EE5353 Program Assignment 8

Coin Versus Scrap Recognition using Convolutional Neural Networks using Keras using Google Colab with Data augmentation

In this assignment we will classify Coin vs Scrap. The character images will be provided to you in the link below. **Students need to build code from scratch in this assignment. No code will be provided**

There are 2 parts to the assignment.

1. CNN without augmentation.
2. Second CNN with augmentation.

CNN structure will be the same. Augmentation code line will be provided.

Image information. Here training folder is **different** from testing folder

Training folder name – Training

Testing folder name – Testing

Each has 2 folders SCRAP and COIN

(trial and trial_test have less images from the training and the testing folder for testing)

Steps for this assignment

1. Input the images from the training folder in proper image and label format(use onehot encoding/to_categorical) as used in program 6 and 7
2. Divide the data into training (80%) and **validation(20%)**.
3. Resize the images to 200 by 200.
4. Convert the images to black and white
5. Normalize the input data
6. Design a Convolutional neural network like program 6,7 but some parameters have to be changed It should have the following layers
   - Convolutional layer with 64 filters, Size of the filters is 3, 3, Strides is 2 and relu activation
   - Pooling layer with pool size 2,2
   - Dropout layer with rate 0.5
   - Flattening
   - Dense layer fully connected with 128 hidden units and relu activations
   - Dropout layer with rate 0.5
   - Final dense fully connected layer with number of classes and softmax activation.

7. Verify if the number of iterations/nb_epochs = 20.
8. Validation data should be used during training(line of code will be provided)
9. Input the images from the testing folder in proper image and label format as used for training. Shuffle the testing data.
10. Test the images. Print results and testing figure

The link to the Coin vs scrap Images is below. Kindly download and put it in your google drive folder

➔ **https://drive.google.com/open?id=1Awfb88YS76yumiF0ZipvmRMWrrGHocMJ**

Code for including validation data in training part

```
model.fit(X_train, y_train, batch_size=32, nb_epoch=20, verbose=1,shuffle=False, validation_data = (X_validation, y_validation))
```

Code for including data augmentation

```
data_generator = ImageDataGenerator(vertical_flip=True,horizontal_flip=True)
data_generator.fit(X_train)
model.fit_generator(data_generator.flow(X_train, y_train, batch_size=32),steps_per_epoch=len(X_train) // 32, epochs=20,validation_data=(X_val, y_val),verbose=1)
```

change (`model.fit(X_train, y_train, batch_size=32`) used in program 6, 7 to above

Submission details

Both code with training and testing results separate.

Explanation – Please explain what different you did in this assignment from Program assignment 6 and 7

Conclusion (separate)

Links for the convolution layers, pooling, dense, dropout

https://keras.io/layers/convolutional/

https://keras.io/layers/core/ (this has dropout too)

https://keras.io/layers/pooling/

Insert the following line of code at the start of the assignment in hope to get exactly same results at every run.

(I was unsuccessful to achieve same results. Maybe it is my system try it and let me know if you could)

```
import tensorflow as tf
import random as rn
import os,cv2
import numpy as np

os.environ['PYTHONHASHSEED'] = '0'

# Setting the seed for numpy-generated random numbers
np.random.seed(37)

# Setting the seed for python random numbers
rn.seed(1254)

# Setting the graph-level random seed.
tf.set_random_seed(89)

from keras import backend as K

session_conf = tf.ConfigProto(
      intra_op_parallelism_threads=1,
      inter_op_parallelism_threads=1)

#Force Tensorflow to use a single thread
sess = tf.Session(graph=tf.get_default_graph(), config=session_conf)

K.set_session(sess)
```