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
# from keras.layers import Conv2D, MaxPooling2D, BatchNormalization
# from keras.layers import Concatenate
# from keras.optimizers import Adam
```

In [2]:

```
# Load necessary libraries
from tensorflow.keras import models, layers
from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization, Activation, Flatten
from tensorflow.keras.optimizers import Adam
from numpy import expand_dims
from keras.preprocessing.image import load_img
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from keras import regularizers
from matplotlib import pyplot
```

```
Using TensorFlow backend.
```

In [0]:

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

In [0]:

```
# I will save model to every epochs in drive
# from google.colab import drive
# drive.mount('/content/gdrive')
```

In [0]:

```
# !ls /content/gdrive/My\ Drive
```

In [0]:

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

In [7]:

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

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

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

In [8]:

```
X_train.shape
```

Out[8]:

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

In [9]:

```
X_test.shape
```

Out[9]:

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

In [0]:

```
# Dense Block
def denseblock(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    temp = input
    for _ in range(l):
        BatchNorm = layers.BatchNormalization()(temp)
        relu = layers.Activation('relu')(BatchNorm)
        Conv2D_3_3 = layers.Conv2D(int(num_filter*compression), (3,3), use_bias=False ,padding='sam
e')(relu)
        if dropout_rate>0:
            Conv2D_3_3 = layers.Dropout(dropout_rate)(Conv2D_3_3)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_3_3])

        temp = concat

    return temp

## transition Blosck
def transition(input, num_filter = 12, dropout_rate = 0.2):
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (1,1), use_bias=False ,padding='
same')(relu)
    if dropout_rate>0:
         Conv2D_BottleNeck = layers.Dropout(dropout_rate)(Conv2D_BottleNeck)
    avg = layers.AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)
    return avg

#output layer
def output_layer(input):
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    AvgPooling = layers.AveragePooling2D(pool_size=(2,2))(relu)
    flat = layers.Flatten()(AvgPooling)
    output = layers.Dense(num_classes, activation='softmax')(flat)
    return output
```

In [0]:

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

First_Block = denseblock(First_Conv2D, num_filter, dropout_rate)
First_Transition = transition(First_Block, num_filter, dropout_rate)

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

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

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

In [10]:

```
#https://arxiv.org/pdf/1608.06993.pdf
from IPython.display import IFrame, YouTubeVideo
YouTubeVideo(id='-W6y8xnd--U', width=600)
```

Out[10]:

In [11]:

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

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_1 (InputLayer)         [(None, 32, 32, 3)]       0
_________________________________________________________________________________________________
conv2d (Conv2D)              (None, 32, 32, 12)        324         input_1[0][0]
_________________________________________________________________________________________________
batch_normalization (BatchNorma (None, 32, 32, 12)     48          conv2d[0][0]
_________________________________________________________________________________________________
activation (Activation)      (None, 32, 32, 12)        0           batch_normalization[0][0]
_________________________________________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 6)         648         activation[0][0]
_________________________________________________________________________________________________
dropout (Dropout)            (None, 32, 32, 6)         0           conv2d_1[0][0]
_________________________________________________________________________________________________
concatenate (Concatenate)    (None, 32, 32, 18)        0           conv2d[0][0]
                                                                   dropout[0][0]
_________________________________________________________________________________________________
batch_normalization_1 (BatchNor (None, 32, 32, 18)     72          concatenate[0][0]
_________________________________________________________________________________________________
activation_1 (Activation)    (None, 32, 32, 18)        0           batch_normalization_1[0][0]
_________________________________________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 6)         972         activation_1[0][0]
_________________________________________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 6)         0           conv2d_2[0][0]
_________________________________________________________________________________________________
concatenate_1 (Concatenate)  (None, 32, 32, 24)        0           concatenate[0][0]
                                                                   dropout_1[0][0]
_________________________________________________________________________________________________
batch_normalization_2 (BatchNor (None, 32, 32, 24)     96          concatenate_1[0][0]
_________________________________________________________________________________________________
activation_2 (Activation)    (None, 32, 32, 24)        0           batch_normalization_2[0][0]
_________________________________________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 32, 6)         1296        activation_2[0][0]
_________________________________________________________________________________________________
dropout_2 (Dropout)          (None, 32, 32, 6)         0           conv2d_3[0][0]
_________________________________________________________________________________________________
concatenate_2 (Concatenate)  (None, 32, 32, 30)        0           concatenate_1[0][0]
```

concatenate_2 (Concatenate)	(None, 32, 32, 30)	0	concatenate_1[0][0] dropout_2[0][0]
batch_normalization_3 (BatchNor	(None, 32, 32, 30)	120	concatenate_2[0][0]
activation_3 (Activation)	(None, 32, 32, 30)	0	batch_normalization_3[0][0]
conv2d_4 (Conv2D)	(None, 32, 32, 6)	1620	activation_3[0][0]
dropout_3 (Dropout)	(None, 32, 32, 6)	0	conv2d_4[0][0]
concatenate_3 (Concatenate)	(None, 32, 32, 36)	0	concatenate_2[0][0] dropout_3[0][0]
batch_normalization_4 (BatchNor	(None, 32, 32, 36)	144	concatenate_3[0][0]
activation_4 (Activation)	(None, 32, 32, 36)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 32, 32, 6)	1944	activation_4[0][0]
dropout_4 (Dropout)	(None, 32, 32, 6)	0	conv2d_5[0][0]
concatenate_4 (Concatenate)	(None, 32, 32, 42)	0	concatenate_3[0][0] dropout_4[0][0]
batch_normalization_5 (BatchNor	(None, 32, 32, 42)	168	concatenate_4[0][0]
activation_5 (Activation)	(None, 32, 32, 42)	0	batch_normalization_5[0][0]
conv2d_6 (Conv2D)	(None, 32, 32, 6)	2268	activation_5[0][0]
dropout_5 (Dropout)	(None, 32, 32, 6)	0	conv2d_6[0][0]
concatenate_5 (Concatenate)	(None, 32, 32, 48)	0	concatenate_4[0][0] dropout_5[0][0]
batch_normalization_6 (BatchNor	(None, 32, 32, 48)	192	concatenate_5[0][0]
activation_6 (Activation)	(None, 32, 32, 48)	0	batch_normalization_6[0][0]
conv2d_7 (Conv2D)	(None, 32, 32, 6)	2592	activation_6[0][0]
dropout_6 (Dropout)	(None, 32, 32, 6)	0	conv2d_7[0][0]
concatenate_6 (Concatenate)	(None, 32, 32, 54)	0	concatenate_5[0][0] dropout_6[0][0]
batch_normalization_7 (BatchNor	(None, 32, 32, 54)	216	concatenate_6[0][0]
activation_7 (Activation)	(None, 32, 32, 54)	0	batch_normalization_7[0][0]
conv2d_8 (Conv2D)	(None, 32, 32, 6)	2916	activation_7[0][0]
dropout_7 (Dropout)	(None, 32, 32, 6)	0	conv2d_8[0][0]
concatenate_7 (Concatenate)	(None, 32, 32, 60)	0	concatenate_6[0][0] dropout_7[0][0]
batch_normalization_8 (BatchNor	(None, 32, 32, 60)	240	concatenate_7[0][0]
activation_8 (Activation)	(None, 32, 32, 60)	0	batch_normalization_8[0][0]
conv2d_9 (Conv2D)	(None, 32, 32, 6)	3240	activation_8[0][0]
dropout_8 (Dropout)	(None, 32, 32, 6)	0	conv2d_9[0][0]
concatenate_8 (Concatenate)	(None, 32, 32, 66)	0	concatenate_7[0][0] dropout_8[0][0]
batch_normalization_9 (BatchNor	(None, 32, 32, 66)	264	concatenate_8[0][0]
activation_9 (Activation)	(None, 32, 32, 66)	0	batch_normalization_9[0][0]
conv2d_10 (Conv2D)	(None, 32, 32, 6)	3564	activation_9[0][0]
dropout_9 (Dropout)	(None, 32, 32, 6)	0	conv2d_10[0][0]
concatenate_9 (Concatenate)	(None, 32, 32, 72)	0	concatenate_8[0][0]

concatenate_9 (Concatenate)	(None, 32, 32, 72)	0	concatenate_8[0][0] dropout_9[0][0]
batch_normalization_10 (BatchNo	(None, 32, 32, 72)	288	concatenate_9[0][0]
activation_10 (Activation)	(None, 32, 32, 72)	0	batch_normalization_10[0][0]
conv2d_11 (Conv2D)	(None, 32, 32, 6)	3888	activation_10[0][0]
dropout_10 (Dropout)	(None, 32, 32, 6)	0	conv2d_11[0][0]
concatenate_10 (Concatenate)	(None, 32, 32, 78)	0	concatenate_9[0][0] dropout_10[0][0]
batch_normalization_11 (BatchNo	(None, 32, 32, 78)	312	concatenate_10[0][0]
activation_11 (Activation)	(None, 32, 32, 78)	0	batch_normalization_11[0][0]
conv2d_12 (Conv2D)	(None, 32, 32, 6)	4212	activation_11[0][0]
dropout_11 (Dropout)	(None, 32, 32, 6)	0	conv2d_12[0][0]
concatenate_11 (Concatenate)	(None, 32, 32, 84)	0	concatenate_10[0][0] dropout_11[0][0]
batch_normalization_12 (BatchNo	(None, 32, 32, 84)	336	concatenate_11[0][0]
activation_12 (Activation)	(None, 32, 32, 84)	0	batch_normalization_12[0][0]
conv2d_13 (Conv2D)	(None, 32, 32, 6)	504	activation_12[0][0]
dropout_12 (Dropout)	(None, 32, 32, 6)	0	conv2d_13[0][0]
average_pooling2d (AveragePooli	(None, 16, 16, 6)	0	dropout_12[0][0]
batch_normalization_13 (BatchNo	(None, 16, 16, 6)	24	average_pooling2d[0][0]
activation_13 (Activation)	(None, 16, 16, 6)	0	batch_normalization_13[0][0]
conv2d_14 (Conv2D)	(None, 16, 16, 6)	324	activation_13[0][0]
dropout_13 (Dropout)	(None, 16, 16, 6)	0	conv2d_14[0][0]
concatenate_12 (Concatenate)	(None, 16, 16, 12)	0	average_pooling2d[0][0] dropout_13[0][0]
batch_normalization_14 (BatchNo	(None, 16, 16, 12)	48	concatenate_12[0][0]
activation_14 (Activation)	(None, 16, 16, 12)	0	batch_normalization_14[0][0]
conv2d_15 (Conv2D)	(None, 16, 16, 6)	648	activation_14[0][0]
dropout_14 (Dropout)	(None, 16, 16, 6)	0	conv2d_15[0][0]
concatenate_13 (Concatenate)	(None, 16, 16, 18)	0	concatenate_12[0][0] dropout_14[0][0]
batch_normalization_15 (BatchNo	(None, 16, 16, 18)	72	concatenate_13[0][0]
activation_15 (Activation)	(None, 16, 16, 18)	0	batch_normalization_15[0][0]
conv2d_16 (Conv2D)	(None, 16, 16, 6)	972	activation_15[0][0]
dropout_15 (Dropout)	(None, 16, 16, 6)	0	conv2d_16[0][0]
concatenate_14 (Concatenate)	(None, 16, 16, 24)	0	concatenate_13[0][0] dropout_15[0][0]
batch_normalization_16 (BatchNo	(None, 16, 16, 24)	96	concatenate_14[0][0]
activation_16 (Activation)	(None, 16, 16, 24)	0	batch_normalization_16[0][0]
conv2d_17 (Conv2D)	(None, 16, 16, 6)	1296	activation_16[0][0]
dropout_16 (Dropout)	(None, 16, 16, 6)	0	conv2d_17[0][0]
concatenate_15 (Concatenate)	(None, 16, 16, 30)	0	concatenate_14[0][0] dropout_16[0][0]

			dropout_16[0][0]
batch_normalization_17 (BatchNo	(None, 16, 16, 30)	120	concatenate_15[0][0]
activation_17 (Activation)	(None, 16, 16, 30)	0	batch_normalization_17[0][0]
conv2d_18 (Conv2D)	(None, 16, 16, 6)	1620	activation_17[0][0]
dropout_17 (Dropout)	(None, 16, 16, 6)	0	conv2d_18[0][0]
concatenate_16 (Concatenate)	(None, 16, 16, 36)	0	concatenate_15[0][0] dropout_17[0][0]
batch_normalization_18 (BatchNo	(None, 16, 16, 36)	144	concatenate_16[0][0]
activation_18 (Activation)	(None, 16, 16, 36)	0	batch_normalization_18[0][0]
conv2d_19 (Conv2D)	(None, 16, 16, 6)	1944	activation_18[0][0]
dropout_18 (Dropout)	(None, 16, 16, 6)	0	conv2d_19[0][0]
concatenate_17 (Concatenate)	(None, 16, 16, 42)	0	concatenate_16[0][0] dropout_18[0][0]
batch_normalization_19 (BatchNo	(None, 16, 16, 42)	168	concatenate_17[0][0]
activation_19 (Activation)	(None, 16, 16, 42)	0	batch_normalization_19[0][0]
conv2d_20 (Conv2D)	(None, 16, 16, 6)	2268	activation_19[0][0]
dropout_19 (Dropout)	(None, 16, 16, 6)	0	conv2d_20[0][0]
concatenate_18 (Concatenate)	(None, 16, 16, 48)	0	concatenate_17[0][0] dropout_19[0][0]
batch_normalization_20 (BatchNo	(None, 16, 16, 48)	192	concatenate_18[0][0]
activation_20 (Activation)	(None, 16, 16, 48)	0	batch_normalization_20[0][0]
conv2d_21 (Conv2D)	(None, 16, 16, 6)	2592	activation_20[0][0]
dropout_20 (Dropout)	(None, 16, 16, 6)	0	conv2d_21[0][0]
concatenate_19 (Concatenate)	(None, 16, 16, 54)	0	concatenate_18[0][0] dropout_20[0][0]
batch_normalization_21 (BatchNo	(None, 16, 16, 54)	216	concatenate_19[0][0]
activation_21 (Activation)	(None, 16, 16, 54)	0	batch_normalization_21[0][0]
conv2d_22 (Conv2D)	(None, 16, 16, 6)	2916	activation_21[0][0]
dropout_21 (Dropout)	(None, 16, 16, 6)	0	conv2d_22[0][0]
concatenate_20 (Concatenate)	(None, 16, 16, 60)	0	concatenate_19[0][0] dropout_21[0][0]
batch_normalization_22 (BatchNo	(None, 16, 16, 60)	240	concatenate_20[0][0]
activation_22 (Activation)	(None, 16, 16, 60)	0	batch_normalization_22[0][0]
conv2d_23 (Conv2D)	(None, 16, 16, 6)	3240	activation_22[0][0]
dropout_22 (Dropout)	(None, 16, 16, 6)	0	conv2d_23[0][0]
concatenate_21 (Concatenate)	(None, 16, 16, 66)	0	concatenate_20[0][0] dropout_22[0][0]
batch_normalization_23 (BatchNo	(None, 16, 16, 66)	264	concatenate_21[0][0]
activation_23 (Activation)	(None, 16, 16, 66)	0	batch_normalization_23[0][0]
conv2d_24 (Conv2D)	(None, 16, 16, 6)	3564	activation_23[0][0]
dropout_23 (Dropout)	(None, 16, 16, 6)	0	conv2d_24[0][0]
concatenate_22 (Concatenate)	(None, 16, 16, 72)	0	concatenate_21[0][0]

			dropout_23[0][0]
batch_normalization_24 (BatchNo	(None, 16, 16, 72)	288	concatenate_22[0][0]
activation_24 (Activation)	(None, 16, 16, 72)	0	batch_normalization_24[0][0]
conv2d_25 (Conv2D)	(None, 16, 16, 6)	3888	activation_24[0][0]
dropout_24 (Dropout)	(None, 16, 16, 6)	0	conv2d_25[0][0]
concatenate_23 (Concatenate)	(None, 16, 16, 78)	0	concatenate_22[0][0] dropout_24[0][0]
batch_normalization_25 (BatchNo	(None, 16, 16, 78)	312	concatenate_23[0][0]
activation_25 (Activation)	(None, 16, 16, 78)	0	batch_normalization_25[0][0]
conv2d_26 (Conv2D)	(None, 16, 16, 6)	468	activation_25[0][0]
dropout_25 (Dropout)	(None, 16, 16, 6)	0	conv2d_26[0][0]
average_pooling2d_1 (AveragePoo	(None, 8, 8, 6)	0	dropout_25[0][0]
batch_normalization_26 (BatchNo	(None, 8, 8, 6)	24	average_pooling2d_1[0][0]
activation_26 (Activation)	(None, 8, 8, 6)	0	batch_normalization_26[0][0]
conv2d_27 (Conv2D)	(None, 8, 8, 6)	324	activation_26[0][0]
dropout_26 (Dropout)	(None, 8, 8, 6)	0	conv2d_27[0][0]
concatenate_24 (Concatenate)	(None, 8, 8, 12)	0	average_pooling2d_1[0][0] dropout_26[0][0]
batch_normalization_27 (BatchNo	(None, 8, 8, 12)	48	concatenate_24[0][0]
activation_27 (Activation)	(None, 8, 8, 12)	0	batch_normalization_27[0][0]
conv2d_28 (Conv2D)	(None, 8, 8, 6)	648	activation_27[0][0]
dropout_27 (Dropout)	(None, 8, 8, 6)	0	conv2d_28[0][0]
concatenate_25 (Concatenate)	(None, 8, 8, 18)	0	concatenate_24[0][0] dropout_27[0][0]
batch_normalization_28 (BatchNo	(None, 8, 8, 18)	72	concatenate_25[0][0]
activation_28 (Activation)	(None, 8, 8, 18)	0	batch_normalization_28[0][0]
conv2d_29 (Conv2D)	(None, 8, 8, 6)	972	activation_28[0][0]
dropout_28 (Dropout)	(None, 8, 8, 6)	0	conv2d_29[0][0]
concatenate_26 (Concatenate)	(None, 8, 8, 24)	0	concatenate_25[0][0] dropout_28[0][0]
batch_normalization_29 (BatchNo	(None, 8, 8, 24)	96	concatenate_26[0][0]
activation_29 (Activation)	(None, 8, 8, 24)	0	batch_normalization_29[0][0]
conv2d_30 (Conv2D)	(None, 8, 8, 6)	1296	activation_29[0][0]
dropout_29 (Dropout)	(None, 8, 8, 6)	0	conv2d_30[0][0]
concatenate_27 (Concatenate)	(None, 8, 8, 30)	0	concatenate_26[0][0] dropout_29[0][0]
batch_normalization_30 (BatchNo	(None, 8, 8, 30)	120	concatenate_27[0][0]
activation_30 (Activation)	(None, 8, 8, 30)	0	batch_normalization_30[0][0]
conv2d_31 (Conv2D)	(None, 8, 8, 6)	1620	activation_30[0][0]
dropout_30 (Dropout)	(None, 8, 8, 6)	0	conv2d_31[0][0]
concatenate_28 (Concatenate)	(None, 8, 8, 36)	0	concatenate_27[0][0] dropout_30[0][0]

batch_normalization_31 (BatchNo	(None, 8, 8, 36)	144	concatenate_28[0][0]
activation_31 (Activation)	(None, 8, 8, 36)	0	batch_normalization_31[0][0]
conv2d_32 (Conv2D)	(None, 8, 8, 6)	1944	activation_31[0][0]
dropout_31 (Dropout)	(None, 8, 8, 6)	0	conv2d_32[0][0]
concatenate_29 (Concatenate)	(None, 8, 8, 42)	0	concatenate_28[0][0] dropout_31[0][0]
batch_normalization_32 (BatchNo	(None, 8, 8, 42)	168	concatenate_29[0][0]
activation_32 (Activation)	(None, 8, 8, 42)	0	batch_normalization_32[0][0]
conv2d_33 (Conv2D)	(None, 8, 8, 6)	2268	activation_32[0][0]
dropout_32 (Dropout)	(None, 8, 8, 6)	0	conv2d_33[0][0]
concatenate_30 (Concatenate)	(None, 8, 8, 48)	0	concatenate_29[0][0] dropout_32[0][0]
batch_normalization_33 (BatchNo	(None, 8, 8, 48)	192	concatenate_30[0][0]
activation_33 (Activation)	(None, 8, 8, 48)	0	batch_normalization_33[0][0]
conv2d_34 (Conv2D)	(None, 8, 8, 6)	2592	activation_33[0][0]
dropout_33 (Dropout)	(None, 8, 8, 6)	0	conv2d_34[0][0]
concatenate_31 (Concatenate)	(None, 8, 8, 54)	0	concatenate_30[0][0] dropout_33[0][0]
batch_normalization_34 (BatchNo	(None, 8, 8, 54)	216	concatenate_31[0][0]
activation_34 (Activation)	(None, 8, 8, 54)	0	batch_normalization_34[0][0]
conv2d_35 (Conv2D)	(None, 8, 8, 6)	2916	activation_34[0][0]
dropout_34 (Dropout)	(None, 8, 8, 6)	0	conv2d_35[0][0]
concatenate_32 (Concatenate)	(None, 8, 8, 60)	0	concatenate_31[0][0] dropout_34[0][0]
batch_normalization_35 (BatchNo	(None, 8, 8, 60)	240	concatenate_32[0][0]
activation_35 (Activation)	(None, 8, 8, 60)	0	batch_normalization_35[0][0]
conv2d_36 (Conv2D)	(None, 8, 8, 6)	3240	activation_35[0][0]
dropout_35 (Dropout)	(None, 8, 8, 6)	0	conv2d_36[0][0]
concatenate_33 (Concatenate)	(None, 8, 8, 66)	0	concatenate_32[0][0] dropout_35[0][0]
batch_normalization_36 (BatchNo	(None, 8, 8, 66)	264	concatenate_33[0][0]
activation_36 (Activation)	(None, 8, 8, 66)	0	batch_normalization_36[0][0]
conv2d_37 (Conv2D)	(None, 8, 8, 6)	3564	activation_36[0][0]
dropout_36 (Dropout)	(None, 8, 8, 6)	0	conv2d_37[0][0]
concatenate_34 (Concatenate)	(None, 8, 8, 72)	0	concatenate_33[0][0] dropout_36[0][0]
batch_normalization_37 (BatchNo	(None, 8, 8, 72)	288	concatenate_34[0][0]
activation_37 (Activation)	(None, 8, 8, 72)	0	batch_normalization_37[0][0]
conv2d_38 (Conv2D)	(None, 8, 8, 6)	3888	activation_37[0][0]
dropout_37 (Dropout)	(None, 8, 8, 6)	0	conv2d_38[0][0]
concatenate_35 (Concatenate)	(None, 8, 8, 78)	0	concatenate_34[0][0] dropout_37[0][0]

batch_normalization_38 (BatchNo	(None, 8, 8, 78)	312	concatenate_35[0][0]
activation_38 (Activation)	(None, 8, 8, 78)	0	batch_normalization_38[0][0]
conv2d_39 (Conv2D)	(None, 8, 8, 6)	468	activation_38[0][0]
dropout_38 (Dropout)	(None, 8, 8, 6)	0	conv2d_39[0][0]
average_pooling2d_2 (AveragePoo	(None, 4, 4, 6)	0	dropout_38[0][0]
batch_normalization_39 (BatchNo	(None, 4, 4, 6)	24	average_pooling2d_2[0][0]
activation_39 (Activation)	(None, 4, 4, 6)	0	batch_normalization_39[0][0]
conv2d_40 (Conv2D)	(None, 4, 4, 6)	324	activation_39[0][0]
dropout_39 (Dropout)	(None, 4, 4, 6)	0	conv2d_40[0][0]
concatenate_36 (Concatenate)	(None, 4, 4, 12)	0	average_pooling2d_2[0][0] dropout_39[0][0]
batch_normalization_40 (BatchNo	(None, 4, 4, 12)	48	concatenate_36[0][0]
activation_40 (Activation)	(None, 4, 4, 12)	0	batch_normalization_40[0][0]
conv2d_41 (Conv2D)	(None, 4, 4, 6)	648	activation_40[0][0]
dropout_40 (Dropout)	(None, 4, 4, 6)	0	conv2d_41[0][0]
concatenate_37 (Concatenate)	(None, 4, 4, 18)	0	concatenate_36[0][0] dropout_40[0][0]
batch_normalization_41 (BatchNo	(None, 4, 4, 18)	72	concatenate_37[0][0]
activation_41 (Activation)	(None, 4, 4, 18)	0	batch_normalization_41[0][0]
conv2d_42 (Conv2D)	(None, 4, 4, 6)	972	activation_41[0][0]
dropout_41 (Dropout)	(None, 4, 4, 6)	0	conv2d_42[0][0]
concatenate_38 (Concatenate)	(None, 4, 4, 24)	0	concatenate_37[0][0] dropout_41[0][0]
batch_normalization_42 (BatchNo	(None, 4, 4, 24)	96	concatenate_38[0][0]
activation_42 (Activation)	(None, 4, 4, 24)	0	batch_normalization_42[0][0]
conv2d_43 (Conv2D)	(None, 4, 4, 6)	1296	activation_42[0][0]
dropout_42 (Dropout)	(None, 4, 4, 6)	0	conv2d_43[0][0]
concatenate_39 (Concatenate)	(None, 4, 4, 30)	0	concatenate_38[0][0] dropout_42[0][0]
batch_normalization_43 (BatchNo	(None, 4, 4, 30)	120	concatenate_39[0][0]
activation_43 (Activation)	(None, 4, 4, 30)	0	batch_normalization_43[0][0]
conv2d_44 (Conv2D)	(None, 4, 4, 6)	1620	activation_43[0][0]
dropout_43 (Dropout)	(None, 4, 4, 6)	0	conv2d_44[0][0]
concatenate_40 (Concatenate)	(None, 4, 4, 36)	0	concatenate_39[0][0] dropout_43[0][0]
batch_normalization_44 (BatchNo	(None, 4, 4, 36)	144	concatenate_40[0][0]
activation_44 (Activation)	(None, 4, 4, 36)	0	batch_normalization_44[0][0]
conv2d_45 (Conv2D)	(None, 4, 4, 6)	1944	activation_44[0][0]
dropout_44 (Dropout)	(None, 4, 4, 6)	0	conv2d_45[0][0]
concatenate_41 (Concatenate)	(None, 4, 4, 42)	0	concatenate_40[0][0] dropout_44[0][0]

```
batch_normalization_45 (BatchNo (None, 4, 4, 42)     168         concatenate_41[0][0]
__________________________________________________________________________________________________
activation_45 (Activation)      (None, 4, 4, 42)     0           batch_normalization_45[0][0]
__________________________________________________________________________________________________
conv2d_46 (Conv2D)              (None, 4, 4, 6)      2268        activation_45[0][0]
__________________________________________________________________________________________________
dropout_45 (Dropout)            (None, 4, 4, 6)      0           conv2d_46[0][0]
__________________________________________________________________________________________________
concatenate_42 (Concatenate)    (None, 4, 4, 48)     0           concatenate_41[0][0]
                                                                 dropout_45[0][0]
__________________________________________________________________________________________________
batch_normalization_46 (BatchNo (None, 4, 4, 48)     192         concatenate_42[0][0]
__________________________________________________________________________________________________
activation_46 (Activation)      (None, 4, 4, 48)     0           batch_normalization_46[0][0]
__________________________________________________________________________________________________
conv2d_47 (Conv2D)              (None, 4, 4, 6)      2592        activation_46[0][0]
__________________________________________________________________________________________________
dropout_46 (Dropout)            (None, 4, 4, 6)      0           conv2d_47[0][0]
__________________________________________________________________________________________________
concatenate_43 (Concatenate)    (None, 4, 4, 54)     0           concatenate_42[0][0]
                                                                 dropout_46[0][0]
__________________________________________________________________________________________________
batch_normalization_47 (BatchNo (None, 4, 4, 54)     216         concatenate_43[0][0]
__________________________________________________________________________________________________
activation_47 (Activation)      (None, 4, 4, 54)     0           batch_normalization_47[0][0]
__________________________________________________________________________________________________
conv2d_48 (Conv2D)              (None, 4, 4, 6)      2916        activation_47[0][0]
__________________________________________________________________________________________________
dropout_47 (Dropout)            (None, 4, 4, 6)      0           conv2d_48[0][0]
__________________________________________________________________________________________________
concatenate_44 (Concatenate)    (None, 4, 4, 60)     0           concatenate_43[0][0]
                                                                 dropout_47[0][0]
__________________________________________________________________________________________________
batch_normalization_48 (BatchNo (None, 4, 4, 60)     240         concatenate_44[0][0]
__________________________________________________________________________________________________
activation_48 (Activation)      (None, 4, 4, 60)     0           batch_normalization_48[0][0]
__________________________________________________________________________________________________
conv2d_49 (Conv2D)              (None, 4, 4, 6)      3240        activation_48[0][0]
__________________________________________________________________________________________________
dropout_48 (Dropout)            (None, 4, 4, 6)      0           conv2d_49[0][0]
__________________________________________________________________________________________________
concatenate_45 (Concatenate)    (None, 4, 4, 66)     0           concatenate_44[0][0]
                                                                 dropout_48[0][0]
__________________________________________________________________________________________________
batch_normalization_49 (BatchNo (None, 4, 4, 66)     264         concatenate_45[0][0]
__________________________________________________________________________________________________
activation_49 (Activation)      (None, 4, 4, 66)     0           batch_normalization_49[0][0]
__________________________________________________________________________________________________
conv2d_50 (Conv2D)              (None, 4, 4, 6)      3564        activation_49[0][0]
__________________________________________________________________________________________________
dropout_49 (Dropout)            (None, 4, 4, 6)      0           conv2d_50[0][0]
__________________________________________________________________________________________________
concatenate_46 (Concatenate)    (None, 4, 4, 72)     0           concatenate_45[0][0]
                                                                 dropout_49[0][0]
__________________________________________________________________________________________________
batch_normalization_50 (BatchNo (None, 4, 4, 72)     288         concatenate_46[0][0]
__________________________________________________________________________________________________
activation_50 (Activation)      (None, 4, 4, 72)     0           batch_normalization_50[0][0]
__________________________________________________________________________________________________
conv2d_51 (Conv2D)              (None, 4, 4, 6)      3888        activation_50[0][0]
__________________________________________________________________________________________________
dropout_50 (Dropout)            (None, 4, 4, 6)      0           conv2d_51[0][0]
__________________________________________________________________________________________________
concatenate_47 (Concatenate)    (None, 4, 4, 78)     0           concatenate_46[0][0]
                                                                 dropout_50[0][0]
__________________________________________________________________________________________________
batch_normalization_51 (BatchNo (None, 4, 4, 78)     312         concatenate_47[0][0]
__________________________________________________________________________________________________
activation_51 (Activation)      (None, 4, 4, 78)     0           batch_normalization_51[0][0]
__________________________________________________________________________________________________
average_pooling2d_3 (AveragePoo (None, 2, 2, 78)     0           activation_51[0][0]
__________________________________________________________________________________________________
flatten (Flatten)               (None, 312)          0           average_pooling2d_3[0][0]
__________________________________________________________________________________________________
dense (Dense)                   (None, 10)           3130        flatten[0][0]
==================================================================================================
Total params: 118,918
```

```
Trainable params: 114,394
Non-trainable params: 4,524
_________________________________________________________________
```

In [0]:

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

In [13]:

```
model.fit(X_train, y_train,
                    batch_size=batch_size,
                    epochs=epochs,
                    verbose=1,
                    validation_data=(X_test, y_test))
```

```
Epoch 1/10
391/391 [==============================] - 25s 65ms/step - loss: 1.7582 - accuracy: 0.3436 - val_l
oss: 1.5876 - val_accuracy: 0.4242
Epoch 2/10
391/391 [==============================] - 24s 61ms/step - loss: 1.4103 - accuracy: 0.4789 - val_l
oss: 1.3950 - val_accuracy: 0.4987
Epoch 3/10
391/391 [==============================] - 24s 61ms/step - loss: 1.2338 - accuracy: 0.5541 - val_l
oss: 1.6258 - val_accuracy: 0.4904
Epoch 4/10
391/391 [==============================] - 24s 61ms/step - loss: 1.1185 - accuracy: 0.5990 - val_l
oss: 1.1344 - val_accuracy: 0.6144
Epoch 5/10
391/391 [==============================] - 24s 61ms/step - loss: 1.0438 - accuracy: 0.6258 - val_l
oss: 1.1873 - val_accuracy: 0.5981
Epoch 6/10
391/391 [==============================] - 24s 61ms/step - loss: 0.9919 - accuracy: 0.6435 - val_l
oss: 1.0949 - val_accuracy: 0.6294
Epoch 7/10
391/391 [==============================] - 24s 61ms/step - loss: 0.9513 - accuracy: 0.6576 - val_l
oss: 1.3143 - val_accuracy: 0.5792
Epoch 8/10
391/391 [==============================] - 24s 61ms/step - loss: 0.9180 - accuracy: 0.6720 - val_l
oss: 1.4231 - val_accuracy: 0.5678
Epoch 9/10
391/391 [==============================] - 24s 61ms/step - loss: 0.8895 - accuracy: 0.6827 - val_l
oss: 1.0864 - val_accuracy: 0.6334
Epoch 10/10
391/391 [==============================] - 24s 61ms/step - loss: 0.8649 - accuracy: 0.6896 - val_l
oss: 1.7100 - val_accuracy: 0.5552
```

Out[13]:

```
<tensorflow.python.keras.callbacks.History at 0x7efc85359240>
```

In [14]:

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

```
313/313 [==============================] - 2s 7ms/step - loss: 1.7172 - accuracy: 0.5552
Test loss: 1.7172198295593262
Test accuracy: 0.5551999807357788
```

In [15]:

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

```
Saved model to disk
```

In [0]:

```
# free model variable
del model
```

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 [0]:

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

In [0]:

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

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

In [12]:

```
X_train.shape
```

Out[12]:

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

In [13]:

```
X_test.shape
```

Out[13]:

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

In [14]:

```
y_train.shape
```

Out[14]:

```
(50000, 10)
```

In [15]:

```
y_test.shape
```

Out[15]:

```
(10000, 10)
```

In [0]:

```
def normalize_pixels(train, test):
    '''
    Normalize data into range of 0 to 1
    '''
    train_norm = train.astype('float32')
    test_norm  = test.astype('float32')

    train_norm /= 255
    test_norm /= 255

    return (train_norm, test_norm)
```

In [0]:

```
X_train,X_test=normalize_pixels(X_train,X_test)
```

In [18]:

```
#https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/
sample_image=X_train[1]
sample_image.shape
```

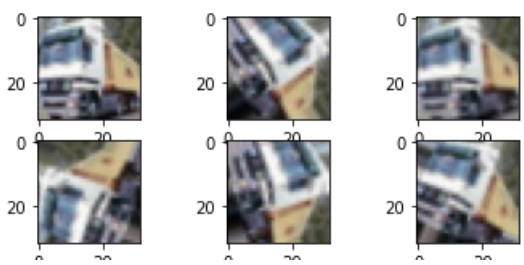
Out[18]:

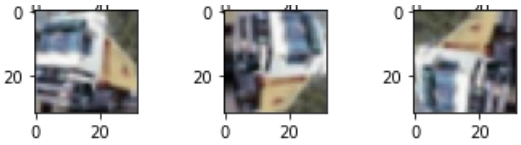
```
(32, 32, 3)
```

In [19]:

```
sample_images = expand_dims(sample_image, 0)
# create image data augmentation generator
datagen = ImageDataGenerator(rotation_range=90)
# prepare iterator
it = datagen.flow(sample_images, batch_size=1)
# generate sample images and plot
for i in range(9):
    pyplot.subplot(330 + 1 + i)
    # generate batch of images
    batch = it.next()
    image = batch[0];
    # plot raw pixel data
    pyplot.imshow(image)

# show the figure
pyplot.show()
```





1.1 Model with dense layer

In [0]:

```
#https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/
def model_summarize(history):
    '''
    Summarize model i.e. print train and test loss
    '''
    # plot loss
    pyplot.subplot(121)
    pyplot.title('Cross Entropy Loss')
    pyplot.plot(history.history['loss'], color='blue', label='train')
    pyplot.plot(history.history['val_loss'], color='orange', label='test')
    pyplot.show()
```

In [0]:

```
# import pickle
```

In [0]:

```
def model_harness(X_train, y_train, X_test, y_test, given_batch_size, given_step_size, given_epochs):
    '''
    define model using data augmentation technique and extend it to it's vertical limit
    '''
    # model = pickle.load('densenet.pkl')
    # create data generator
    datagen = ImageDataGenerator(width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True)
    # prepare iterator
    iterator_train = datagen.flow(X_train, y_train, batch_size=given_batch_size)
    # fit model
    steps = int(X_train.shape[0] / given_step_size)
    history = model.fit_generator(iterator_train, steps_per_epoch=steps, epochs=given_epochs, validation_data=(X_test, y_test), verbose=1)
    # evaluate model
    _, acc = model.evaluate(X_test, y_test, verbose=1)
    print('> %.3f' % (acc * 100.0))
    # file = open('/content/gdrive/My Drive/densenet.pkl', 'wb')
    # pickle.dumps(model)#, file
    # learning curves
    model_summarize(history)
```

In [0]:

```
def denseblock(input, num_filter = 64, dropout_rate = 0):
    '''
    Create dense block
    '''
    global compression
    temp = input
    for _ in range(l):
        BatchNorm = layers.BatchNormalization()(temp)
        relu = layers.Activation('relu')(BatchNorm)
        Conv2D_5_5 = layers.Conv2D(int(num_filter*compression),
(5,5),kernel_initializer="he_uniform" ,padding='same')(relu)
        if dropout_rate>0:
            Conv2D_5_5 = layers.Dropout(dropout_rate)(Conv2D_5_5)
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_5_5])

        temp = concat
```

```
    return temp

def transition(input, num_filter = 32, dropout_rate = 0):
    '''
    Create transition block
    '''
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (5,5), kernel_initializer="he_un
iform" ,padding='same')(relu)
    if dropout_rate>0:
         Conv2D_BottleNeck = layers.Dropout(dropout_rate)(Conv2D_BottleNeck)
    avg = layers.AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)

    return avg

def output_layer(input):
    '''
    define output layer
    '''
    global compression
    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    AvgPooling = layers.AveragePooling2D(pool_size=(2,2))(relu)
    flat = layers.Flatten()(AvgPooling)
    output = layers.Dense(num_classes, activation='softmax')(flat)

    return output
```

In [0]:

```
num_filter = 10
dropout_rate = 0
l = 12
input = layers.Input(shape=(img_height, img_width, channel))
First_Conv2D = layers.Conv2D(num_filter, (5,5), use_bias=False ,padding='same')(input)
BatchNorm = layers.BatchNormalization()(First_Conv2D)

First_Block = denseblock(BatchNorm,32, dropout_rate)
First_Transition = transition(First_Block, num_filter, dropout_rate)

Second_Block = denseblock(First_Transition, 16, dropout_rate)
Second_Transition = transition(Second_Block, num_filter, dropout_rate)

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

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

In [31]:

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

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 32, 32, 3)]  0
__________________________________________________________________________________________________
conv2d_52 (Conv2D)              (None, 32, 32, 10)   750         input_2[0][0]
__________________________________________________________________________________________________
batch_normalization_52 (BatchNo (None, 32, 32, 10)   40          conv2d_52[0][0]
__________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, 32, 32, 10)   40          batch_normalization_52[0][0]
__________________________________________________________________________________________________
activation_52 (Activation)      (None, 32, 32, 10)   0           batch_normalization_53[0][0]
__________________________________________________________________________________________________
conv2d_53 (Conv2D)              (None, 32, 32, 16)   4016        activation_52[0][0]
__________________________________________________________________________________________________
concatenate_48 (Concatenate)    (None, 32, 32, 26)   0           batch_normalization_52[0][0]
                                                                 conv2d_53[0][0]
```

batch_normalization_54 (BatchNo	(None, 32, 32, 26)	104	concatenate_48[0][0]
activation_53 (Activation)	(None, 32, 32, 26)	0	batch_normalization_54[0][0]
conv2d_54 (Conv2D)	(None, 32, 32, 16)	10416	activation_53[0][0]
concatenate_49 (Concatenate)	(None, 32, 32, 42)	0	concatenate_48[0][0] conv2d_54[0][0]
batch_normalization_55 (BatchNo	(None, 32, 32, 42)	168	concatenate_49[0][0]
activation_54 (Activation)	(None, 32, 32, 42)	0	batch_normalization_55[0][0]
conv2d_55 (Conv2D)	(None, 32, 32, 16)	16816	activation_54[0][0]
concatenate_50 (Concatenate)	(None, 32, 32, 58)	0	concatenate_49[0][0] conv2d_55[0][0]
batch_normalization_56 (BatchNo	(None, 32, 32, 58)	232	concatenate_50[0][0]
activation_55 (Activation)	(None, 32, 32, 58)	0	batch_normalization_56[0][0]
conv2d_56 (Conv2D)	(None, 32, 32, 16)	23216	activation_55[0][0]
concatenate_51 (Concatenate)	(None, 32, 32, 74)	0	concatenate_50[0][0] conv2d_56[0][0]
batch_normalization_57 (BatchNo	(None, 32, 32, 74)	296	concatenate_51[0][0]
activation_56 (Activation)	(None, 32, 32, 74)	0	batch_normalization_57[0][0]
conv2d_57 (Conv2D)	(None, 32, 32, 16)	29616	activation_56[0][0]
concatenate_52 (Concatenate)	(None, 32, 32, 90)	0	concatenate_51[0][0] conv2d_57[0][0]
batch_normalization_58 (BatchNo	(None, 32, 32, 90)	360	concatenate_52[0][0]
activation_57 (Activation)	(None, 32, 32, 90)	0	batch_normalization_58[0][0]
conv2d_58 (Conv2D)	(None, 32, 32, 16)	36016	activation_57[0][0]
concatenate_53 (Concatenate)	(None, 32, 32, 106)	0	concatenate_52[0][0] conv2d_58[0][0]
batch_normalization_59 (BatchNo	(None, 32, 32, 106)	424	concatenate_53[0][0]
activation_58 (Activation)	(None, 32, 32, 106)	0	batch_normalization_59[0][0]
conv2d_59 (Conv2D)	(None, 32, 32, 16)	42416	activation_58[0][0]
concatenate_54 (Concatenate)	(None, 32, 32, 122)	0	concatenate_53[0][0] conv2d_59[0][0]
batch_normalization_60 (BatchNo	(None, 32, 32, 122)	488	concatenate_54[0][0]
activation_59 (Activation)	(None, 32, 32, 122)	0	batch_normalization_60[0][0]
conv2d_60 (Conv2D)	(None, 32, 32, 16)	48816	activation_59[0][0]
concatenate_55 (Concatenate)	(None, 32, 32, 138)	0	concatenate_54[0][0] conv2d_60[0][0]
batch_normalization_61 (BatchNo	(None, 32, 32, 138)	552	concatenate_55[0][0]
activation_60 (Activation)	(None, 32, 32, 138)	0	batch_normalization_61[0][0]
conv2d_61 (Conv2D)	(None, 32, 32, 16)	55216	activation_60[0][0]
concatenate_56 (Concatenate)	(None, 32, 32, 154)	0	concatenate_55[0][0] conv2d_61[0][0]
batch_normalization_62 (BatchNo	(None, 32, 32, 154)	616	concatenate_56[0][0]
activation_61 (Activation)	(None, 32, 32, 154)	0	batch_normalization_62[0][0]

conv2d_62 (Conv2D)	(None, 32, 32, 16)	61616	activation_61[0][0]
concatenate_57 (Concatenate)	(None, 32, 32, 170)	0	concatenate_56[0][0] conv2d_62[0][0]
batch_normalization_63 (BatchNo	(None, 32, 32, 170)	680	concatenate_57[0][0]
activation_62 (Activation)	(None, 32, 32, 170)	0	batch_normalization_63[0][0]
conv2d_63 (Conv2D)	(None, 32, 32, 16)	68016	activation_62[0][0]
concatenate_58 (Concatenate)	(None, 32, 32, 186)	0	concatenate_57[0][0] conv2d_63[0][0]
batch_normalization_64 (BatchNo	(None, 32, 32, 186)	744	concatenate_58[0][0]
activation_63 (Activation)	(None, 32, 32, 186)	0	batch_normalization_64[0][0]
conv2d_64 (Conv2D)	(None, 32, 32, 16)	74416	activation_63[0][0]
concatenate_59 (Concatenate)	(None, 32, 32, 202)	0	concatenate_58[0][0] conv2d_64[0][0]
batch_normalization_65 (BatchNo	(None, 32, 32, 202)	808	concatenate_59[0][0]
activation_64 (Activation)	(None, 32, 32, 202)	0	batch_normalization_65[0][0]
conv2d_65 (Conv2D)	(None, 32, 32, 5)	25255	activation_64[0][0]
average_pooling2d_4 (AveragePoo	(None, 16, 16, 5)	0	conv2d_65[0][0]
batch_normalization_66 (BatchNo	(None, 16, 16, 5)	20	average_pooling2d_4[0][0]
activation_65 (Activation)	(None, 16, 16, 5)	0	batch_normalization_66[0][0]
conv2d_66 (Conv2D)	(None, 16, 16, 8)	1008	activation_65[0][0]
concatenate_60 (Concatenate)	(None, 16, 16, 13)	0	average_pooling2d_4[0][0] conv2d_66[0][0]
batch_normalization_67 (BatchNo	(None, 16, 16, 13)	52	concatenate_60[0][0]
activation_66 (Activation)	(None, 16, 16, 13)	0	batch_normalization_67[0][0]
conv2d_67 (Conv2D)	(None, 16, 16, 8)	2608	activation_66[0][0]
concatenate_61 (Concatenate)	(None, 16, 16, 21)	0	concatenate_60[0][0] conv2d_67[0][0]
batch_normalization_68 (BatchNo	(None, 16, 16, 21)	84	concatenate_61[0][0]
activation_67 (Activation)	(None, 16, 16, 21)	0	batch_normalization_68[0][0]
conv2d_68 (Conv2D)	(None, 16, 16, 8)	4208	activation_67[0][0]
concatenate_62 (Concatenate)	(None, 16, 16, 29)	0	concatenate_61[0][0] conv2d_68[0][0]
batch_normalization_69 (BatchNo	(None, 16, 16, 29)	116	concatenate_62[0][0]
activation_68 (Activation)	(None, 16, 16, 29)	0	batch_normalization_69[0][0]
conv2d_69 (Conv2D)	(None, 16, 16, 8)	5808	activation_68[0][0]
concatenate_63 (Concatenate)	(None, 16, 16, 37)	0	concatenate_62[0][0] conv2d_69[0][0]
batch_normalization_70 (BatchNo	(None, 16, 16, 37)	148	concatenate_63[0][0]
activation_69 (Activation)	(None, 16, 16, 37)	0	batch_normalization_70[0][0]
conv2d_70 (Conv2D)	(None, 16, 16, 8)	7408	activation_69[0][0]
concatenate_64 (Concatenate)	(None, 16, 16, 45)	0	concatenate_63[0][0] conv2d_70[0][0]
batch_normalization_71 (BatchNo	(None, 16, 16, 45)	180	concatenate_64[0][0]

activation_70 (Activation)	(None, 16, 16, 45)	0	batch_normalization_71[0][0]
conv2d_71 (Conv2D)	(None, 16, 16, 8)	9008	activation_70[0][0]
concatenate_65 (Concatenate)	(None, 16, 16, 53)	0	concatenate_64[0][0] conv2d_71[0][0]
batch_normalization_72 (BatchNo	(None, 16, 16, 53)	212	concatenate_65[0][0]
activation_71 (Activation)	(None, 16, 16, 53)	0	batch_normalization_72[0][0]
conv2d_72 (Conv2D)	(None, 16, 16, 8)	10608	activation_71[0][0]
concatenate_66 (Concatenate)	(None, 16, 16, 61)	0	concatenate_65[0][0] conv2d_72[0][0]
batch_normalization_73 (BatchNo	(None, 16, 16, 61)	244	concatenate_66[0][0]
activation_72 (Activation)	(None, 16, 16, 61)	0	batch_normalization_73[0][0]
conv2d_73 (Conv2D)	(None, 16, 16, 8)	12208	activation_72[0][0]
concatenate_67 (Concatenate)	(None, 16, 16, 69)	0	concatenate_66[0][0] conv2d_73[0][0]
batch_normalization_74 (BatchNo	(None, 16, 16, 69)	276	concatenate_67[0][0]
activation_73 (Activation)	(None, 16, 16, 69)	0	batch_normalization_74[0][0]
conv2d_74 (Conv2D)	(None, 16, 16, 8)	13808	activation_73[0][0]
concatenate_68 (Concatenate)	(None, 16, 16, 77)	0	concatenate_67[0][0] conv2d_74[0][0]
batch_normalization_75 (BatchNo	(None, 16, 16, 77)	308	concatenate_68[0][0]
activation_74 (Activation)	(None, 16, 16, 77)	0	batch_normalization_75[0][0]
conv2d_75 (Conv2D)	(None, 16, 16, 8)	15408	activation_74[0][0]
concatenate_69 (Concatenate)	(None, 16, 16, 85)	0	concatenate_68[0][0] conv2d_75[0][0]
batch_normalization_76 (BatchNo	(None, 16, 16, 85)	340	concatenate_69[0][0]
activation_75 (Activation)	(None, 16, 16, 85)	0	batch_normalization_76[0][0]
conv2d_76 (Conv2D)	(None, 16, 16, 8)	17008	activation_75[0][0]
concatenate_70 (Concatenate)	(None, 16, 16, 93)	0	concatenate_69[0][0] conv2d_76[0][0]
batch_normalization_77 (BatchNo	(None, 16, 16, 93)	372	concatenate_70[0][0]
activation_76 (Activation)	(None, 16, 16, 93)	0	batch_normalization_77[0][0]
conv2d_77 (Conv2D)	(None, 16, 16, 8)	18608	activation_76[0][0]
concatenate_71 (Concatenate)	(None, 16, 16, 101)	0	concatenate_70[0][0] conv2d_77[0][0]
batch_normalization_78 (BatchNo	(None, 16, 16, 101)	404	concatenate_71[0][0]
activation_77 (Activation)	(None, 16, 16, 101)	0	batch_normalization_78[0][0]
conv2d_78 (Conv2D)	(None, 16, 16, 5)	12630	activation_77[0][0]
average_pooling2d_5 (AveragePoo	(None, 8, 8, 5)	0	conv2d_78[0][0]
batch_normalization_79 (BatchNo	(None, 8, 8, 5)	20	average_pooling2d_5[0][0]
activation_78 (Activation)	(None, 8, 8, 5)	0	batch_normalization_79[0][0]
conv2d_79 (Conv2D)	(None, 8, 8, 5)	630	activation_78[0][0]
concatenate_72 (Concatenate)	(None, 8, 8, 10)	0	average_pooling2d_5[0][0]

			conv2d_79[0][0]
batch_normalization_80 (BatchNo	(None, 8, 8, 10)	40	concatenate_72[0][0]
activation_79 (Activation)	(None, 8, 8, 10)	0	batch_normalization_80[0][0]
conv2d_80 (Conv2D)	(None, 8, 8, 5)	1255	activation_79[0][0]
concatenate_73 (Concatenate)	(None, 8, 8, 15)	0	concatenate_72[0][0] conv2d_80[0][0]
batch_normalization_81 (BatchNo	(None, 8, 8, 15)	60	concatenate_73[0][0]
activation_80 (Activation)	(None, 8, 8, 15)	0	batch_normalization_81[0][0]
conv2d_81 (Conv2D)	(None, 8, 8, 5)	1880	activation_80[0][0]
concatenate_74 (Concatenate)	(None, 8, 8, 20)	0	concatenate_73[0][0] conv2d_81[0][0]
batch_normalization_82 (BatchNo	(None, 8, 8, 20)	80	concatenate_74[0][0]
activation_81 (Activation)	(None, 8, 8, 20)	0	batch_normalization_82[0][0]
conv2d_82 (Conv2D)	(None, 8, 8, 5)	2505	activation_81[0][0]
concatenate_75 (Concatenate)	(None, 8, 8, 25)	0	concatenate_74[0][0] conv2d_82[0][0]
batch_normalization_83 (BatchNo	(None, 8, 8, 25)	100	concatenate_75[0][0]
activation_82 (Activation)	(None, 8, 8, 25)	0	batch_normalization_83[0][0]
conv2d_83 (Conv2D)	(None, 8, 8, 5)	3130	activation_82[0][0]
concatenate_76 (Concatenate)	(None, 8, 8, 30)	0	concatenate_75[0][0] conv2d_83[0][0]
batch_normalization_84 (BatchNo	(None, 8, 8, 30)	120	concatenate_76[0][0]
activation_83 (Activation)	(None, 8, 8, 30)	0	batch_normalization_84[0][0]
conv2d_84 (Conv2D)	(None, 8, 8, 5)	3755	activation_83[0][0]
concatenate_77 (Concatenate)	(None, 8, 8, 35)	0	concatenate_76[0][0] conv2d_84[0][0]
batch_normalization_85 (BatchNo	(None, 8, 8, 35)	140	concatenate_77[0][0]
activation_84 (Activation)	(None, 8, 8, 35)	0	batch_normalization_85[0][0]
conv2d_85 (Conv2D)	(None, 8, 8, 5)	4380	activation_84[0][0]
concatenate_78 (Concatenate)	(None, 8, 8, 40)	0	concatenate_77[0][0] conv2d_85[0][0]
batch_normalization_86 (BatchNo	(None, 8, 8, 40)	160	concatenate_78[0][0]
activation_85 (Activation)	(None, 8, 8, 40)	0	batch_normalization_86[0][0]
conv2d_86 (Conv2D)	(None, 8, 8, 5)	5005	activation_85[0][0]
concatenate_79 (Concatenate)	(None, 8, 8, 45)	0	concatenate_78[0][0] conv2d_86[0][0]
batch_normalization_87 (BatchNo	(None, 8, 8, 45)	180	concatenate_79[0][0]
activation_86 (Activation)	(None, 8, 8, 45)	0	batch_normalization_87[0][0]
conv2d_87 (Conv2D)	(None, 8, 8, 5)	5630	activation_86[0][0]
concatenate_80 (Concatenate)	(None, 8, 8, 50)	0	concatenate_79[0][0] conv2d_87[0][0]
batch_normalization_88 (BatchNo	(None, 8, 8, 50)	200	concatenate_80[0][0]
activation_87 (Activation)	(None, 8, 8, 50)	0	batch_normalization_88[0][0]

conv2d_88 (Conv2D)	(None, 8, 8, 5)	6255	activation_87[0][0]
concatenate_81 (Concatenate)	(None, 8, 8, 55)	0	concatenate_80[0][0] conv2d_88[0][0]
batch_normalization_89 (BatchNo	(None, 8, 8, 55)	220	concatenate_81[0][0]
activation_88 (Activation)	(None, 8, 8, 55)	0	batch_normalization_89[0][0]
conv2d_89 (Conv2D)	(None, 8, 8, 5)	6880	activation_88[0][0]
concatenate_82 (Concatenate)	(None, 8, 8, 60)	0	concatenate_81[0][0] conv2d_89[0][0]
batch_normalization_90 (BatchNo	(None, 8, 8, 60)	240	concatenate_82[0][0]
activation_89 (Activation)	(None, 8, 8, 60)	0	batch_normalization_90[0][0]
conv2d_90 (Conv2D)	(None, 8, 8, 5)	7505	activation_89[0][0]
concatenate_83 (Concatenate)	(None, 8, 8, 65)	0	concatenate_82[0][0] conv2d_90[0][0]
batch_normalization_91 (BatchNo	(None, 8, 8, 65)	260	concatenate_83[0][0]
activation_90 (Activation)	(None, 8, 8, 65)	0	batch_normalization_91[0][0]
conv2d_91 (Conv2D)	(None, 8, 8, 5)	8130	activation_90[0][0]
average_pooling2d_6 (AveragePoo	(None, 4, 4, 5)	0	conv2d_91[0][0]
batch_normalization_92 (BatchNo	(None, 4, 4, 5)	20	average_pooling2d_6[0][0]
activation_91 (Activation)	(None, 4, 4, 5)	0	batch_normalization_92[0][0]
conv2d_92 (Conv2D)	(None, 4, 4, 5)	630	activation_91[0][0]
concatenate_84 (Concatenate)	(None, 4, 4, 10)	0	average_pooling2d_6[0][0] conv2d_92[0][0]
batch_normalization_93 (BatchNo	(None, 4, 4, 10)	40	concatenate_84[0][0]
activation_92 (Activation)	(None, 4, 4, 10)	0	batch_normalization_93[0][0]
conv2d_93 (Conv2D)	(None, 4, 4, 5)	1255	activation_92[0][0]
concatenate_85 (Concatenate)	(None, 4, 4, 15)	0	concatenate_84[0][0] conv2d_93[0][0]
batch_normalization_94 (BatchNo	(None, 4, 4, 15)	60	concatenate_85[0][0]
activation_93 (Activation)	(None, 4, 4, 15)	0	batch_normalization_94[0][0]
conv2d_94 (Conv2D)	(None, 4, 4, 5)	1880	activation_93[0][0]
concatenate_86 (Concatenate)	(None, 4, 4, 20)	0	concatenate_85[0][0] conv2d_94[0][0]
batch_normalization_95 (BatchNo	(None, 4, 4, 20)	80	concatenate_86[0][0]
activation_94 (Activation)	(None, 4, 4, 20)	0	batch_normalization_95[0][0]
conv2d_95 (Conv2D)	(None, 4, 4, 5)	2505	activation_94[0][0]
concatenate_87 (Concatenate)	(None, 4, 4, 25)	0	concatenate_86[0][0] conv2d_95[0][0]
batch_normalization_96 (BatchNo	(None, 4, 4, 25)	100	concatenate_87[0][0]
activation_95 (Activation)	(None, 4, 4, 25)	0	batch_normalization_96[0][0]
conv2d_96 (Conv2D)	(None, 4, 4, 5)	3130	activation_95[0][0]
concatenate_88 (Concatenate)	(None, 4, 4, 30)	0	concatenate_87[0][0] conv2d_96[0][0]

batch_normalization_97 (BatchNo	(None, 4, 4, 30)	120	concatenate_88[0][0]
activation_96 (Activation)	(None, 4, 4, 30)	0	batch_normalization_97[0][0]
conv2d_97 (Conv2D)	(None, 4, 4, 5)	3755	activation_96[0][0]
concatenate_89 (Concatenate)	(None, 4, 4, 35)	0	concatenate_88[0][0] conv2d_97[0][0]
batch_normalization_98 (BatchNo	(None, 4, 4, 35)	140	concatenate_89[0][0]
activation_97 (Activation)	(None, 4, 4, 35)	0	batch_normalization_98[0][0]
conv2d_98 (Conv2D)	(None, 4, 4, 5)	4380	activation_97[0][0]
concatenate_90 (Concatenate)	(None, 4, 4, 40)	0	concatenate_89[0][0] conv2d_98[0][0]
batch_normalization_99 (BatchNo	(None, 4, 4, 40)	160	concatenate_90[0][0]
activation_98 (Activation)	(None, 4, 4, 40)	0	batch_normalization_99[0][0]
conv2d_99 (Conv2D)	(None, 4, 4, 5)	5005	activation_98[0][0]
concatenate_91 (Concatenate)	(None, 4, 4, 45)	0	concatenate_90[0][0] conv2d_99[0][0]
batch_normalization_100 (BatchN	(None, 4, 4, 45)	180	concatenate_91[0][0]
activation_99 (Activation)	(None, 4, 4, 45)	0	batch_normalization_100[0][0]
conv2d_100 (Conv2D)	(None, 4, 4, 5)	5630	activation_99[0][0]
concatenate_92 (Concatenate)	(None, 4, 4, 50)	0	concatenate_91[0][0] conv2d_100[0][0]
batch_normalization_101 (BatchN	(None, 4, 4, 50)	200	concatenate_92[0][0]
activation_100 (Activation)	(None, 4, 4, 50)	0	batch_normalization_101[0][0]
conv2d_101 (Conv2D)	(None, 4, 4, 5)	6255	activation_100[0][0]
concatenate_93 (Concatenate)	(None, 4, 4, 55)	0	concatenate_92[0][0] conv2d_101[0][0]
batch_normalization_102 (BatchN	(None, 4, 4, 55)	220	concatenate_93[0][0]
activation_101 (Activation)	(None, 4, 4, 55)	0	batch_normalization_102[0][0]
conv2d_102 (Conv2D)	(None, 4, 4, 5)	6880	activation_101[0][0]
concatenate_94 (Concatenate)	(None, 4, 4, 60)	0	concatenate_93[0][0] conv2d_102[0][0]
batch_normalization_103 (BatchN	(None, 4, 4, 60)	240	concatenate_94[0][0]
activation_102 (Activation)	(None, 4, 4, 60)	0	batch_normalization_103[0][0]
conv2d_103 (Conv2D)	(None, 4, 4, 5)	7505	activation_102[0][0]
concatenate_95 (Concatenate)	(None, 4, 4, 65)	0	concatenate_94[0][0] conv2d_103[0][0]
batch_normalization_104 (BatchN	(None, 4, 4, 65)	260	concatenate_95[0][0]
activation_103 (Activation)	(None, 4, 4, 65)	0	batch_normalization_104[0][0]
average_pooling2d_7 (AveragePoo	(None, 2, 2, 65)	0	activation_103[0][0]
flatten_1 (Flatten)	(None, 260)	0	average_pooling2d_7[0][0]
dense_1 (Dense)	(None, 10)	2610	flatten_1[0][0]

Total params: 747,231
Trainable params: 741,257
Non-trainable params: 5,974

In [0]:

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

In [0]:

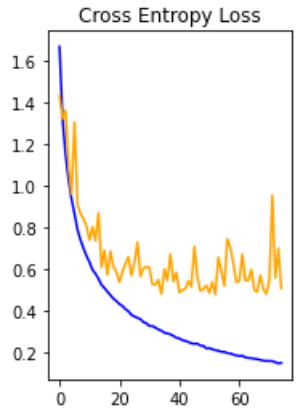
```
# entry point, run the test harness
model_harness(X_train, y_train, X_test, y_test, 64, 64, 75)
```

```
WARNING:tensorflow:From <ipython-input-28-46f7e69e7015>:11: Model.fit_generator (from
tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
Epoch 1/75
781/781 [==============================] - 83s 107ms/step - loss: 1.6684 - accuracy: 0.3792 - val_
loss: 1.4304 - val_accuracy: 0.4914
Epoch 2/75
781/781 [==============================] - 82s 105ms/step - loss: 1.3260 - accuracy: 0.5175 - val_
loss: 1.3210 - val_accuracy: 0.5277
Epoch 3/75
781/781 [==============================] - 82s 105ms/step - loss: 1.1522 - accuracy: 0.5854 - val_
loss: 1.3607 - val_accuracy: 0.5235
Epoch 4/75
781/781 [==============================] - 84s 107ms/step - loss: 1.0313 - accuracy: 0.6312 - val_
loss: 1.0919 - val_accuracy: 0.6118
Epoch 5/75
781/781 [==============================] - 84s 107ms/step - loss: 0.9370 - accuracy: 0.6667 - val_
loss: 0.9598 - val_accuracy: 0.6625
Epoch 6/75
781/781 [==============================] - 84s 107ms/step - loss: 0.8641 - accuracy: 0.6973 - val_
loss: 1.3034 - val_accuracy: 0.5862
Epoch 7/75
781/781 [==============================] - 82s 105ms/step - loss: 0.7877 - accuracy: 0.7217 - val_
loss: 0.9087 - val_accuracy: 0.6839
Epoch 8/75
781/781 [==============================] - 81s 104ms/step - loss: 0.7386 - accuracy: 0.7411 - val_
loss: 0.8664 - val_accuracy: 0.6985
Epoch 9/75
781/781 [==============================] - 82s 104ms/step - loss: 0.6978 - accuracy: 0.7547 - val_
loss: 0.8392 - val_accuracy: 0.7202
Epoch 10/75
781/781 [==============================] - 81s 104ms/step - loss: 0.6620 - accuracy: 0.7706 - val_
loss: 0.8044 - val_accuracy: 0.7289
Epoch 11/75
781/781 [==============================] - 81s 104ms/step - loss: 0.6322 - accuracy: 0.7796 - val_
loss: 0.7374 - val_accuracy: 0.7495
Epoch 12/75
781/781 [==============================] - 81s 104ms/step - loss: 0.5946 - accuracy: 0.7935 - val_
loss: 0.8011 - val_accuracy: 0.7338
Epoch 13/75
781/781 [==============================] - 81s 104ms/step - loss: 0.5766 - accuracy: 0.7996 - val_
loss: 0.7339 - val_accuracy: 0.7555
Epoch 14/75
781/781 [==============================] - 81s 104ms/step - loss: 0.5516 - accuracy: 0.8103 - val_
loss: 0.8674 - val_accuracy: 0.7169
Epoch 15/75
781/781 [==============================] - 81s 103ms/step - loss: 0.5223 - accuracy: 0.8179 - val_
loss: 0.6069 - val_accuracy: 0.7943
Epoch 16/75
781/781 [==============================] - 81s 103ms/step - loss: 0.5071 - accuracy: 0.8237 - val_
loss: 0.6881 - val_accuracy: 0.7710
Epoch 17/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4891 - accuracy: 0.8303 - val_
loss: 0.5705 - val_accuracy: 0.8093
Epoch 18/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4725 - accuracy: 0.8363 - val_
loss: 0.6845 - val_accuracy: 0.7786
Epoch 19/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4561 - accuracy: 0.8412 - val_
loss: 0.6089 - val_accuracy: 0.7979
```

```
Epoch 20/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4437 - accuracy: 0.8458 - val_
loss: 0.5828 - val_accuracy: 0.8022
Epoch 21/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4296 - accuracy: 0.8504 - val_
loss: 0.5345 - val_accuracy: 0.8205
Epoch 22/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4202 - accuracy: 0.8547 - val_
loss: 0.5862 - val_accuracy: 0.8037
Epoch 23/75
781/781 [==============================] - 81s 104ms/step - loss: 0.4054 - accuracy: 0.8597 - val_
loss: 0.6250 - val_accuracy: 0.8024
Epoch 24/75
781/781 [==============================] - 81s 103ms/step - loss: 0.3950 - accuracy: 0.8628 - val_
loss: 0.6574 - val_accuracy: 0.7918
Epoch 25/75
781/781 [==============================] - 81s 103ms/step - loss: 0.3778 - accuracy: 0.8674 - val_
loss: 0.5681 - val_accuracy: 0.8149
Epoch 26/75
781/781 [==============================] - 82s 105ms/step - loss: 0.3702 - accuracy: 0.8708 - val_
loss: 0.6275 - val_accuracy: 0.8054
Epoch 27/75
781/781 [==============================] - 83s 106ms/step - loss: 0.3632 - accuracy: 0.8725 - val_
loss: 0.7269 - val_accuracy: 0.7794
Epoch 28/75
781/781 [==============================] - 83s 106ms/step - loss: 0.3571 - accuracy: 0.8765 - val_
loss: 0.5631 - val_accuracy: 0.8246
Epoch 29/75
781/781 [==============================] - 82s 106ms/step - loss: 0.3428 - accuracy: 0.8809 - val_
loss: 0.6024 - val_accuracy: 0.8129
Epoch 30/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3372 - accuracy: 0.8832 - val_
loss: 0.6088 - val_accuracy: 0.8147
Epoch 31/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3238 - accuracy: 0.8881 - val_
loss: 0.6050 - val_accuracy: 0.8153
Epoch 32/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3243 - accuracy: 0.8857 - val_
loss: 0.5257 - val_accuracy: 0.8331
Epoch 33/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3156 - accuracy: 0.8901 - val_
loss: 0.5227 - val_accuracy: 0.8336
Epoch 34/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3077 - accuracy: 0.8927 - val_
loss: 0.5468 - val_accuracy: 0.8283
Epoch 35/75
781/781 [==============================] - 81s 104ms/step - loss: 0.3012 - accuracy: 0.8941 - val_
loss: 0.4772 - val_accuracy: 0.8463
Epoch 36/75
781/781 [==============================] - 81s 104ms/step - loss: 0.2919 - accuracy: 0.8990 - val_
loss: 0.5982 - val_accuracy: 0.8135
Epoch 37/75
781/781 [==============================] - 81s 104ms/step - loss: 0.2878 - accuracy: 0.9013 - val_
loss: 0.5440 - val_accuracy: 0.8317
Epoch 38/75
781/781 [==============================] - 81s 104ms/step - loss: 0.2846 - accuracy: 0.9001 - val_
loss: 0.6696 - val_accuracy: 0.7951
Epoch 39/75
781/781 [==============================] - 80s 103ms/step - loss: 0.2762 - accuracy: 0.9034 - val_
loss: 0.5379 - val_accuracy: 0.8301
Epoch 40/75
781/781 [==============================] - 81s 103ms/step - loss: 0.2674 - accuracy: 0.9072 - val_
loss: 0.5816 - val_accuracy: 0.8275
Epoch 41/75
781/781 [==============================] - 81s 103ms/step - loss: 0.2639 - accuracy: 0.9082 - val_
loss: 0.4855 - val_accuracy: 0.8527
Epoch 42/75
781/781 [==============================] - 81s 103ms/step - loss: 0.2562 - accuracy: 0.9113 - val_
loss: 0.4963 - val_accuracy: 0.8397
Epoch 43/75
781/781 [==============================] - 81s 103ms/step - loss: 0.2523 - accuracy: 0.9114 - val_
loss: 0.5028 - val_accuracy: 0.8428
Epoch 44/75
781/781 [==============================] - 81s 103ms/step - loss: 0.2468 - accuracy: 0.9143 - val_
loss: 0.5409 - val_accuracy: 0.8351
Epoch 45/75
781/781 [==============================] - 80s 103ms/step - loss: 0.2411 - accuracy: 0.9161 - val_
```

```
loss: 0.5131 - val_accuracy: 0.8446
Epoch 46/75
781/781 [==============================] - 81s 104ms/step - loss: 0.2379 - accuracy: 0.9169 - val_
loss: 0.7075 - val_accuracy: 0.8006
Epoch 47/75
781/781 [==============================] - 81s 104ms/step - loss: 0.2392 - accuracy: 0.9171 - val_
loss: 0.5598 - val_accuracy: 0.8328
Epoch 48/75
781/781 [==============================] - 82s 104ms/step - loss: 0.2299 - accuracy: 0.9192 - val_
loss: 0.4924 - val_accuracy: 0.8491
Epoch 49/75
781/781 [==============================] - 83s 107ms/step - loss: 0.2287 - accuracy: 0.9211 - val_
loss: 0.5023 - val_accuracy: 0.8455
Epoch 50/75
781/781 [==============================] - 83s 107ms/step - loss: 0.2158 - accuracy: 0.9245 - val_
loss: 0.5197 - val_accuracy: 0.8456
Epoch 51/75
781/781 [==============================] - 83s 106ms/step - loss: 0.2181 - accuracy: 0.9228 - val_
loss: 0.4835 - val_accuracy: 0.8552
Epoch 52/75
781/781 [==============================] - 82s 105ms/step - loss: 0.2116 - accuracy: 0.9252 - val_
loss: 0.5345 - val_accuracy: 0.8464
Epoch 53/75
781/781 [==============================] - 82s 104ms/step - loss: 0.2081 - accuracy: 0.9268 - val_
loss: 0.4759 - val_accuracy: 0.8608
Epoch 54/75
781/781 [==============================] - 82s 104ms/step - loss: 0.2055 - accuracy: 0.9272 - val_
loss: 0.6548 - val_accuracy: 0.8200
Epoch 55/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1993 - accuracy: 0.9306 - val_
loss: 0.5874 - val_accuracy: 0.8344
Epoch 56/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1996 - accuracy: 0.9289 - val_
loss: 0.5164 - val_accuracy: 0.8448
Epoch 57/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1934 - accuracy: 0.9326 - val_
loss: 0.7423 - val_accuracy: 0.8139
Epoch 58/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1925 - accuracy: 0.9340 - val_
loss: 0.7007 - val_accuracy: 0.8165
Epoch 59/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1862 - accuracy: 0.9347 - val_
loss: 0.6281 - val_accuracy: 0.8325
Epoch 60/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1826 - accuracy: 0.9346 - val_
loss: 0.5355 - val_accuracy: 0.8479
Epoch 61/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1794 - accuracy: 0.9368 - val_
loss: 0.5398 - val_accuracy: 0.8470
Epoch 62/75
781/781 [==============================] - 80s 103ms/step - loss: 0.1814 - accuracy: 0.9369 - val_
loss: 0.6696 - val_accuracy: 0.8233
Epoch 63/75
781/781 [==============================] - 82s 104ms/step - loss: 0.1728 - accuracy: 0.9399 - val_
loss: 0.5428 - val_accuracy: 0.8474
Epoch 64/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1708 - accuracy: 0.9401 - val_
loss: 0.5434 - val_accuracy: 0.8498
Epoch 65/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1689 - accuracy: 0.9401 - val_
loss: 0.5965 - val_accuracy: 0.8339
Epoch 66/75
781/781 [==============================] - 84s 107ms/step - loss: 0.1667 - accuracy: 0.9417 - val_
loss: 0.4968 - val_accuracy: 0.8565
Epoch 67/75
781/781 [==============================] - 84s 108ms/step - loss: 0.1643 - accuracy: 0.9426 - val_
loss: 0.4887 - val_accuracy: 0.8629
Epoch 68/75
781/781 [==============================] - 84s 108ms/step - loss: 0.1609 - accuracy: 0.9443 - val_
loss: 0.5673 - val_accuracy: 0.8451
Epoch 69/75
781/781 [==============================] - 85s 108ms/step - loss: 0.1570 - accuracy: 0.9446 - val_
loss: 0.5022 - val_accuracy: 0.8585
Epoch 70/75
781/781 [==============================] - 81s 104ms/step - loss: 0.1561 - accuracy: 0.9449 - val_
loss: 0.4776 - val_accuracy: 0.8687
Epoch 71/75
```

```
781/781 [==============================] - 85s 109ms/step - loss: 0.1557 - accuracy: 0.9449 - val_
loss: 0.5449 - val_accuracy: 0.8586
Epoch 72/75
781/781 [==============================] - 85s 109ms/step - loss: 0.1554 - accuracy: 0.9436 - val_
loss: 0.9523 - val_accuracy: 0.7867
Epoch 73/75
781/781 [==============================] - 84s 108ms/step - loss: 0.1497 - accuracy: 0.9466 - val_
loss: 0.5549 - val_accuracy: 0.8565
Epoch 74/75
781/781 [==============================] - 85s 109ms/step - loss: 0.1442 - accuracy: 0.9486 - val_
loss: 0.6982 - val_accuracy: 0.8181
Epoch 75/75
781/781 [==============================] - 82s 105ms/step - loss: 0.1466 - accuracy: 0.9478 - val_
loss: 0.5055 - val_accuracy: 0.8603
313/313 [==============================] - 5s 17ms/step - loss: 0.5055 - accuracy: 0.8603
> 86.030
```



In [0]:

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

```
Saved model to disk
```

In [0]:

```
# free model variable
del model
```

1.2 Model without Dense layer and implemented all tweaks mentioned in the assignment task

In [0]:

```
def denseblock(input, num_filter = 12, dropout_rate = 0.2):
    '''
    Create Dense Block
    '''
    global compression
    temp = input
    for _ in range(l):

        BatchNorm = layers.BatchNormalization()(temp)
        relu = layers.Activation('relu')(BatchNorm)

        Conv2D_5_5 = layers.Conv2D(int(num_filter*compression), (5,5), use_bias=False ,padding='sam
e')(relu)

        if dropout_rate>0:
            Conv2D_5_5 = layers.Dropout(dropout_rate)(Conv2D_5_5)
```

```
        concat = layers.Concatenate(axis=-1)([temp,Conv2D_5_5])

        temp = concat

    return temp

def transition(input, num_filter = 12, dropout_rate = 0.2):
    '''
    Create transition block
    '''
    global compression

    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)

    Conv2D_BottleNeck = layers.Conv2D(int(num_filter*compression), (5,5), use_bias=False ,padding='
same')(relu)

    if dropout_rate>0:
         Conv2D_BottleNeck = layers.Dropout(dropout_rate)(Conv2D_BottleNeck)

    avg = layers.AveragePooling2D(pool_size=(2,2))(Conv2D_BottleNeck)

    return avg

def output_layer(input):
    '''
    Define output layer
    '''
    global compression

    BatchNorm = layers.BatchNormalization()(input)
    relu = layers.Activation('relu')(BatchNorm)
    AvgPooling = layers. MaxPooling2D(pool_size=(2,2))(relu)

    output = layers.Conv2D(filters=10,kernel_size=(2,2),activation='softmax')(AvgPooling)

    flat = layers.Flatten()(output)

    return flat
```

In [0]:

```
num_filter = 12
dropout_rate = 0
l = 12
input = layers.Input(shape=(img_height, img_width, channel,))
First_Conv2D = layers.Conv2D(32, (3,3), use_bias=False ,padding='same')(input)

First_Block = denseblock(First_Conv2D,10, dropout_rate)
First_Transition = transition(First_Block, 64, dropout_rate)

Second_Block = denseblock(First_Transition, 10, dropout_rate)
Second_Transition = transition(Second_Block, 32, dropout_rate)

Third_Block = denseblock(Second_Transition, num_filter, dropout_rate)
Third_Transition = transition(Third_Block, 32, dropout_rate)

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

In [27]:

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

```
Model: "model"
_________________________________________________________________________________________________
Layer (type)                 Output Shape              Param #     Connected to
=================================================================================================
input_2 (InputLayer)         [(None, 32, 32, 3)]       0
_________________________________________________________________________________________________
conv2d_52 (Conv2D)           (None, 32, 32, 32)        864         input_2[0][0]
_________________________________________________________________________________________________
batch_normalization_53 (BatchNo (None, 32, 32, 32)     128         conv2d_52[0][0]
```

activation_52 (Activation)	(None, 32, 32, 32)	0	batch_normalization_53[0][0]
conv2d_53 (Conv2D)	(None, 32, 32, 5)	4000	activation_52[0][0]
concatenate_48 (Concatenate)	(None, 32, 32, 37)	0	conv2d_52[0][0] conv2d_53[0][0]
batch_normalization_54 (BatchNo	(None, 32, 32, 37)	148	concatenate_48[0][0]
activation_53 (Activation)	(None, 32, 32, 37)	0	batch_normalization_54[0][0]
conv2d_54 (Conv2D)	(None, 32, 32, 5)	4625	activation_53[0][0]
concatenate_49 (Concatenate)	(None, 32, 32, 42)	0	concatenate_48[0][0] conv2d_54[0][0]
batch_normalization_55 (BatchNo	(None, 32, 32, 42)	168	concatenate_49[0][0]
activation_54 (Activation)	(None, 32, 32, 42)	0	batch_normalization_55[0][0]
conv2d_55 (Conv2D)	(None, 32, 32, 5)	5250	activation_54[0][0]
concatenate_50 (Concatenate)	(None, 32, 32, 47)	0	concatenate_49[0][0] conv2d_55[0][0]
batch_normalization_56 (BatchNo	(None, 32, 32, 47)	188	concatenate_50[0][0]
activation_55 (Activation)	(None, 32, 32, 47)	0	batch_normalization_56[0][0]
conv2d_56 (Conv2D)	(None, 32, 32, 5)	5875	activation_55[0][0]
concatenate_51 (Concatenate)	(None, 32, 32, 52)	0	concatenate_50[0][0] conv2d_56[0][0]
batch_normalization_57 (BatchNo	(None, 32, 32, 52)	208	concatenate_51[0][0]
activation_56 (Activation)	(None, 32, 32, 52)	0	batch_normalization_57[0][0]
conv2d_57 (Conv2D)	(None, 32, 32, 5)	6500	activation_56[0][0]
concatenate_52 (Concatenate)	(None, 32, 32, 57)	0	concatenate_51[0][0] conv2d_57[0][0]
batch_normalization_58 (BatchNo	(None, 32, 32, 57)	228	concatenate_52[0][0]
activation_57 (Activation)	(None, 32, 32, 57)	0	batch_normalization_58[0][0]
conv2d_58 (Conv2D)	(None, 32, 32, 5)	7125	activation_57[0][0]
concatenate_53 (Concatenate)	(None, 32, 32, 62)	0	concatenate_52[0][0] conv2d_58[0][0]
batch_normalization_59 (BatchNo	(None, 32, 32, 62)	248	concatenate_53[0][0]
activation_58 (Activation)	(None, 32, 32, 62)	0	batch_normalization_59[0][0]
conv2d_59 (Conv2D)	(None, 32, 32, 5)	7750	activation_58[0][0]
concatenate_54 (Concatenate)	(None, 32, 32, 67)	0	concatenate_53[0][0] conv2d_59[0][0]
batch_normalization_60 (BatchNo	(None, 32, 32, 67)	268	concatenate_54[0][0]
activation_59 (Activation)	(None, 32, 32, 67)	0	batch_normalization_60[0][0]
conv2d_60 (Conv2D)	(None, 32, 32, 5)	8375	activation_59[0][0]
concatenate_55 (Concatenate)	(None, 32, 32, 72)	0	concatenate_54[0][0] conv2d_60[0][0]
batch_normalization_61 (BatchNo	(None, 32, 32, 72)	288	concatenate_55[0][0]
activation_60 (Activation)	(None, 32, 32, 72)	0	batch_normalization_61[0][0]
conv2d_61 (Conv2D)	(None, 32, 32, 5)	9000	activation_60[0][0]

concatenate_56 (Concatenate)	(None, 32, 32, 77)	0	concatenate_55[0][0] conv2d_61[0][0]
batch_normalization_62 (BatchNo	(None, 32, 32, 77)	308	concatenate_56[0][0]
activation_61 (Activation)	(None, 32, 32, 77)	0	batch_normalization_62[0][0]
conv2d_62 (Conv2D)	(None, 32, 32, 5)	9625	activation_61[0][0]
concatenate_57 (Concatenate)	(None, 32, 32, 82)	0	concatenate_56[0][0] conv2d_62[0][0]
batch_normalization_63 (BatchNo	(None, 32, 32, 82)	328	concatenate_57[0][0]
activation_62 (Activation)	(None, 32, 32, 82)	0	batch_normalization_63[0][0]
conv2d_63 (Conv2D)	(None, 32, 32, 5)	10250	activation_62[0][0]
concatenate_58 (Concatenate)	(None, 32, 32, 87)	0	concatenate_57[0][0] conv2d_63[0][0]
batch_normalization_64 (BatchNo	(None, 32, 32, 87)	348	concatenate_58[0][0]
activation_63 (Activation)	(None, 32, 32, 87)	0	batch_normalization_64[0][0]
conv2d_64 (Conv2D)	(None, 32, 32, 5)	10875	activation_63[0][0]
concatenate_59 (Concatenate)	(None, 32, 32, 92)	0	concatenate_58[0][0] conv2d_64[0][0]
batch_normalization_65 (BatchNo	(None, 32, 32, 92)	368	concatenate_59[0][0]
activation_64 (Activation)	(None, 32, 32, 92)	0	batch_normalization_65[0][0]
conv2d_65 (Conv2D)	(None, 32, 32, 32)	73600	activation_64[0][0]
average_pooling2d_4 (AveragePoo	(None, 16, 16, 32)	0	conv2d_65[0][0]
batch_normalization_66 (BatchNo	(None, 16, 16, 32)	128	average_pooling2d_4[0][0]
activation_65 (Activation)	(None, 16, 16, 32)	0	batch_normalization_66[0][0]
conv2d_66 (Conv2D)	(None, 16, 16, 5)	4000	activation_65[0][0]
concatenate_60 (Concatenate)	(None, 16, 16, 37)	0	average_pooling2d_4[0][0] conv2d_66[0][0]
batch_normalization_67 (BatchNo	(None, 16, 16, 37)	148	concatenate_60[0][0]
activation_66 (Activation)	(None, 16, 16, 37)	0	batch_normalization_67[0][0]
conv2d_67 (Conv2D)	(None, 16, 16, 5)	4625	activation_66[0][0]
concatenate_61 (Concatenate)	(None, 16, 16, 42)	0	concatenate_60[0][0] conv2d_67[0][0]
batch_normalization_68 (BatchNo	(None, 16, 16, 42)	168	concatenate_61[0][0]
activation_67 (Activation)	(None, 16, 16, 42)	0	batch_normalization_68[0][0]
conv2d_68 (Conv2D)	(None, 16, 16, 5)	5250	activation_67[0][0]
concatenate_62 (Concatenate)	(None, 16, 16, 47)	0	concatenate_61[0][0] conv2d_68[0][0]
batch_normalization_69 (BatchNo	(None, 16, 16, 47)	188	concatenate_62[0][0]
activation_68 (Activation)	(None, 16, 16, 47)	0	batch_normalization_69[0][0]
conv2d_69 (Conv2D)	(None, 16, 16, 5)	5875	activation_68[0][0]
concatenate_63 (Concatenate)	(None, 16, 16, 52)	0	concatenate_62[0][0] conv2d_69[0][0]
batch_normalization_70 (BatchNo	(None, 16, 16, 52)	208	concatenate_63[0][0]
activation_69 (Activation)	(None, 16, 16, 52)	0	batch_normalization_70[0][0]

activation_69 (Activation)	(None, 16, 16, 52)	0	batch_normalization_70[0][0]
conv2d_70 (Conv2D)	(None, 16, 16, 5)	6500	activation_69[0][0]
concatenate_64 (Concatenate)	(None, 16, 16, 57)	0	concatenate_63[0][0] conv2d_70[0][0]
batch_normalization_71 (BatchNo	(None, 16, 16, 57)	228	concatenate_64[0][0]
activation_70 (Activation)	(None, 16, 16, 57)	0	batch_normalization_71[0][0]
conv2d_71 (Conv2D)	(None, 16, 16, 5)	7125	activation_70[0][0]
concatenate_65 (Concatenate)	(None, 16, 16, 62)	0	concatenate_64[0][0] conv2d_71[0][0]
batch_normalization_72 (BatchNo	(None, 16, 16, 62)	248	concatenate_65[0][0]
activation_71 (Activation)	(None, 16, 16, 62)	0	batch_normalization_72[0][0]
conv2d_72 (Conv2D)	(None, 16, 16, 5)	7750	activation_71[0][0]
concatenate_66 (Concatenate)	(None, 16, 16, 67)	0	concatenate_65[0][0] conv2d_72[0][0]
batch_normalization_73 (BatchNo	(None, 16, 16, 67)	268	concatenate_66[0][0]
activation_72 (Activation)	(None, 16, 16, 67)	0	batch_normalization_73[0][0]
conv2d_73 (Conv2D)	(None, 16, 16, 5)	8375	activation_72[0][0]
concatenate_67 (Concatenate)	(None, 16, 16, 72)	0	concatenate_66[0][0] conv2d_73[0][0]
batch_normalization_74 (BatchNo	(None, 16, 16, 72)	288	concatenate_67[0][0]
activation_73 (Activation)	(None, 16, 16, 72)	0	batch_normalization_74[0][0]
conv2d_74 (Conv2D)	(None, 16, 16, 5)	9000	activation_73[0][0]
concatenate_68 (Concatenate)	(None, 16, 16, 77)	0	concatenate_67[0][0] conv2d_74[0][0]
batch_normalization_75 (BatchNo	(None, 16, 16, 77)	308	concatenate_68[0][0]
activation_74 (Activation)	(None, 16, 16, 77)	0	batch_normalization_75[0][0]
conv2d_75 (Conv2D)	(None, 16, 16, 5)	9625	activation_74[0][0]
concatenate_69 (Concatenate)	(None, 16, 16, 82)	0	concatenate_68[0][0] conv2d_75[0][0]
batch_normalization_76 (BatchNo	(None, 16, 16, 82)	328	concatenate_69[0][0]
activation_75 (Activation)	(None, 16, 16, 82)	0	batch_normalization_76[0][0]
conv2d_76 (Conv2D)	(None, 16, 16, 5)	10250	activation_75[0][0]
concatenate_70 (Concatenate)	(None, 16, 16, 87)	0	concatenate_69[0][0] conv2d_76[0][0]
batch_normalization_77 (BatchNo	(None, 16, 16, 87)	348	concatenate_70[0][0]
activation_76 (Activation)	(None, 16, 16, 87)	0	batch_normalization_77[0][0]
conv2d_77 (Conv2D)	(None, 16, 16, 5)	10875	activation_76[0][0]
concatenate_71 (Concatenate)	(None, 16, 16, 92)	0	concatenate_70[0][0] conv2d_77[0][0]
batch_normalization_78 (BatchNo	(None, 16, 16, 92)	368	concatenate_71[0][0]
activation_77 (Activation)	(None, 16, 16, 92)	0	batch_normalization_78[0][0]
conv2d_78 (Conv2D)	(None, 16, 16, 16)	36800	activation_77[0][0]
average_pooling2d_5 (AveragePoo	(None, 8, 8, 16)	0	conv2d_78[0][0]

batch_normalization_79 (BatchNo	(None, 8, 8, 16)	64	average_pooling2d_5[0][0]
activation_78 (Activation)	(None, 8, 8, 16)	0	batch_normalization_79[0][0]
conv2d_79 (Conv2D)	(None, 8, 8, 6)	2400	activation_78[0][0]
concatenate_72 (Concatenate)	(None, 8, 8, 22)	0	average_pooling2d_5[0][0] conv2d_79[0][0]
batch_normalization_80 (BatchNo	(None, 8, 8, 22)	88	concatenate_72[0][0]
activation_79 (Activation)	(None, 8, 8, 22)	0	batch_normalization_80[0][0]
conv2d_80 (Conv2D)	(None, 8, 8, 6)	3300	activation_79[0][0]
concatenate_73 (Concatenate)	(None, 8, 8, 28)	0	concatenate_72[0][0] conv2d_80[0][0]
batch_normalization_81 (BatchNo	(None, 8, 8, 28)	112	concatenate_73[0][0]
activation_80 (Activation)	(None, 8, 8, 28)	0	batch_normalization_81[0][0]
conv2d_81 (Conv2D)	(None, 8, 8, 6)	4200	activation_80[0][0]
concatenate_74 (Concatenate)	(None, 8, 8, 34)	0	concatenate_73[0][0] conv2d_81[0][0]
batch_normalization_82 (BatchNo	(None, 8, 8, 34)	136	concatenate_74[0][0]
activation_81 (Activation)	(None, 8, 8, 34)	0	batch_normalization_82[0][0]
conv2d_82 (Conv2D)	(None, 8, 8, 6)	5100	activation_81[0][0]
concatenate_75 (Concatenate)	(None, 8, 8, 40)	0	concatenate_74[0][0] conv2d_82[0][0]
batch_normalization_83 (BatchNo	(None, 8, 8, 40)	160	concatenate_75[0][0]
activation_82 (Activation)	(None, 8, 8, 40)	0	batch_normalization_83[0][0]
conv2d_83 (Conv2D)	(None, 8, 8, 6)	6000	activation_82[0][0]
concatenate_76 (Concatenate)	(None, 8, 8, 46)	0	concatenate_75[0][0] conv2d_83[0][0]
batch_normalization_84 (BatchNo	(None, 8, 8, 46)	184	concatenate_76[0][0]
activation_83 (Activation)	(None, 8, 8, 46)	0	batch_normalization_84[0][0]
conv2d_84 (Conv2D)	(None, 8, 8, 6)	6900	activation_83[0][0]
concatenate_77 (Concatenate)	(None, 8, 8, 52)	0	concatenate_76[0][0] conv2d_84[0][0]
batch_normalization_85 (BatchNo	(None, 8, 8, 52)	208	concatenate_77[0][0]
activation_84 (Activation)	(None, 8, 8, 52)	0	batch_normalization_85[0][0]
conv2d_85 (Conv2D)	(None, 8, 8, 6)	7800	activation_84[0][0]
concatenate_78 (Concatenate)	(None, 8, 8, 58)	0	concatenate_77[0][0] conv2d_85[0][0]
batch_normalization_86 (BatchNo	(None, 8, 8, 58)	232	concatenate_78[0][0]
activation_85 (Activation)	(None, 8, 8, 58)	0	batch_normalization_86[0][0]
conv2d_86 (Conv2D)	(None, 8, 8, 6)	8700	activation_85[0][0]
concatenate_79 (Concatenate)	(None, 8, 8, 64)	0	concatenate_78[0][0] conv2d_86[0][0]
batch_normalization_87 (BatchNo	(None, 8, 8, 64)	256	concatenate_79[0][0]
activation_86 (Activation)	(None, 8, 8, 64)	0	batch_normalization_87[0][0]
conv2d_87 (Conv2D)	(None, 8, 8, 6)	9600	activation_86[0][0]

conv2d_87 (Conv2D)	(None, 8, 8, 6)	9600	activation_86[0][0]
concatenate_80 (Concatenate)	(None, 8, 8, 70)	0	concatenate_79[0][0] conv2d_87[0][0]
batch_normalization_88 (BatchNo	(None, 8, 8, 70)	280	concatenate_80[0][0]
activation_87 (Activation)	(None, 8, 8, 70)	0	batch_normalization_88[0][0]
conv2d_88 (Conv2D)	(None, 8, 8, 6)	10500	activation_87[0][0]
concatenate_81 (Concatenate)	(None, 8, 8, 76)	0	concatenate_80[0][0] conv2d_88[0][0]
batch_normalization_89 (BatchNo	(None, 8, 8, 76)	304	concatenate_81[0][0]
activation_88 (Activation)	(None, 8, 8, 76)	0	batch_normalization_89[0][0]
conv2d_89 (Conv2D)	(None, 8, 8, 6)	11400	activation_88[0][0]
concatenate_82 (Concatenate)	(None, 8, 8, 82)	0	concatenate_81[0][0] conv2d_89[0][0]
batch_normalization_90 (BatchNo	(None, 8, 8, 82)	328	concatenate_82[0][0]
activation_89 (Activation)	(None, 8, 8, 82)	0	batch_normalization_90[0][0]
conv2d_90 (Conv2D)	(None, 8, 8, 6)	12300	activation_89[0][0]
concatenate_83 (Concatenate)	(None, 8, 8, 88)	0	concatenate_82[0][0] conv2d_90[0][0]
batch_normalization_91 (BatchNo	(None, 8, 8, 88)	352	concatenate_83[0][0]
activation_90 (Activation)	(None, 8, 8, 88)	0	batch_normalization_91[0][0]
conv2d_91 (Conv2D)	(None, 8, 8, 16)	35200	activation_90[0][0]
average_pooling2d_6 (AveragePoo	(None, 4, 4, 16)	0	conv2d_91[0][0]
batch_normalization_92 (BatchNo	(None, 4, 4, 16)	64	average_pooling2d_6[0][0]
activation_91 (Activation)	(None, 4, 4, 16)	0	batch_normalization_92[0][0]
conv2d_92 (Conv2D)	(None, 4, 4, 6)	2400	activation_91[0][0]
concatenate_84 (Concatenate)	(None, 4, 4, 22)	0	average_pooling2d_6[0][0] conv2d_92[0][0]
batch_normalization_93 (BatchNo	(None, 4, 4, 22)	88	concatenate_84[0][0]
activation_92 (Activation)	(None, 4, 4, 22)	0	batch_normalization_93[0][0]
conv2d_93 (Conv2D)	(None, 4, 4, 6)	3300	activation_92[0][0]
concatenate_85 (Concatenate)	(None, 4, 4, 28)	0	concatenate_84[0][0] conv2d_93[0][0]
batch_normalization_94 (BatchNo	(None, 4, 4, 28)	112	concatenate_85[0][0]
activation_93 (Activation)	(None, 4, 4, 28)	0	batch_normalization_94[0][0]
conv2d_94 (Conv2D)	(None, 4, 4, 6)	4200	activation_93[0][0]
concatenate_86 (Concatenate)	(None, 4, 4, 34)	0	concatenate_85[0][0] conv2d_94[0][0]
batch_normalization_95 (BatchNo	(None, 4, 4, 34)	136	concatenate_86[0][0]
activation_94 (Activation)	(None, 4, 4, 34)	0	batch_normalization_95[0][0]
conv2d_95 (Conv2D)	(None, 4, 4, 6)	5100	activation_94[0][0]
concatenate_87 (Concatenate)	(None, 4, 4, 40)	0	concatenate_86[0][0] conv2d_95[0][0]
batch_normalization_96 (BatchNo	(None, 4, 4, 40)	160	concatenate_87[0][0]

activation_95 (Activation)	(None, 4, 4, 40)	0	batch_normalization_96[0][0]
conv2d_96 (Conv2D)	(None, 4, 4, 6)	6000	activation_95[0][0]
concatenate_88 (Concatenate)	(None, 4, 4, 46)	0	concatenate_87[0][0] conv2d_96[0][0]
batch_normalization_97 (BatchNo	(None, 4, 4, 46)	184	concatenate_88[0][0]
activation_96 (Activation)	(None, 4, 4, 46)	0	batch_normalization_97[0][0]
conv2d_97 (Conv2D)	(None, 4, 4, 6)	6900	activation_96[0][0]
concatenate_89 (Concatenate)	(None, 4, 4, 52)	0	concatenate_88[0][0] conv2d_97[0][0]
batch_normalization_98 (BatchNo	(None, 4, 4, 52)	208	concatenate_89[0][0]
activation_97 (Activation)	(None, 4, 4, 52)	0	batch_normalization_98[0][0]
conv2d_98 (Conv2D)	(None, 4, 4, 6)	7800	activation_97[0][0]
concatenate_90 (Concatenate)	(None, 4, 4, 58)	0	concatenate_89[0][0] conv2d_98[0][0]
batch_normalization_99 (BatchNo	(None, 4, 4, 58)	232	concatenate_90[0][0]
activation_98 (Activation)	(None, 4, 4, 58)	0	batch_normalization_99[0][0]
conv2d_99 (Conv2D)	(None, 4, 4, 6)	8700	activation_98[0][0]
concatenate_91 (Concatenate)	(None, 4, 4, 64)	0	concatenate_90[0][0] conv2d_99[0][0]
batch_normalization_100 (BatchN	(None, 4, 4, 64)	256	concatenate_91[0][0]
activation_99 (Activation)	(None, 4, 4, 64)	0	batch_normalization_100[0][0]
conv2d_100 (Conv2D)	(None, 4, 4, 6)	9600	activation_99[0][0]
concatenate_92 (Concatenate)	(None, 4, 4, 70)	0	concatenate_91[0][0] conv2d_100[0][0]
batch_normalization_101 (BatchN	(None, 4, 4, 70)	280	concatenate_92[0][0]
activation_100 (Activation)	(None, 4, 4, 70)	0	batch_normalization_101[0][0]
conv2d_101 (Conv2D)	(None, 4, 4, 6)	10500	activation_100[0][0]
concatenate_93 (Concatenate)	(None, 4, 4, 76)	0	concatenate_92[0][0] conv2d_101[0][0]
batch_normalization_102 (BatchN	(None, 4, 4, 76)	304	concatenate_93[0][0]
activation_101 (Activation)	(None, 4, 4, 76)	0	batch_normalization_102[0][0]
conv2d_102 (Conv2D)	(None, 4, 4, 6)	11400	activation_101[0][0]
concatenate_94 (Concatenate)	(None, 4, 4, 82)	0	concatenate_93[0][0] conv2d_102[0][0]
batch_normalization_103 (BatchN	(None, 4, 4, 82)	328	concatenate_94[0][0]
activation_102 (Activation)	(None, 4, 4, 82)	0	batch_normalization_103[0][0]
conv2d_103 (Conv2D)	(None, 4, 4, 6)	12300	activation_102[0][0]
concatenate_95 (Concatenate)	(None, 4, 4, 88)	0	concatenate_94[0][0] conv2d_103[0][0]
batch_normalization_104 (BatchN	(None, 4, 4, 88)	352	concatenate_95[0][0]
activation_103 (Activation)	(None, 4, 4, 88)	0	batch_normalization_104[0][0]
max_pooling2d (MaxPooling2D)	(None, 2, 2, 88)	0	activation_103[0][0]

```
conv2d_104 (Conv2D)             (None, 1, 1, 10)     3530        max_pooling2d[0][0]
__________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 10)           0           conv2d_104[0][0]
==================================================================================================
Total params: 516,750
Trainable params: 510,822
Non-trainable params: 5,928
__________________________________________________________________________________________________
```

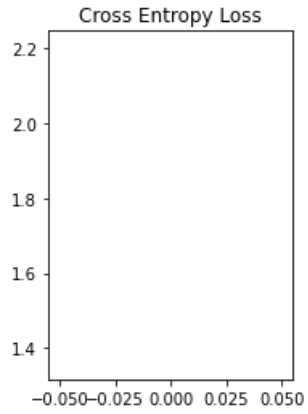
In [0]:

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

In [29]:

```
# sample run
model_harness(X_train, y_train, X_test, y_test, 60, 39, 1)
```

```
WARNING:tensorflow:From <ipython-input-22-9641928ef676>:12: Model.fit_generator (from
tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.
Instructions for updating:
Please use Model.fit, which supports generators.
1282/1282 [==============================] - 125s 97ms/step - loss: 1.3586 - accuracy: 0.5066 - va
l_loss: 2.2045 - val_accuracy: 0.4037
313/313 [==============================] - 5s 16ms/step - loss: 2.2045 - accuracy: 0.4037
> 40.370
```



In [30]:

```
# entry point, run the test harness for 50 * 5 iterations, 1st slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 50)
```

```
Epoch 1/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.8660 - accuracy: 0.6932 - va
l_loss: 1.0409 - val_accuracy: 0.6576
Epoch 2/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.6867 - accuracy: 0.7593 - va
l_loss: 1.0192 - val_accuracy: 0.6963
Epoch 3/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.5855 - accuracy: 0.7983 - va
l_loss: 0.7289 - val_accuracy: 0.7644
Epoch 4/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.5207 - accuracy: 0.8198 - va
l_loss: 0.7642 - val_accuracy: 0.7520
Epoch 5/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.4703 - accuracy: 0.8362 - va
l_loss: 0.8759 - val_accuracy: 0.7475
Epoch 6/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.4331 - accuracy: 0.8500 - va
l_loss: 0.5181 - val_accuracy: 0.8315
Epoch 7/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.3976 - accuracy: 0.8632 - va
l loss: 0.9855 - val accuracy: 0.7000
```

```
Epoch 8/50
1282/1282 [==============================] - 122s 96ms/step - loss: 0.3685 - accuracy: 0.8735 - va
l_loss: 0.5827 - val_accuracy: 0.8098
Epoch 9/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.3502 - accuracy: 0.8785 - va
l_loss: 0.5268 - val_accuracy: 0.8274
Epoch 10/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.3253 - accuracy: 0.8871 - va
l_loss: 0.4344 - val_accuracy: 0.8566
Epoch 11/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.3065 - accuracy: 0.8937 - va
l_loss: 0.4581 - val_accuracy: 0.8505
Epoch 12/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.2877 - accuracy: 0.8993 - va
l_loss: 0.5677 - val_accuracy: 0.8288
Epoch 13/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.2827 - accuracy: 0.9016 - va
l_loss: 0.4890 - val_accuracy: 0.8505
Epoch 14/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.2626 - accuracy: 0.9077 - va
l_loss: 0.5302 - val_accuracy: 0.8353
Epoch 15/50
1282/1282 [==============================] - 124s 97ms/step - loss: 0.2538 - accuracy: 0.9110 - va
l_loss: 0.5711 - val_accuracy: 0.8269
Epoch 16/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.2365 - accuracy: 0.9180 - va
l_loss: 0.4538 - val_accuracy: 0.8590
Epoch 17/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.2280 - accuracy: 0.9198 - va
l_loss: 0.5031 - val_accuracy: 0.8533
Epoch 18/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.2166 - accuracy: 0.9251 - va
l_loss: 0.4858 - val_accuracy: 0.8552
Epoch 19/50
1282/1282 [==============================] - 124s 96ms/step - loss: 0.2117 - accuracy: 0.9260 - va
l_loss: 0.4953 - val_accuracy: 0.8528
Epoch 20/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.2027 - accuracy: 0.9283 - va
l_loss: 0.3930 - val_accuracy: 0.8749
Epoch 21/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1952 - accuracy: 0.9313 - va
l_loss: 0.4831 - val_accuracy: 0.8607
Epoch 22/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1877 - accuracy: 0.9334 - va
l_loss: 0.5390 - val_accuracy: 0.8521
Epoch 23/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1831 - accuracy: 0.9354 - va
l_loss: 0.4747 - val_accuracy: 0.8638
Epoch 24/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1742 - accuracy: 0.9390 - va
l_loss: 0.4574 - val_accuracy: 0.8703
Epoch 25/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1716 - accuracy: 0.9393 - va
l_loss: 0.4609 - val_accuracy: 0.8690
Epoch 26/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1640 - accuracy: 0.9422 - va
l_loss: 0.5551 - val_accuracy: 0.8495
Epoch 27/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1590 - accuracy: 0.9433 - va
l_loss: 0.4133 - val_accuracy: 0.8815
Epoch 28/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1518 - accuracy: 0.9466 - va
l_loss: 0.4696 - val_accuracy: 0.8667
Epoch 29/50
1282/1282 [==============================] - 124s 96ms/step - loss: 0.1470 - accuracy: 0.9478 - va
l_loss: 0.4961 - val_accuracy: 0.8623
Epoch 30/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1470 - accuracy: 0.9475 - va
l_loss: 0.4960 - val_accuracy: 0.8689
Epoch 31/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1375 - accuracy: 0.9512 - va
l_loss: 0.5330 - val_accuracy: 0.8587
Epoch 32/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1376 - accuracy: 0.9514 - va
l_loss: 0.4404 - val_accuracy: 0.8777
Epoch 33/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1308 - accuracy: 0.9542 - va
```

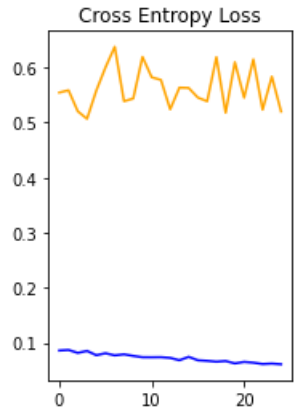
```
l_loss: 0.4889 - val_accuracy: 0.8737
Epoch 34/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1287 - accuracy: 0.9541 - va
l_loss: 0.4713 - val_accuracy: 0.8770
Epoch 35/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1238 - accuracy: 0.9560 - va
l_loss: 0.5709 - val_accuracy: 0.8584
Epoch 36/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1233 - accuracy: 0.9556 - va
l_loss: 0.4822 - val_accuracy: 0.8761
Epoch 37/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1165 - accuracy: 0.9585 - va
l_loss: 0.4925 - val_accuracy: 0.8726
Epoch 38/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1167 - accuracy: 0.9589 - va
l_loss: 0.5539 - val_accuracy: 0.8639
Epoch 39/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1145 - accuracy: 0.9595 - va
l_loss: 0.4832 - val_accuracy: 0.8771
Epoch 40/50
1282/1282 [==============================] - 123s 96ms/step - loss: 0.1083 - accuracy: 0.9616 - va
l_loss: 0.4617 - val_accuracy: 0.8788
Epoch 41/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.1087 - accuracy: 0.9608 - va
l_loss: 0.5368 - val_accuracy: 0.8675
Epoch 42/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.1060 - accuracy: 0.9617 - va
l_loss: 0.4694 - val_accuracy: 0.8841
Epoch 43/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.1060 - accuracy: 0.9626 - va
l_loss: 0.4511 - val_accuracy: 0.8831
Epoch 44/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0991 - accuracy: 0.9649 - va
l_loss: 0.4976 - val_accuracy: 0.8785
Epoch 45/50
1282/1282 [==============================] - 122s 95ms/step - loss: 0.1013 - accuracy: 0.9643 - va
l_loss: 0.4611 - val_accuracy: 0.8802
Epoch 46/50
 379/1282 [=======>.....................] - ETA: 1:22 - loss: 0.0964 - accuracy: 0.9647Buffered d
ata was truncated after reaching the output size limit.
```

In [31]:

```
# entry point, run the test harness for 25 * 8 iterations, 2nd slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0862 - accuracy: 0.9690 - va
l_loss: 0.5547 - val_accuracy: 0.8716
Epoch 2/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0872 - accuracy: 0.9692 - va
l_loss: 0.5588 - val_accuracy: 0.8696
Epoch 3/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0818 - accuracy: 0.9715 - va
l_loss: 0.5206 - val_accuracy: 0.8823
Epoch 4/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0854 - accuracy: 0.9698 - va
l_loss: 0.5070 - val_accuracy: 0.8804
Epoch 5/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0776 - accuracy: 0.9723 - va
l_loss: 0.5580 - val_accuracy: 0.8731
Epoch 6/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0812 - accuracy: 0.9714 - va
l_loss: 0.6004 - val_accuracy: 0.8623
Epoch 7/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0774 - accuracy: 0.9727 - va
l_loss: 0.6378 - val_accuracy: 0.8578
Epoch 8/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0793 - accuracy: 0.9718 - va
l_loss: 0.5389 - val_accuracy: 0.8770
Epoch 9/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0764 - accuracy: 0.9731 - va
l_loss: 0.5438 - val_accuracy: 0.8797
Epoch 10/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0740 - accuracy: 0.9733 - va
```

```
l_loss: 0.6194 - val_accuracy: 0.8651
Epoch 11/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0739 - accuracy: 0.9738 - va
l_loss: 0.5824 - val_accuracy: 0.8760
Epoch 12/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0740 - accuracy: 0.9734 - va
l_loss: 0.5777 - val_accuracy: 0.8773
Epoch 13/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0728 - accuracy: 0.9742 - va
l_loss: 0.5242 - val_accuracy: 0.8804
Epoch 14/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0685 - accuracy: 0.9756 - va
l_loss: 0.5636 - val_accuracy: 0.8787
Epoch 15/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0746 - accuracy: 0.9742 - va
l_loss: 0.5630 - val_accuracy: 0.8764
Epoch 16/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0684 - accuracy: 0.9755 - va
l_loss: 0.5455 - val_accuracy: 0.8827
Epoch 17/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0674 - accuracy: 0.9760 - va
l_loss: 0.5388 - val_accuracy: 0.8845
Epoch 18/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0659 - accuracy: 0.9769 - va
l_loss: 0.6190 - val_accuracy: 0.8756
Epoch 19/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0670 - accuracy: 0.9764 - va
l_loss: 0.5180 - val_accuracy: 0.8855
Epoch 20/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0628 - accuracy: 0.9778 - va
l_loss: 0.6098 - val_accuracy: 0.8751
Epoch 21/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0655 - accuracy: 0.9768 - va
l_loss: 0.5453 - val_accuracy: 0.8832
Epoch 22/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0641 - accuracy: 0.9777 - va
l_loss: 0.6150 - val_accuracy: 0.8684
Epoch 23/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0616 - accuracy: 0.9786 - va
l_loss: 0.5235 - val_accuracy: 0.8843
Epoch 24/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0625 - accuracy: 0.9774 - va
l_loss: 0.5837 - val_accuracy: 0.8748
Epoch 25/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0612 - accuracy: 0.9784 - va
l_loss: 0.5202 - val_accuracy: 0.8815
313/313 [==============================] - 5s 16ms/step - loss: 0.5202 - accuracy: 0.8815
> 88.150
```

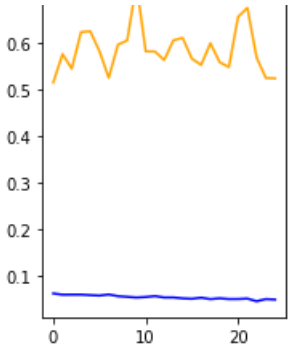


In [32]:

```
# entry point, run the test harness for 25 * 8 iterations, 3rd slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0615 - accuracy: 0.9785 - va
l_loss: 0.5160 - val_accuracy: 0.8866
Epoch 2/25
```

```
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0589 - accuracy: 0.9795 - va
l_loss: 0.5773 - val_accuracy: 0.8808
Epoch 3/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0590 - accuracy: 0.9792 - va
l_loss: 0.5458 - val_accuracy: 0.8872
Epoch 4/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0590 - accuracy: 0.9787 - va
l_loss: 0.6247 - val_accuracy: 0.8708
Epoch 5/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0579 - accuracy: 0.9794 - va
l_loss: 0.6266 - val_accuracy: 0.8707
Epoch 6/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0569 - accuracy: 0.9798 - va
l_loss: 0.5833 - val_accuracy: 0.8807
Epoch 7/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0593 - accuracy: 0.9789 - va
l_loss: 0.5262 - val_accuracy: 0.8887
Epoch 8/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0556 - accuracy: 0.9799 - va
l_loss: 0.5977 - val_accuracy: 0.8761
Epoch 9/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0542 - accuracy: 0.9806 - va
l_loss: 0.6068 - val_accuracy: 0.8763
Epoch 10/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0527 - accuracy: 0.9820 - va
l_loss: 0.7288 - val_accuracy: 0.8625
Epoch 11/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0538 - accuracy: 0.9804 - va
l_loss: 0.5832 - val_accuracy: 0.8825
Epoch 12/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0559 - accuracy: 0.9808 - va
l_loss: 0.5828 - val_accuracy: 0.8811
Epoch 13/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0528 - accuracy: 0.9815 - va
l_loss: 0.5644 - val_accuracy: 0.8845
Epoch 14/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0529 - accuracy: 0.9810 - va
l_loss: 0.6071 - val_accuracy: 0.8819
Epoch 15/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0509 - accuracy: 0.9819 - va
l_loss: 0.6121 - val_accuracy: 0.8763
Epoch 16/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0500 - accuracy: 0.9823 - va
l_loss: 0.5674 - val_accuracy: 0.8834
Epoch 17/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0523 - accuracy: 0.9818 - va
l_loss: 0.5539 - val_accuracy: 0.8847
Epoch 18/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0493 - accuracy: 0.9825 - va
l_loss: 0.6008 - val_accuracy: 0.8840
Epoch 19/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0510 - accuracy: 0.9822 - va
l_loss: 0.5597 - val_accuracy: 0.8870
Epoch 20/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0494 - accuracy: 0.9828 - va
l_loss: 0.5493 - val_accuracy: 0.8870
Epoch 21/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0495 - accuracy: 0.9827 - va
l_loss: 0.6576 - val_accuracy: 0.8703
Epoch 22/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0505 - accuracy: 0.9828 - va
l_loss: 0.6769 - val_accuracy: 0.8726
Epoch 23/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0447 - accuracy: 0.9848 - va
l_loss: 0.5699 - val_accuracy: 0.8856
Epoch 24/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0494 - accuracy: 0.9826 - va
l_loss: 0.5257 - val_accuracy: 0.8931
Epoch 25/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0483 - accuracy: 0.9834 - va
l_loss: 0.5249 - val_accuracy: 0.8915
313/313 [==============================] - 5s 17ms/step - loss: 0.5249 - accuracy: 0.8915
> 89.150
```

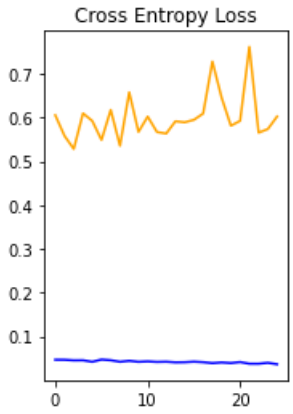


In [33]:

```
# entry point, run the test harness for 25 * 8 iterations, 4th slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0470 - accuracy: 0.9830 - va
l_loss: 0.6054 - val_accuracy: 0.8792
Epoch 2/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0469 - accuracy: 0.9839 - va
l_loss: 0.5582 - val_accuracy: 0.8873
Epoch 3/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0456 - accuracy: 0.9838 - va
l_loss: 0.5284 - val_accuracy: 0.8914
Epoch 4/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0457 - accuracy: 0.9837 - va
l_loss: 0.6097 - val_accuracy: 0.8796
Epoch 5/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0422 - accuracy: 0.9850 - va
l_loss: 0.5924 - val_accuracy: 0.8871
Epoch 6/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0474 - accuracy: 0.9836 - va
l_loss: 0.5490 - val_accuracy: 0.8897
Epoch 7/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0458 - accuracy: 0.9841 - va
l_loss: 0.6175 - val_accuracy: 0.8808
Epoch 8/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0423 - accuracy: 0.9851 - va
l_loss: 0.5357 - val_accuracy: 0.8906
Epoch 9/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0445 - accuracy: 0.9841 - va
l_loss: 0.6579 - val_accuracy: 0.8774
Epoch 10/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0422 - accuracy: 0.9852 - va
l_loss: 0.5673 - val_accuracy: 0.8876
Epoch 11/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0432 - accuracy: 0.9847 - va
l_loss: 0.6024 - val_accuracy: 0.8802
Epoch 12/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0420 - accuracy: 0.9853 - va
l_loss: 0.5671 - val_accuracy: 0.8874
Epoch 13/25
1282/1282 [==============================] - 122s 96ms/step - loss: 0.0425 - accuracy: 0.9855 - va
l_loss: 0.5635 - val_accuracy: 0.8876
Epoch 14/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0410 - accuracy: 0.9856 - va
l_loss: 0.5916 - val_accuracy: 0.8878
Epoch 15/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0413 - accuracy: 0.9857 - va
l_loss: 0.5893 - val_accuracy: 0.8880
Epoch 16/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0426 - accuracy: 0.9851 - va
l_loss: 0.5949 - val_accuracy: 0.8833
Epoch 17/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0414 - accuracy: 0.9855 - va
l_loss: 0.6093 - val_accuracy: 0.8818
Epoch 18/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0393 - accuracy: 0.9859 - va
l_loss: 0.7280 - val_accuracy: 0.8714
Epoch 19/25
1282/1282 [==============================] - 124s 96ms/step - loss: 0.0408 - accuracy: 0.9858 - va
l_loss: 0.6453 - val_accuracy: 0.8824
```

```
l_loss: 0.6455 - val_accuracy: 0.8824
Epoch 20/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0395 - accuracy: 0.9859 - va
l_loss: 0.5815 - val_accuracy: 0.8898
Epoch 21/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0417 - accuracy: 0.9852 - va
l_loss: 0.5925 - val_accuracy: 0.8871
Epoch 22/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0379 - accuracy: 0.9870 - va
l_loss: 0.7617 - val_accuracy: 0.8626
Epoch 23/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0378 - accuracy: 0.9862 - va
l_loss: 0.5656 - val_accuracy: 0.8946
Epoch 24/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0400 - accuracy: 0.9858 - va
l_loss: 0.5738 - val_accuracy: 0.8890
Epoch 25/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0367 - accuracy: 0.9876 - va
l_loss: 0.6025 - val_accuracy: 0.8875
313/313 [==============================] - 5s 16ms/step - loss: 0.6025 - accuracy: 0.8875
> 88.750
```

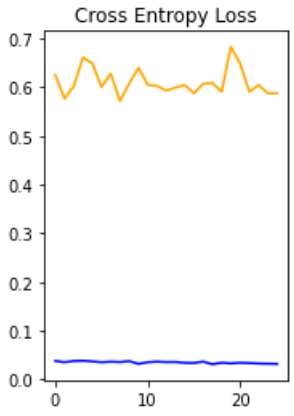


In [34]:

```
# entry point, run the test harness for 25 * 8 iterations, 5th slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0383 - accuracy: 0.9865 - va
l_loss: 0.6244 - val_accuracy: 0.8817
Epoch 2/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0357 - accuracy: 0.9873 - va
l_loss: 0.5762 - val_accuracy: 0.8912
Epoch 3/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0382 - accuracy: 0.9864 - va
l_loss: 0.6020 - val_accuracy: 0.8875
Epoch 4/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0387 - accuracy: 0.9866 - va
l_loss: 0.6606 - val_accuracy: 0.8794
Epoch 5/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0374 - accuracy: 0.9870 - va
l_loss: 0.6488 - val_accuracy: 0.8820
Epoch 6/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0356 - accuracy: 0.9875 - va
l_loss: 0.5999 - val_accuracy: 0.8901
Epoch 7/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0368 - accuracy: 0.9870 - va
l_loss: 0.6274 - val_accuracy: 0.8846
Epoch 8/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0361 - accuracy: 0.9878 - va
l_loss: 0.5713 - val_accuracy: 0.8934
Epoch 9/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0379 - accuracy: 0.9867 - va
l_loss: 0.6074 - val_accuracy: 0.8841
Epoch 10/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0323 - accuracy: 0.9890 - va
l_loss: 0.6395 - val_accuracy: 0.8849
Epoch 11/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0357 - accuracy: 0.9874 - va
```

```
l_loss: 0.6045 - val_accuracy: 0.8877
Epoch 12/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0369 - accuracy: 0.9873 - va
l_loss: 0.6021 - val_accuracy: 0.8900
Epoch 13/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0360 - accuracy: 0.9877 - va
l_loss: 0.5927 - val_accuracy: 0.8916
Epoch 14/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0362 - accuracy: 0.9876 - va
l_loss: 0.5987 - val_accuracy: 0.8886
Epoch 15/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0345 - accuracy: 0.9879 - va
l_loss: 0.6042 - val_accuracy: 0.8854
Epoch 16/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0342 - accuracy: 0.9880 - va
l_loss: 0.5871 - val_accuracy: 0.8915
Epoch 17/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0370 - accuracy: 0.9872 - va
l_loss: 0.6068 - val_accuracy: 0.8897
Epoch 18/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0315 - accuracy: 0.9888 - va
l_loss: 0.6081 - val_accuracy: 0.8890
Epoch 19/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0347 - accuracy: 0.9878 - va
l_loss: 0.5905 - val_accuracy: 0.8886
Epoch 20/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0333 - accuracy: 0.9884 - va
l_loss: 0.6826 - val_accuracy: 0.8785
Epoch 21/25
1282/1282 [==============================] - 124s 97ms/step - loss: 0.0347 - accuracy: 0.9883 - va
l_loss: 0.6477 - val_accuracy: 0.8812
Epoch 22/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0340 - accuracy: 0.9882 - va
l_loss: 0.5902 - val_accuracy: 0.8880
Epoch 23/25
1282/1282 [==============================] - 124s 97ms/step - loss: 0.0330 - accuracy: 0.9885 - va
l_loss: 0.6042 - val_accuracy: 0.8906
Epoch 24/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0326 - accuracy: 0.9889 - va
l_loss: 0.5871 - val_accuracy: 0.8881
Epoch 25/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0323 - accuracy: 0.9887 - va
l_loss: 0.5873 - val_accuracy: 0.8919
313/313 [==============================] - 5s 17ms/step - loss: 0.5873 - accuracy: 0.8919
> 89.190
```

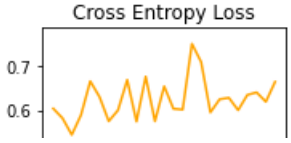


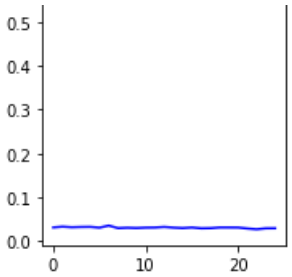
In [35]:

```
# entry point, run the test harness for 25 * 8 iterations, 6th slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0309 - accuracy: 0.9892 - va
l_loss: 0.6036 - val_accuracy: 0.8879
Epoch 2/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0327 - accuracy: 0.9885 - va
l_loss: 0.5816 - val_accuracy: 0.8914
Epoch 3/25
```

```
1282/1282 [==============================] - 122s 96ms/step - loss: 0.0313 - accuracy: 0.9895 - va
l_loss: 0.5437 - val_accuracy: 0.9013
Epoch 4/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0320 - accuracy: 0.9887 - va
l_loss: 0.5891 - val_accuracy: 0.8911
Epoch 5/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0322 - accuracy: 0.9888 - va
l_loss: 0.6661 - val_accuracy: 0.8830
Epoch 6/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0303 - accuracy: 0.9898 - va
l_loss: 0.6304 - val_accuracy: 0.8880
Epoch 7/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0351 - accuracy: 0.9879 - va
l_loss: 0.5748 - val_accuracy: 0.8930
Epoch 8/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0293 - accuracy: 0.9894 - va
l_loss: 0.5999 - val_accuracy: 0.8884
Epoch 9/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0302 - accuracy: 0.9899 - va
l_loss: 0.6692 - val_accuracy: 0.8831
Epoch 10/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0295 - accuracy: 0.9897 - va
l_loss: 0.5741 - val_accuracy: 0.8935
Epoch 11/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0304 - accuracy: 0.9895 - va
l_loss: 0.6766 - val_accuracy: 0.8801
Epoch 12/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0307 - accuracy: 0.9893 - va
l_loss: 0.5748 - val_accuracy: 0.8928
Epoch 13/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0320 - accuracy: 0.9889 - va
l_loss: 0.6544 - val_accuracy: 0.8794
Epoch 14/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0304 - accuracy: 0.9895 - va
l_loss: 0.6032 - val_accuracy: 0.8912
Epoch 15/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0294 - accuracy: 0.9897 - va
l_loss: 0.6013 - val_accuracy: 0.8939
Epoch 16/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0308 - accuracy: 0.9893 - va
l_loss: 0.7516 - val_accuracy: 0.8721
Epoch 17/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0289 - accuracy: 0.9895 - va
l_loss: 0.7099 - val_accuracy: 0.8790
Epoch 18/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0292 - accuracy: 0.9898 - va
l_loss: 0.5949 - val_accuracy: 0.8956
Epoch 19/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0308 - accuracy: 0.9893 - va
l_loss: 0.6249 - val_accuracy: 0.8874
Epoch 20/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0308 - accuracy: 0.9893 - va
l_loss: 0.6283 - val_accuracy: 0.8812
Epoch 21/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0307 - accuracy: 0.9893 - va
l_loss: 0.6000 - val_accuracy: 0.8871
Epoch 22/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0284 - accuracy: 0.9902 - va
l_loss: 0.6349 - val_accuracy: 0.8871
Epoch 23/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0267 - accuracy: 0.9906 - va
l_loss: 0.6406 - val_accuracy: 0.8880
Epoch 24/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0288 - accuracy: 0.9902 - va
l_loss: 0.6191 - val_accuracy: 0.8869
Epoch 25/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0290 - accuracy: 0.9901 - va
l_loss: 0.6647 - val_accuracy: 0.8855
313/313 [==============================] - 5s 16ms/step - loss: 0.6647 - accuracy: 0.8855
> 88.550
```



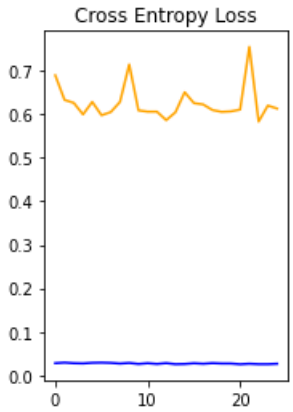


In [36]:

```
# entry point, run the test harness for 25 * 8 iterations, 7th slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0280 - accuracy: 0.9900 - va
l_loss: 0.6893 - val_accuracy: 0.8854
Epoch 2/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0294 - accuracy: 0.9895 - va
l_loss: 0.6329 - val_accuracy: 0.8846
Epoch 3/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0279 - accuracy: 0.9903 - va
l_loss: 0.6261 - val_accuracy: 0.8937
Epoch 4/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0274 - accuracy: 0.9909 - va
l_loss: 0.5996 - val_accuracy: 0.8953
Epoch 5/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0289 - accuracy: 0.9901 - va
l_loss: 0.6283 - val_accuracy: 0.8869
Epoch 6/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0293 - accuracy: 0.9900 - va
l_loss: 0.5977 - val_accuracy: 0.8935
Epoch 7/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0287 - accuracy: 0.9904 - va
l_loss: 0.6050 - val_accuracy: 0.8931
Epoch 8/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0271 - accuracy: 0.9907 - va
l_loss: 0.6283 - val_accuracy: 0.8916
Epoch 9/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0285 - accuracy: 0.9899 - va
l_loss: 0.7145 - val_accuracy: 0.8820
Epoch 10/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0262 - accuracy: 0.9907 - va
l_loss: 0.6089 - val_accuracy: 0.8917
Epoch 11/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0279 - accuracy: 0.9902 - va
l_loss: 0.6059 - val_accuracy: 0.8944
Epoch 12/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0262 - accuracy: 0.9909 - va
l_loss: 0.6059 - val_accuracy: 0.8910
Epoch 13/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0281 - accuracy: 0.9902 - va
l_loss: 0.5864 - val_accuracy: 0.8952
Epoch 14/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0257 - accuracy: 0.9914 - va
l_loss: 0.6045 - val_accuracy: 0.8925
Epoch 15/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0261 - accuracy: 0.9907 - va
l_loss: 0.6506 - val_accuracy: 0.8907
Epoch 16/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0278 - accuracy: 0.9907 - va
l_loss: 0.6256 - val_accuracy: 0.8900
Epoch 17/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0266 - accuracy: 0.9908 - va
l_loss: 0.6225 - val_accuracy: 0.8914
Epoch 18/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0281 - accuracy: 0.9904 - va
l_loss: 0.6097 - val_accuracy: 0.8933
Epoch 19/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0272 - accuracy: 0.9904 - va
l_loss: 0.6055 - val_accuracy: 0.8909
Epoch 20/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0272 - accuracy: 0.9906 - va
```

```
l_loss: 0.6066 - val_accuracy: 0.8905
Epoch 21/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0255 - accuracy: 0.9913 - va
l_loss: 0.6107 - val_accuracy: 0.8929
Epoch 22/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0267 - accuracy: 0.9906 - va
l_loss: 0.7550 - val_accuracy: 0.8729
Epoch 23/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0257 - accuracy: 0.9909 - va
l_loss: 0.5831 - val_accuracy: 0.8956
Epoch 24/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0257 - accuracy: 0.9911 - va
l_loss: 0.6199 - val_accuracy: 0.8930
Epoch 25/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0266 - accuracy: 0.9908 - va
l_loss: 0.6131 - val_accuracy: 0.8959
313/313 [==============================] - 5s 16ms/step - loss: 0.6131 - accuracy: 0.8959
> 89.590
```

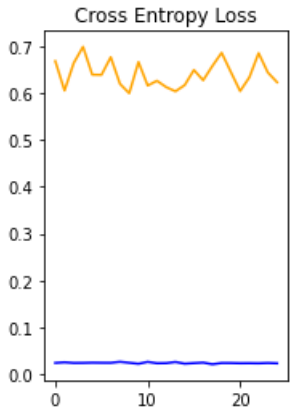


In [37]:

```
# entry point, run the test harness for 25 * 8 iterations, 8th slot
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0243 - accuracy: 0.9915 - va
l_loss: 0.6682 - val_accuracy: 0.8857
Epoch 2/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0255 - accuracy: 0.9911 - va
l_loss: 0.6053 - val_accuracy: 0.8943
Epoch 3/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0242 - accuracy: 0.9914 - va
l_loss: 0.6644 - val_accuracy: 0.8916
Epoch 4/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0243 - accuracy: 0.9915 - va
l_loss: 0.6988 - val_accuracy: 0.8882
Epoch 5/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0248 - accuracy: 0.9918 - va
l_loss: 0.6393 - val_accuracy: 0.8927
Epoch 6/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0245 - accuracy: 0.9914 - va
l_loss: 0.6389 - val_accuracy: 0.8947
Epoch 7/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0245 - accuracy: 0.9913 - va
l_loss: 0.6766 - val_accuracy: 0.8863
Epoch 8/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0266 - accuracy: 0.9907 - va
l_loss: 0.6196 - val_accuracy: 0.8919
Epoch 9/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0245 - accuracy: 0.9918 - va
l_loss: 0.5995 - val_accuracy: 0.8934
Epoch 10/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0224 - accuracy: 0.9921 - va
l_loss: 0.6663 - val_accuracy: 0.8884
Epoch 11/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0265 - accuracy: 0.9913 - va
l_loss: 0.6160 - val_accuracy: 0.8922
Epoch 12/25
```

```
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0235 - accuracy: 0.9919 - va
l_loss: 0.6263 - val_accuracy: 0.8947
Epoch 13/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0237 - accuracy: 0.9918 - va
l_loss: 0.6127 - val_accuracy: 0.8968
Epoch 14/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0262 - accuracy: 0.9909 - va
l_loss: 0.6037 - val_accuracy: 0.8960
Epoch 15/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0226 - accuracy: 0.9922 - va
l_loss: 0.6169 - val_accuracy: 0.9000
Epoch 16/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0238 - accuracy: 0.9918 - va
l_loss: 0.6493 - val_accuracy: 0.8910
Epoch 17/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0249 - accuracy: 0.9913 - va
l_loss: 0.6273 - val_accuracy: 0.8899
Epoch 18/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0214 - accuracy: 0.9926 - va
l_loss: 0.6578 - val_accuracy: 0.8895
Epoch 19/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0241 - accuracy: 0.9914 - va
l_loss: 0.6861 - val_accuracy: 0.8858
Epoch 20/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0240 - accuracy: 0.9920 - va
l_loss: 0.6449 - val_accuracy: 0.8936
Epoch 21/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0235 - accuracy: 0.9921 - va
l_loss: 0.6042 - val_accuracy: 0.8946
Epoch 22/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0236 - accuracy: 0.9919 - va
l_loss: 0.6336 - val_accuracy: 0.8888
Epoch 23/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0234 - accuracy: 0.9915 - va
l_loss: 0.6853 - val_accuracy: 0.8849
Epoch 24/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0241 - accuracy: 0.9918 - va
l_loss: 0.6438 - val_accuracy: 0.8914
Epoch 25/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0234 - accuracy: 0.9916 - va
l_loss: 0.6231 - val_accuracy: 0.8932
313/313 [==============================] - 5s 16ms/step - loss: 0.6231 - accuracy: 0.8932
> 89.320
```



In [38]:

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

```
Saved model to disk
```

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

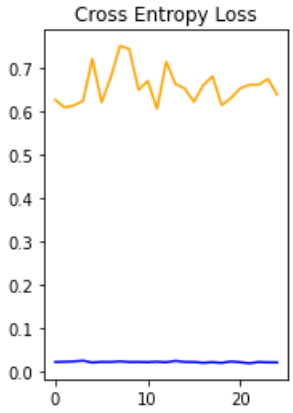
and Train from that epoch

In [39]:

```
# entry point, run the test harness
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 117s 92ms/step - loss: 0.0218 - accuracy: 0.9923 - va
l_loss: 0.6264 - val_accuracy: 0.8953
Epoch 2/25
1282/1282 [==============================] - 120s 93ms/step - loss: 0.0226 - accuracy: 0.9921 - va
l_loss: 0.6088 - val_accuracy: 0.8963
Epoch 3/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0233 - accuracy: 0.9921 - va
l_loss: 0.6136 - val_accuracy: 0.8960
Epoch 4/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0250 - accuracy: 0.9912 - va
l_loss: 0.6241 - val_accuracy: 0.8925
Epoch 5/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0206 - accuracy: 0.9925 - va
l_loss: 0.7215 - val_accuracy: 0.8810
Epoch 6/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0222 - accuracy: 0.9919 - va
l_loss: 0.6209 - val_accuracy: 0.8916
Epoch 7/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0220 - accuracy: 0.9925 - va
l_loss: 0.6788 - val_accuracy: 0.8928
Epoch 8/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0234 - accuracy: 0.9922 - va
l_loss: 0.7513 - val_accuracy: 0.8795
Epoch 9/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0219 - accuracy: 0.9926 - va
l_loss: 0.7435 - val_accuracy: 0.8785
Epoch 10/25
1282/1282 [==============================] - 121s 95ms/step - loss: 0.0220 - accuracy: 0.9924 - va
l_loss: 0.6497 - val_accuracy: 0.8896
Epoch 11/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0216 - accuracy: 0.9926 - va
l_loss: 0.6699 - val_accuracy: 0.8912
Epoch 12/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0226 - accuracy: 0.9925 - va
l_loss: 0.6055 - val_accuracy: 0.8988
Epoch 13/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0213 - accuracy: 0.9928 - va
l_loss: 0.7143 - val_accuracy: 0.8877
Epoch 14/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0245 - accuracy: 0.9918 - va
l_loss: 0.6632 - val_accuracy: 0.8891
Epoch 15/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0220 - accuracy: 0.9927 - va
l_loss: 0.6528 - val_accuracy: 0.8935
Epoch 16/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0219 - accuracy: 0.9926 - va
l_loss: 0.6218 - val_accuracy: 0.8942
Epoch 17/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0201 - accuracy: 0.9932 - va
l_loss: 0.6603 - val_accuracy: 0.8897
Epoch 18/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0215 - accuracy: 0.9924 - va
l_loss: 0.6808 - val_accuracy: 0.8882
Epoch 19/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0198 - accuracy: 0.9933 - va
l_loss: 0.6142 - val_accuracy: 0.8993
Epoch 20/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0232 - accuracy: 0.9921 - va
l_loss: 0.6307 - val_accuracy: 0.8958
Epoch 21/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0214 - accuracy: 0.9927 - va
l_loss: 0.6532 - val_accuracy: 0.8918
Epoch 22/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0190 - accuracy: 0.9934 - va
l_loss: 0.6614 - val_accuracy: 0.8913
Epoch 23/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0219 - accuracy: 0.9923 - va
l_loss: 0.6615 - val_accuracy: 0.8926
```

```
l_loss: 0.6619 - val_accuracy: 0.8926
Epoch 24/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0211 - accuracy: 0.9928 - va
l_loss: 0.6747 - val_accuracy: 0.8855
Epoch 25/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0209 - accuracy: 0.9925 - va
l_loss: 0.6387 - val_accuracy: 0.8897
313/313 [==============================] - 5s 16ms/step - loss: 0.6387 - accuracy: 0.8897
> 88.970
```

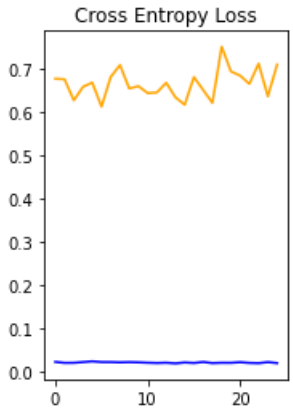


In [40]:

```
# entry point, run the test harness
model_harness(X_train, y_train, X_test, y_test, 60, 39, 25)
```

```
Epoch 1/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0218 - accuracy: 0.9925 - va
l_loss: 0.6757 - val_accuracy: 0.8907
Epoch 2/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0195 - accuracy: 0.9934 - va
l_loss: 0.6745 - val_accuracy: 0.8895
Epoch 3/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0197 - accuracy: 0.9929 - va
l_loss: 0.6257 - val_accuracy: 0.8967
Epoch 4/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0213 - accuracy: 0.9925 - va
l_loss: 0.6570 - val_accuracy: 0.8946
Epoch 5/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0231 - accuracy: 0.9919 - va
l_loss: 0.6668 - val_accuracy: 0.8933
Epoch 6/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0214 - accuracy: 0.9928 - va
l_loss: 0.6113 - val_accuracy: 0.8952
Epoch 7/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0214 - accuracy: 0.9929 - va
l_loss: 0.6800 - val_accuracy: 0.8897
Epoch 8/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0207 - accuracy: 0.9929 - va
l_loss: 0.7073 - val_accuracy: 0.8896
Epoch 9/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0213 - accuracy: 0.9928 - va
l_loss: 0.6532 - val_accuracy: 0.8942
Epoch 10/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0207 - accuracy: 0.9928 - va
l_loss: 0.6585 - val_accuracy: 0.8885
Epoch 11/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0199 - accuracy: 0.9936 - va
l_loss: 0.6424 - val_accuracy: 0.8955
Epoch 12/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0192 - accuracy: 0.9935 - va
l_loss: 0.6440 - val_accuracy: 0.8903
Epoch 13/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0199 - accuracy: 0.9931 - va
l_loss: 0.6663 - val_accuracy: 0.8928
Epoch 14/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0183 - accuracy: 0.9940 - va
l_loss: 0.6328 - val_accuracy: 0.8925
Epoch 15/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0205 - accuracy: 0.9924 - va
```

```
l_loss: 0.6156 - val_accuracy: 0.8969
Epoch 16/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0193 - accuracy: 0.9935 - va
l_loss: 0.6795 - val_accuracy: 0.8907
Epoch 17/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0216 - accuracy: 0.9927 - va
l_loss: 0.6497 - val_accuracy: 0.8944
Epoch 18/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0190 - accuracy: 0.9935 - va
l_loss: 0.6199 - val_accuracy: 0.8984
Epoch 19/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0198 - accuracy: 0.9934 - va
l_loss: 0.7498 - val_accuracy: 0.8850
Epoch 20/25
1282/1282 [==============================] - 120s 94ms/step - loss: 0.0197 - accuracy: 0.9932 - va
l_loss: 0.6927 - val_accuracy: 0.8908
Epoch 21/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0211 - accuracy: 0.9929 - va
l_loss: 0.6829 - val_accuracy: 0.8853
Epoch 22/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0194 - accuracy: 0.9936 - va
l_loss: 0.6638 - val_accuracy: 0.8933
Epoch 23/25
1282/1282 [==============================] - 121s 94ms/step - loss: 0.0188 - accuracy: 0.9934 - va
l_loss: 0.7107 - val_accuracy: 0.8870
Epoch 24/25
1282/1282 [==============================] - 122s 95ms/step - loss: 0.0213 - accuracy: 0.9927 - va
l_loss: 0.6348 - val_accuracy: 0.8933
Epoch 25/25
1282/1282 [==============================] - 123s 96ms/step - loss: 0.0190 - accuracy: 0.9936 - va
l_loss: 0.7084 - val_accuracy: 0.8873
313/313 [==============================] - 5s 16ms/step - loss: 0.7084 - accuracy: 0.8873
> 88.730
```



Summarize the model output

In [1]:

```
from prettytable import PrettyTable

table = PrettyTable()
table.field_names = ['Model', 'Epochs', 'Train Accuracy', 'Test Accuracy']
table.add_row(['With dense layer ', 75, 0.9478, 0.8603])
table.add_row(['Without dense layer', '300', 0.9936,  0.8873])

print(table)
```

```
+--------------------+--------+----------------+---------------+
|        Model       | Epochs | Train Accuracy | Test Accuracy |
+--------------------+--------+----------------+---------------+
|  With dense layer  |   75   |     0.9478     |     0.8603    |
| Without dense layer |  300   |     0.9936     |     0.8873    |
+--------------------+--------+----------------+---------------+
```