

Deep Generative Models/Advanced Computer Vision Assignment 2 (100 points)

Deadline : until May. 27, 23:29

- Please download the data using the link below:

<https://drive.google.com/drive/folders/1--Dtsk9-j5V0Zo4FnvLkxZBwILfBks2C?usp=sharing>

- Please use the data in the `train` folder as the training data when training your model (e.g. in Problem 2) while using the data in the `test` folder as the testing data for evaluating your model (e.g. in Problem 1 and 2).
- Please download the network architecture, *.py and weight file(*.pth), contained in the link below:
<https://drive.google.com/drive/folders/1yOho8txY1QF3MQyxziQFajLmbad3dKel?usp=sharing>

If you are not using the CoLab, accompanying `requirements.txt` (or `environment.yaml`) is required with your submission. You can have the penalty if your code is not executable in the evaluator's PC, due to the environmental difference etc. Please use the CoLab to prevent this.

- If you have any questions, please post it on the blackboard/discussion/Q&A board.
 - You can write reports either in Korean or English.
1. **[20 points]** We provided the `Tester` class in main.py that initializes the network with the weights `model_30.pth` provided. Please do the following things for this problem.

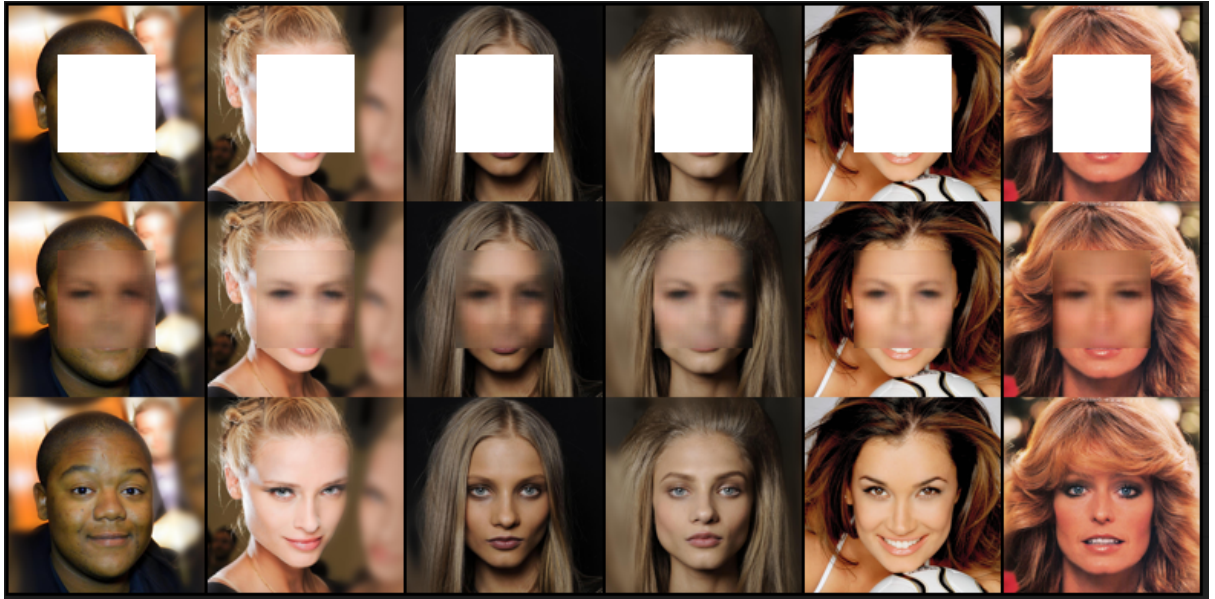
- Please upload datasets in your own environment (google drive is recommended).
- Please upload .pth files in your own environment (google drive is recommended).
- Please upload .py files in your local colab environment.
- Please confirm the folder path and settings in main.py.

```
from google.colab import drive
drive.mount('/content/gdrive')
```

Drive already mounted at /content/gdrive

```
[9] !python main.py
```

- You can use above codes for mounting the google drive and running the main.py. If you set all the things correctly, you can obtain the visualization in 'test.png' as below, for erased image, reconstructed image and original images in each row.



- Please attach plots for 0.png through 5.png located in the test folder in the report file (You can use 'test.png' obtained from 'save_img' method of 'Tester' Class for this). **[10 points]**
 - Please describe the testing/training methods of the context encoder in the report file. **[10 points]**
2. **[75 points]** 'Trainer' class is reserved for training your context encoder on the training data provided, when called from the main() function as follows:

```
trainer = Trainer(epochs, batchSize, learningRate)
trainer.train()
```

- Output of the network is the original image in the 'train' folder and input is the erased image.
- It uses the generator/discriminator architectures implemented in 'model.py'.
- It uses the dataloader implemented in FaceDataset class. It returns random-erased images for training (via 'apply_random_mask' method) while returns center-erased images for testing (via 'apply_center_mask' method).

- Please execute the initial training code and attach plots for 0.png through 5.png located in the 'test' folder in the report file using weight files obtained at 3 epoch and at 50 epoch, respectively. **[10 points]**

- The initial code implements the context encoder training only with pixel-wise loss (torch.nn.L1Loss()). Please extend the implementation by adding the adversarial loss

in the 'Trainer' class to train your network using both pixel-wise and adversarial losses. (Please properly tune your learning rate, batchSize and # of epochs etc. to properly train your network.) **[35 points]**

- Please visualize the generated images with and without the adversarial loss and discuss the quality of the results. **[10 points]**

- When implementing additional adversarial loss, one may need to combine two losses with the constant lambda, as follows:

$$\text{adv_loss} * \text{lambda} + \text{pixel_loss}$$

Please visualize the generated images at epoch 50 by changing the value of 'lambda' and please discuss the possible reasons for that. **[20 points]**

3. **[5 points]** Please note and apply the below details.

- a. Send all files(.zip) via **Blackboard**.
- b. In **20200401_seungryulbaek_ass2.zip** file, you need to include the main.py and trained weights (.pth) and report files.
- c. Please provide a report document summarizing answers and execution methods for your program for each problem in the form of 'report.pdf' or 'report.docx'.
- d. Please attach the best weights obtained from Problem 2 with the name of 'model.pth'. Please contain codes for all problems in main.py while setting it executable for problem 2 by commenting out codes.

<Example>

```
20200401_seungryulbaek_ass2.zip
|-- model.pth
|-- main.py
|-- model.py
|-- report.pdf or .docx
```

<Tip>

Maximum keeping time for one session in the CoLab is 12 hours when you use GPU mode, but if you remain in your CoLab session without typing anything over 90 mins, the session would be automatically shut down. So, please use the code below to prevent it.

<https://naenjun.tistory.com/18>