DL Lab6: Image Generative Models

Lab Objective:

In this assignment, you will need to implement image generative model (GAN).

Rules:

(1) This assignment should be done individually. Plagiarism is strictly prohibited. Once the T.A. finds plagiarism, you will receive a score of **0** on this assignment.

(2) Only **PyTorch** frameworks are allowed in this lab, beyond that you can only use numpy, matplotlib, and other Python standard library.

(3) The assignment format and files are not in accordance with the regulations, the report **score × 0.9**.

(4) If the assignment is missing or incomplete training for any item, the assignment score will be deducted proportionally to the incompleteness.

(5) If you submit your assignment late, your score will be **multiplied by 0.9 for each day of delay.**

Submission:

(1) Please write your code on Jupyter notebook.

(2) **Only allowed to use the specified model according to each task**. Otherwise, no points will be awarded.

(3) The report can only be handed in **Four A4 pages** at most. You should explain all the implications of all the programs you write and post them in the report, and if you compare any different results, please present the changes and their effects in the report, but it is strictly forbidden to post the entire code.

(4) Upload the compressed file (**.zip**) of "the report (**.pdf**), all program files (**.ipynb**), and all the best weight of model which you wrote." to the E3 platform. The file name is **A6_studentID_studentName.zip**.

(5) Sample submission format: A6_studentID_studentName/

```
|       A6_studentID_studentName.pdf
|       gan.ipynb
|       bonus.ipynb     (Not necessary)
|       model_weight/
|       |       Generator weights.pth
|       |       Discriminator weights.pth
```

Deadline: 2024/11/18 (Mon.) 12:00

Requirements:

(1) In this assignment, you have to finish following task:

Task: RGB image generation by GAN model.

In this task, it is required to implement a **Generative Adversarial Network (GAN)**, including different types **except "original dcGAN", but you can modify dcGAN**, and train the model by **FLOWER-102** dataset.

(2) **Plot the training loss of GAN (generator & discriminator)** during training. And in each of model you should output a **result image with 8x8 generate images** or more.

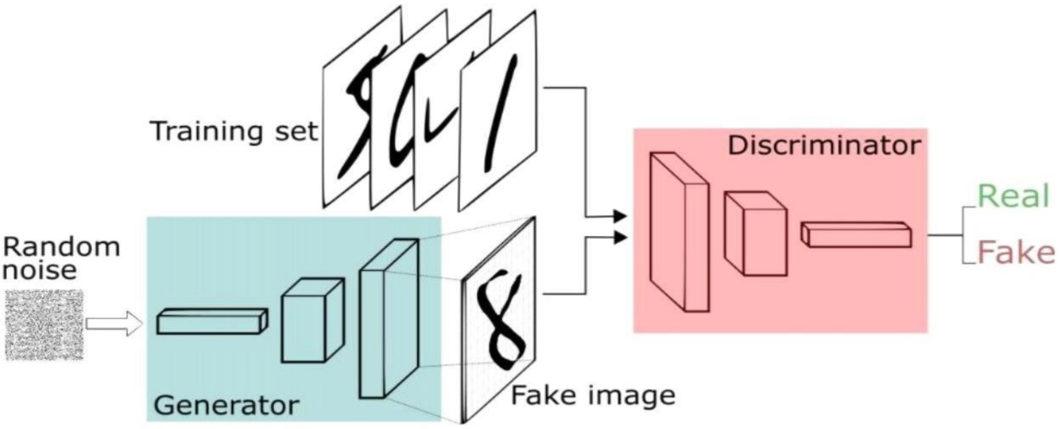
(3) **Compare performance** changes due to different parameters and model structures **and write them into reports.**

(4) Set "**torch.manual_seed(12)**" and "**torch.backends.cudnn.deterministic = True**" in your code for model's training reproducibility.

Descriptions:

(1) Model Architecture

i. GAN



(2) Dataset

i. Task RGB: FLOWER-102

(3) Bonus

i. Training using different generative models, such as other types of GAN, or utilizing additional datasets, choose one from the options mentioned above.

Reference:

(1) https://github.com/eriklindernoren/PyTorch-GAN/tree/master/implementations

(2) https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

Assignment Evaluation:

(1) Code & model performances (60%)

(2) Report (40%)

(3) Bonus (5%+10%)

Please contact TA if you have any questions.

sk774325.ai12@nycu.edu.tw
andy90123.ai12@nycu.edu.tw