

GeneFacePlusPlus

使用系统: **Ubuntu22.04** 需要把常用的Linux命令包安装好

```
sudo apt update
```

工具准备: VPN节点

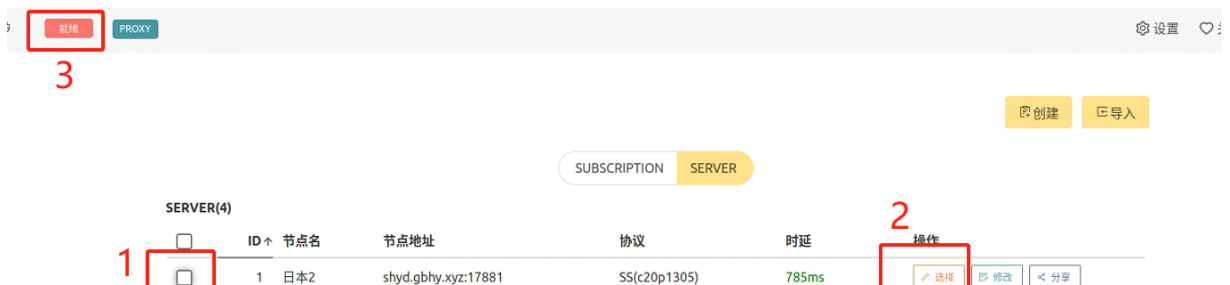
这是举例用的 实际应该连不上或者不稳定

1. 配置Linux系统下的VPN代理:

1. `sudo mkdir -p /etc/apt/keyrings/`
2. `wget -qO - https://apt.v2raya.org/key/public-key.asc | sudo tee /etc/apt/keyrings/v2raya.asc`
3. `echo "deb [signed-by=/etc/apt/keyrings/v2raya.asc] https://apt.v2raya.org/v2raya main" | sudo tee /etc/apt/sources.list.d/v2raya.list sudo apt update`
4. `sudo apt update`
5. `sudo apt install v2raya v2ray`
6. `sudo systemctl start v2raya.service`
7. 打开火狐 地址栏填入 localhost:2017 输入用户名密码 进行登录
8. 右上角点击 导入 把可用的VPN节点进行导入



9. 对节点进行勾选 然后右侧选择 左上方点就绪



10. 打开Youtube进行科学上网测试

2. 配置对应版本的CUDA文件

`nvidia-smi` 查看本机GPU状态

更新gcc 并在安装完毕后查看确认

```
sudo add-apt-repository ppa:ubuntu-toolchain-r/test sudo apt update sudo apt
install gcc-12 g++-12 sudo update-alternatives --install /usr/bin/gcc gcc
/usr/bin/gcc-12 120 --slave /usr/bin/g++ g++ /usr/bin/g++-12 sudo update-
alternatives --config gcc gcc --version
```

安装CUDA 11.7.0

1. 访问NVIDIA开发者网站的CUDA Toolkit Archive 把run文件下载到本地

```
cuda_11.7.0_465.19.01_linux.run
```

2. 关闭图形界面 `Ctrl+Alt+F3`

3. login 登录管理员账号

4. 禁用 Nouveau 驱动

`vi /etc/modprobe.d/blacklist-nouveau.conf` 添加以下内容:

```
blacklist nouveau options nouveau modeset=0
```

执行: `sudo update-initramfs -u`

5. 安装

```
chmod +x cuda_11.7.0_465.19.01_linux.run sudo ./cuda_11.7.0_465.19.01_linux.run
```

6. 设置环境变量

```
echo "export CUDA_HOME=/usr/local/cuda-11.7" >> ~/.bashrc echo "export
PATH=\\$CUDA_HOME/bin:\\$PATH" >> ~/.bashrc echo "export
LD_LIBRARY_PATH=\\$CUDA_HOME/lib64:\\$LD_LIBRARY_PATH" >> ~/.bashrc source
~/.bashrc
...
```

3. 安装Conda环境

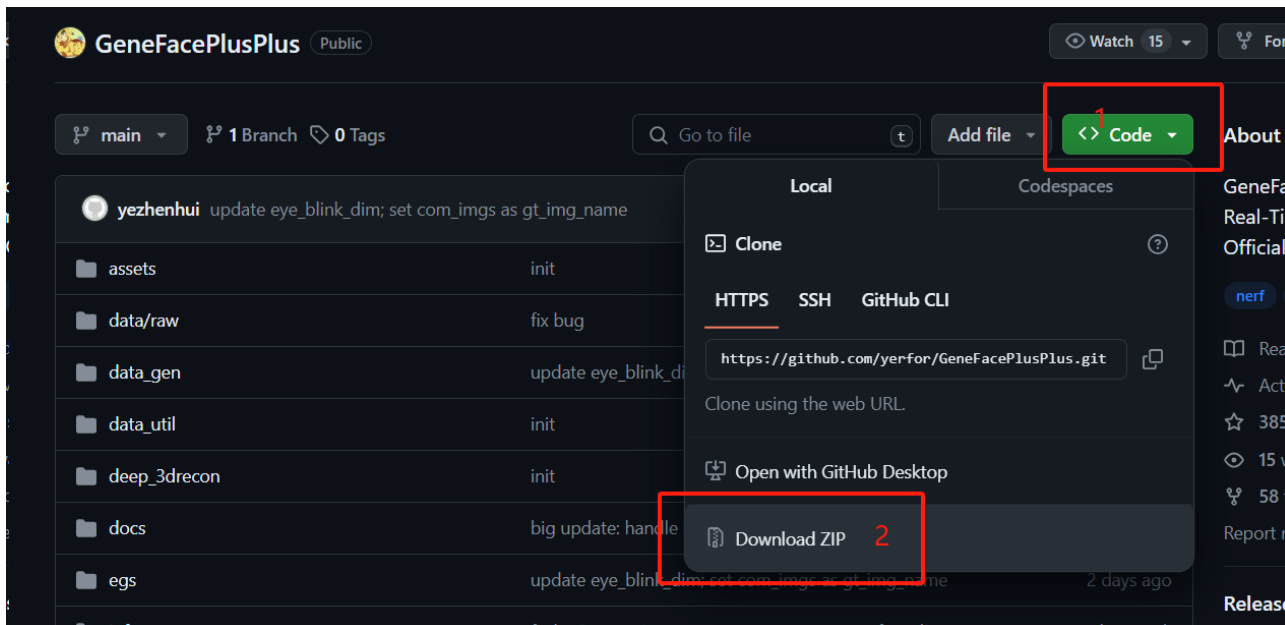
1. 进入下载路径 `cd /home/zb`

2. 下载驱动 `bash Anaconda-latest-Linux-x86_64.sh`

3. 重启配置环境 `source ~/.bashrc`

4. 检测是否安装成功 `conda --version`

4. 科学上网 打开[GeneFace](#)的开源项目仓库 将项目下载到本地 并对其进行解压



并在其目录下打开终端

5. 创建并激活虚拟环境

1. `source <CondaRoot>/bin/activate` 其中 `<CondaRoot>` 为Conda需要替换安装路径如 `/home/zb`
2. `conda create -n geneface python=3.9`
3. `conda activate geneface`

6. 安装所需依赖文件

```
conda install conda-forge::ffmpeg
conda install pytorch==2.0.1 torchvision==0.15.2 torchaudio==2.0.2 pytorch-cuda=11.7 -c pytorch -c nvidia
```

7. 安装pytorch3d文件

两种方法，但是第一种经常因为网络原因报超时错误

1. 命令行安装：
`pip install "git+https://github.com/facebookresearch/pytorch3d.git@stable"`
2. 手动安装：
 1. 在GitHub上下载[pytorch3d](#)压缩文件
 2. 进入Conda中虚拟环境的安装包，参考目录为：
`/home/zb/conda/env/geneface/python3.9/lib/site-packages`
 3. 在文件目录下新建一个终端 `python setup.py install`

8. 安装依赖文件

```
pip install cython
pip install openmim==0.3.9
mim install mmcv==2.1.0
sudo apt-get install libasound2-dev portaudio19-dev
pip install -r docs/prepare_env/requirements.txt -v
```

9. 执行配置文件

```
bash docs/prepare_env/install_ext.sh
```

10. 准备数据模型

1. [百度云](#)下载解压后，将文件夹内的8个文件移动到 <GeneFaceRoot>/deep_3drecon/BFM/ 目录下
2. [百度云](#)下载解压后，将 trainval_dataset.npy 放到 data/binary/videos/May/trainval_dataset.npy 路径下，将 audio2motion_vae.zip 和 motion2video_nerf.zip 放到 /checkpoints/ 目录下

其中文件层级需要注意，应为：

```
checkpoints
audio2moyion_vae
motion2video_nerf
may_head
may_torso
data
binary
videos
May
trainval_dataset.npy
```

11. 激活python环境

```
export PYTHONPATH=./
```

12. 运行demo并保存生成文件

```
python inference/genefacepp_infer.py --a2m_ckpt=checkpoints/audio2motion_vae --
head_ckpt= --torso_ckpt=checkpoints/motion2video_nerf/may_torso --
drv_aud=data/raw/val_wavs/MacronSpeech.wav --out_name=may_demo.mp4
```

使用说明（均在项目环境中运行）

数据准备

1. 准备一段视频 必须确保每一帧都有目标人脸
2. 将视频主动调参（512x512 25FPS）

```
ffmpeg -i data/raw/videos/${VIDEO_ID}.mp4 -vf fps=25,scale=w=512:h=512 -qmin 1
-q:v 1 data/raw/videos/${VIDEO_ID}_512.mp4
mv data/raw/videos/${VIDEO_ID}.mp4 data/raw/videos/${VIDEO_ID}_to_rm.mp4
mv data/raw/videos/${VIDEO_ID}_512.mp4 data/raw/videos/${VIDEO_ID}.mp4
```

3. 提取音频特征

```
export CUDA_VISIBLE_DEVICES=0
export PYTHONPATH=./
export VIDEO_ID=May
mkdir -p data/processed/videos/${VIDEO_ID}
ffmpeg -i data/raw/videos/${VIDEO_ID}.mp4 -f wav -ar 16000
data/processed/videos/${VIDEO_ID}/aud.wav
python data_gen/utils/process_audio/extract_hubert.py --video_id=${VIDEO_ID}
python data_gen/utils/process_audio/extract_mel_f0.py --video_id=${VIDEO_ID}
```

4. 提取图片信息

```
export PYTHONPATH=./
export VIDEO_ID=May
export CUDA_VISIBLE_DEVICES=0
mkdir -p data/processed/videos/${VIDEO_ID}/gt_imgs
ffmpeg -i data/raw/videos/${VIDEO_ID}.mp4 -vf fps=25,scale=w=512:h=512 -qmin 1
-q:v 1 -start_number 0 data/processed/videos/${VIDEO_ID}/gt_imgs/%08d.jpg
python data_gen/utils/process_video/extract_segment_imgs.py --ds_name=nerf --
vid_dir=data/raw/videos/${VIDEO_ID}.mp4 # extract image, segmap, and background
```

5. LM2D & Fit 3D

```
export PYTHONPATH=./
export VIDEO_ID=May
python data_gen/utils/process_video/extract_lm2d.py --ds_name=nerf --
vid_dir=data/raw/videos/${VIDEO_ID}.mp4
export PYTHONPATH=./
export VIDEO_ID=May
export CUDA_VISIBLE_DEVICES=0
```

```
python data_gen/utils/process_video/fit_3dmm_landmark.py --ds_name=nerf --  
vid_dir=data/raw/videos/${VIDEO_ID}.mp4 --reset --debug --id_mode=global
```

6. 数据打包

```
export PYTHONPATH=./  
export VIDEO_ID=May  
python data_gen/runs/binarizer_nerf.py --video_id=${VIDEO_ID}
```

7. 训练 Head NeRF 模型 和 Torso NeRF 模型

```
CUDA_VISIBLE_DEVICES=0 python tasks/run.py --  
config=egs/datasets/May/lm3d_radnerf_sr.yaml --  
exp_name=motion2video_nerf/may_head --reset
```

```
CUDA_VISIBLE_DEVICES=0 python tasks/run.py --  
config=egs/datasets/May/lm3d_radnerf_torso_sr.yaml --  
exp_name=motion2video_nerf/may_torso --  
hparams=head_model_dir=checkpoints/motion2video_nerf/may_head --reset
```

8. 将得到的数据集和模型放到对应文件夹中