

数字内容安全 实验报告



姓 名 詹冲
学 号 2023211616
指导教师 杨震
学 院 网络空间安全学院

2025 年 4 月 13 日

实验名称 GAN 的生成器产生图片及换脸检测 实验日期: 4 月 13 日 指导老师 杨震 得分
学院 网络空间安全学院 专业 信息安全 班次 2023211801 姓名 詹冲 学号 2023211616

一、实验目的

1. 掌握在 WINDOWS 下安装和使用 GAN 图片生成系统、换脸检测系统
2. 掌握 GAN 图片生成系统主要功能模块、换脸检测系统主要功能模块
3. GAN 图片生成系统的原理、换脸检测系统的原理

二、实验内容

1. GAN 生成图片:

- (1) 调试 GAN 图片生成系统程序 (DCGAN_6858) 主要功能模块
- (2) 选取 DCGAN_6858.zip 中实验数据集
- (3) 运行 WINDOWS 下的 GAN 图片生成系统
- (4) 用 GAN 图片生成系统对实验数据集进行图片生成实验

2. 换脸检测:

- (1) 选取实验数据集 (用实验二: CNN-dection 补充数据集.zip 替换 dataset 路径下载数据集代码)
- (2) 调试换脸检测系统程序 (DeepfakesCNN-Detection) 主要功能模块
- (3) 加载 DeepfakesCNN 换脸检测模型权重 (CNN-Detection-modelweights.rar)
- (4) 运行 WINDOWS 下的 GAN 换脸检测系统
- (5) 用换脸检测系统对实验数据集进行换脸检测实验

三、系统整体描述和分功能描述

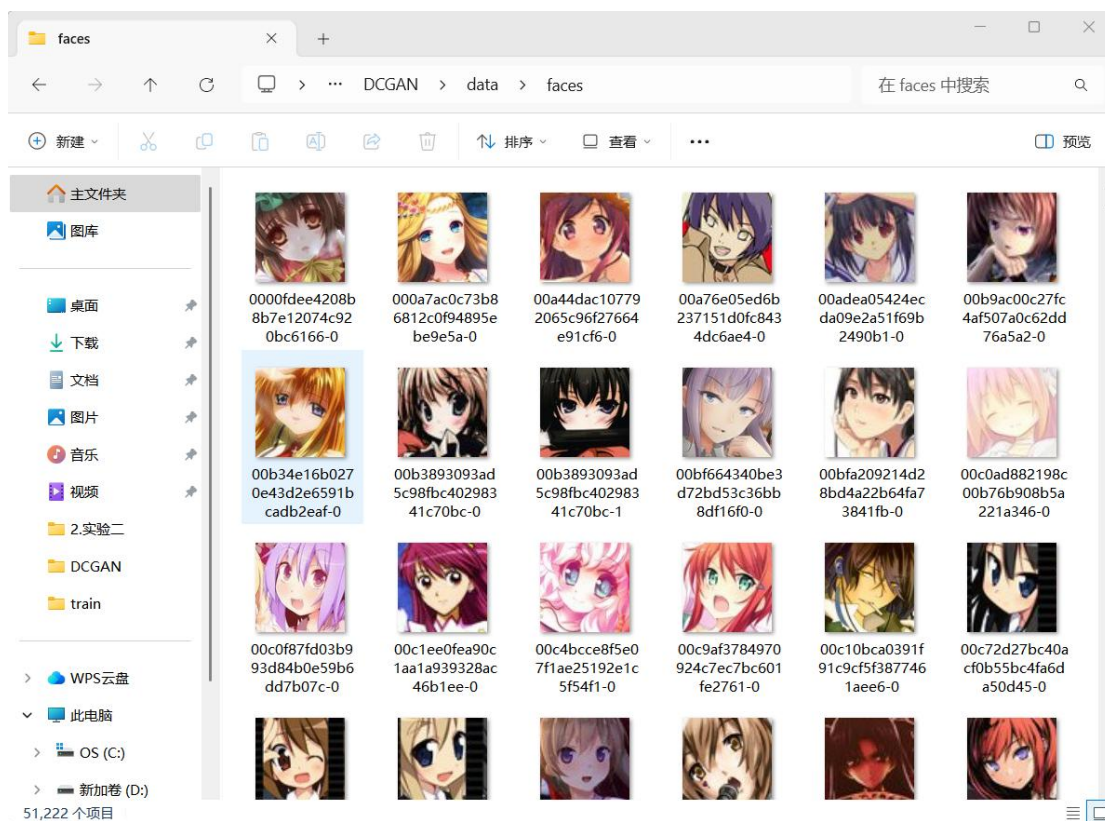
1. 系统整体描述

- (1) DCGAN 系统: 通过 GAN 来生成图片
- (2) CNN-Detection 系统: 通过 CNN 算法对图像进行鉴别, 辨别是否是伪造信息。

2. 分功能描述

- (1) 分功能 1: 训练生成图片模型

为了完成训练, 我们需要先建立一个动漫人物头像数据库, 可以通过爬虫获取大量原始动漫人物形象素材, 使用 opencv 工具将图片进行剪裁, 仅留出动漫人物面部图像, 按照 1: 1 随机划分为训练集和测试集, 并给每个图片唯一的 ID。 (此次实验直接使用老师给出的数据集)。



为了证明各参数设置对网络训练效果的影响，在不同的对照组中按照比例设置参数。通过改变生成器训练次数，判别器训练次数，生成器学习率，判别器学习率，总训练周期，一次迭代时输入的样本数量，经过多次实验与测试，得出尽可能好的超参数。（本实验使用预设的超参数进行训练）。

```
python main.py train --gpu --vis=False
```

（2）分功能 2：生成图片

通过已经训练好的模型，对指定数据库进行图片生成。

```
python main.py generate --nogpu --vis=False --netd-path=checkpoints/netd_200.pth --netg-path=checkpoints/netg_200.pth --gen-img=result.png --gen-num=64
```

（3）分功能 3：检测一张图片的真伪

读取一张图片，对它们进行真伪检测。

```
python demo.py -f examples/real.png -m weights/blur_jpg_prob0.5.pth
python demo.py -f examples/fake.png -m weights/blur_jpg_prob0.5.pth
```

（4）分功能 4：批量检测图片的真伪

批量读取图片，对它们进行真伪检测。

```
python demo_dir.py -d examples/realfakedir -m weights/blur_jpg_prob0.5.pth
```

四、实验步骤、结果及分析

1. 实验步骤

(1) 查看自己的 CUDA 版本，以便下载适配的 pytorch，win+R 打开终端，输入 `nvcc -V`，发现自己的 CUDA 版本为 12.4。

```
C:\WINDOWS\system32\CMD. X + v

Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation。保留所有权利。

C:\Users\asus>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2024 NVIDIA Corporation
Built on Tue_Feb_27_16:28:36_Pacific_Standard_Time_2024
Cuda compilation tools, release 12.4, V12.4.99
Build cuda_12.4.r12.4/compiler.33961263_0
```

(2) 打开 pytorch 官网，找到适配 CUDA 12.4 的 pytorch，并复制其 pip 命令至终端在相应环境中安装如图所示，此处我均已下载完毕，所有都显示 `already satisfied`。

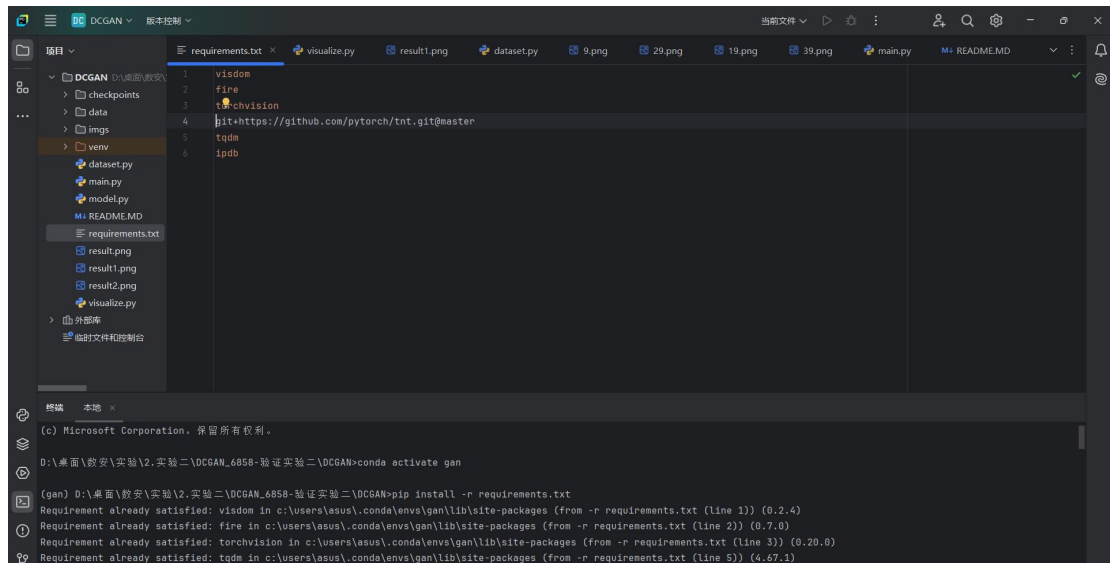
```
C:\WINDOWS\system32\CMD. X + v

Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation。保留所有权利。

C:\Users\asus>conda activate gan

(gan) C:\Users\asus>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu124 -i http://mirrors.cloud.aliyuncs.com/pypi/simple/
Looking in indexes: http://mirrors.cloud.aliyuncs.com/pypi/simple/
Requirement already satisfied: torch in c:\users\asus\conda\envs\gan\lib\site-packages (2.4.1)
Requirement already satisfied: torchvision in c:\users\asus\conda\envs\gan\lib\site-packages (0.20.0)
Requirement already satisfied: torchaudio in c:\users\asus\conda\envs\gan\lib\site-packages (2.4.1)
Requirement already satisfied: filelock in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (4.11.0)
Requirement already satisfied: sympy in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.1)
Requirement already satisfied: Jinja2 in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (2025.3.0)
Requirement already satisfied: numpy in c:\users\asus\conda\envs\gan\lib\site-packages (from torchvision) (1.24.3)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from torchvision) (10.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from Jinja2->torch) (2.1.3)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from sympy->torch) (1.3.0)
```

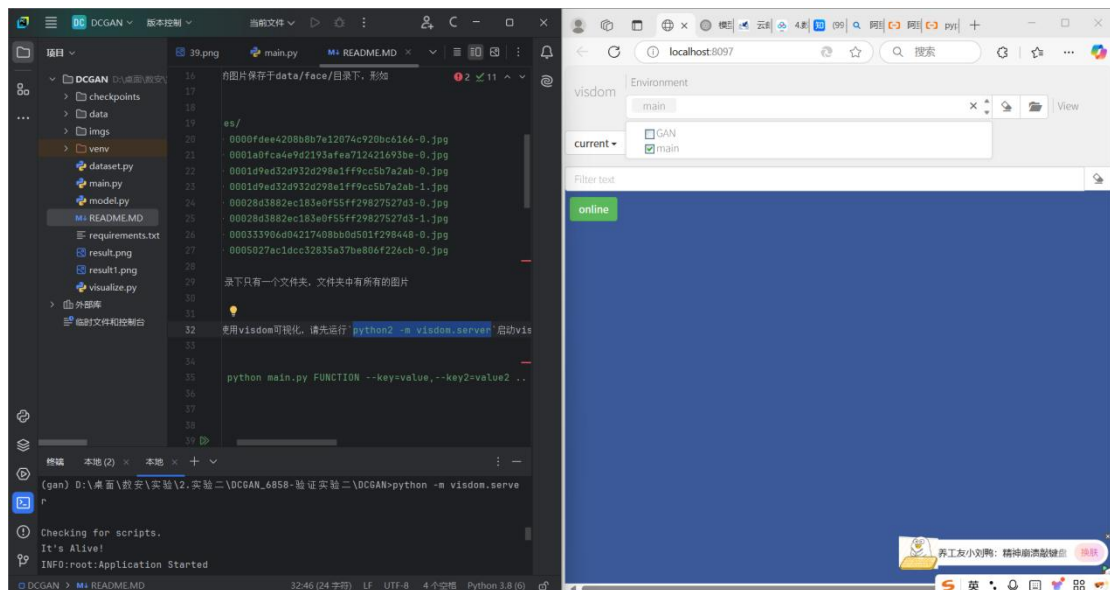
(3) 下载系统所需要的第三方库，在终端输入 `pip install -r requirements.txt` 即可，此处我均已下载完毕，所有都显示 `already satisfied`。



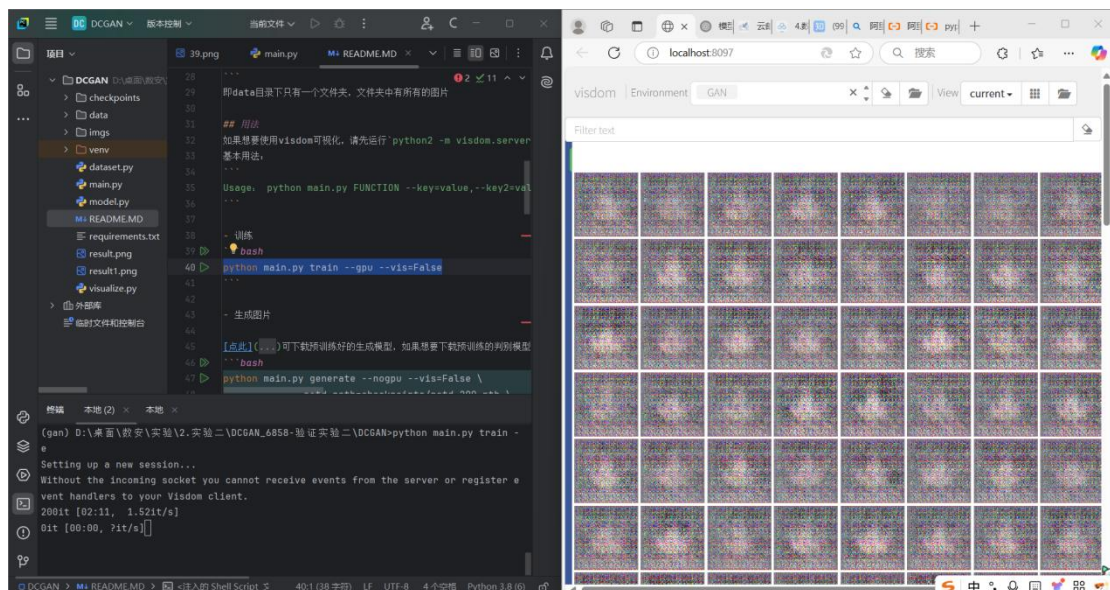
```
requirements.txt
1 visdom
2 fire
3 chvision
4 git+https://github.com/pytorch/tnt.git@master
5 tqdm
6 ipdb

(gan) D:\桌面\数安\实验\2_实验二\DCGAN_6858-验证实验二\DCGAN>conda activate gan
(gan) D:\桌面\数安\实验\2_实验二\DCGAN_6858-验证实验二\DCGAN>pip install -r requirements.txt
Requirement already satisfied: visdom in c:\users\asus\conda\envs\gan\lib\site-packages (from -r requirements.txt (line 1)) (0.2.4)
Requirement already satisfied: fire in c:\users\asus\conda\envs\gan\lib\site-packages (from -r requirements.txt (line 2)) (0.7.0)
Requirement already satisfied: torch in c:\users\asus\conda\envs\gan\lib\site-packages (from -r requirements.txt (line 3)) (2.4.1)
Requirement already satisfied: torchvision in c:\users\asus\conda\envs\gan\lib\site-packages (from -r requirements.txt (line 3)) (0.20.0)
Requirement already satisfied: torchaudio in c:\users\asus\conda\envs\gan\lib\site-packages (from -r requirements.txt (line 3)) (2.4.1)
Requirement already satisfied: filelock in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.13.1)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (4.11.0)
Requirement already satisfied: sympy in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (1.13.3)
Requirement already satisfied: networkx in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.1)
Requirement already satisfied: Jinja2 in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (3.1.4)
Requirement already satisfied: fsspec in c:\users\asus\conda\envs\gan\lib\site-packages (from torch) (2025.3.0)
Requirement already satisfied: numpy in c:\users\asus\conda\envs\gan\lib\site-packages (from torchvision) (1.24.3)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from torchvision) (10.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from Jinja2->torch) (2.1.3)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\asus\conda\envs\gan\lib\site-packages (from sympy->torch) (1.3.0)
```

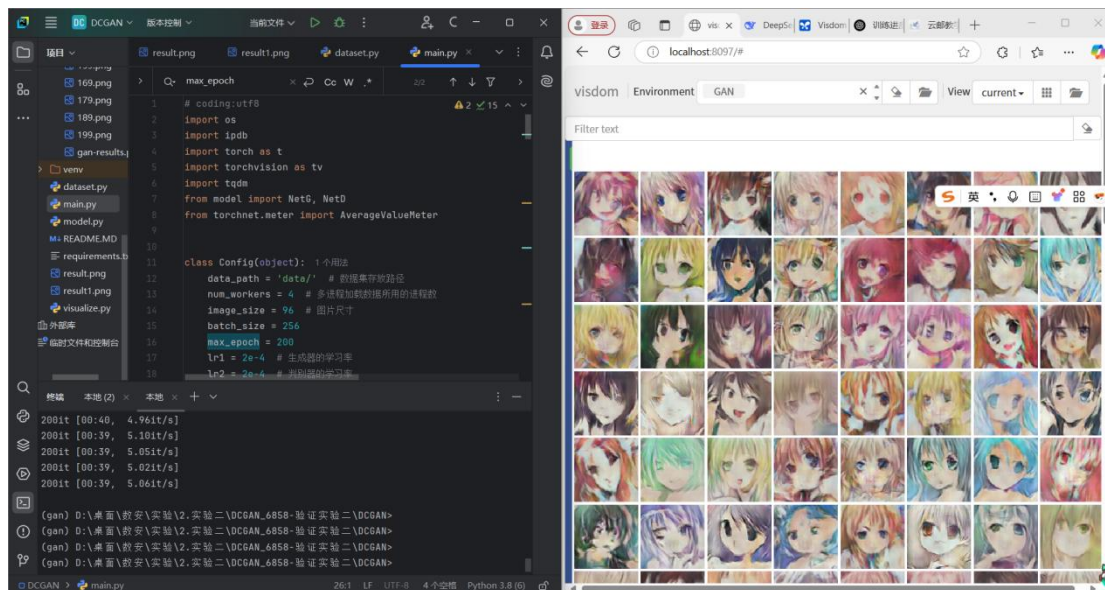
(4) 至此 GAN 生成图片的环境已经全部配置完毕，接下来进行模型训练，首先打开可视化窗口，在终端输入 `python -m visdom.server`，并在可视化窗口中，将环境从默认的主改为 GAN。



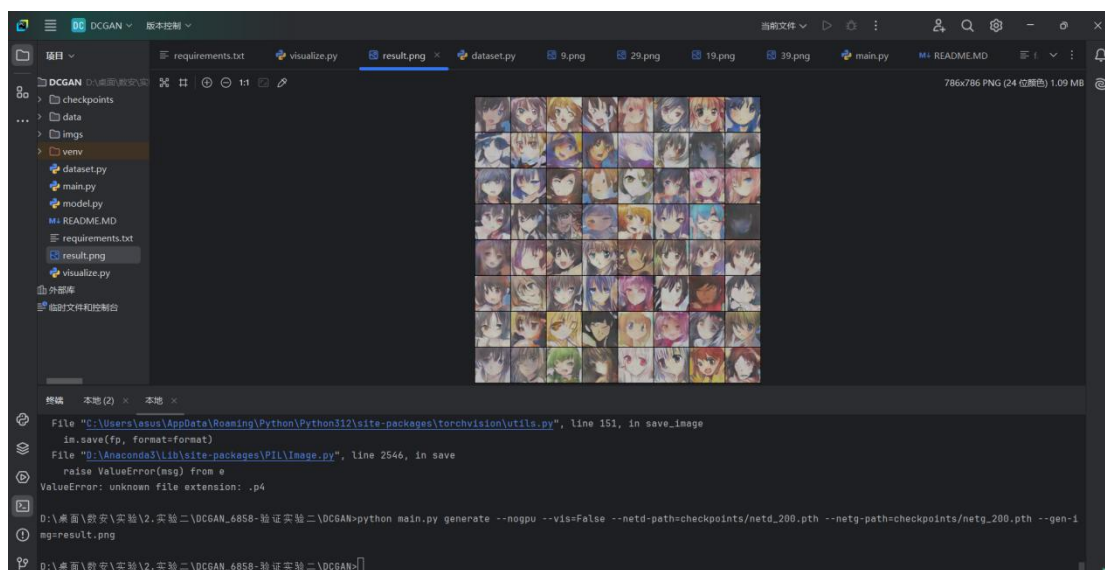
(5) 开始训练模型，在终端输入 `python main.py train --gpu --vis=True` (`vis=True` 为使用可视化，`False` 为不使用可视化)。

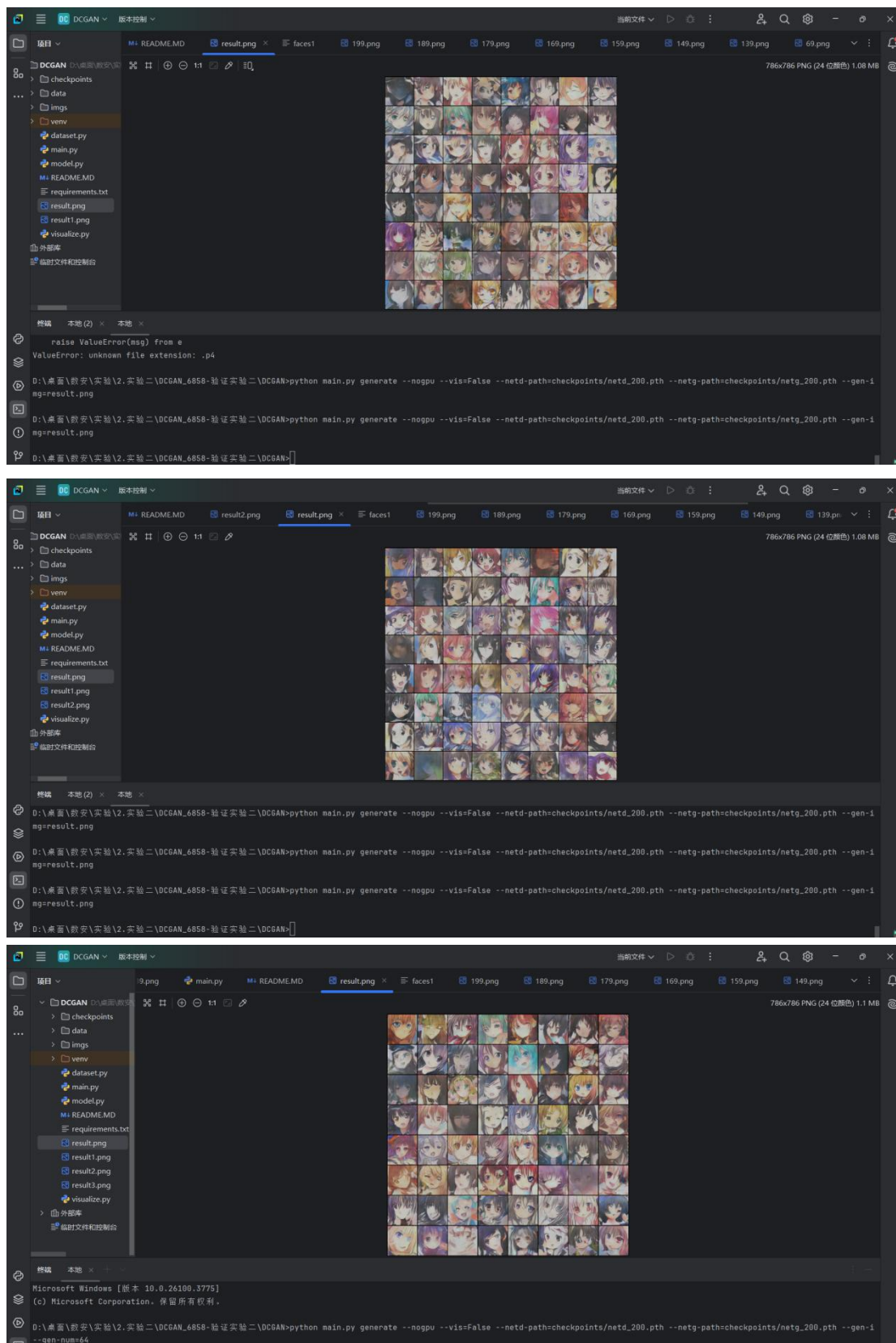


(6) 经过一个漫长的时间，训练完成，其中各个 epoch 的参数，均已存储在了 checkpoints 文件夹中。



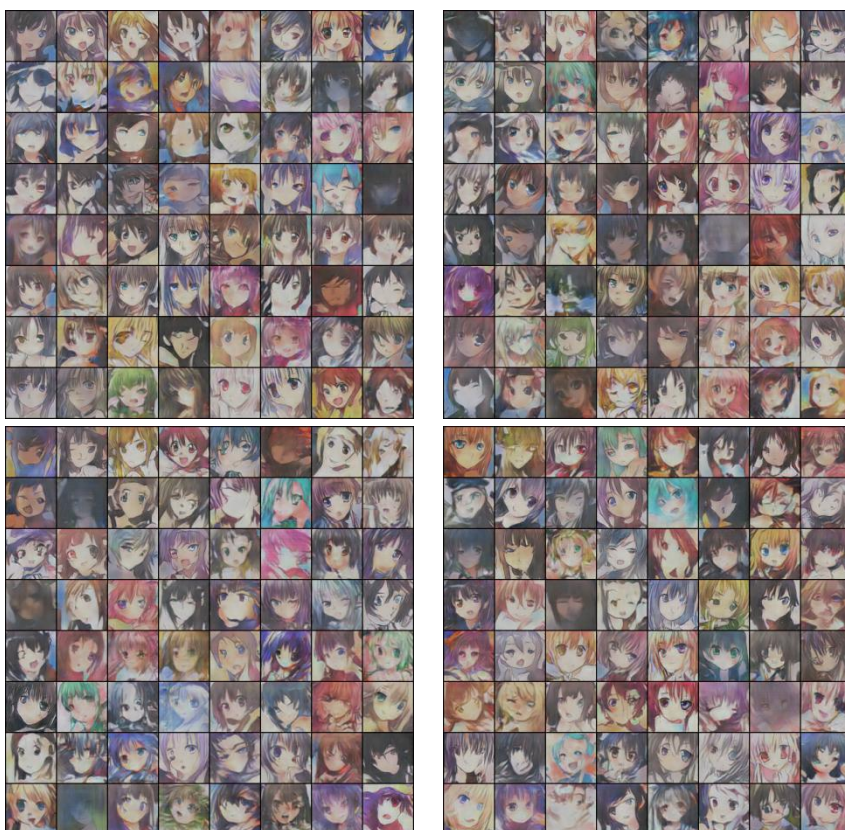
(7) 接下来通过已经训练好的参数进行多次的图片生成，使用语句
`python main.py generate --nogpu --vis=False --netd-path=checkpoints/netd_200.pth --netg-path=checkpoints/netg_200.pth --gen-img=result.png` 即可生成图片，并将其保存在目录下的 `result.png` 中。





至此第一个实验完成。

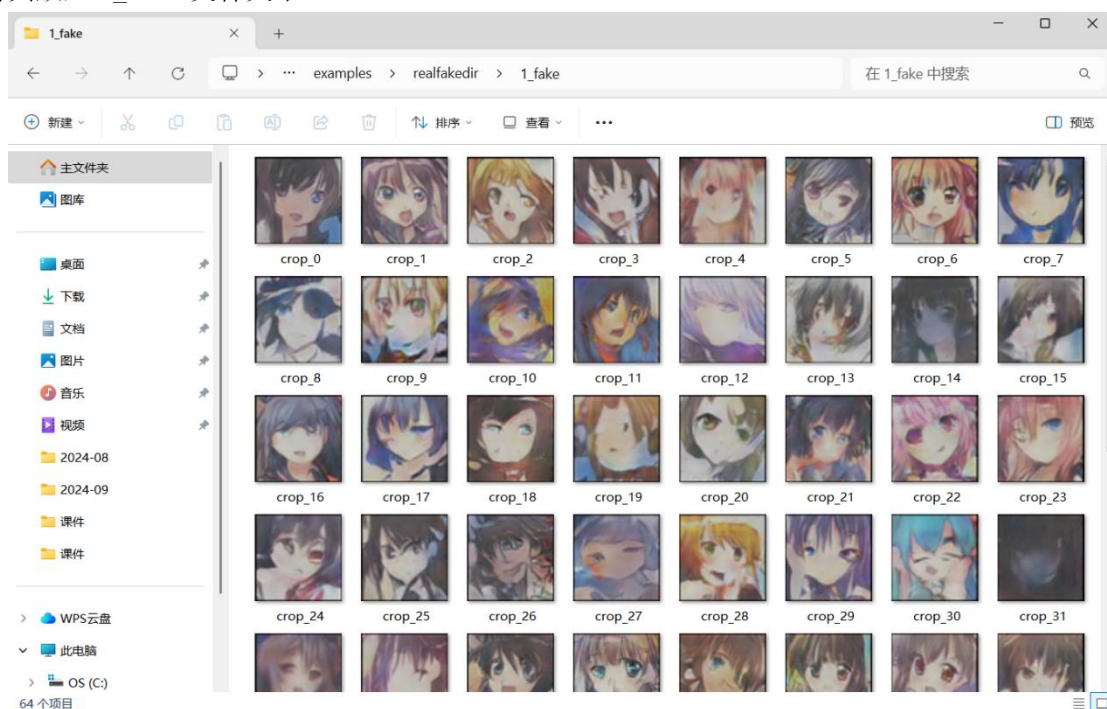
（8）配置换脸检测系统环境，首先需要安装必要的库，由于换脸检测系统



(2) 换脸检测系统

1) 对实验一中图片生成系统中的图片进行换脸检测。

对图片生成系统生成的第一张结果图运行 python 脚本进行剪裁,得到如下图所示结果,并将其放入 1_fake 文件夹中。



并且随机在生成图片的数据集中选取 64 张真图片放入 0_real 文件夹中。

对这些图片进行换脸检测,结果如下图所示,发现对于真图片的检测效果很好,正确率

可以到达 100%，但是对于假图片的检测效果有所下降，只能达到 73.44%，说明图片生成系统中，仍有些图片可以足够真到骗过换脸检测系统，而换脸检测系统仍有改进的空间。

```
(gsen) D:\桌面\数安\实验二\Deepfakes CNN-Detection验证实验二>python demo_dir.py -d examples/realfakedir -m weights/blurr_prob0.6.pth  
Not cropping  
loading [1] datasets  
100%|#####| 4/4 [00:01:08:00, 3.21it/s]  
  
Average sizes: [106.50+/-10.50] x [106.50+/-10.50] = [0.01+/-0.00 Mpix]  
Num reals: 64, Num fakes: 64  
AP: 98.10, Acc: 86.72, Acc (real): 100.00, Acc (fake): 73.44
```

2) 对单个图片进行换脸检测时, 能够很好的识别给定的两张图片。对于真实的图片, 发现识别它为生成图片的概率为 0%。



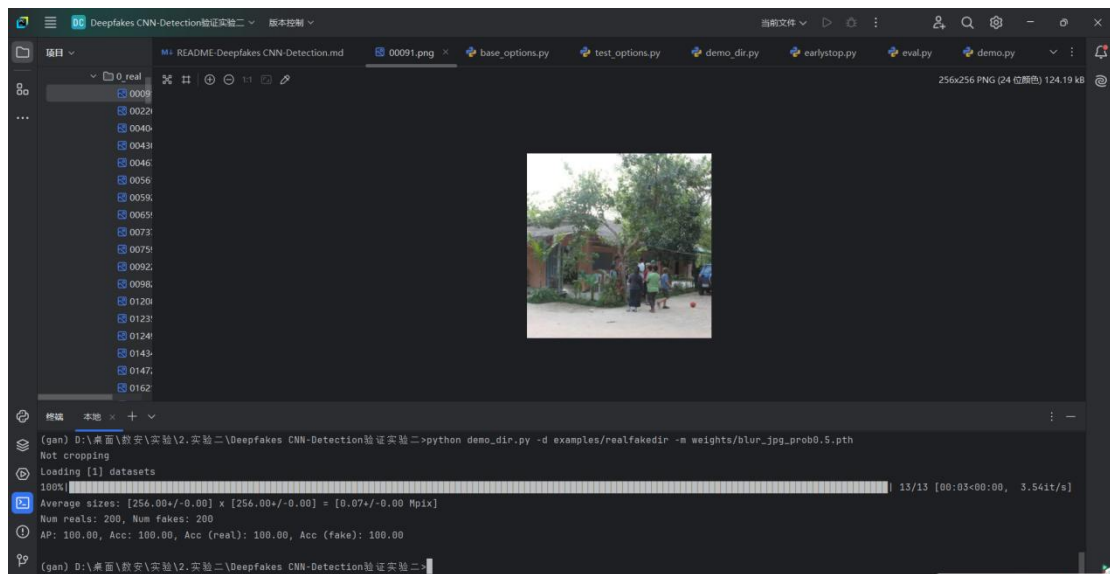
```
(Gan) D:\桌面\数安\实验\2.实验二\Deepfakes CNN-Detection验证实验二>python demo.py -f examples/real.png -m weights/blvr_jpg_prob0.5.pth
Not cropping
probability of being synthetic: 0.00%
```

对于虚假的图片，发现识别它为生成图片的概率为 99.86%。

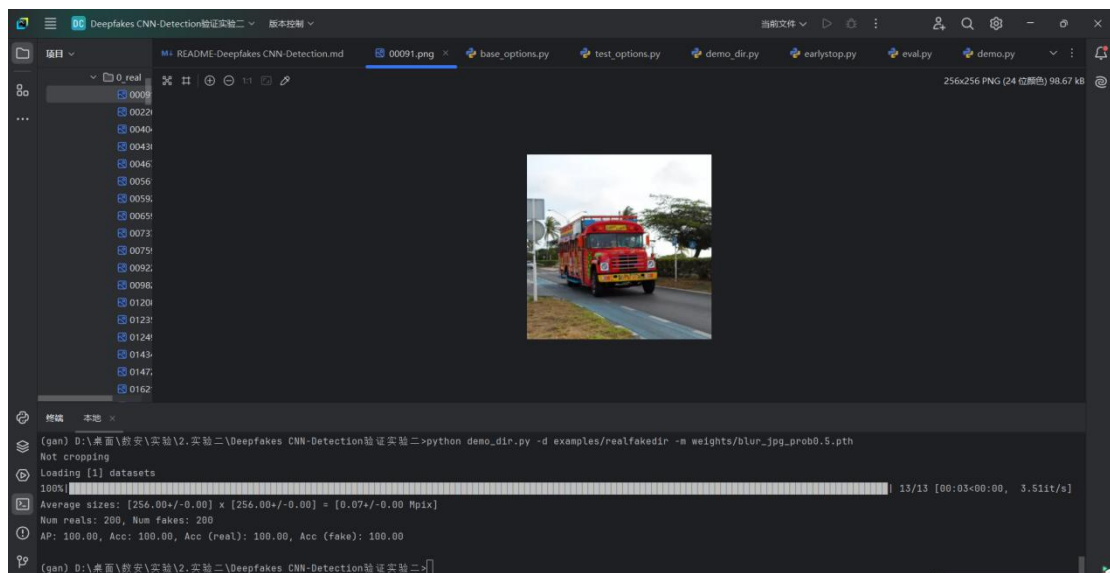


```
(gan) D:\桌面\数安\实验\2.实验二\Deepfakes CNN-Detection验证实验二>python demo.py -f examples/fake.png -m weights/blur_jpg_prob0.5.pth
Not cropping
probability of being synthetic: 99.86%
```

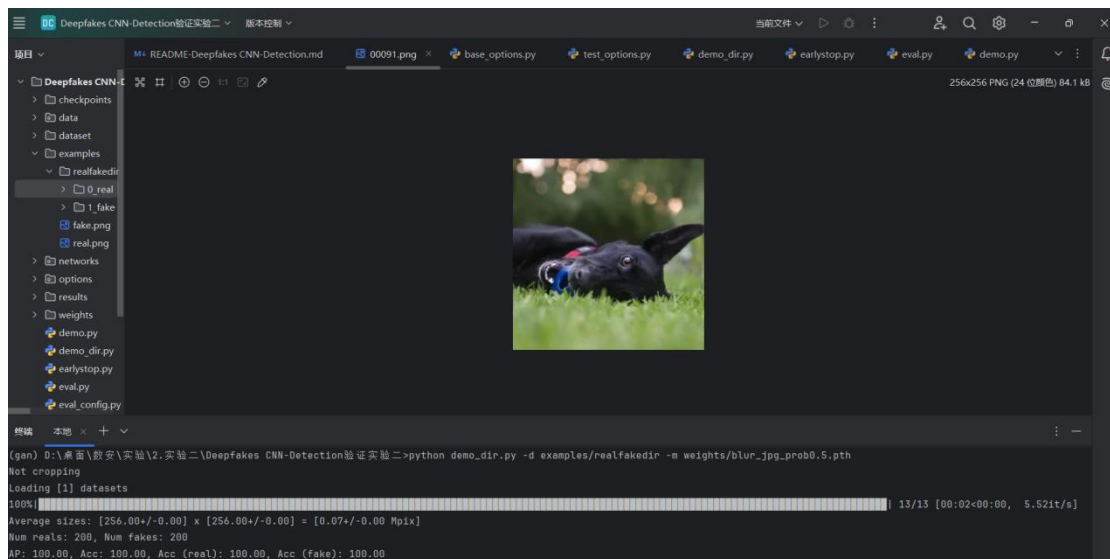
3) 对数据集进行检测时,发现结果仍旧很好,对于虚假和真实的图片识别率均达 100%。
对 person 数据集



对 bus 数据集



对 dog 数据集



五、实验中遇到的问题及改正的方法

1. CUDA 版本太高（一开始为 12.7）导致没有与之适配的 pytorch 版本，只能将 CUDA 卸载之后，重新下载低版本的 CUDA，并下载相关适配版本的 pytorch。

具体操作主要参考 https://blog.csdn.net/qq_43308156/article/details/127479544

```
C:\WINDOWS\system32\CMD. x + v
Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\asus>nvidia-smi
Fri Apr 11 12:10:49 2025

+-----+
| NVIDIA-SMI 566.07              Driver Version: 566.07      CUDA Version: 12.7     |
+-----+-----+
| GPU   Name           Driver-Model  Bus-Id          Disp.A    Volatile Uncorr. ECC   |
| Fan  Temp            Perf             Pwr:Usage/Cap     Memory-Usage  GPU-Util  Compute M.  |
|                                             MIG M.         |
+-----+-----+
|    0   NVIDIA GeForce RTX 4060 ... WDDM      00000000:01:00.0 Off |
| N/A    50C           P0              16W /  93W      0MiB /  8188MiB     0%        Default |
|                                             N/A          |
+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI        PID   Type   Process name                      GPU Memory |
| ID     ID     ID                             Usage      |
+-----+-----+
| No running processes found |
+-----+

C:\Users\asus>
```

2. 在安装必要库时，发现如下报错，原因是开了 VPN 之后，代理中断了 SSL 连接，所以需要关闭 VPN，但是不使用 VPN 下载速度又太慢了，所以使用阿里 pip 镜像源进行下载。

```

(gan) D:\桌面\网络安全\实验\2_实验二\Deepfakes\CNN-Detection验证实验二>psip install tqdm
WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLError[urllib3.exceptions.SSLError], 'TLS/SSL connection has been closed (EOF) (ssl.c:1149)')': /simple/tqdm/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLError[urllib3.exceptions.SSLError], 'TLS/SSL connection has been closed (EOF) (ssl.c:1149)')': /simple/tqdm/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLError[urllib3.exceptions.SSLError], 'TLS/SSL connection has been closed (EOF) (ssl.c:1149)')': /simple/tqdm/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLError[urllib3.exceptions.SSLError], 'TLS/SSL connection has been closed (EOF) (ssl.c:1149)')': /simple/tqdm/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'SSLError(SSLError[urllib3.exceptions.SSLError], 'TLS/SSL connection has been closed (EOF) (ssl.c:1149)')': /simple/tqdm/

```

3. 训练生成图片模型时，在尝试打开可视化界面时，在很长一段时间内都无法打开，后面发现是 VPN 的原因，后续将 VPN 关闭，再次尝试打开，由于“墙”的原因，需要较久的时间才能加载完毕。