# 23rd_April_meet_updates

1. **Reason for switch from DenseNet-161 to Mobilenetv2 encoder based model:**

   In Upsample_1.convA layer, the time taken for 1 iteration = ~320s
   => time taken for inference over 1 image = 552 * 320 = ~49 hrs > 2 days!
   => time taken for inference over 5 images = ~245 hrs = ~10 days

```
Loading test data...Test data loaded.

Testing...
  0%|                                                                | 0/1 [00:00<?, ?it/s]
time_taken for iter 0 / 552 = 320.8581383228302
time_taken for iter 1 / 552 = 328.3324122428894
time_taken for iter 2 / 552 = 334.26970767974854
```

   Also, as we go further into the decoder layers, the number of MAC operations only increase (as the no. of convolutions increase). There for further layers, it would have taken even more time to perform inference over 5 images.

   Hence, I decided to shift to a model which had a smaller encoder architecture, so that the inference can be done in a practical time-frame.

   Hence, the Mobilenet_v2 model was chosen.

2. **Which activation function is used in Mobilenetv2 encoder and why?**
   ReLU6 is used as the activation function in MobileNEtv2 encoder as it is robust when used with low-precision computation.

   This is significant because Mobilenetv2 architecture is made for edge devices, in which memory has to be used judiciously and the weights are always quantised to lower precision to reduce the trained model size.

   Link to paper: https://arxiv.org/abs/1801.04381 - MobileNet_v2 paper (also shared in dropbox folder)

3. **Color coding of output to represent depth values.**
   The matplotlib's get_cmap function is used to map the output result to a 'Perceptually Uniform Sequential colormap" which is frequently used to represent data that have an ordering.

In this case, the ordering is in the depth values of the predicted output. So the get_cmap function makes use of the Plasma colormap to map the output tensor values to its range.


The progression from darker to lighter shades represents a shift from low-to-high values.

Reference: https://matplotlib.org/2.0.2/examples/color/colormaps_reference.html

4. **HLS Reference paper - for custom Depthwise, Pointwise and Standard Convolutions**
   https://kth.diva-portal.org/smash/get/diva2:1394764/FULLTEXT01.pdf