BM582 - Homework 2

Emre Girgin - 2016400099

BM582 - Homework 2 **1**

Emre Girgin - 2016400099 **1**

Remarks: 2

Questions Answered: 3

Question 10: 3

Question 11: 4

Question 12: 5

Remarks:

- While creating Dataset II and Dataset IV, only the diagnostic slide of TCGA-42-2590 is used. The reason is, both the training and testing is done in the Colab, due to lack of personal GPU. Thus, the larger diagnostic slides of the other two cases fill the disk and it is impossible to work on them in the Colab environment. However, since the technique is the same, this modification does not affect the learning objectives of the homework.
- For question 8, the "result_dir" flag works as it is wanted to be. Thus any modification was not made on the test file and not included to the deliverables.
- For question 9, the index.html files only contain the paths to the results of the models. All the datasets contain thousands of images and it was impossible for me to deliver all of them. Thus I only included some of the results from the test I made but not all of them. You can find them under the folder named like "samples/dataset2/images_d2_c1". This naming represents the samples from dataset 2 as the output of the case 1.
- I also provided the notebooks I run on Colab to give you an intuition about what I did. I hope it helps you to understand the path I followed. The notebook called train is used to train four cases, and the dataset notebook is for testing. However, note that, the notebooks do not work when you run them in your environment because I mounted my google drive account to transfer trained models from one notebook to the other. The code is based on the data stored in the drive, so this is why the notebooks do not run in your environment. However, they are pretty straightforward.

Questions Answered:

- Question 10:

I made my modifications on the CASE1. The outputs of the modifications are compared with the CASE1.

To decrease the loss, I tried several configurations but none of them performed remarkably. Some of the results are listed below:

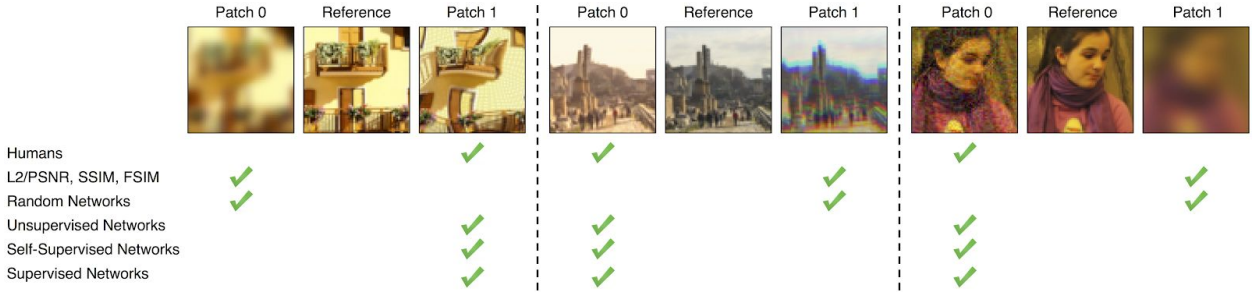
  - Learning rate : 0.001 - Did not changed
  - Norm : Batch - Did not changed
  - Norm : Batch, Lamda_L1 : 70 - Did not changed
  - Norm : Batch, Lamda_L1 : 70, Epochs : 200+200 = 400 - Slightly improvement in the G_L1 loss.

The one I included in the log_loss folder is the latest due to little improvement in G_L1 loss. **(log_loss_improved.txt)**

- Question 11:

In the NucleiSegmentatin/models/network.py file, under the GANLoss module (line 188) it is seen that MSE or BCE is used. This is for the discriminator. L1 loss is used for the generator. **1.** For the generator LPIPS loss can be used. Metrics like L1 can not determine the visual similarity between two images like humans do. For example in terms of MSE all the images below have the same MSE score. These pixel-wise losses are suitable for human level perceptual similarity.

On the other hand LPIPS measures the perceptual similarity. It uses a pre-trained feature extractor inside (AlexNet or VGG) and compares the embedding it obtained. Thus the similarity between real samples and synthetically generated fake samples means a lot. This allows us to generate more realistic samples.



2. For the discriminator it is a binary classification problem so the suggested metrics work well. However the other metrics like RMSE, Hinge Loss can be used similarly.

- Question 12:

To apply the loss I mentioned in Q11 (LPIPS), go to "**NucleiSegmentation/models/pix2pix_model.py**" file. Find the **line 96** and replace it with the following lines:

```
import lpips
loss_fn = lpips.LPIPS(net='alex')
self.loss_LPIPS = loss_fn.forward(self.fake_B, self.real_B)
self.loss_G = self.loss_G_GAN + self.loss_G_L1 + self.loss_LPIPS
```

This will calculate the perceptual similarity between "fake_B" and "real_B".

PS: Although the name suggest the similarity, the score is lower when two images are similar. Thus we do not subtract the LPIPS loss rather we add it.