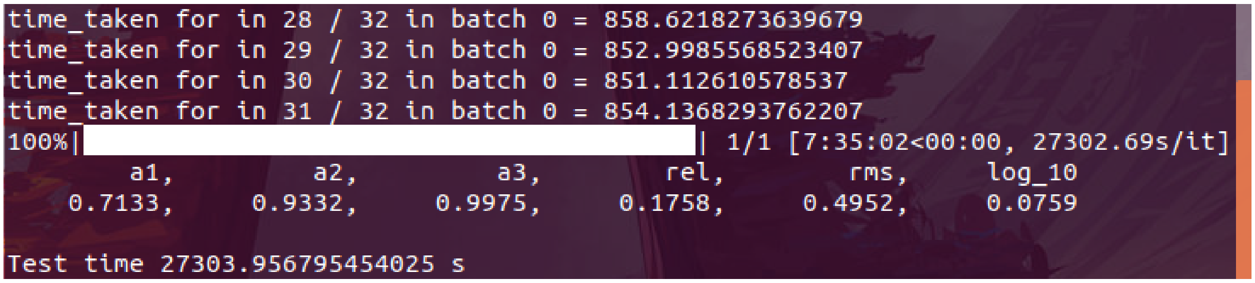
7_8_9th_April_Update

1. **Up5.convA - 5th image** (refer previous update for the 1st four images)



```
time_taken for in 28 / 32 in batch 0 = 858.6218273639679
time_taken for in 29 / 32 in batch 0 = 852.9985568523407
time_taken for in 30 / 32 in batch 0 = 851.112610578537
time_taken for in 31 / 32 in batch 0 = 854.1368293762207
100%|████████████████████████████████████████| 1/1 [7:35:02<00:00, 27302.69s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.7133,     0.9332,     0.9975,     0.1758,     0.4952,     0.0759

Test time 27303.956795454025 s
```

Taking the mean value of error metrics over 5 images gives the following avg error metric value over 5 images:

a1	**a2**	**a3**	**rel**	**rms**	**log10**
0.5946	0.8108	0.9188	0.2564	0.6199	0.1103

2. In order to improve the error metric results of the layers up1.convB and up2.convA, the kernel and image multiplier values were altered and inference runs were given once again over 1st five images from the NYUDepthv2 test dataset.

Before corrections were made to the kernel and image multiplier values

Layer	**a1 deviation**	**a2 deviation**	**a3 deviation**	**rel deviation**	**rms deviation**	**log10 deviation**
up1.convB	-45.60%	-7.26%	16.22%	169%	59%	82.79%
up2.convA	-82%	-30.18%	3.14%	265%	103%	130.53%

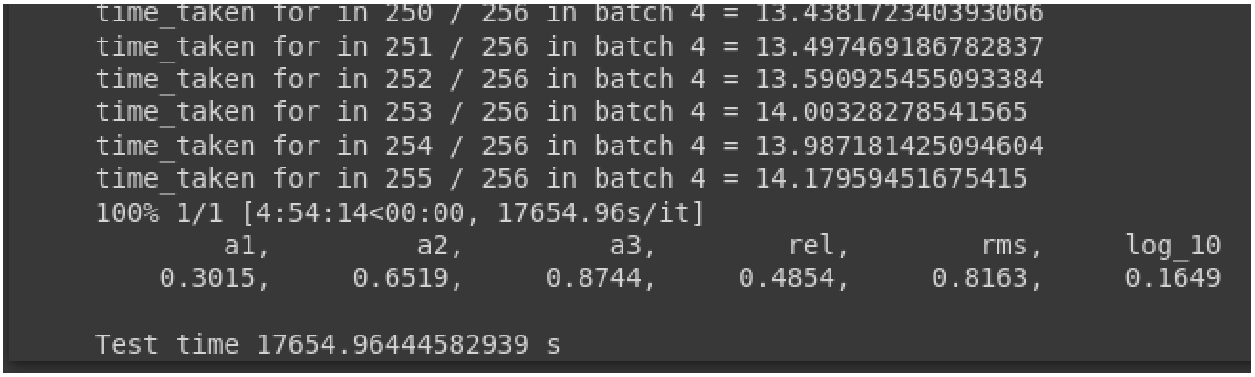
After corrections were made to kernel and image multipliers values

Layer	**a1 deviation**	**a2 deviation**	**a3 deviation**	**rel deviation**	**rms deviation**	**log10 deviation**
up1.convB	-39.28%	-4.30%	19.09%	135.56%	51.67%	73.03%
up2.convA	-28.21%	2.12%	23.04%	120.20%	38.08%	57.80%

It is observed that the deviations from expected values are lesser after applying correct image and kernel multiplier values.

3. **Inference run details**

decoder.up1.convB - google colab - (runtime = 4hrs 54 min)

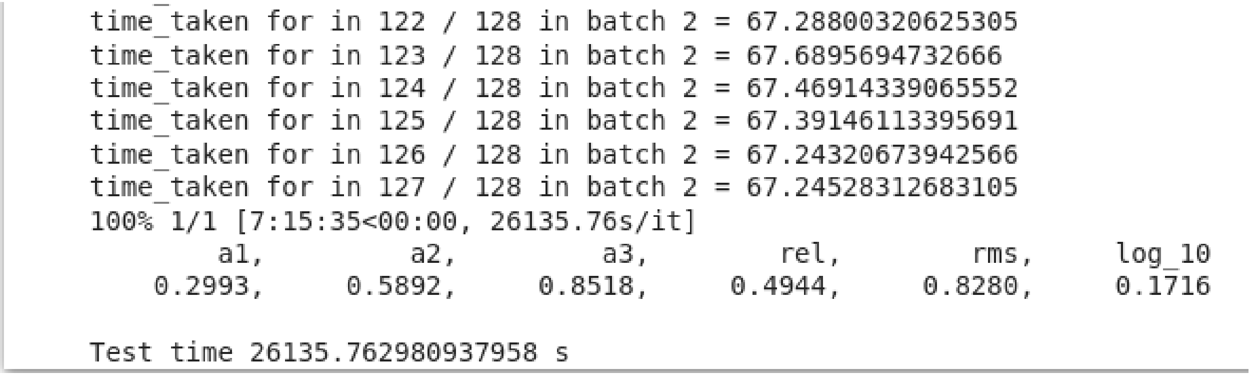


```
time_taken for in 250 / 256 in batch 4 = 13.438172340393066
time_taken for in 251 / 256 in batch 4 = 13.497469186782837
time_taken for in 252 / 256 in batch 4 = 13.590925455093384
time_taken for in 253 / 256 in batch 4 = 14.00328278541565
time_taken for in 254 / 256 in batch 4 = 13.987181425094604
time_taken for in 255 / 256 in batch 4 = 14.17959451675415
100% 1/1 [4:54:14<00:00, 17654.96s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.3015,     0.6519,     0.8744,     0.4854,     0.8163,     0.1649

Test time 17654.96444582939 s
```

decoder.up2.convA

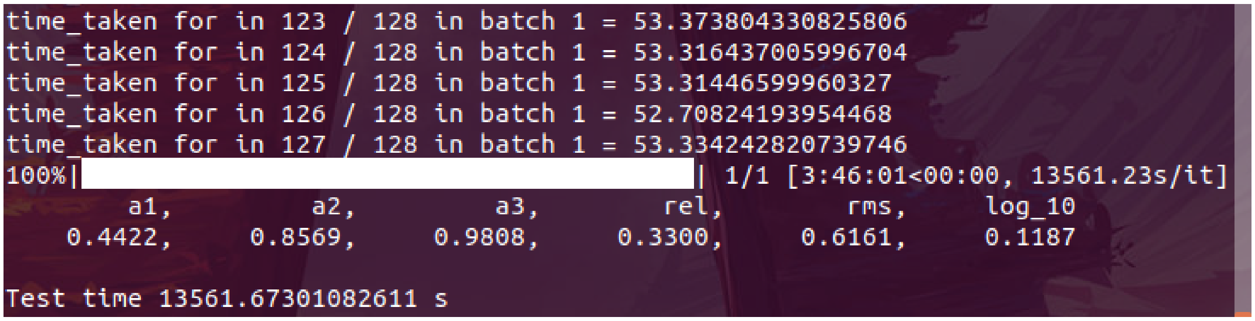
Img[0:3] - google colab - (runtime = 7hrs 15min)



```
time_taken for in 122 / 128 in batch 2 = 67.28800320625305
time_taken for in 123 / 128 in batch 2 = 67.6895694732666
time_taken for in 124 / 128 in batch 2 = 67.46914339065552
time_taken for in 125 / 128 in batch 2 = 67.39146113395691
time_taken for in 126 / 128 in batch 2 = 67.24320673942566
time_taken for in 127 / 128 in batch 2 = 67.24528312683105
100% 1/1 [7:15:35<00:00, 26135.76s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.2993,     0.5892,     0.8518,     0.4944,     0.8280,     0.1716

Test time 26135.762980937958 s
```

Img[3:5] - local machine - (runtime = 3hrs 46min)



```
time_taken for in 123 / 128 in batch 1 = 53.373804330825806
time_taken for in 124 / 128 in batch 1 = 53.316437005996704
time_taken for in 125 / 128 in batch 1 = 53.31446599960327
time_taken for in 126 / 128 in batch 1 = 52.70824193954468
time_taken for in 127 / 128 in batch 1 = 53.334242820739746
100%|████████████████████████████████████████| 1/1 [3:46:01<00:00, 13561.23s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.4422,     0.8569,     0.9808,     0.3300,     0.6161,     0.1187

Test time 13561.67301082611 s
```

The weighted mean for these error metric results were computed to obtain the avg error metric values for 5 images:

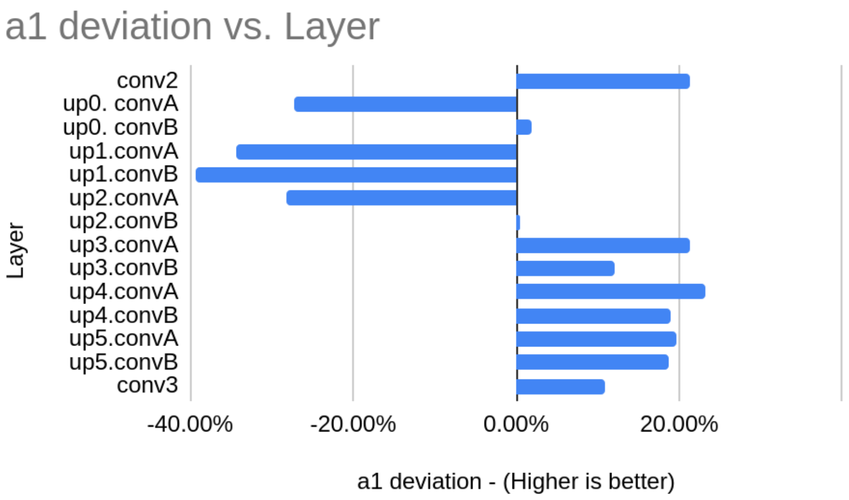
a1	a2	a3	rel	rms	log10
0.3565	0.6963	0.9034	0.4286	0.7432	0.1504

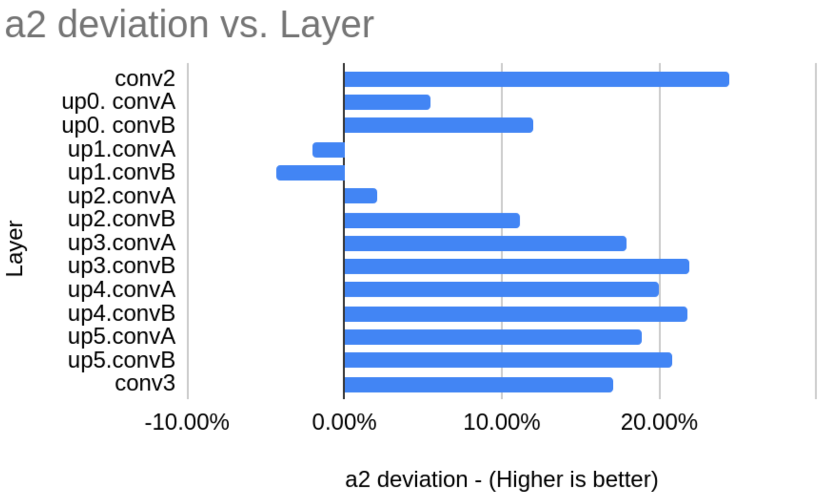
4. The amount of deviation from the *expected-result can be seen below.

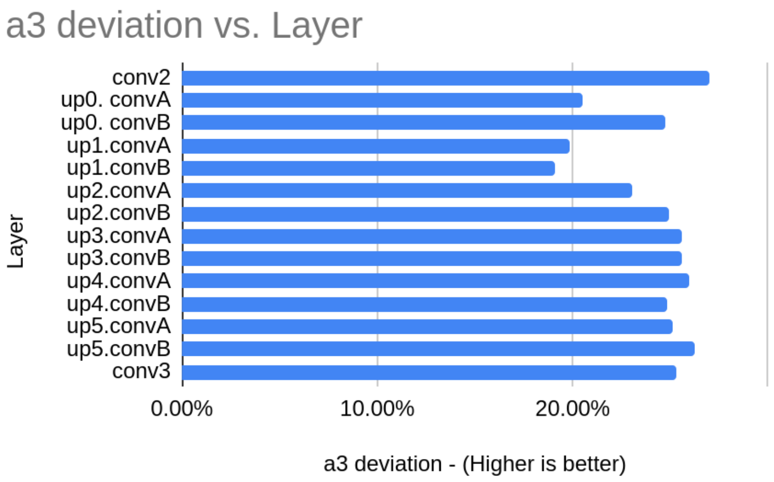
Layer	**a1 deviation**	**a2 deviation**	**a3 deviation**	**rel deviation**	**rms deviation**	**log10 deviation**
conv2	21.26%	24.39%	26.99%	22.61%	20.38%	21.93%
up0. convA	-27.14%	5.50%	20.49%	111.25%	41.76%	53.93%
up0. convB	1.83%	12.04%	24.77%	62.53%	22.96%	28.50%
up1.convA	-34.35%	-2.05%	19.83%	136.22%	46.09%	67.05%
up1.convB	-39.28%	-4.30%	19.09%	135.56%	51.67%	73.03%
up2.convA	-28.21%	2.12%	23.04%	120.20%	38.08%	57.80%
up2.convB	0.42%	11.20%	24.93%	69.73%	24.17%	30.22%
up3.convA	21.24%	17.96%	25.54%	36.28%	12.96%	14.48%
up3.convB	12.16%	21.91%	25.59%	24.71%	16.75%	18.15%
up4.convA	23.18%	19.96%	26.01%	29.13%	12.43%	13.01%
up4.convB	18.99%	21.79%	24.83%	36.07%	17.15%	18.36%
up5.convA	19.73%	18.92%	25.14%	31.75%	15.18%	15.73%
up5.convB	18.71%	20.81%	26.29%	32.32%	12.36%	13.54%
conv3	10.82%	17.03%	25.35%	44.55%	17.24%	20.67%

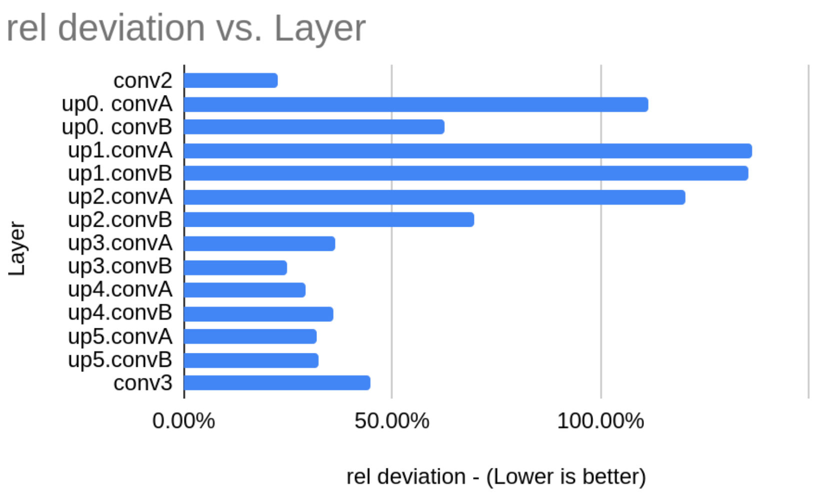
*expected-result values can be found here:
https://docs.google.com/spreadsheets/d/1tmXCuR8P1yGrYK8_bC07wBkrz_x6DhPl-a2OtFICZnw/edit?usp=sharing

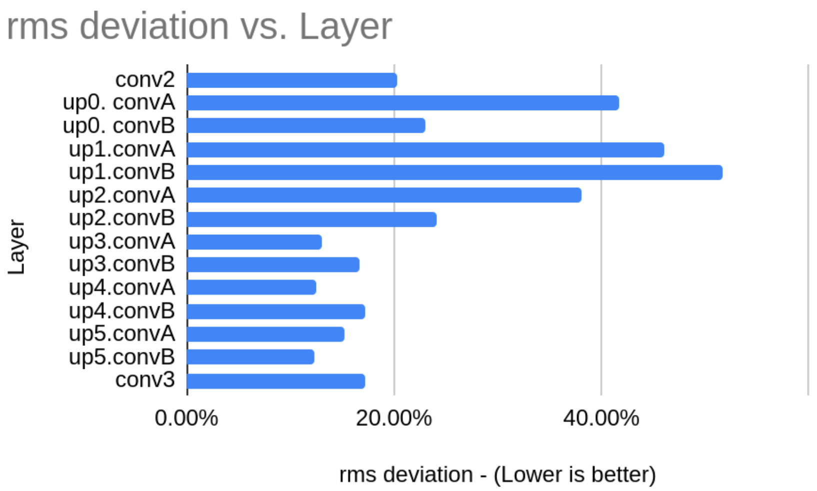
Layer vs Error-metric bar graphs

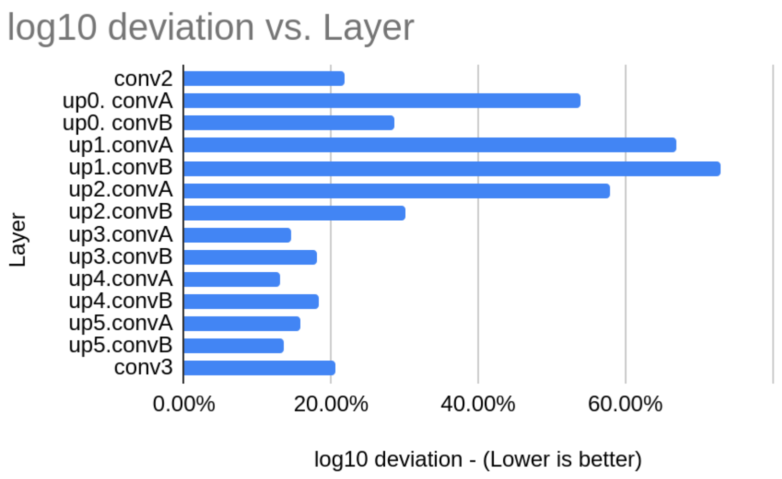


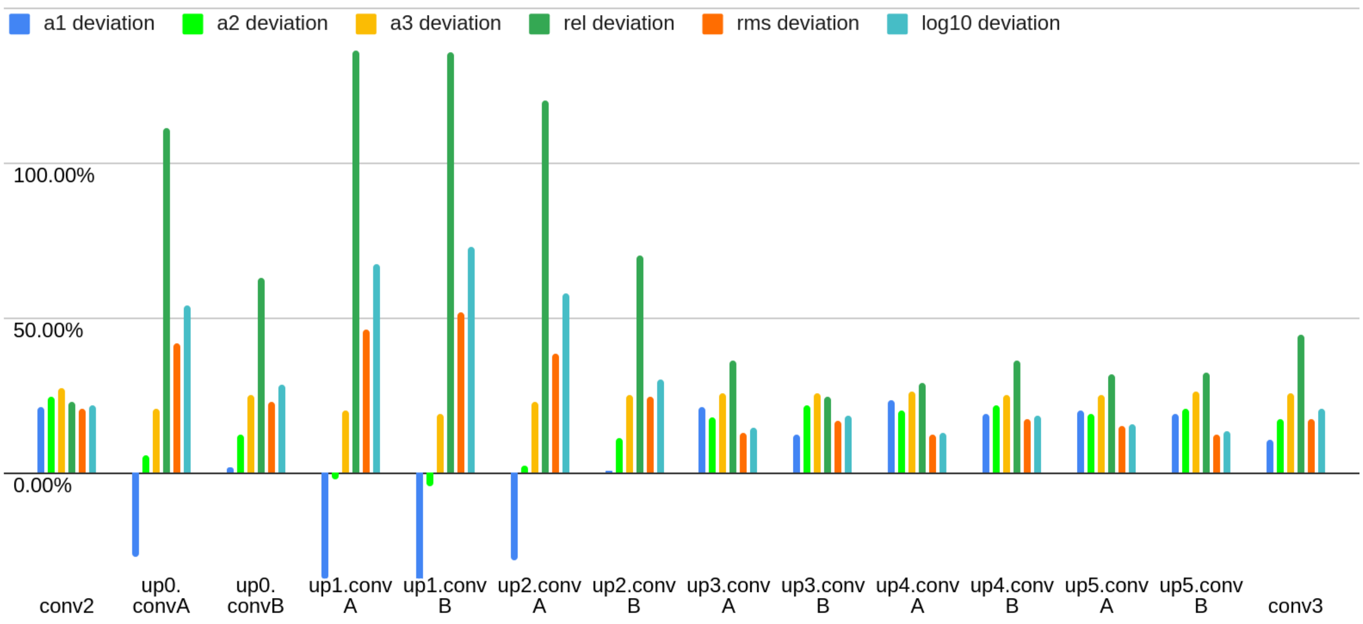





rel deviation vs. Layer
Layer
conv2
up0. convA
up0. convB
up1.convA
up1.convB
up2.convA
up2.convB
up3.convA
up3.convB
up4.convA
up4.convB
up5.convA
up5.convB
conv3
0.00%
50.00%
100.00%
rel deviation - (Lower is better)


rms deviation vs. Layer
Layer
conv2
up0. convA
up0. convB
up1.convA
up1.convB
up2.convA
up2.convB
up3.convA
up3.convB
up4.convA
up4.convB
up5.convA
up5.convB
conv3
0.00%
20.00%
40.00%
rms deviation - (Lower is better)


log10 deviation vs. Layer
Layer
conv2
up0. convA
up0. convB
up1.convA
up1.convB
up2.convA
up2.convB
up3.convA
up3.convB
up4.convA
up4.convB
up5.convA
up5.convB
conv3
0.00%
20.00%
40.00%
60.00%
log10 deviation - (Lower is better)



Observations:

1. In most of the layers, there is an increase in accuracy in the error metrics a1, a2, a3 (higher is better). In the layers up0.convA, up1.convA, up1.convB and up2.convA, there is a decrease in a1 error metric value than expected.

2. In all the decoder layers, after performing approximate MBM multiplication lookup, there is a decrease in accuracy with respect to the rel, rms and log10 error metric values than expected (lower is better). However, in all the layers (except *SET*) the decrease in rms, rel and log10 accuracy can be seen paired with increased accuracy in a1, a2 and s3 error metrics.

3. The reduced error metrics values in the *SET* can either be due to -
   1) Need for further calibration/ trials with the image and kernel multiplier values before inference,
   2) The feature maps of input image (over which the approximate convolution is applied) in these layers may be more sensitive to the approximation than other layers.

   ***SET = {up1.convA, up1.convB, up2.convA}**

5. Server environment setup

The environment setup was done on the Athena workstation as described in the previous update.