April_1_2_3_Update

The inference runs on the following layers were given on the respective days as mentioned below:

1. The layers with relatively less inference time (~51 min per image) were initially chosen on this day. (ie *up0.convB* and *up1.convB* layers). Followed by a layer which has large inference time was given a run overnight (*up2.convA* layer)

    a. **decoder.up0.convB (runtime = 5hrs 59min for 5 imgs)**

```
time_taken for in 252 / 256 in batch 4 = 17.16431999206543
time_taken for in 253 / 256 in batch 4 = 17.259791374206543
time_taken for in 254 / 256 in batch 4 = 17.287111043930054
time_taken for in 255 / 256 in batch 4 = 17.238867044448853
100% 1/1 [5:58:49<00:00, 21529.22s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.5057,     0.7639,     0.9161,     0.3163,     0.6618,     0.1225

Test time 21529.22363805771 s
```

    b. **decoder.up1.convB (runtime = 6hrs 52 min for 5 images)**

```
time_taken for in 252 / 256 in batch 4 = 18.530148744583l3
time_taken for in 253 / 256 in batch 4 = 19.46601390838623
time_taken for in 254 / 256 in batch 4 = 18.50220489501953
time_taken for in 255 / 256 in batch 4 = 18.455303192138672
100% 1/1 [6:52:50<00:00, 24770.46s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.2697,     0.6323,     0.8533,     0.5238,     0.8574,     0.1742

Test time 24770.46419596672 s
```

    c. **decoder.up2.convA (runtime = 10hrs 5 min for 5 images)**

```
Time taken for iteration:126 / 128 in batch:4 is:63
Time taken for iteration:127 / 128 in batch:4 is:63
100%|
                | 1/1 [10:05:33<00:00, 36333.70s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.0912,     0.4760,     0.7573,     0.7103,     1.0929,     0.2197

Test time 36333.6978700161 s
```

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%

*deviation = change in the observed result from the expected result.

The deviation from expected results were too high compared to other layers such as conv2 and conv3. Hence a better way of assigning image and kernel multipliers was implemented

2. Incorporating multiplier values in an effective manner

    a. **Method to determine scaling factors**
    The *torch.mean*, *torch.max*, *torch.min* functions were used to find the max, min and mean of the values present in the image and kernel tensors. In this manner it is easier to decide the kernel and image multiplier coefficients. *(This is done in order to scale up the values in the range of 0-255 so that MBM lookup can be performed, and after computation it is scaled down)*.

    For example, for the layer *decoder.up3.convA,* the following were determined:

    For kernel: max = 0.1207, min = 2.0177 * 10^{-7},mean = 0.0147

    For image: max = 142.98, min = 3.8 * 10^{-7}, mean = 8.04

    Hence, the kernel multiplier = 1000, image multiplier = 1

    Reasoning: It is more accurate to round off values in the range of $10^{-1}\ to\ 10^{-7}$ to 0 than to truncate larger values to 255 (because the range in the lookup table is 0 - 255)

    In this similar fashion, the multiplier/ scaling values were determined for the following layers, and is mentioned in the excel sheet.

    b. Inference -> decoder.up3.convA
    The inference was split into 3 different runs as following:
    1) image [0:2] - (2 images) - google colab (runtime = 4hrs 44 min)

```
time_taken for in 61 / 64 in batch 1 = 133.00910758972168
time_taken for in 62 / 64 in batch 1 = 133.0547318458557
time_taken for in 63 / 64 in batch 1 = 132.798574924469
100% 1/1 [4:44:16<00:00, 17056.45s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.4425,     0.7167,     0.9468,     0.3592,     0.5876,     0.1332

Test time 17056.452590942383 s
```

2) image [2:4] - (2 images) - google colab (runtime = 4hr 46 min)

```
time_taken for in 60 / 64 in batch 1 = 134.3627953529358
time_taken for in 61 / 64 in batch 1 = 136.28779339790344
time_taken for in 62 / 64 in batch 1 = 134.7961995601654
time_taken for in 63 / 64 in batch 1 = 135.15464615821838
100% 1/1 [4:46:15<00:00, 17175.34s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.7135,     0.8164,     0.8580,     0.2128,     0.6903,     0.1016

Test time 17175.33689045906 s
```

3) image [4:5] - (1 image) - local machine (runtime = 1hr 49 min)

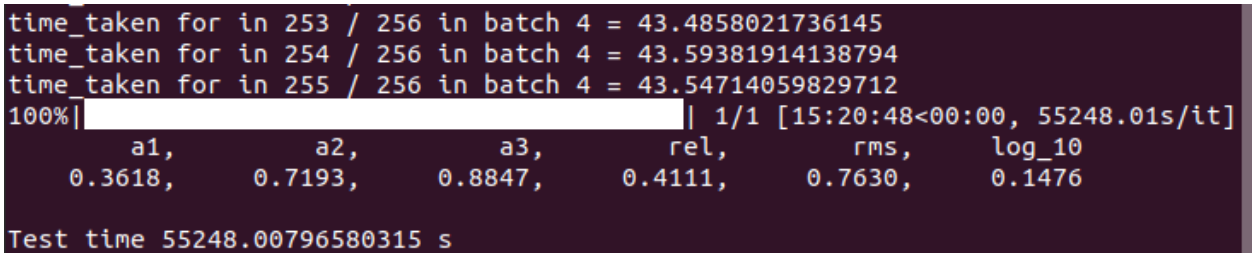
```
time_taken for in 61 / 64 in batch 0 = 102.52170968055725
time_taken for in 62 / 64 in batch 0 = 103.61117148399353
time_taken for in 63 / 64 in batch 0 = 103.8316855430603
100%|                                                  | 1/1 [1:49:52<00:00, 6592.66s/it]
        a1,         a2,         a3,        rel,        rms,     log_10
    0.6984,     0.9555,     0.9990,     0.1815,     0.4842,     0.0759

Test time 6592.6598563194275 s
```

The weighted mean of these three runs were taken and the result is as mentioned below (which is also updated in the excel sheet)

a1	a2	a3	rel	rms	log10
0.6021	0.8043	0.9217	0.2651	0.608	0.1091

c. Inference -> decoder.up0.convA (runtime = 15 hrs 20 min)

The inference run was given overnight. (April 2nd - 3rd)



The google sheet where the updates are given on a daily basis:
https://docs.google.com/spreadsheets/d/1tmXCuR8P1yGrYK8_bC07wBkrz_x6DhPl-a2OtFICZnw/edit?usp=sharing

The github repo: (for google colab inference run script)
https://github.com/bALAJi-aDItHYa/MBM_implementation.git