    BatchNorm = BatchNormalization()(input)
    relu = Activation('relu')(BatchNorm)
    Conv2D_BottleNeck = Conv2D(int(num_filter*compression), (1,1), use_bias=False,
kernel_regularizer = regularizers.l1() ,padding='same')(relu)
    #if dropout_rate>0:
      #Conv2D_BottleNeck = Dropout2D(dropout_rate)(Conv2D_BottleNeck)
    avg = AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)

    return avg
```

In [0]:

```
# converted the last Dense Layer to a Fully Convolution N/w as use of Dense Layer was prohibited
def output_layer(input):
    global compression
    BatchNorm = BatchNormalization()(input)
    relu = Activation('relu')(BatchNorm)
    AvgPooling = AveragePooling2D(pool_size=(2,2))(relu)
    temp = Conv2D(num_classes, kernel_size = (2,2))(AvgPooling)
    output = Activation('softmax')(temp)
    flat = Flatten()(output)

    return flat
```

In [0]:

```
num_filter = 36
dropout_rate = 0.2
l= 12
input = Input(shape=(img_height, img_width, channel,))
First_Conv2D = Conv2D(num_filter, (3,3), use_bias=False ,padding='same')(input)

First_Block = add_denseblock(First_Conv2D, num_filter, dropout_rate)
First_Transition = add_transition(First_Block, num_filter, dropout_rate)

Second_Block = add_denseblock(First_Transition, num_filter, dropout_rate)
Second_Transition = add_transition(Second_Block, num_filter, dropout_rate)

Third_Block = add_denseblock(Second_Transition, num_filter, dropout_rate)
Third_Transition = add_transition(Third_Block, num_filter, dropout_rate)

Last_Block = add_denseblock(Third_Transition,  num_filter, dropout_rate)
output = output_layer(Last_Block)
```

In [0]:

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

```
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
input_2 (InputLayer)         (None, 32, 32, 3)         0
```

input_2 (InputLayer)	(None, 32, 32, 3)	0	
conv2d_54 (Conv2D)	(None, 32, 32, 36)	972	input_2[0][0]
batch_normalization_53 (BatchNo	(None, 32, 32, 36)	144	conv2d_54[0][0]
activation_54 (Activation)	(None, 32, 32, 36)	0	batch_normalization_53[0][0]
conv2d_55 (Conv2D)	(None, 32, 32, 18)	5832	activation_54[0][0]
concatenate_49 (Concatenate)	(None, 32, 32, 54)	0	conv2d_54[0][0] conv2d_55[0][0]
batch_normalization_54 (BatchNo	(None, 32, 32, 54)	216	concatenate_49[0][0]
activation_55 (Activation)	(None, 32, 32, 54)	0	batch_normalization_54[0][0]
conv2d_56 (Conv2D)	(None, 32, 32, 18)	8748	activation_55[0][0]
concatenate_50 (Concatenate)	(None, 32, 32, 72)	0	concatenate_49[0][0] conv2d_56[0][0]
batch_normalization_55 (BatchNo	(None, 32, 32, 72)	288	concatenate_50[0][0]
activation_56 (Activation)	(None, 32, 32, 72)	0	batch_normalization_55[0][0]
conv2d_57 (Conv2D)	(None, 32, 32, 18)	11664	activation_56[0][0]
concatenate_51 (Concatenate)	(None, 32, 32, 90)	0	concatenate_50[0][0] conv2d_57[0][0]
batch_normalization_56 (BatchNo	(None, 32, 32, 90)	360	concatenate_51[0][0]
activation_57 (Activation)	(None, 32, 32, 90)	0	batch_normalization_56[0][0]
conv2d_58 (Conv2D)	(None, 32, 32, 18)	14580	activation_57[0][0]
concatenate_52 (Concatenate)	(None, 32, 32, 108)	0	concatenate_51[0][0] conv2d_58[0][0]
batch_normalization_57 (BatchNo	(None, 32, 32, 108)	432	concatenate_52[0][0]
activation_58 (Activation)	(None, 32, 32, 108)	0	batch_normalization_57[0][0]
conv2d_59 (Conv2D)	(None, 32, 32, 18)	17496	activation_58[0][0]
concatenate_53 (Concatenate)	(None, 32, 32, 126)	0	concatenate_52[0][0] conv2d_59[0][0]
batch_normalization_58 (BatchNo	(None, 32, 32, 126)	504	concatenate_53[0][0]
activation_59 (Activation)	(None, 32, 32, 126)	0	batch_normalization_58[0][0]
conv2d_60 (Conv2D)	(None, 32, 32, 18)	20412	activation_59[0][0]
concatenate_54 (Concatenate)	(None, 32, 32, 144)	0	concatenate_53[0][0] conv2d_60[0][0]
batch_normalization_59 (BatchNo	(None, 32, 32, 144)	576	concatenate_54[0][0]
activation_60 (Activation)	(None, 32, 32, 144)	0	batch_normalization_59[0][0]
conv2d_61 (Conv2D)	(None, 32, 32, 18)	23328	activation_60[0][0]
concatenate_55 (Concatenate)	(None, 32, 32, 162)	0	concatenate_54[0][0] conv2d_61[0][0]
batch_normalization_60 (BatchNo	(None, 32, 32, 162)	648	concatenate_55[0][0]
activation_61 (Activation)	(None, 32, 32, 162)	0	batch_normalization_60[0][0]
conv2d_62 (Conv2D)	(None, 32, 32, 18)	26244	activation_61[0][0]
concatenate_56 (Concatenate)	(None, 32, 32, 180)	0	concatenate_55[0][0] conv2d_62[0][0]
batch_normalization_61 (BatchNo	(None, 32, 32, 180)	720	concatenate_56[0][0]

activation_62 (Activation)	(None, 32, 32, 180)	0	batch_normalization_61[0][0]
conv2d_63 (Conv2D)	(None, 32, 32, 18)	29160	activation_62[0][0]
concatenate_57 (Concatenate)	(None, 32, 32, 198)	0	concatenate_56[0][0] conv2d_63[0][0]
batch_normalization_62 (BatchNo	(None, 32, 32, 198)	792	concatenate_57[0][0]
activation_63 (Activation)	(None, 32, 32, 198)	0	batch_normalization_62[0][0]
conv2d_64 (Conv2D)	(None, 32, 32, 18)	32076	activation_63[0][0]
concatenate_58 (Concatenate)	(None, 32, 32, 216)	0	concatenate_57[0][0] conv2d_64[0][0]
batch_normalization_63 (BatchNo	(None, 32, 32, 216)	864	concatenate_58[0][0]
activation_64 (Activation)	(None, 32, 32, 216)	0	batch_normalization_63[0][0]
conv2d_65 (Conv2D)	(None, 32, 32, 18)	34992	activation_64[0][0]
concatenate_59 (Concatenate)	(None, 32, 32, 234)	0	concatenate_58[0][0] conv2d_65[0][0]
batch_normalization_64 (BatchNo	(None, 32, 32, 234)	936	concatenate_59[0][0]
activation_65 (Activation)	(None, 32, 32, 234)	0	batch_normalization_64[0][0]
conv2d_66 (Conv2D)	(None, 32, 32, 18)	37908	activation_65[0][0]
concatenate_60 (Concatenate)	(None, 32, 32, 252)	0	concatenate_59[0][0] conv2d_66[0][0]
batch_normalization_65 (BatchNo	(None, 32, 32, 252)	1008	concatenate_60[0][0]
activation_66 (Activation)	(None, 32, 32, 252)	0	batch_normalization_65[0][0]
conv2d_67 (Conv2D)	(None, 32, 32, 18)	4536	activation_66[0][0]
average_pooling2d_5 (AveragePoo	(None, 16, 16, 18)	0	conv2d_67[0][0]
batch_normalization_66 (BatchNo	(None, 16, 16, 18)	72	average_pooling2d_5[0][0]
activation_67 (Activation)	(None, 16, 16, 18)	0	batch_normalization_66[0][0]
conv2d_68 (Conv2D)	(None, 16, 16, 18)	2916	activation_67[0][0]
concatenate_61 (Concatenate)	(None, 16, 16, 36)	0	average_pooling2d_5[0][0] conv2d_68[0][0]
batch_normalization_67 (BatchNo	(None, 16, 16, 36)	144	concatenate_61[0][0]
activation_68 (Activation)	(None, 16, 16, 36)	0	batch_normalization_67[0][0]
conv2d_69 (Conv2D)	(None, 16, 16, 18)	5832	activation_68[0][0]
concatenate_62 (Concatenate)	(None, 16, 16, 54)	0	concatenate_61[0][0] conv2d_69[0][0]
batch_normalization_68 (BatchNo	(None, 16, 16, 54)	216	concatenate_62[0][0]
activation_69 (Activation)	(None, 16, 16, 54)	0	batch_normalization_68[0][0]
conv2d_70 (Conv2D)	(None, 16, 16, 18)	8748	activation_69[0][0]
concatenate_63 (Concatenate)	(None, 16, 16, 72)	0	concatenate_62[0][0] conv2d_70[0][0]
batch_normalization_69 (BatchNo	(None, 16, 16, 72)	288	concatenate_63[0][0]
activation_70 (Activation)	(None, 16, 16, 72)	0	batch_normalization_69[0][0]
conv2d_71 (Conv2D)	(None, 16, 16, 18)	11664	activation_70[0][0]
concatenate_64 (Concatenate)	(None, 16, 16, 90)	0	concatenate_63[0][0] conv2d_71[0][0]

				conv2d_71[0][0]
batch_normalization_70 (BatchNo	(None, 16, 16, 90)	360	concatenate_64[0][0]	
activation_71 (Activation)	(None, 16, 16, 90)	0	batch_normalization_70[0][0]	
conv2d_72 (Conv2D)	(None, 16, 16, 18)	14580	activation_71[0][0]	
concatenate_65 (Concatenate)	(None, 16, 16, 108)	0	concatenate_64[0][0] conv2d_72[0][0]	
batch_normalization_71 (BatchNo	(None, 16, 16, 108)	432	concatenate_65[0][0]	
activation_72 (Activation)	(None, 16, 16, 108)	0	batch_normalization_71[0][0]	
conv2d_73 (Conv2D)	(None, 16, 16, 18)	17496	activation_72[0][0]	
concatenate_66 (Concatenate)	(None, 16, 16, 126)	0	concatenate_65[0][0] conv2d_73[0][0]	
batch_normalization_72 (BatchNo	(None, 16, 16, 126)	504	concatenate_66[0][0]	
activation_73 (Activation)	(None, 16, 16, 126)	0	batch_normalization_72[0][0]	
conv2d_74 (Conv2D)	(None, 16, 16, 18)	20412	activation_73[0][0]	
concatenate_67 (Concatenate)	(None, 16, 16, 144)	0	concatenate_66[0][0] conv2d_74[0][0]	
batch_normalization_73 (BatchNo	(None, 16, 16, 144)	576	concatenate_67[0][0]	
activation_74 (Activation)	(None, 16, 16, 144)	0	batch_normalization_73[0][0]	
conv2d_75 (Conv2D)	(None, 16, 16, 18)	23328	activation_74[0][0]	
concatenate_68 (Concatenate)	(None, 16, 16, 162)	0	concatenate_67[0][0] conv2d_75[0][0]	
batch_normalization_74 (BatchNo	(None, 16, 16, 162)	648	concatenate_68[0][0]	
activation_75 (Activation)	(None, 16, 16, 162)	0	batch_normalization_74[0][0]	
conv2d_76 (Conv2D)	(None, 16, 16, 18)	26244	activation_75[0][0]	
concatenate_69 (Concatenate)	(None, 16, 16, 180)	0	concatenate_68[0][0] conv2d_76[0][0]	
batch_normalization_75 (BatchNo	(None, 16, 16, 180)	720	concatenate_69[0][0]	
activation_76 (Activation)	(None, 16, 16, 180)	0	batch_normalization_75[0][0]	
conv2d_77 (Conv2D)	(None, 16, 16, 18)	29160	activation_76[0][0]	
concatenate_70 (Concatenate)	(None, 16, 16, 198)	0	concatenate_69[0][0] conv2d_77[0][0]	
batch_normalization_76 (BatchNo	(None, 16, 16, 198)	792	concatenate_70[0][0]	
activation_77 (Activation)	(None, 16, 16, 198)	0	batch_normalization_76[0][0]	
conv2d_78 (Conv2D)	(None, 16, 16, 18)	32076	activation_77[0][0]	
concatenate_71 (Concatenate)	(None, 16, 16, 216)	0	concatenate_70[0][0] conv2d_78[0][0]	
batch_normalization_77 (BatchNo	(None, 16, 16, 216)	864	concatenate_71[0][0]	
activation_78 (Activation)	(None, 16, 16, 216)	0	batch_normalization_77[0][0]	
conv2d_79 (Conv2D)	(None, 16, 16, 18)	34992	activation_78[0][0]	
concatenate_72 (Concatenate)	(None, 16, 16, 234)	0	concatenate_71[0][0] conv2d_79[0][0]	
batch_normalization_78 (BatchNo	(None, 16, 16, 234)	936	concatenate_72[0][0]	
activation_79 (Activation)	(None, 16, 16, 234)	0	batch_normalization_78[0][0]	

conv2d_80 (Conv2D)	(None, 16, 16, 18)	4212	activation_79[0][0]
average_pooling2d_6 (AveragePoo	(None, 8, 8, 18)	0	conv2d_80[0][0]
batch_normalization_79 (BatchNo	(None, 8, 8, 18)	72	average_pooling2d_6[0][0]
activation_80 (Activation)	(None, 8, 8, 18)	0	batch_normalization_79[0][0]
conv2d_81 (Conv2D)	(None, 8, 8, 18)	2916	activation_80[0][0]
concatenate_73 (Concatenate)	(None, 8, 8, 36)	0	average_pooling2d_6[0][0] conv2d_81[0][0]
batch_normalization_80 (BatchNo	(None, 8, 8, 36)	144	concatenate_73[0][0]
activation_81 (Activation)	(None, 8, 8, 36)	0	batch_normalization_80[0][0]
conv2d_82 (Conv2D)	(None, 8, 8, 18)	5832	activation_81[0][0]
concatenate_74 (Concatenate)	(None, 8, 8, 54)	0	concatenate_73[0][0] conv2d_82[0][0]
batch_normalization_81 (BatchNo	(None, 8, 8, 54)	216	concatenate_74[0][0]
activation_82 (Activation)	(None, 8, 8, 54)	0	batch_normalization_81[0][0]
conv2d_83 (Conv2D)	(None, 8, 8, 18)	8748	activation_82[0][0]
concatenate_75 (Concatenate)	(None, 8, 8, 72)	0	concatenate_74[0][0] conv2d_83[0][0]
batch_normalization_82 (BatchNo	(None, 8, 8, 72)	288	concatenate_75[0][0]
activation_83 (Activation)	(None, 8, 8, 72)	0	batch_normalization_82[0][0]
conv2d_84 (Conv2D)	(None, 8, 8, 18)	11664	activation_83[0][0]
concatenate_76 (Concatenate)	(None, 8, 8, 90)	0	concatenate_75[0][0] conv2d_84[0][0]
batch_normalization_83 (BatchNo	(None, 8, 8, 90)	360	concatenate_76[0][0]
activation_84 (Activation)	(None, 8, 8, 90)	0	batch_normalization_83[0][0]
conv2d_85 (Conv2D)	(None, 8, 8, 18)	14580	activation_84[0][0]
concatenate_77 (Concatenate)	(None, 8, 8, 108)	0	concatenate_76[0][0] conv2d_85[0][0]
batch_normalization_84 (BatchNo	(None, 8, 8, 108)	432	concatenate_77[0][0]
activation_85 (Activation)	(None, 8, 8, 108)	0	batch_normalization_84[0][0]
conv2d_86 (Conv2D)	(None, 8, 8, 18)	17496	activation_85[0][0]
concatenate_78 (Concatenate)	(None, 8, 8, 126)	0	concatenate_77[0][0] conv2d_86[0][0]
batch_normalization_85 (BatchNo	(None, 8, 8, 126)	504	concatenate_78[0][0]
activation_86 (Activation)	(None, 8, 8, 126)	0	batch_normalization_85[0][0]
conv2d_87 (Conv2D)	(None, 8, 8, 18)	20412	activation_86[0][0]
concatenate_79 (Concatenate)	(None, 8, 8, 144)	0	concatenate_78[0][0] conv2d_87[0][0]
batch_normalization_86 (BatchNo	(None, 8, 8, 144)	576	concatenate_79[0][0]
activation_87 (Activation)	(None, 8, 8, 144)	0	batch_normalization_86[0][0]
conv2d_88 (Conv2D)	(None, 8, 8, 18)	23328	activation_87[0][0]
concatenate_80 (Concatenate)	(None, 8, 8, 162)	0	concatenate_79[0][0] conv2d_88[0][0]
batch_normalization_87 (BatchNo	(None, 8, 8, 162)	648	concatenate_80[0][0]

batch_normalization_87 (BatchNo	(None, 8, 8, 162)	648	concatenate_80[0][0]
activation_88 (Activation)	(None, 8, 8, 162)	0	batch_normalization_87[0][0]
conv2d_89 (Conv2D)	(None, 8, 8, 18)	26244	activation_88[0][0]
concatenate_81 (Concatenate)	(None, 8, 8, 180)	0	concatenate_80[0][0] conv2d_89[0][0]
batch_normalization_88 (BatchNo	(None, 8, 8, 180)	720	concatenate_81[0][0]
activation_89 (Activation)	(None, 8, 8, 180)	0	batch_normalization_88[0][0]
conv2d_90 (Conv2D)	(None, 8, 8, 18)	29160	activation_89[0][0]
concatenate_82 (Concatenate)	(None, 8, 8, 198)	0	concatenate_81[0][0] conv2d_90[0][0]
batch_normalization_89 (BatchNo	(None, 8, 8, 198)	792	concatenate_82[0][0]
activation_90 (Activation)	(None, 8, 8, 198)	0	batch_normalization_89[0][0]
conv2d_91 (Conv2D)	(None, 8, 8, 18)	32076	activation_90[0][0]
concatenate_83 (Concatenate)	(None, 8, 8, 216)	0	concatenate_82[0][0] conv2d_91[0][0]
batch_normalization_90 (BatchNo	(None, 8, 8, 216)	864	concatenate_83[0][0]
activation_91 (Activation)	(None, 8, 8, 216)	0	batch_normalization_90[0][0]
conv2d_92 (Conv2D)	(None, 8, 8, 18)	34992	activation_91[0][0]
concatenate_84 (Concatenate)	(None, 8, 8, 234)	0	concatenate_83[0][0] conv2d_92[0][0]
batch_normalization_91 (BatchNo	(None, 8, 8, 234)	936	concatenate_84[0][0]
activation_92 (Activation)	(None, 8, 8, 234)	0	batch_normalization_91[0][0]
conv2d_93 (Conv2D)	(None, 8, 8, 18)	4212	activation_92[0][0]
average_pooling2d_7 (AveragePoo	(None, 4, 4, 18)	0	conv2d_93[0][0]
batch_normalization_92 (BatchNo	(None, 4, 4, 18)	72	average_pooling2d_7[0][0]
activation_93 (Activation)	(None, 4, 4, 18)	0	batch_normalization_92[0][0]
conv2d_94 (Conv2D)	(None, 4, 4, 18)	2916	activation_93[0][0]
concatenate_85 (Concatenate)	(None, 4, 4, 36)	0	average_pooling2d_7[0][0] conv2d_94[0][0]
batch_normalization_93 (BatchNo	(None, 4, 4, 36)	144	concatenate_85[0][0]
activation_94 (Activation)	(None, 4, 4, 36)	0	batch_normalization_93[0][0]
conv2d_95 (Conv2D)	(None, 4, 4, 18)	5832	activation_94[0][0]
concatenate_86 (Concatenate)	(None, 4, 4, 54)	0	concatenate_85[0][0] conv2d_95[0][0]
batch_normalization_94 (BatchNo	(None, 4, 4, 54)	216	concatenate_86[0][0]
activation_95 (Activation)	(None, 4, 4, 54)	0	batch_normalization_94[0][0]
conv2d_96 (Conv2D)	(None, 4, 4, 18)	8748	activation_95[0][0]
concatenate_87 (Concatenate)	(None, 4, 4, 72)	0	concatenate_86[0][0] conv2d_96[0][0]
batch_normalization_95 (BatchNo	(None, 4, 4, 72)	288	concatenate_87[0][0]
activation_96 (Activation)	(None, 4, 4, 72)	0	batch_normalization_95[0][0]
conv2d_97 (Conv2D)	(None, 4, 4, 18)	11664	activation_96[0][0]
concatenate_88 (Concatenate)	(None, 4, 4, 90)	0	concatenate_87[0][0]

concatenate_88 (Concatenate)	(None, 4, 4, 90)	0	concatenate_87[0][0] conv2d_97[0][0]
batch_normalization_96 (BatchNo	(None, 4, 4, 90)	360	concatenate_88[0][0]
activation_97 (Activation)	(None, 4, 4, 90)	0	batch_normalization_96[0][0]
conv2d_98 (Conv2D)	(None, 4, 4, 18)	14580	activation_97[0][0]
concatenate_89 (Concatenate)	(None, 4, 4, 108)	0	concatenate_88[0][0] conv2d_98[0][0]
batch_normalization_97 (BatchNo	(None, 4, 4, 108)	432	concatenate_89[0][0]
activation_98 (Activation)	(None, 4, 4, 108)	0	batch_normalization_97[0][0]
conv2d_99 (Conv2D)	(None, 4, 4, 18)	17496	activation_98[0][0]
concatenate_90 (Concatenate)	(None, 4, 4, 126)	0	concatenate_89[0][0] conv2d_99[0][0]
batch_normalization_98 (BatchNo	(None, 4, 4, 126)	504	concatenate_90[0][0]
activation_99 (Activation)	(None, 4, 4, 126)	0	batch_normalization_98[0][0]
conv2d_100 (Conv2D)	(None, 4, 4, 18)	20412	activation_99[0][0]
concatenate_91 (Concatenate)	(None, 4, 4, 144)	0	concatenate_90[0][0] conv2d_100[0][0]
batch_normalization_99 (BatchNo	(None, 4, 4, 144)	576	concatenate_91[0][0]
activation_100 (Activation)	(None, 4, 4, 144)	0	batch_normalization_99[0][0]
conv2d_101 (Conv2D)	(None, 4, 4, 18)	23328	activation_100[0][0]
concatenate_92 (Concatenate)	(None, 4, 4, 162)	0	concatenate_91[0][0] conv2d_101[0][0]
batch_normalization_100 (BatchN	(None, 4, 4, 162)	648	concatenate_92[0][0]
activation_101 (Activation)	(None, 4, 4, 162)	0	batch_normalization_100[0][0]
conv2d_102 (Conv2D)	(None, 4, 4, 18)	26244	activation_101[0][0]
concatenate_93 (Concatenate)	(None, 4, 4, 180)	0	concatenate_92[0][0] conv2d_102[0][0]
batch_normalization_101 (BatchN	(None, 4, 4, 180)	720	concatenate_93[0][0]
activation_102 (Activation)	(None, 4, 4, 180)	0	batch_normalization_101[0][0]
conv2d_103 (Conv2D)	(None, 4, 4, 18)	29160	activation_102[0][0]
concatenate_94 (Concatenate)	(None, 4, 4, 198)	0	concatenate_93[0][0] conv2d_103[0][0]
batch_normalization_102 (BatchN	(None, 4, 4, 198)	792	concatenate_94[0][0]
activation_103 (Activation)	(None, 4, 4, 198)	0	batch_normalization_102[0][0]
conv2d_104 (Conv2D)	(None, 4, 4, 18)	32076	activation_103[0][0]
concatenate_95 (Concatenate)	(None, 4, 4, 216)	0	concatenate_94[0][0] conv2d_104[0][0]
batch_normalization_103 (BatchN	(None, 4, 4, 216)	864	concatenate_95[0][0]
activation_104 (Activation)	(None, 4, 4, 216)	0	batch_normalization_103[0][0]
conv2d_105 (Conv2D)	(None, 4, 4, 18)	34992	activation_104[0][0]
concatenate_96 (Concatenate)	(None, 4, 4, 234)	0	concatenate_95[0][0] conv2d_105[0][0]
batch_normalization_104 (BatchN	(None, 4, 4, 234)	936	concatenate_96[0][0]
activation_105 (Activation)	(None, 4, 4, 234)	0	batch_normalization_104[0][0]

```
activation_105 (Activation)     (None, 4, 4, 234)    0           batch_normalization_104[0][0]
__________________________________________________________________________________________________
average_pooling2d_8 (AveragePoo (None, 2, 2, 234)    0           activation_105[0][0]
__________________________________________________________________________________________________
conv2d_106 (Conv2D)             (None, 1, 1, 10)     9370        average_pooling2d_8[0][0]
__________________________________________________________________________________________________
activation_106 (Activation)     (None, 1, 1, 10)     0           conv2d_106[0][0]
__________________________________________________________________________________________________
flatten_2 (Flatten)             (None, 10)           0           activation_106[0][0]
==================================================================================================
Total params: 995,230
Trainable params: 981,658
Non-trainable params: 13,572
__________________________________________________________________________________________________
```

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]:

```
from keras.callbacks import ModelCheckpoint, CSVLogger
ckpt = ModelCheckpoint(os.path.join(path, 'model.hdf5'), monitor = 'val_acc')
csv = CSVLogger(os.path.join(path, 'log.csv'), append = True)
```

In [0]:

```
# determine Loss function and Optimizer
model.compile(loss='categorical_crossentropy',
              optimizer=SGD(0.01, momentum = 0.7),
              metrics=['accuracy'])
```

In [0]:

```
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 = [csv, ckpt])
model.save_weights(os.path.join(path, '30epochs.h5'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is
deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is depr
ecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Epoch 1/30
391/390 [==============================] - 301s 769ms/step - loss: 7.0762 - acc: 0.4017 -
val_loss: 2.7243 - val_acc: 0.2101
Epoch 2/30
391/390 [==============================] - 282s 720ms/step - loss: 1.9663 - acc: 0.4695 -
val_loss: 2.2608 - val_acc: 0.3896
Epoch 3/30
391/390 [==============================] - 281s 720ms/step - loss: 1.7373 - acc: 0.5277 -
val_loss: 1.9559 - val_acc: 0.4850
Epoch 4/30
391/390 [==============================] - 281s 720ms/step - loss: 1.6122 - acc: 0.5716 -
val_loss: 2.2660 - val_acc: 0.4709
Epoch 5/30
391/390 [==============================] - 281s 720ms/step - loss: 1.5173 - acc: 0.5992 -
val_loss: 1.9089 - val_acc: 0.5104
Epoch 6/30
391/390 [==============================] - 282s 720ms/step - loss: 1.4472 - acc: 0.6283 -
val_loss: 1.7864 - val_acc: 0.5552
Epoch 7/30
391/390 [==============================] - 282s 720ms/step - loss: 1.3962 - acc: 0.6436 -
val_loss: 1.6632 - val_acc: 0.5665
Epoch 8/30
```

```
391/390 [==============================] - 282s 720ms/step - loss: 1.3499 - acc: 0.6632 -
val_loss: 1.5832 - val_acc: 0.6090
Epoch 9/30
391/390 [==============================] - 282s 720ms/step - loss: 1.3047 - acc: 0.6794 -
val_loss: 1.4688 - val_acc: 0.6398
Epoch 10/30
391/390 [==============================] - 282s 720ms/step - loss: 1.2727 - acc: 0.6916 -
val_loss: 1.8136 - val_acc: 0.5927
Epoch 11/30
391/390 [==============================] - 281s 720ms/step - loss: 1.2387 - acc: 0.7044 -
val_loss: 1.3662 - val_acc: 0.6650
Epoch 12/30
391/390 [==============================] - 282s 720ms/step - loss: 1.2134 - acc: 0.7111 -
val_loss: 1.2340 - val_acc: 0.7158
Epoch 13/30
391/390 [==============================] - 281s 720ms/step - loss: 1.1783 - acc: 0.7251 -
val_loss: 1.3248 - val_acc: 0.6928
Epoch 14/30
391/390 [==============================] - 281s 720ms/step - loss: 1.1575 - acc: 0.7331 -
val_loss: 1.2079 - val_acc: 0.7246
Epoch 15/30
391/390 [==============================] - 281s 719ms/step - loss: 1.1282 - acc: 0.7422 -
val_loss: 1.4078 - val_acc: 0.6715
Epoch 16/30
391/390 [==============================] - 280s 717ms/step - loss: 1.1073 - acc: 0.7491 -
val_loss: 1.3001 - val_acc: 0.6969
Epoch 17/30
391/390 [==============================] - 280s 717ms/step - loss: 1.0949 - acc: 0.7543 -
val_loss: 1.3848 - val_acc: 0.6918
Epoch 18/30
391/390 [==============================] - 280s 715ms/step - loss: 1.0689 - acc: 0.7614 -
val_loss: 1.4064 - val_acc: 0.6932
Epoch 19/30
391/390 [==============================] - 279s 714ms/step - loss: 1.0511 - acc: 0.7685 -
val_loss: 1.2036 - val_acc: 0.7261
Epoch 20/30
391/390 [==============================] - 279s 713ms/step - loss: 1.0426 - acc: 0.7706 -
val_loss: 1.1953 - val_acc: 0.7335
Epoch 21/30
391/390 [==============================] - 278s 712ms/step - loss: 1.0226 - acc: 0.7785 -
val_loss: 1.2224 - val_acc: 0.7262
Epoch 22/30
391/390 [==============================] - 278s 712ms/step - loss: 1.0009 - acc: 0.7881 -
val_loss: 1.2863 - val_acc: 0.7080
Epoch 23/30
391/390 [==============================] - 278s 712ms/step - loss: 0.9891 - acc: 0.7892 -
val_loss: 1.1319 - val_acc: 0.7507
Epoch 24/30
391/390 [==============================] - 278s 712ms/step - loss: 0.9776 - acc: 0.7923 -
val_loss: 1.0027 - val_acc: 0.7850
Epoch 25/30
391/390 [==============================] - 279s 714ms/step - loss: 0.9666 - acc: 0.7987 -
val_loss: 1.1532 - val_acc: 0.7573
Epoch 26/30
391/390 [==============================] - 280s 717ms/step - loss: 0.9483 - acc: 0.8030 -
val_loss: 0.9879 - val_acc: 0.7959
Epoch 27/30
391/390 [==============================] - 281s 719ms/step - loss: 0.9394 - acc: 0.8037 -
val_loss: 0.9873 - val_acc: 0.7953
Epoch 28/30
391/390 [==============================] - 281s 720ms/step - loss: 0.9305 - acc: 0.8096 -
val_loss: 1.0487 - val_acc: 0.7711
Epoch 29/30
391/390 [==============================] - 281s 719ms/step - loss: 0.9175 - acc: 0.8133 -
val_loss: 1.1011 - val_acc: 0.7561
Epoch 30/30
391/390 [==============================] - 281s 720ms/step - loss: 0.9140 - acc: 0.8159 -
val_loss: 1.0169 - val_acc: 0.7874
```

In [0]:

```
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 = [csv, ckpt])
model.save_weights(os.path.join(path, '60epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 281s 719ms/step - loss: 0.9025 - acc: 0.8154 -
val_loss: 0.8638 - val_acc: 0.8250
Epoch 2/30
391/390 [==============================] - 282s 721ms/step - loss: 0.8880 - acc: 0.8205 -
val_loss: 1.1500 - val_acc: 0.7579
Epoch 3/30
391/390 [==============================] - 281s 720ms/step - loss: 0.8771 - acc: 0.8245 -
val_loss: 1.1167 - val_acc: 0.7605
Epoch 4/30
391/390 [==============================] - 281s 720ms/step - loss: 0.8728 - acc: 0.8261 -
val_loss: 1.0960 - val_acc: 0.7643
Epoch 5/30
391/390 [==============================] - 281s 718ms/step - loss: 0.8609 - acc: 0.8320 -
val_loss: 1.3143 - val_acc: 0.7280
Epoch 6/30
391/390 [==============================] - 281s 719ms/step - loss: 0.8508 - acc: 0.8326 -
val_loss: 1.1231 - val_acc: 0.7702
Epoch 7/30
391/390 [==============================] - 281s 719ms/step - loss: 0.8448 - acc: 0.8333 -
val_loss: 0.8784 - val_acc: 0.8293
Epoch 8/30
391/390 [==============================] - 281s 720ms/step - loss: 0.8355 - acc: 0.8358 -
val_loss: 1.0412 - val_acc: 0.7764
Epoch 9/30
391/390 [==============================] - 281s 720ms/step - loss: 0.8227 - acc: 0.8398 -
val_loss: 1.1882 - val_acc: 0.7495
Epoch 10/30
391/390 [==============================] - 282s 721ms/step - loss: 0.8187 - acc: 0.8413 -
val_loss: 0.9523 - val_acc: 0.8095
Epoch 11/30
391/390 [==============================] - 281s 719ms/step - loss: 0.8109 - acc: 0.8447 -
val_loss: 0.9656 - val_acc: 0.8019
Epoch 12/30
391/390 [==============================] - 282s 720ms/step - loss: 0.8043 - acc: 0.8463 -
val_loss: 1.0320 - val_acc: 0.7952
Epoch 13/30
305/390 [======================>.......] - ETA: 58s - loss: 0.7942 - acc: 0.8495
```

In [0]:

```
#restoring the last model
from keras.models import load_model
model = load_model(os.path.join(path, 'model.hdf5'))
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)
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 = [csv, ckpt])
model.save_weights(os.path.join(path, '72epochs.h5'))
```

```
Epoch 1/30
391/390 [==============================] - 289s 739ms/step - loss: 0.7186 - acc: 0.8692 -
val_loss: 0.8482 - val_acc: 0.8320
Epoch 2/30
391/390 [==============================] - 274s 699ms/step - loss: 0.7068 - acc: 0.8725 -
val_loss: 0.7481 - val_acc: 0.8645
Epoch 3/30
391/390 [==============================] - 273s 697ms/step - loss: 0.7072 - acc: 0.8747 -
val_loss: 0.8450 - val_acc: 0.8378
Epoch 4/30
391/390 [==============================] - 273s 699ms/step - loss: 0.7020 - acc: 0.8749 -
val_loss: 1.3341 - val_acc: 0.7237
Epoch 5/30
391/390 [==============================] - 273s 698ms/step - loss: 0.7001 - acc: 0.8760 -
val_loss: 0.9239 - val_acc: 0.8147
Epoch 6/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6851 - acc: 0.8795 -
val_loss: 0.8432 - val_acc: 0.8421
Epoch 7/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6799 - acc: 0.8812 -
val_loss: 0.7917 - val_acc: 0.8431
Epoch 8/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6827 - acc: 0.8793 -
val_loss: 0.9111 - val_acc: 0.8182
```

```
val_loss: 0.9111 - val_acc: 0.8162
Epoch 9/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6756 - acc: 0.8824 -
val_loss: 0.7613 - val_acc: 0.8548
Epoch 10/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6764 - acc: 0.8819 -
val_loss: 0.9727 - val_acc: 0.8091
Epoch 11/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6635 - acc: 0.8840 -
val_loss: 0.8100 - val_acc: 0.8507
Epoch 12/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6630 - acc: 0.8863 -
val_loss: 0.8723 - val_acc: 0.8302
Epoch 13/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6583 - acc: 0.8868 -
val_loss: 0.9680 - val_acc: 0.8054
Epoch 14/30
391/390 [==============================] - 272s 697ms/step - loss: 0.6538 - acc: 0.8895 -
val_loss: 0.7967 - val_acc: 0.8537
Epoch 15/30
391/390 [==============================] - 273s 697ms/step - loss: 0.6515 - acc: 0.8880 -
val_loss: 0.9328 - val_acc: 0.8135
Epoch 16/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6461 - acc: 0.8907 -
val_loss: 0.8955 - val_acc: 0.8235
Epoch 17/30
391/390 [==============================] - 273s 699ms/step - loss: 0.6437 - acc: 0.8912 -
val_loss: 0.8498 - val_acc: 0.8352
Epoch 18/30
391/390 [==============================] - 272s 697ms/step - loss: 0.6446 - acc: 0.8905 -
val_loss: 1.0233 - val_acc: 0.8077
Epoch 19/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6399 - acc: 0.8913 -
val_loss: 0.8660 - val_acc: 0.8320
Epoch 20/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6378 - acc: 0.8926 -
val_loss: 0.7881 - val_acc: 0.8627
Epoch 21/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6334 - acc: 0.8930 -
val_loss: 0.8520 - val_acc: 0.8367
Epoch 22/30
391/390 [==============================] - 273s 698ms/step - loss: 0.6295 - acc: 0.8935 -
val_loss: 0.7744 - val_acc: 0.8552
Epoch 23/30
391/390 [==============================] - 273s 699ms/step - loss: 0.6360 - acc: 0.8926 -
val_loss: 0.7599 - val_acc: 0.8617
Epoch 24/30
391/390 [==============================] - 272s 697ms/step - loss: 0.6204 - acc: 0.8969 -
val_loss: 1.0240 - val_acc: 0.7970
Epoch 25/30
391/390 [==============================] - 273s 697ms/step - loss: 0.6234 - acc: 0.8969 -
val_loss: 0.8029 - val_acc: 0.8475
Epoch 26/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6203 - acc: 0.8971 -
val_loss: 0.8687 - val_acc: 0.8317
Epoch 27/30
391/390 [==============================] - 273s 697ms/step - loss: 0.6237 - acc: 0.8978 -
val_loss: 0.7873 - val_acc: 0.8573
Epoch 28/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6140 - acc: 0.8991 -
val_loss: 0.8532 - val_acc: 0.8428
Epoch 29/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6160 - acc: 0.8993 -
val_loss: 0.7863 - val_acc: 0.8588
Epoch 30/30
391/390 [==============================] - 272s 696ms/step - loss: 0.6055 - acc: 0.9002 -
val_loss: 0.9144 - val_acc: 0.8296
```

In [0]:

```
model.load_weights(os.path.join(path, '72epochs.h5'))
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)
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 = [csv, ckpt])
```

```
batch_size, epochs = 50, validation_data =(x_test, y_test), callbacks = [csv, ckpt])
model.save_weights(os.path.join(path, '102epochs.h5'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is
deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is depr
ecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Epoch 1/30
391/390 [==============================] - 293s 749ms/step - loss: 0.6085 - acc: 0.9015 -
val_loss: 0.8522 - val_acc: 0.8419
Epoch 2/30
391/390 [==============================] - 276s 706ms/step - loss: 0.6065 - acc: 0.9018 -
val_loss: 0.8206 - val_acc: 0.8466
Epoch 3/30
391/390 [==============================] - 274s 701ms/step - loss: 0.6005 - acc: 0.9033 -
val_loss: 0.8312 - val_acc: 0.8405
Epoch 4/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5962 - acc: 0.9037 -
val_loss: 0.6590 - val_acc: 0.8825
Epoch 5/30
391/390 [==============================] - 274s 700ms/step - loss: 0.5922 - acc: 0.9054 -
val_loss: 1.0431 - val_acc: 0.7991
Epoch 6/30
391/390 [==============================] - 273s 699ms/step - loss: 0.5945 - acc: 0.9043 -
val_loss: 0.9608 - val_acc: 0.8028
Epoch 7/30
391/390 [==============================] - 273s 699ms/step - loss: 0.5865 - acc: 0.9057 -
val_loss: 0.8287 - val_acc: 0.8488
Epoch 8/30
391/390 [==============================] - 274s 700ms/step - loss: 0.5983 - acc: 0.9040 -
val_loss: 0.8926 - val_acc: 0.8376
Epoch 9/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5905 - acc: 0.9044 -
val_loss: 0.7270 - val_acc: 0.8633
Epoch 10/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5825 - acc: 0.9074 -
val_loss: 0.8707 - val_acc: 0.8284
Epoch 11/30
391/390 [==============================] - 275s 704ms/step - loss: 0.5773 - acc: 0.9088 -
val_loss: 0.8716 - val_acc: 0.8333
Epoch 12/30
391/390 [==============================] - 276s 705ms/step - loss: 0.5806 - acc: 0.9083 -
val_loss: 0.8137 - val_acc: 0.8616
Epoch 13/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5785 - acc: 0.9078 -
val_loss: 0.8176 - val_acc: 0.8561
Epoch 14/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5778 - acc: 0.9083 -
val_loss: 1.0768 - val_acc: 0.7976
Epoch 15/30
391/390 [==============================] - 275s 702ms/step - loss: 0.5750 - acc: 0.9089 -
val_loss: 0.8099 - val_acc: 0.8494
Epoch 16/30
391/390 [==============================] - 274s 701ms/step - loss: 0.5719 - acc: 0.9115 -
val_loss: 0.7910 - val_acc: 0.8520
Epoch 17/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5748 - acc: 0.9101 -
val_loss: 0.7055 - val_acc: 0.8726
Epoch 18/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5643 - acc: 0.9120 -
val_loss: 0.7967 - val_acc: 0.8578
Epoch 19/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5606 - acc: 0.9131 -
val_loss: 0.7192 - val_acc: 0.8735
Epoch 20/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5627 - acc: 0.9114 -
val_loss: 0.7588 - val_acc: 0.8616
Epoch 21/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5582 - acc: 0.9135 -
val_loss: 0.8157 - val_acc: 0.8477
Epoch 22/30
```

```
391/390 [==============================] - 275s 704ms/step - loss: 0.5502 - acc: 0.9166 -
val_loss: 0.8063 - val_acc: 0.8579
Epoch 23/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5523 - acc: 0.9154 -
val_loss: 0.7784 - val_acc: 0.8606
Epoch 24/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5618 - acc: 0.9128 -
val_loss: 0.8646 - val_acc: 0.8372
Epoch 25/30
391/390 [==============================] - 274s 702ms/step - loss: 0.5533 - acc: 0.9163 -
val_loss: 0.7414 - val_acc: 0.8715
Epoch 26/30
391/390 [==============================] - 275s 704ms/step - loss: 0.5468 - acc: 0.9167 -
val_loss: 0.8146 - val_acc: 0.8442
Epoch 27/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5505 - acc: 0.9159 -
val_loss: 0.6578 - val_acc: 0.8875
Epoch 28/30
391/390 [==============================] - 275s 704ms/step - loss: 0.5440 - acc: 0.9173 -
val_loss: 0.7130 - val_acc: 0.8736
Epoch 29/30
391/390 [==============================] - 275s 703ms/step - loss: 0.5515 - acc: 0.9153 -
val_loss: 0.8489 - val_acc: 0.8469
Epoch 30/30
391/390 [==============================] - 274s 701ms/step - loss: 0.5407 - acc: 0.9189 -
val_loss: 1.0301 - val_acc: 0.8035
```

In [0]:

```
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch =
1.5*x_train.shape[0]/batch_size, epochs = 30, validation_data =(x_test, y_test), callbacks = [csv,
ckpt])
model.save_weights(os.path.join(path, '132epochs.h5'))
```

```
Epoch 1/30
586/585 [==============================] - 404s 689ms/step - loss: 0.5365 - acc: 0.9203 -
val_loss: 0.7786 - val_acc: 0.8584
Epoch 2/30
586/585 [==============================] - 403s 688ms/step - loss: 0.5404 - acc: 0.9175 -
val_loss: 0.7118 - val_acc: 0.8775
Epoch 3/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5311 - acc: 0.9210 -
val_loss: 0.7592 - val_acc: 0.8632
Epoch 4/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5346 - acc: 0.9207 -
val_loss: 0.7722 - val_acc: 0.8618
Epoch 5/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5259 - acc: 0.9235 -
val_loss: 0.9573 - val_acc: 0.8196
Epoch 6/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5279 - acc: 0.9210 -
val_loss: 0.7815 - val_acc: 0.8568
Epoch 7/30
586/585 [==============================] - 402s 687ms/step - loss: 0.5223 - acc: 0.9234 -
val_loss: 0.6805 - val_acc: 0.8873
Epoch 8/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5227 - acc: 0.9236 -
val_loss: 1.0442 - val_acc: 0.8167
Epoch 9/30
586/585 [==============================] - 402s 686ms/step - loss: 0.5188 - acc: 0.9248 -
val_loss: 0.7324 - val_acc: 0.8698
Epoch 10/30
586/585 [==============================] - 403s 688ms/step - loss: 0.5198 - acc: 0.9238 -
val_loss: 0.7930 - val_acc: 0.8499
Epoch 11/30
586/585 [==============================] - 404s 689ms/step - loss: 0.5200 - acc: 0.9236 -
val_loss: 0.8599 - val_acc: 0.8485
Epoch 12/30
586/585 [==============================] - 404s 689ms/step - loss: 0.5092 - acc: 0.9266 -
val_loss: 0.7357 - val_acc: 0.8754
Epoch 13/30
586/585 [==============================] - 403s 688ms/step - loss: 0.5063 - acc: 0.9278 -
val_loss: 0.6651 - val_acc: 0.8794
Epoch 14/30
586/585 [==============================] - 404s 689ms/step - loss: 0.5132 - acc: 0.9260 -
```

```
val_loss: 0.7859 - val_acc: 0.8623
Epoch 15/30
586/585 [==============================] - 403s 689ms/step - loss: 0.5073 - acc: 0.9270 -
val_loss: 0.7578 - val_acc: 0.8609
Epoch 16/30
586/585 [==============================] - 403s 688ms/step - loss: 0.5027 - acc: 0.9294 -
val_loss: 0.7870 - val_acc: 0.8594
Epoch 17/30
586/585 [==============================] - 401s 685ms/step - loss: 0.4996 - acc: 0.9290 -
val_loss: 0.7860 - val_acc: 0.8559
Epoch 18/30
586/585 [==============================] - 400s 682ms/step - loss: 0.5067 - acc: 0.9285 -
val_loss: 0.7413 - val_acc: 0.8640
Epoch 19/30
586/585 [==============================] - 399s 681ms/step - loss: 0.4989 - acc: 0.9302 -
val_loss: 0.7454 - val_acc: 0.8699
Epoch 20/30
586/585 [==============================] - 398s 680ms/step - loss: 0.4985 - acc: 0.9303 -
val_loss: 0.6877 - val_acc: 0.8870
Epoch 21/30
586/585 [==============================] - 401s 684ms/step - loss: 0.4924 - acc: 0.9306 -
val_loss: 0.6924 - val_acc: 0.8767
Epoch 22/30
586/585 [==============================] - 400s 682ms/step - loss: 0.4974 - acc: 0.9304 -
val_loss: 0.7213 - val_acc: 0.8742
Epoch 23/30
586/585 [==============================] - 399s 681ms/step - loss: 0.4871 - acc: 0.9328 -
val_loss: 0.9453 - val_acc: 0.8232
Epoch 24/30
586/585 [==============================] - 400s 683ms/step - loss: 0.4889 - acc: 0.9326 -
val_loss: 0.7593 - val_acc: 0.8672
Epoch 25/30
586/585 [==============================] - 400s 682ms/step - loss: 0.4856 - acc: 0.9333 -
val_loss: 0.7471 - val_acc: 0.8652
Epoch 26/30
586/585 [==============================] - 400s 683ms/step - loss: 0.4842 - acc: 0.9333 -
val_loss: 0.8415 - val_acc: 0.8462
Epoch 27/30
586/585 [==============================] - 400s 682ms/step - loss: 0.4870 - acc: 0.9333 -
val_loss: 0.7794 - val_acc: 0.8569
Epoch 28/30
586/585 [==============================] - 399s 681ms/step - loss: 0.4854 - acc: 0.9334 -
val_loss: 0.6650 - val_acc: 0.8830
Epoch 29/30
586/585 [==============================] - 400s 683ms/step - loss: 0.4736 - acc: 0.9355 -
val_loss: 0.6903 - val_acc: 0.8859
Epoch 30/30
586/585 [==============================] - 400s 683ms/step - loss: 0.4832 - acc: 0.9341 -
val_loss: 0.8638 - val_acc: 0.8426
```

In [0]:

```
keras.backend.set_value(model.optimizer.lr, .001)
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 3*x_train.shape[0
]/batch_size, epochs = 20, validation_data =(x_test, y_test), callbacks = [csv, ckpt])
model.save_weights(os.path.join(path, '157epochs.h5'))#157 because it ran for 5 epochs before, as
i forgot to update the learning rate
```

```
Epoch 1/20
1172/1171 [==============================] - 785s 669ms/step - loss: 0.2054 - acc: 0.9635 - val_lo
ss: 0.3784 - val_acc: 0.9125
Epoch 2/20
1172/1171 [==============================] - 782s 667ms/step - loss: 0.1892 - acc: 0.9646 - val_lo
ss: 0.4052 - val_acc: 0.9070
Epoch 3/20
1172/1171 [==============================] - 783s 668ms/step - loss: 0.1935 - acc: 0.9645 - val_lo
ss: 0.4143 - val_acc: 0.9061
Epoch 4/20
1172/1171 [==============================] - 782s 667ms/step - loss: 0.2025 - acc: 0.9633 - val_lo
ss: 0.4531 - val_acc: 0.9028
Epoch 5/20
1172/1171 [==============================] - 780s 666ms/step - loss: 0.2001 - acc: 0.9644 - val_lo
ss: 0.4063 - val_acc: 0.9105
Epoch 6/20
1172/1171 [==============================] - 779s 665ms/step - loss: 0.1990 - acc: 0.9645 - val_lo
ss: 0.4300 - val acc: 0.9067
```

```
Epoch 7/20
1172/1171 [==============================] - 775s 661ms/step - loss: 0.1998 - acc: 0.9646 - val_lo
ss: 0.4415 - val_acc: 0.9029
Epoch 8/20
1172/1171 [==============================] - 779s 664ms/step - loss: 0.1969 - acc: 0.9660 - val_lo
ss: 0.4012 - val_acc: 0.9125
Epoch 9/20
1172/1171 [==============================] - 779s 665ms/step - loss: 0.1973 - acc: 0.9656 - val_lo
ss: 0.4686 - val_acc: 0.8975
Epoch 10/20
1172/1171 [==============================] - 778s 664ms/step - loss: 0.1984 - acc: 0.9649 - val_lo
ss: 0.4428 - val_acc: 0.9036
Epoch 11/20
1172/1171 [==============================] - 773s 659ms/step - loss: 0.1947 - acc: 0.9663 - val_lo
ss: 0.4368 - val_acc: 0.9089
Epoch 12/20
1172/1171 [==============================] - 777s 663ms/step - loss: 0.1957 - acc: 0.9664 - val_lo
ss: 0.4116 - val_acc: 0.9116
Epoch 13/20
1172/1171 [==============================] - 776s 662ms/step - loss: 0.1941 - acc: 0.9667 - val_lo
ss: 0.4601 - val_acc: 0.9014
Epoch 14/20
1172/1171 [==============================] - 774s 660ms/step - loss: 0.1938 - acc: 0.9667 - val_lo
ss: 0.4160 - val_acc: 0.9142
Epoch 15/20
1172/1171 [==============================] - 776s 662ms/step - loss: 0.1954 - acc: 0.9662 - val_lo
ss: 0.4113 - val_acc: 0.9121
Epoch 16/20
1172/1171 [==============================] - 775s 662ms/step - loss: 0.1901 - acc: 0.9679 - val_lo
ss: 0.4877 - val_acc: 0.8948
Epoch 17/20
1172/1171 [==============================] - 775s 661ms/step - loss: 0.1910 - acc: 0.9678 - val_lo
ss: 0.4471 - val_acc: 0.9068
Epoch 18/20
1172/1171 [==============================] - 777s 663ms/step - loss: 0.1949 - acc: 0.9667 - val_lo
ss: 0.5133 - val_acc: 0.8906
Epoch 19/20
1172/1171 [==============================] - 775s 661ms/step - loss: 0.1911 - acc: 0.9684 - val_lo
ss: 0.4508 - val_acc: 0.9065
Epoch 20/20
1172/1171 [==============================] - 775s 661ms/step - loss: 0.1925 - acc: 0.9677 - val_lo
ss: 0.4517 - val_acc: 0.9006
```

In [0]:

```
model.load_weights(os.path.join(path, '157epochs.h5'))
keras.backend.set_value(model.optimizer.momentum, 0.7)
keras.backend.set_value(model.optimizer.lr, 0.001)
best_ckpt = ModelCheckpoint(os.path.join(path, 'best_model.h5'), monitor = 'val_acc',
save_best_only = True)
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)
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 3*x_train.shape[0
]/batch_size, epochs = 20, validation_data =(x_test, y_test), callbacks = [csv, ckpt, best_ckpt])
model.save_weights(os.path.join(path, '177epochs.h5'))
```

```
Epoch 1/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1953 - acc: 0.9678 - val_lo
ss: 0.4215 - val_acc: 0.9101
Epoch 2/20
1172/1171 [==============================] - 802s 685ms/step - loss: 0.1928 - acc: 0.9677 - val_lo
ss: 0.4599 - val_acc: 0.9045
Epoch 3/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1898 - acc: 0.9684 - val_lo
ss: 0.4914 - val_acc: 0.8930
Epoch 4/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1897 - acc: 0.9688 - val_lo
ss: 0.4312 - val_acc: 0.9123
Epoch 5/20
1172/1171 [==============================] - 801s 683ms/step - loss: 0.1883 - acc: 0.9692 - val_lo
ss: 0.4124 - val_acc: 0.9129
Epoch 6/20
1172/1171 [==============================] - 801s 684ms/step - loss: 0.1909 - acc: 0.9688 - val_lo
```

```
ss: 0.4092 - val_acc: 0.9140
Epoch 7/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1898 - acc: 0.9688 - val_lo
ss: 0.4962 - val_acc: 0.8972
Epoch 8/20
1172/1171 [==============================] - 803s 685ms/step - loss: 0.1897 - acc: 0.9692 - val_lo
ss: 0.4710 - val_acc: 0.9019
Epoch 9/20
1172/1171 [==============================] - 801s 683ms/step - loss: 0.1864 - acc: 0.9699 - val_lo
ss: 0.4319 - val_acc: 0.9105
Epoch 10/20
1172/1171 [==============================] - 800s 683ms/step - loss: 0.1866 - acc: 0.9700 - val_lo
ss: 0.4423 - val_acc: 0.9069
Epoch 11/20
1172/1171 [==============================] - 800s 682ms/step - loss: 0.1883 - acc: 0.9696 - val_lo
ss: 0.4416 - val_acc: 0.9097
Epoch 12/20
1172/1171 [==============================] - 800s 683ms/step - loss: 0.1897 - acc: 0.9699 - val_lo
ss: 0.4437 - val_acc: 0.9083
Epoch 13/20
1172/1171 [==============================] - 800s 683ms/step - loss: 0.1859 - acc: 0.9704 - val_lo
ss: 0.4420 - val_acc: 0.9123
Epoch 14/20
1172/1171 [==============================] - 800s 683ms/step - loss: 0.1893 - acc: 0.9693 - val_lo
ss: 0.4634 - val_acc: 0.9031
Epoch 15/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1877 - acc: 0.9705 - val_lo
ss: 0.5390 - val_acc: 0.8916
Epoch 16/20
1172/1171 [==============================] - 803s 685ms/step - loss: 0.1882 - acc: 0.9695 - val_lo
ss: 0.4416 - val_acc: 0.9088
Epoch 17/20
1172/1171 [==============================] - 801s 684ms/step - loss: 0.1847 - acc: 0.9711 - val_lo
ss: 0.4303 - val_acc: 0.9083
Epoch 18/20
1172/1171 [==============================] - 802s 684ms/step - loss: 0.1851 - acc: 0.9707 - val_lo
ss: 0.4674 - val_acc: 0.9016
Epoch 19/20
1172/1171 [==============================] - 799s 682ms/step - loss: 0.1836 - acc: 0.9714 - val_lo
ss: 0.4599 - val_acc: 0.9078
Epoch 20/20
1172/1171 [==============================] - 799s 681ms/step - loss: 0.1862 - acc: 0.9707 - val_lo
ss: 0.4531 - val_acc: 0.9072
```

In [0]:

```
model.load_weights(os.path.join(path, '177epochs.h5'))
from keras.preprocessing.image import ImageDataGenerator
keras.backend.set_value(model.optimizer.momentum, 0.7)
keras.backend.set_value(model.optimizer.lr, 0.001)
best_ckpt = ModelCheckpoint(os.path.join(path, 'best_model.h5'), monitor = 'val_acc',
save_best_only = True)
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 =
3.5*x_train.shape[0]/batch_size, epochs = 10, validation_data =(x_test, y_test), callbacks = [csv,
ckpt, best_ckpt])
model.save_weights(os.path.join(path, '187epochs.h5'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is
deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is depr
ecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Epoch 1/10
1368/1367 [==============================] - 942s 689ms/step - loss: 0.1854 - acc: 0.9706 - val_lo
ss: 0.4150 - val_acc: 0.9147
Epoch 2/10
1368/1367 [==============================] - 926s 677ms/step - loss: 0.1840 - acc: 0.9710 - val_lo
ss: 0.4621 - val_acc: 0.9051
```

```
ss: 0.4021 - val_acc: 0.9051
Epoch 3/10
1368/1367 [==============================] - 921s 673ms/step - loss: 0.1822 - acc: 0.9717 - val_lo
ss: 0.4534 - val_acc: 0.9077
Epoch 4/10
1368/1367 [==============================] - 923s 675ms/step - loss: 0.1812 - acc: 0.9724 - val_lo
ss: 0.4615 - val_acc: 0.9039
Epoch 5/10
1368/1367 [==============================] - 922s 674ms/step - loss: 0.1863 - acc: 0.9709 - val_lo
ss: 0.4345 - val_acc: 0.9122
Epoch 6/10
1368/1367 [==============================] - 924s 676ms/step - loss: 0.1828 - acc: 0.9715 - val_lo
ss: 0.4552 - val_acc: 0.9066
Epoch 7/10
1368/1367 [==============================] - 920s 672ms/step - loss: 0.1829 - acc: 0.9719 - val_lo
ss: 0.4504 - val_acc: 0.9085
Epoch 8/10
1368/1367 [==============================] - 916s 670ms/step - loss: 0.1817 - acc: 0.9717 - val_lo
ss: 0.4907 - val_acc: 0.8999
Epoch 9/10
1368/1367 [==============================] - 917s 670ms/step - loss: 0.1828 - acc: 0.9717 - val_lo
ss: 0.4499 - val_acc: 0.9081
Epoch 10/10
1368/1367 [==============================] - 915s 669ms/step - loss: 0.1806 - acc: 0.9724 - val_lo
ss: 0.4282 - val_acc: 0.9140
```

In [0]:

```
model.load_weights(os.path.join(path, '187epochs.h5'))
datagen = ImageDataGenerator(rotation_range = 10, horizontal_flip = True, width_shift_range = 0.08,
height_shift_range = 0.08, zoom_range = 0.15, shear_range = 10)
datagen.fit(x_train)
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 4*x_train.shape[0
]/batch_size, epochs = 10, validation_data =(x_test, y_test), callbacks = [csv, ckpt, best_ckpt])
model.save_weights(os.path.join(path, '197epochs.h5'))
```

```
Epoch 1/10
1563/1562 [==============================] - 1057s 676ms/step - loss: 0.1200 - acc: 0.9864 - val_l
oss: 0.4136 - val_acc: 0.9134
Epoch 2/10
1563/1562 [==============================] - 1055s 675ms/step - loss: 0.1280 - acc: 0.9853 - val_l
oss: 0.4497 - val_acc: 0.9056
Epoch 3/10
1563/1562 [==============================] - 1053s 674ms/step - loss: 0.1266 - acc: 0.9857 - val_l
oss: 0.4673 - val_acc: 0.9032
Epoch 4/10
1563/1562 [==============================] - 1057s 676ms/step - loss: 0.1277 - acc: 0.9854 - val_l
oss: 0.4837 - val_acc: 0.8996
Epoch 5/10
1563/1562 [==============================] - 1064s 681ms/step - loss: 0.1270 - acc: 0.9862 - val_l
oss: 0.4422 - val_acc: 0.9090
Epoch 6/10
1563/1562 [==============================] - 1063s 680ms/step - loss: 0.1260 - acc: 0.9864 - val_l
oss: 0.4395 - val_acc: 0.9095
Epoch 7/10
1563/1562 [==============================] - 1064s 680ms/step - loss: 0.1272 - acc: 0.9860 - val_l
oss: 0.5276 - val_acc: 0.8968
Epoch 8/10
1563/1562 [==============================] - 1062s 680ms/step - loss: 0.1263 - acc: 0.9861 - val_l
oss: 0.4768 - val_acc: 0.9052
Epoch 9/10
1563/1562 [==============================] - 1059s 678ms/step - loss: 0.1269 - acc: 0.9861 - val_l
oss: 0.4302 - val_acc: 0.9125
Epoch 10/10
1563/1562 [==============================] - 1060s 678ms/step - loss: 0.1269 - acc: 0.9863 - val_l
oss: 0.4558 - val_acc: 0.9082
```

In [0]:

```
model.load_weights(os.path.join(path, '197epochs.h5'))
keras.backend.set_value(model.optimizer.lr, 0.0001)
bacth_size = 256
from keras.preprocessing.image import ImageDataGenerator
datagen = ImageDataGenerator(rotation_range = 5, horizontal_flip = True, width_shift_range = 0.05,
height_shift_range = 0.05, shear_range = 5)
```

```
datagen.fit(x_train)
best_ckpt = ModelCheckpoint(os.path.join(path, 'best_model.h5'), monitor = 'val_acc',
save_best_only = True)
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 3*x_train.shape[0
]/batch_size, epochs = 10, validation_data =(x_test, y_test), callbacks = [csv, ckpt, best_ckpt])
model.save_weights(os.path.join(path, '207epochs.h5'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is
deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-
packages/tensorflow/python/ops/math_grad.py:102: div (from tensorflow.python.ops.math_ops) is depr
ecated and will be removed in a future version.
Instructions for updating:
Deprecated in favor of operator or tf.math.divide.
Epoch 1/10
1172/1171 [==============================] - 822s 702ms/step - loss: 0.0498 - acc: 0.9969 - val_lo
ss: 0.3443 - val_acc: 0.9235
Epoch 2/10
1172/1171 [==============================] - 806s 687ms/step - loss: 0.0418 - acc: 0.9972 - val_lo
ss: 0.3436 - val_acc: 0.9238
Epoch 3/10
1172/1171 [==============================] - 805s 687ms/step - loss: 0.0398 - acc: 0.9973 - val_lo
ss: 0.3457 - val_acc: 0.9222
Epoch 4/10
1172/1171 [==============================] - 805s 687ms/step - loss: 0.0384 - acc: 0.9972 - val_lo
ss: 0.3404 - val_acc: 0.9242
Epoch 5/10
1172/1171 [==============================] - 806s 687ms/step - loss: 0.0373 - acc: 0.9972 - val_lo
ss: 0.3404 - val_acc: 0.9234
Epoch 6/10
1172/1171 [==============================] - 806s 687ms/step - loss: 0.0372 - acc: 0.9969 - val_lo
ss: 0.3386 - val_acc: 0.9223
Epoch 7/10
 740/1171 [=================>............] - ETA: 4:51 - loss: 0.0367 - acc: 0.9970


---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-15-3aaabc2d6cc6> in <module>()
      6 datagen.fit(x_train)
      7 best_ckpt = ModelCheckpoint(os.path.join(path, 'best_model.h5'), monitor = 'val_acc', save_
best_only = True)
----> 8 model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 3*x_train
.shape[0]/batch_size, epochs = 10, validation_data =(x_test, y_test), callbacks = [csv, ckpt,
best_ckpt])
      9 model.save_weights(os.path.join(path, '177epochs.h5'))

/usr/local/lib/python3.6/dist-packages/keras/legacy/interfaces.py in wrapper(*args, **kwargs)
     89                 warnings.warn('Update your `' + object_name + '` call to the ' +
     90                               'Keras 2 API: ' + signature, stacklevel=2)
---> 91             return func(*args, **kwargs)
     92         wrapper._original_function = func
     93         return wrapper

/usr/local/lib/python3.6/dist-packages/keras/engine/training.py in fit_generator(self, generator,
steps_per_epoch, epochs, verbose, callbacks, validation_data, validation_steps, class_weight,
max_queue_size, workers, use_multiprocessing, shuffle, initial_epoch)
   1416             use_multiprocessing=use_multiprocessing,
   1417             shuffle=shuffle,
-> 1418             initial_epoch=initial_epoch)
   1419
   1420     @interfaces.legacy_generator_methods_support

/usr/local/lib/python3.6/dist-packages/keras/engine/training_generator.py in fit_generator(model,
generator, steps_per_epoch, epochs, verbose, callbacks, validation_data, validation_steps,
class_weight, max_queue_size, workers, use_multiprocessing, shuffle, initial_epoch)
    215                 outs = model.train_on_batch(x, y,
    216                                             sample_weight=sample_weight,
--> 217                                             class_weight=class_weight)
    218
    219                 outs = to_list(outs)

/usr/local/lib/python3.6/dist-packages/keras/engine/training.py in train_on_batch(self, x, y,
sample_weight, class_weight)
```

```
   1215             ins = x + y + sample_weights
   1216         self._make_train_function()
-> 1217         outputs = self.train_function(ins)
   1218         return unpack_singleton(outputs)
   1219

/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py in __call__(self,
inputs)
   2713                 return self._legacy_call(inputs)
   2714
-> 2715             return self._call(inputs)
   2716         else:
   2717             if py_any(is_tensor(x) for x in inputs):

/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py in _call(self, inputs)
   2673             fetched = self._callable_fn(*array_vals, run_metadata=self.run_metadata)
   2674         else:
-> 2675             fetched = self._callable_fn(*array_vals)
   2676         return fetched[:len(self.outputs)]
   2677

/usr/local/lib/python3.6/dist-packages/tensorflow/python/client/session.py in __call__(self,
*args, **kwargs)
   1437           ret = tf_session.TF_SessionRunCallable(
   1438               self._session._session, self._handle, args, status,
-> 1439               run_metadata_ptr)
   1440         if run_metadata:
   1441           proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

KeyboardInterrupt:
```

In [0]:

```
keras.backend.set_value(model.optimizer.lr, 0.0001)
keras.backend.set_value(model.optimizer.momentum, 0.5)
bacth_size = 512
datagen = ImageDataGenerator(rotation_range = 5, horizontal_flip = True, width_shift_range = 0.05,
height_shift_range = 0.05, shear_range = 5)
datagen.fit(x_train)
model.fit_generator(datagen.flow(x_train, y_train, batch_size), steps_per_epoch = 3*x_train.shape[0
]/batch_size, epochs = 5 validation_data =(x_test, y_test), callbacks = [csv, ckpt, best_ckpt])
```

```
Epoch 1/5
1172/1171 [==============================] - 807s 688ms/step - loss: 0.0342 - acc: 0.9971 - val_lo
ss: 0.3361 - val_acc: 0.9227
Epoch 2/5
1172/1171 [==============================] - 804s 686ms/step - loss: 0.0335 - acc: 0.9973 - val_lo
ss: 0.3415 - val_acc: 0.9216
Epoch 3/5
1172/1171 [==============================] - 804s 686ms/step - loss: 0.0335 - acc: 0.9970 - val_lo
ss: 0.3467 - val_acc: 0.9207
Epoch 4/5
1172/1171 [==============================] - 805s 687ms/step - loss: 0.0338 - acc: 0.9970 - val_lo
ss: 0.3481 - val_acc: 0.9211
Epoch 5/5
1172/1171 [==============================] - 805s 687ms/step - loss: 0.0334 - acc: 0.9969 - val_lo
ss: 0.3397 - val_acc: 0.9218
```

In [0]:

```
model.save_weights(os.path.join(path, '208epochs.h5'))
```

Applied evaluate on test data

In [0]:

```
model.evaluate(x_test, y_test)
```

```
10000/10000 [==============================] - 20s 2ms/step
```

Out[0]:

Out[6]:

```
[0.33972805423736574, 0.9218]
```

Got accuracy of 92 % on test data

Summary

1. I reach over 90 percent classificaion accuracy on CIFAR_10, a dataset with 50 000 training images in 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck). I tried different operations, the best seemed to be rotations, horizontal flipping and vertical or horizontal shifts. The model is a VGG-type convolutional network with 6 conv layers and one dense fully connected layer before the output. Apart from image augmentation, batch normalisation (in each layer) and dropout in the dense layer is used.