15_to_18th_April_Update

01. The shift from Densenet-161 based encoder to MobileNet_v2 based encoder was done for the following reason:
    - The time taken for performing inference over images using DenseNet-161 as encoder was very high (~ 30hrs per image) due to the large number of computations to be performed.
    - In order to be able to run inference over all layers of architecture, a smaller and light-weight model was chosen - Mobilenetv2 which reduces the number of computations. (so as to be able to complete inferences in a practical time frame)
    - The shift was made on March 27th (As mentioned earlier in ***27_March_Update.pdf*** which has been uploaded in the dropbox already)

02. Server setup - packages, files and necessary commands:
    - Command to connect to vpn - */opt/cisco/anyconnect/bin/vpnui*

    - Enter necessary details to login to VPN and be able to access TUD server

    - Ssh into the TUD server -
      command = ssh -X username>+dom@<workstation>.pd.inf.tu-dresden.de

    - Copy the necessary project files from the local machine into the server using the ***scp*** command.

    - Copy the necessary environment setup files - shell.nix, condasetup_ProjName.sh, condaenv_ProjName.yml
      *The contents of these files have been uploaded on the dropbox (Pytorch_Env_Setup). Note that all the files were created only in the ~/tmp directory on the server.

    - Commands to install the required packages:
      Go to the directory in which the above 3 files are present and:
      > nix-shell
      > conda-shell
      > . ./condasetup_ProjName.sh

      To be able to run conda activate command the next time without building the conda environment from scratch each time, we can give the foll:
      > conda init bash

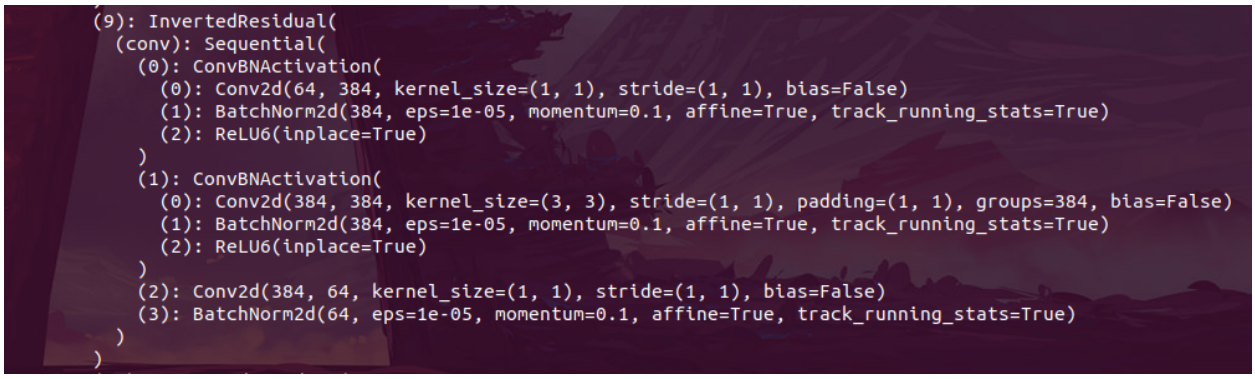
Now, for further logins, we can directly follow the steps to enter the desired conda environment:
> nix-shell
> conda-shell
> conda activate <path/to/folder>/CONDA_ENV

- Jupyter setup for easy editing/ testing:
  From remote: jupyter notebook --no-browser --port=8888
  From local : ssh -NL 8888:localhost:8888 <username>+dom@<workstation>.pd.inf.tu-dresden.de
  Open browser: Copy link as mentioned in URL

  To be able to determine if the preferred port is under use or not (for ssh/ listening to the port of server)
  > netstat -lep -tcp
  > sudo kill <PID> (of the required port)

03. Naming schemes for the model architecture:

A snippet of the Inverted residual block as defined in the model's state_dict() is provided below:



As it can be seen, this snippet describes the contents of the 9th Inverted Residual block.

- When referring to the 1st Conv2d layer within the block ,we can observe that it is present within (0): ConvBNActivation layer - hence it's naming scheme will be **9.conv.0.0**
- Similarly the 2nd Conv2d layer is within (1): ConvBNActivation layer - hence it's named as **9.conv.1.0**

- The 3rd conv2d layer within the block is not within any ConvBNActivation layer. Hence it's named as **9.conv.2**

The initial and final conv layers which aren't present inside a Inverted Residual block are named as **0.0** and **18.0** respectively.
In this manner all the layers are named in a shorthand format for easier reference.

04. **Inference Runs**

Inference runs over the following layers were given.

- 0.0 (Initial conv layer) - **local + gcolab**
- 1.conv.0.0 - **local**
- 2.conv.0.0 - **local**
- 2.conv.1.0 - **TUD server**
- 2.conv.2 - **TUD server**
- 3.conv.1.0 - **gcolab**
- 4.conv.1.0 - **gcolab**
- 5.conv.1.0 - **local**
- 6.conv.1.0 - **TUD server**
- 7.conv.1.0 - **gcolab**
- 8.conv.1.0 - **local**
- 9.conv.1.0 - **gcolab**
- 10.conv.1.0 - **local**
- 17.conv.2 - **gcolab**
- 18.0 (final layer) - **local + gcolab**

The details regarding image, kernel multiplier values and the error metrics results have been updated in the shared Google sheet under ***results-Encoder*** tab.