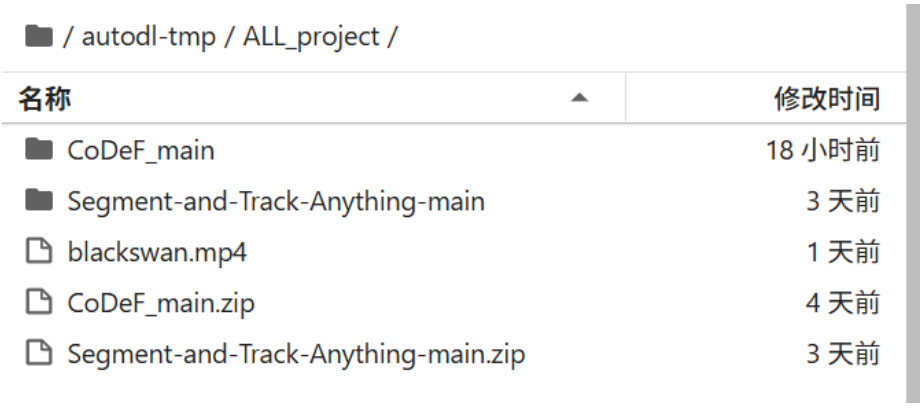
一．环境准备

在 autodl 上租的，配置如下；

项目整体运行跑通了，系统盘镜像已经保存

然后大概我把数据盘打包成压缩包就可以完全移植这个实例了吧



二．Segment-and-Track-Anything-main

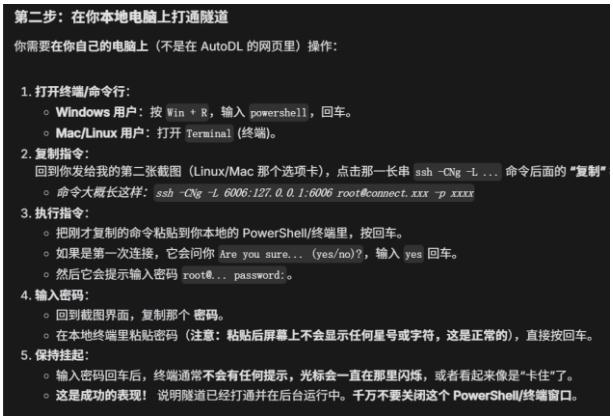
新建一个虚拟环境单独安装依赖

```
(base) root@autodl-container-c0d64989c1-c3c508de:~/autodl-tmp/ALL_project# conda env list
# conda environments:
#
base                  *  /root/miniconda3
codef_env                /root/miniconda3/envs/codef_env
sam_track                /root/miniconda3/envs/sam_track
```

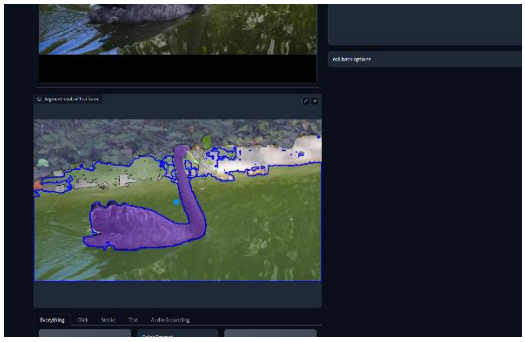
脚本运行过程有一些库版本和缺失，一个个装就行

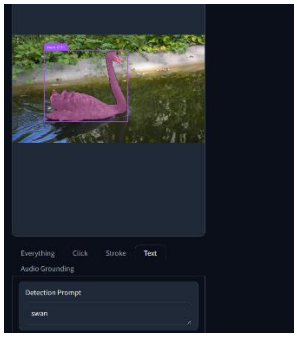
Autodl 貌似不能直接通过浏览器打开这个项目的网页 ui，需要在控制台自定义服务那找到下面这个

再下面这样，本地浏览器就可以打开网页了



然后效果如下：

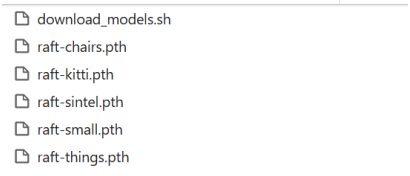


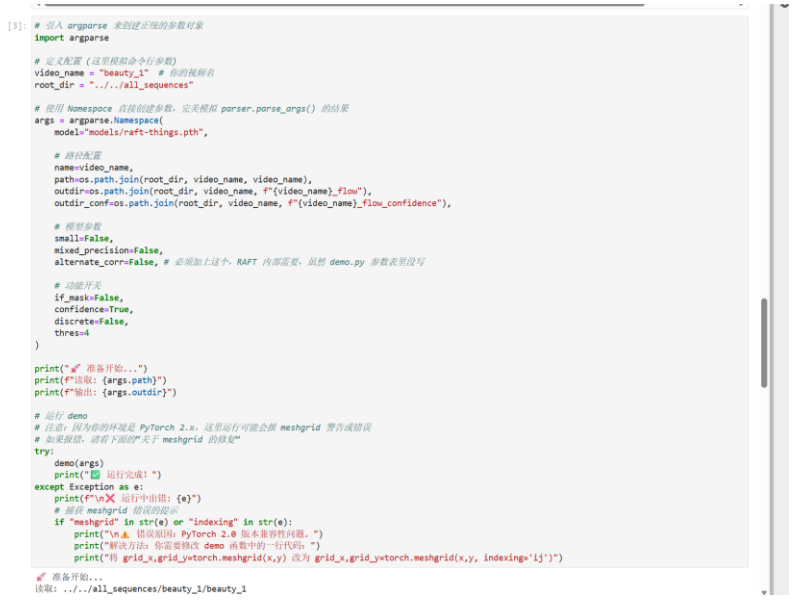


这个项目是在终端执行的，下一个光流是在 notebook 执行

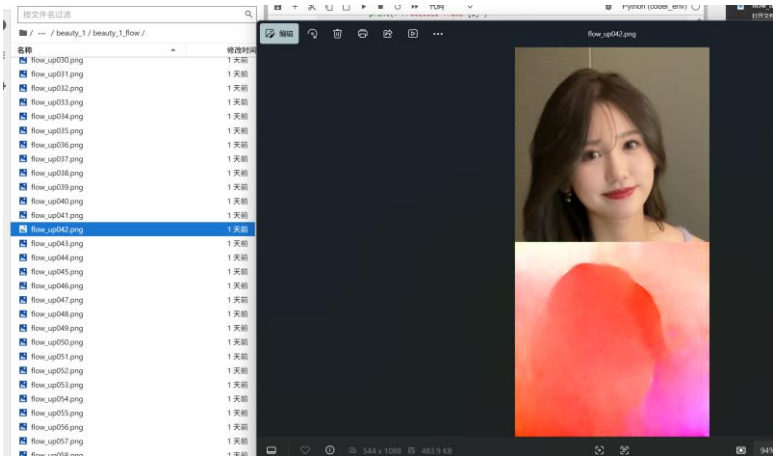
三．光流

这个问题要少一些，网络问题不好下模型，本地下完传上去就好了





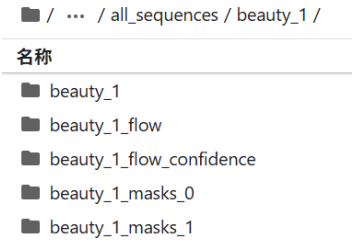
然后获得给光流文件

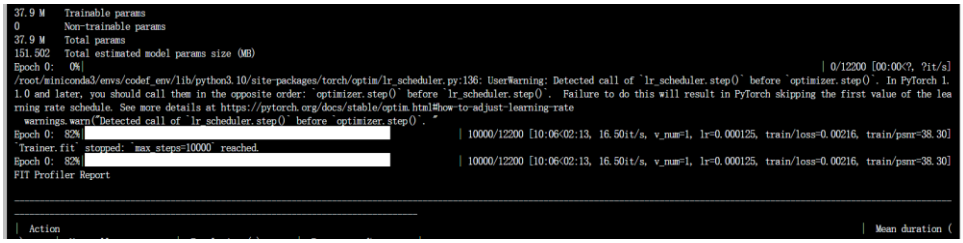


四．Codef

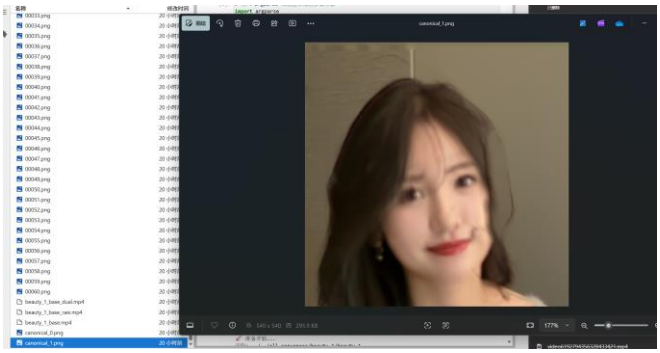
数据预处理完成后，我是在终端运行和测试的

文件结构 训练

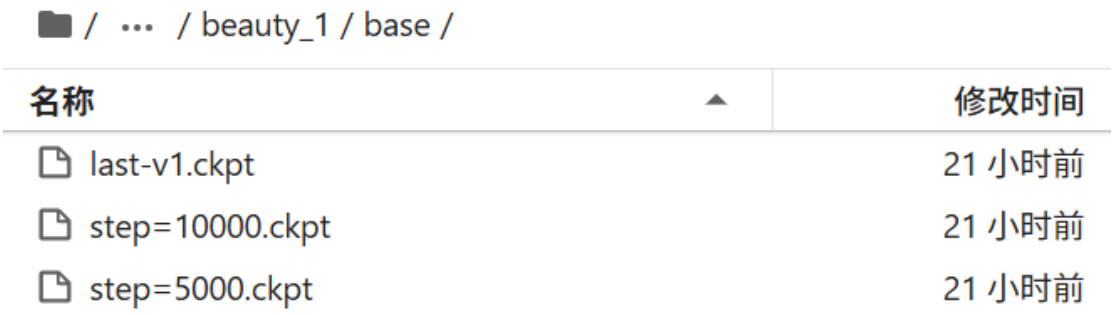




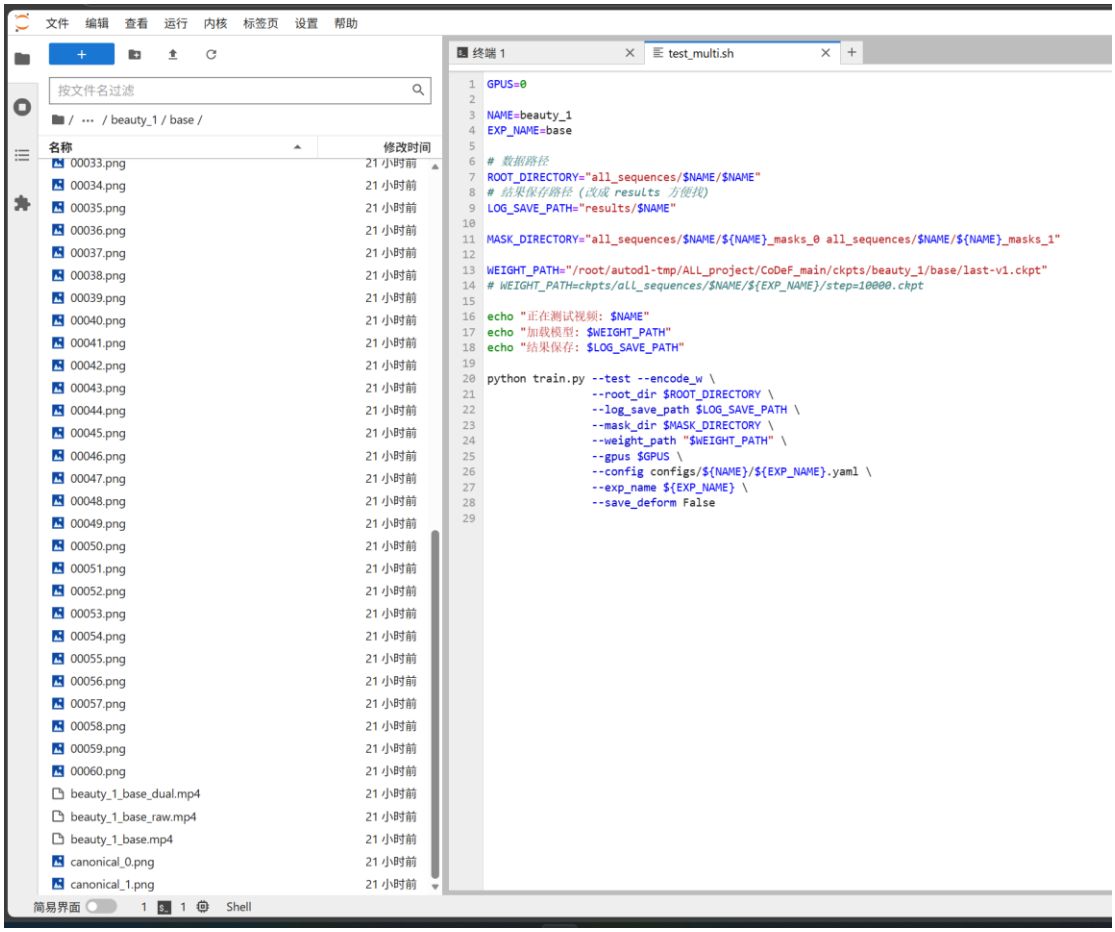
获得规范场图



保存的模型



最后测试和视频重建


文件 编辑 查看 运行 内核 标签页 设置 帮助
按文件名过滤
/ ··· / beauty_1 / base /
名称
修改时间
00033.png 21 小时前
00034.png 21 小时前
00035.png 21 小时前
00036.png 21 小时前
00037.png 21 小时前
00038.png 21 小时前
00039.png 21 小时前
00040.png 21 小时前
00041.png 21 小时前
00042.png 21 小时前
00043.png 21 小时前
00044.png 21 小时前
00045.png 21 小时前
00046.png 21 小时前
00047.png 21 小时前
00048.png 21 小时前
00049.png 21 小时前
00050.png 21 小时前
00051.png 21 小时前
00052.png 21 小时前
00053.png 21 小时前
00054.png 21 小时前
00055.png 21 小时前
00056.png 21 小时前
00057.png 21 小时前
00058.png 21 小时前
00059.png 21 小时前
00060.png 21 小时前
beauty_1_base_dual.mp4 21 小时前
beauty_1_base_raw.mp4 21 小时前
beauty_1_base.mp4 21 小时前
canonical_0.png 21 小时前
canonical_1.png 21 小时前
简易界面
1 1 Shell
终端 1
test_multi.sh
GPUS=0
NAME=beauty_1
EXP_NAME=base
# 数据路径
ROOT_DIRECTORY="all_sequences/$NAME/$NAME"
# 结果保存路径（改成 results 方便找）
LOG_SAVE_PATH="results/$NAME"
MASK_DIRECTORY="all_sequences/$NAME/${NAME}_masks_0 all_sequences/$NAME/${NAME}_masks_1"
WEIGHT_PATH="/root/autodl-tmp/ALL_project/CoDeF_main/ckpts/beauty_1/base/last-v1.ckpt"
# WEIGHT_PATH=ckpts/all_sequences/$NAME/${EXP_NAME}/step=10000.ckpt
echo "正在测试视频: $NAME"
echo "加载模型: $WEIGHT_PATH"
echo "结果保存: $LOG_SAVE_PATH"
python train.py --test --encode_w \
--root_dir $ROOT_DIRECTORY \
--log_save_path $LOG_SAVE_PATH \
--mask_dir $MASK_DIRECTORY \
--weight_path "$WEIGHT_PATH" \
--gpus $GPUS \
--config configs/${NAME}/${EXP_NAME}.yaml \
--exp_name ${EXP_NAME} \
--save_deform False