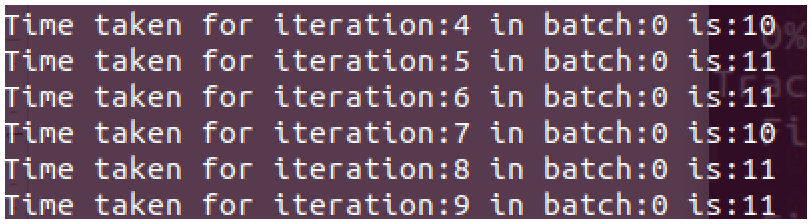
Update_23rd_March

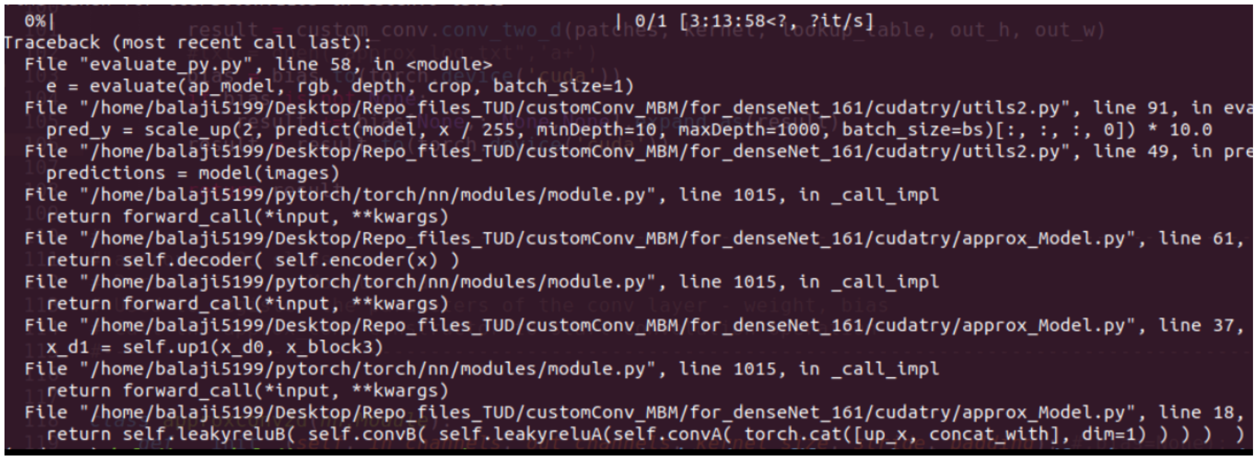
1. Currently working on trying to reduce the inference time as much as possible. Implemented custom convolution on c++ backend.
   Gave a sample run for inference over 1 image, to check the functionality.

   The inference time has now decreased to ~3hrs 14min from ~3hrs 45min and time taken for each iteration is ~10-11 sec on an average.



```
Time taken for iteration:4 in batch:0 is:10
Time taken for iteration:5 in batch:0 is:11
Time taken for iteration:6 in batch:0 is:11
Time taken for iteration:7 in batch:0 is:10
Time taken for iteration:8 in batch:0 is:11
Time taken for iteration:9 in batch:0 is:11
```

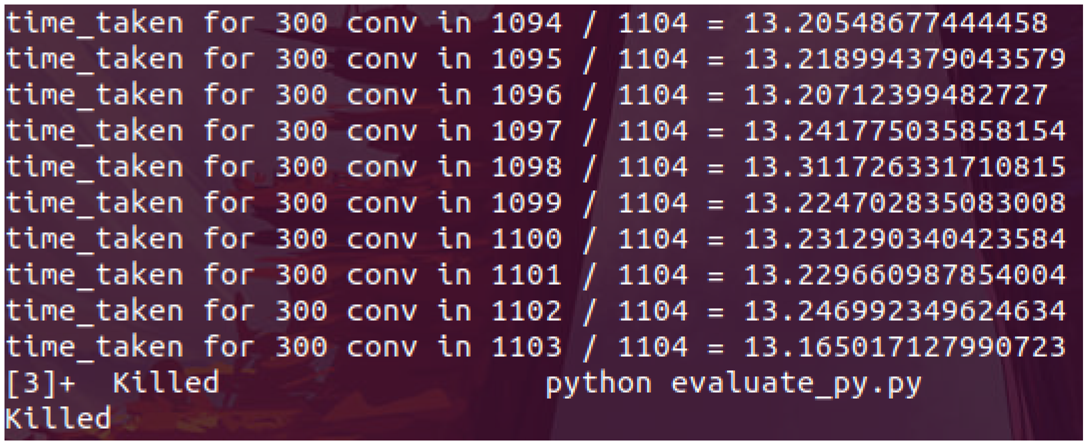
   After having given run over 1 image, I encountered some errors and am currently working on rectifying them.



```
  0%|                                                    | 0/1 [3:13:58<?, ?it/s]
Traceback (most recent call last):
  File "evaluate_py.py", line 58, in <module>
    e = evaluate(ap_model, rgb, depth, crop, batch_size=1)
  File "/home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/for_denseNet_161/cudatry/utils2.py", line 91, in eva
    pred_y = scale_up(2, predict(model, x / 255, minDepth=10, maxDepth=1000, batch_size=bs)[:, :, :, 0]) * 10.0
  File "/home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/for_denseNet_161/cudatry/utils2.py", line 49, in pre
    predictions = model(images)
  File "/home/balaji5199/pytorch/torch/nn/modules/module.py", line 1015, in _call_impl
    return forward_call(*input, **kwargs)
  File "/home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/for_denseNet_161/cudatry/approx_Model.py", line 61,
    return self.decoder( self.encoder(x) )
  File "/home/balaji5199/pytorch/torch/nn/modules/module.py", line 1015, in _call_impl
    return forward_call(*input, **kwargs)
  File "/home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/for_denseNet_161/cudatry/approx_Model.py", line 37,
    x_d1 = self.up1(x_d0, x_block3)
  File "/home/balaji5199/pytorch/torch/nn/modules/module.py", line 1015, in _call_impl
    return forward_call(*input, **kwargs)
  File "/home/balaji5199/Desktop/Repo_files_TUD/customConv_MBM/for_denseNet_161/cudatry/approx_Model.py", line 18,
    return self.leakyreluB( self.convB( self.leakyreluA(self.convA( torch.cat([up_x, concat_with], dim=1) ) ) )  )
```

   I have pushed the codes into the github repo under the subfolder ‘cudatry’.
   Kindly look into the same for your reference.

2. **Runtime Killed - Error:**
   I ran an inference over the next 4 images from the nyu_test dataset. After ~16 hours, the run got the following error:



```
time_taken for 300 conv in 1094 / 1104 = 13.20548677444458
time_taken for 300 conv in 1095 / 1104 = 13.218994379043579
time_taken for 300 conv in 1096 / 1104 = 13.20712399482727
time_taken for 300 conv in 1097 / 1104 = 13.241775035858154
time_taken for 300 conv in 1098 / 1104 = 13.311726331710815
time_taken for 300 conv in 1099 / 1104 = 13.224702835083008
time_taken for 300 conv in 1100 / 1104 = 13.231290340423584
time_taken for 300 conv in 1101 / 1104 = 13.229660987854004
time_taken for 300 conv in 1102 / 1104 = 13.246992349624634
time_taken for 300 conv in 1103 / 1104 = 13.165017127990723
[3]+  Killed                  python evaluate_py.py
Killed
```

This is probably due to browser tabs that I had left open alongside the run overnight which was taking up memory.

My reasoning behind running an inference over the next 4 images was to get the error metrics over them and take an average with the earlier results so that effectively we would have error metric results over 8 images.