Update_21st_March

1. Gave an inference run over 4 test images using the pre-trained model *nyu_5000_e15.h5* and observed the results. (the training was done on 5000 images, 15 epochs, lr=0.0001) The total inference time is - 15hrs 20min 42s. Images of the error metrics are provided below.

   (a) Inference results using approximate model in the **decoder.conv2** layer

```
time_taken for 300 conv in 1101 / 1104 = 12.338353872299194
time_taken for 300 conv in 1102 / 1104 = 12.370011568069458
time_taken for 300 conv in 1103 / 1104 = 12.308685541152954
100%|                                                  | 1/1 [15:20:42<00:00, 55242.71s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.6122,     0.8262,     0.8896,     0.2542,     0.7091,     0.1298

Test time 55243.421884298325 s
```

   (b) Inference results using accurate model

```
D/customConv_MBM/for_denseNet_161/Pytorch_for_161$ python evaluate_py.py
Loading test data...Test data loaded.

Testing...
100%|                                                  | 1/1 [00:01<00:00,  1.88s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.6026,     0.8197,     0.8884,     0.2543,     0.7175,     0.1322

Test time 1.8786110877990723 s
```

   It can be observed yet again that the error metric values are comparable between the accurate and the approximate model.

2. **Request for provision of access to TU Dresden's Server to perform Inference over larger batch of images:**

   While running the inference on my local machine, the time taken for 1 iteration increases by upto ~1-2 seconds even if I have to open a browser window and hence increasing the inference time significantly (~2.5 hrs). Also as we go deeper into the decoder layers to perform inference, the time for each iteration would increase.

   So if I am given access to the university server/ VM, I would be able to give inference runs for larger batches of images and not worry about increasing the inference time while working in parallel.

   As mentioned earlier in my Update_18th_March.pdf, use of Google Colab has been unreliable in my previous 3 runs.

3. The following are the required dependencies and packages I'm using, as of now, on my local machine to run inferences.
   a. Pytorch built from source - 1.9.0a0+gitdc29604

b. Torchvision built from source - 0.9.0a0+afc502b
c. Nvidia Driver version - 460.32.03
d. Cudatookit version - 10.2.89; cudnn version - 7.6.5

The complete installation procedure is provided in the ENVIRONMENT SETUP.pdf that I have provided as an update last week. If provided with access to the server I will be downloading the following dependencies and start the inference over larger batches of images.

4. **Error metrics definitions**:

Rel - average relative error = $1/n * \sum |y_p - y'_p| / y$

RMS - root mean squared error = $\sqrt{(1/n * \sum (y_p - y'_p)^2)}$

Log10 - average log error = $1/n * \sum |log_{10}(y_p) - log_{10}(y'_p)|$

a_i - Threshold accuracy = % of y_p s.t.

$max(y_p/y'_p, y'_p/y_p) = a_i < thr, for\ thr = 1.25, 1.25^2, 1.25^3$

The definitions are referred from here: https://arxiv.org/abs/1812.11941