National University
Of Computer and Emerging Sciences

Deep Learning

(MS-DS)

PROJECT 2B REPORT

Submitted By:

1. **FAHAD MALIK**
   18L-2097

SUBMITTED TO:

Sir. Usman Sadiq

DATE OF SUBMISSION:

17/12/2018

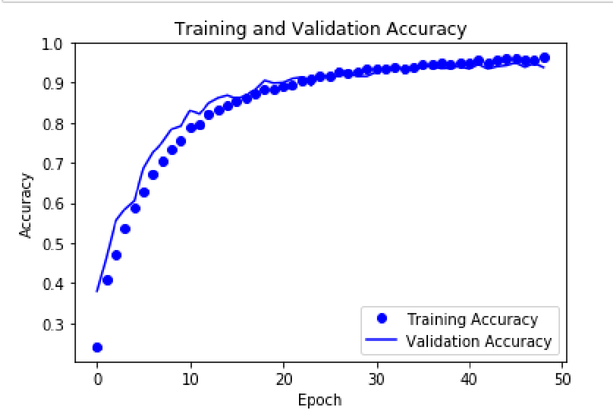
CLASSIFICATION:

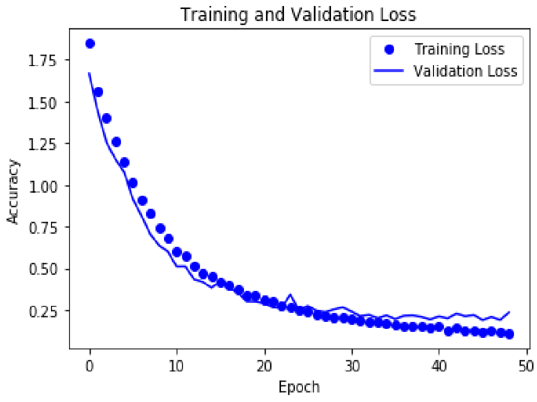
For the classification, I have used Alex Net model to classify it since it is simpler to implement.

For the classification to succeed,

- I have written code to pick images of each folder and rename it and store it in new folder.
- Since we are having less data in other folders as compared to 6th folder therefore I have used image generator to generate new images using a the provided images having shift of 2% and storing the generated images in the new folder.
- Then I have implemented Alex Net model to classify images with respect to classes which gives me top1 accuracy of almost 95% and a loss of approximately 4.23% within the working environment of my machine with total parameters of architecture of approximately 2 to 3 lac. Since it depends on the depth of each layer of the architecture.
- I have implemented VGG16 model as well which gives me top1 accuracy of approximately 85% with a loss of 14.253% loss keeping the architecture layers minimum to make it runnable.

Accuracy Vs Loss Plot:





The plots show that initially the model over fits but it converges after 30 epochs resulting in a uniform accuracy and loss plot.

Factors Effecting Accuracy:

The following factors/techniques improve accuracy;

- Removing the noisy data improves accuracy and avoids under/over fitting.
- Data imbalance is a major factor that effects the accuracy. Since the model shifts towards that class that has more data therefore each directory should contain equal data. To remove imbalance, data generator function is used to generate new data given the previous data and store those images in the directory.
- After removing imbalance query, the class labels are one hot encoded to increase computation speed since the one hot encoding transforms the categorical/nominal labels to row vectors that are easy to use in processing.
- Decreasing the batch size and image size improves accuracy.
- Adding dense layers increases the model authority to learn new features in the data and learn from it. This technique increases the accuracy and a smooth plot is achieved.
- Using Alex Net provides me an ease to run it on my machine with less parameters since my machine gets stuck when using VGG 16 or inception model. Therefore Alex Net is comfortable for my machine.
- After 35 epochs, the model converges till 90-93% but running it till 49 epochs, the accuracy results in 93-95% with its relative loss.

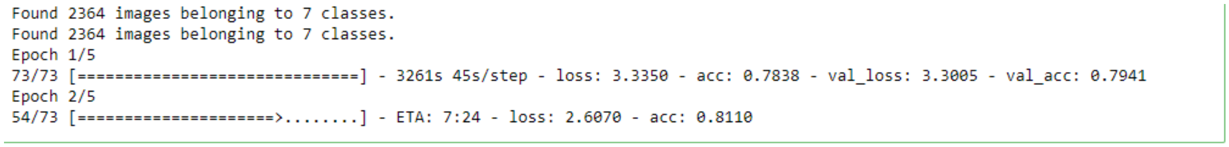
TRANSFER LEARNING:

For transfer learning ResNet50 model is being used in which the number of epochs are set to 5, input shape is 224x224 since decreasing the input shape decreases accuracy along with categorical classification of labels.

The first 141 layers are kept fixed and the rest of the layer retrained to transfer the learning with 7 class labels.

Each epoch takes approximately 30 mins or above on my machine therefore I have tested it for just two epochs and the accuracy is above 80% for the first 2 epochs since running it on large epochs freeze’s my machine and it gets stuck due to cpu and disk limit.

The transfer learning on ResNet50 model is good since its first epoch gives accuracy close to 80% therefore it starts with a good one.



```
Found 2364 images belonging to 7 classes.
Found 2364 images belonging to 7 classes.
Epoch 1/5
73/73 [==============================] - 3261s 45s/step - loss: 3.3350 - acc: 0.7838 - val_loss: 3.3005 - val_acc: 0.7941
Epoch 2/5
54/73 [=====================>........] - ETA: 7:24 - loss: 2.6070 - acc: 0.8110
```

I have also tried Google Net model but due to its parameters, memory issue is raised therefore I cannot implement that model on my machine.