**Approach:**

Problem : Semantinc segmentation with multiclass

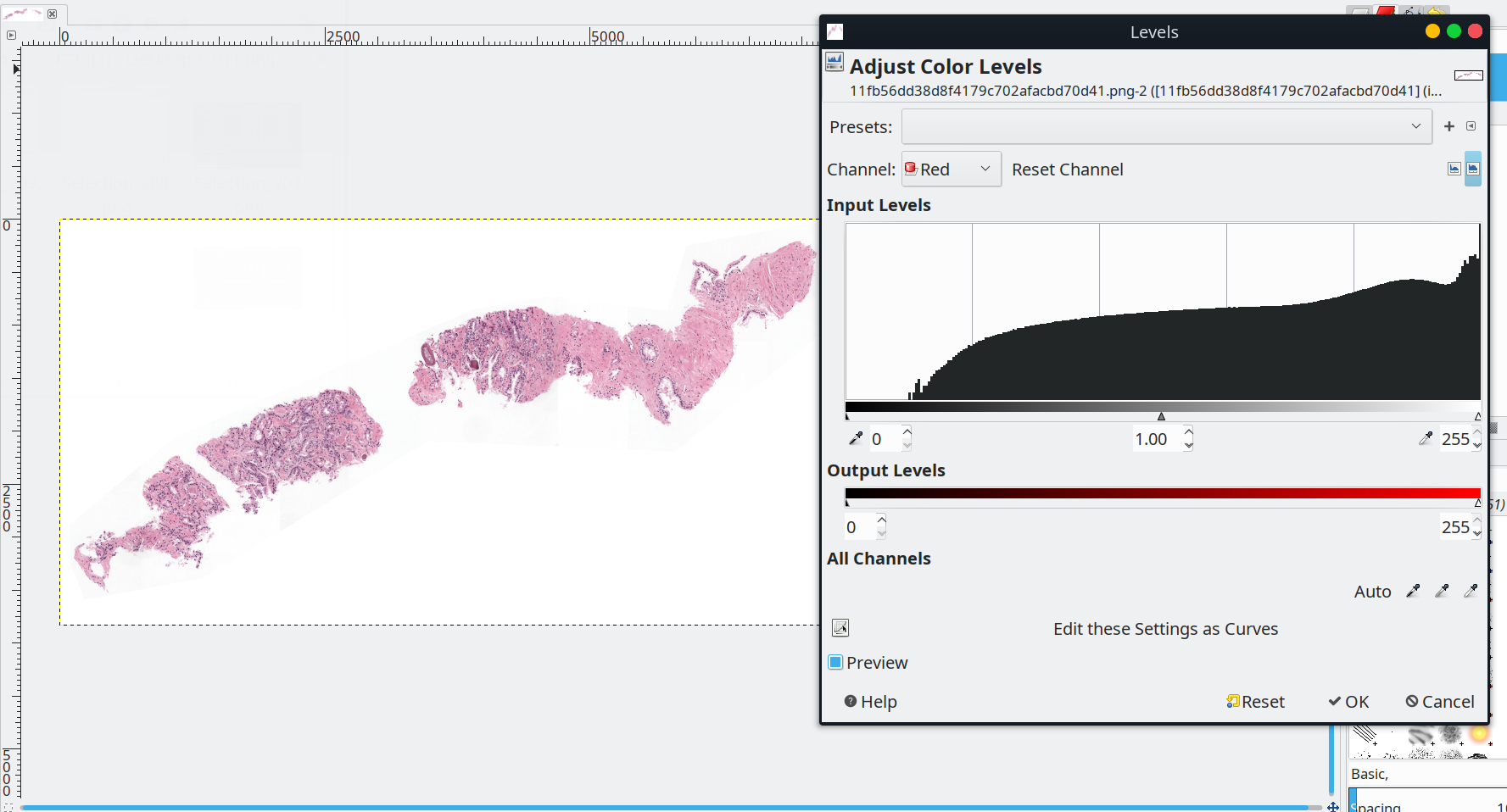
My image segmentation study notes: <https://www.notion.so/Image-Segmentation-29b80b8e00e04c53adc6e25a3abc1a01>

Model link: <https://drive.google.com/file/d/1aaaI3PyuEYZjzwJ6HpSCBFuJGv5eth0I/view?usp=sharing>

Colab file: <https://drive.google.com/file/d/1F58mb83-zu9SGOH2JgR7liy6R3JDdRNn/view?usp=sharing>

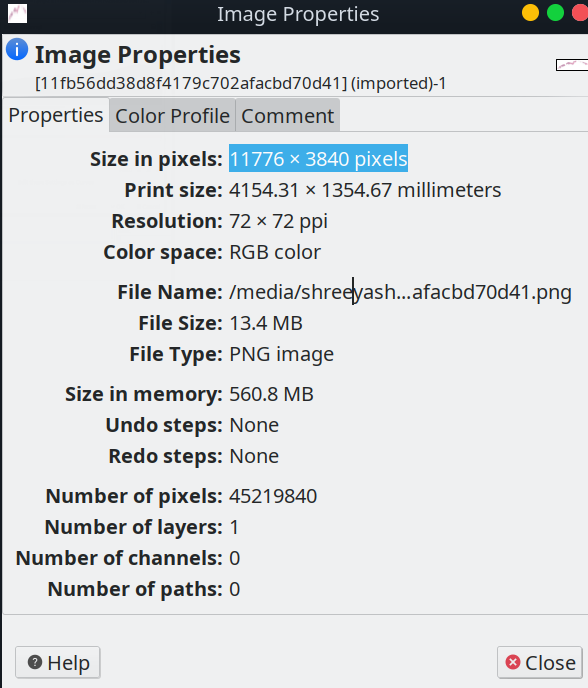
Patched dataset: <https://drive.google.com/file/d/1cGaWHClVC7cSx5L-3zR3rdfdyT3XOiT2/view?usp=sharing> (Train: 4376, Val: 5281)

First did the qualitative image analysis, using handy GIMP



Played with histogram values for RGB. Observed thresholding effects using contrast enchancement.

Noticed that images are very hi-resolution:



Hence the fist appraoch was to use sliding window pre-processing operation.

Used patchify to create 254x254 non-overlapping patches.

Since one of first times dealing with such kind segmentation of dataset, 80% of time was spent on data pipelines.

Since we had to get the one segmentation map per class, we had to defined our ground truth the same.

Used extensive transformation operations as given in notebook attached.

Assumed that the backround class and “black” class was same for now.

Since less data, used a UNET with pre-trained ResNet32 encoder. And followed the transfer learning approach. Will also be trying more models in future.

For the best model, on **validation set**:

dice\_loss = 0.09517

**dice\_coeff = 0.9048**

**IOU/Jaccard = 0.8693**

More ToDo:

1. Reguralisations: Augementations, more dropouts etc.
2. GAP to remove size constraints
3. Cropping as preprocessing step (remoe the white backround)
4. Better patching algorithm (can see squares in output)
5. Backround black different from class "black", especially image 2 in training
6. Class balance using class\_weight
7. Biomedical pretrained weights
8. LR schedular

More models to try:

<https://github.com/PingoLH/FCHarDNet>

<https://github.com/DengPingFan/PraNet>

<https://github.com/jnkl314/DeepLabV3FineTuning>

<https://github.com/fregu856/deeplabv3>

<https://github.com/YudeWang/deeplabv3plus-pytorch>

<https://github.com/YudeWang/semantic-segmentation-codebase/tree/main/experiment/deeplabv3%2Bvoc>

Transformer+Unet: <https://github.com/Beckschen/TransUNet/blob/main/trainer.py>

Rest of info, per cell is given in attached jupyter notbook. Thank you for this amazing learning opportuninty!