ASSIGNMENT 1

Building your own CNN Image Classifier

This assignment will be focused on using the concept of transfer learning on a pre-trained CNN to build a dog/cat classifier. By the end of this assignment, one should have a clear idea of how to use CNN for Image classification from scratch.

The final deliverable should consist of the following:

1. A JNB ipynb file consisting of the Python code used to build the classifier with the naming convention "CNN_(uncc_emailID)".
2. A word document with a stepwise description and supported screenshots explaining the steps undertaken to build the classifier. The naming convention for the same would be Doc_(uncc_emailID). For example: CNN_malam8 and Doc_malam8
3. Zip the two documents together and upload the folder with the naming convention Assignment1_(uncc_emailID) onto canvas. Example: Assignment1_malam8

Dataset:

Download the dataset from https://www.kaggle.com/c/dogs-vs-cats/data and click the "*Download All*" button. This will download the 850-megabyte file "*dogs-vs-cats.zip*" to your workstation.

Unzip the file and you will see *train.zip*, *train1.zip* and a *.csv* file. Unzip the *train.zip* file, as we will be focusing only on this dataset.

You will now have a folder called '*train/*"that contains 25,000 .jpg files of dogs and cats.

(If you do not have a Kaggle account, sign up first in order to download the dataset)

The following steps will be considered for grading purposes:

1. Dividing the above dataset with 25,000 images into train and test datasets.
2. Preprocess and augment the training data.
3. Perform transfer learning by using the various pre-trained models (VGG/MobileNetV2/ResNet) offered by Keras to use the new data on a pretrained model and training a classifier
4. Fine-tuning the model for improved accuracy.
5. Repeat steps 1-4, but this time, don't use the datagenerator or dataloader functions directly from Keras or Tensorflow. Instead, write your own custom-function called 'ImageLoader' that will take in images and apply on the fly augmentations for model training. It is mandatory to specify 3-4 augmentation methods.

6. Bonus marks will be given if you can show Tensorboard to track loss and accuracy visualization.
7. Provide explanation and justification for hyperparameters in the jupyter notebooks.
8. For model training, use google collab or personal computer.