Git lone这个地址

<https://github.com/youyuge34/Anime-InPainting>

存在了f盘edge-ours数据

英伟达的补全网络对比看看

https://www.nvidia.com/research/inpainting/result

# 首先配置环境

建议新建一个conda虚拟环境

要求pytorch1.0.0 gpu版本的安装，不然各种报错。

可以离线安装，出错概率小一些

<https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/win-64/>

可以在上面找对应的版本装上

https://pytorch.org/get-started/previous-versions/

Torchvision安装0.2.2版本

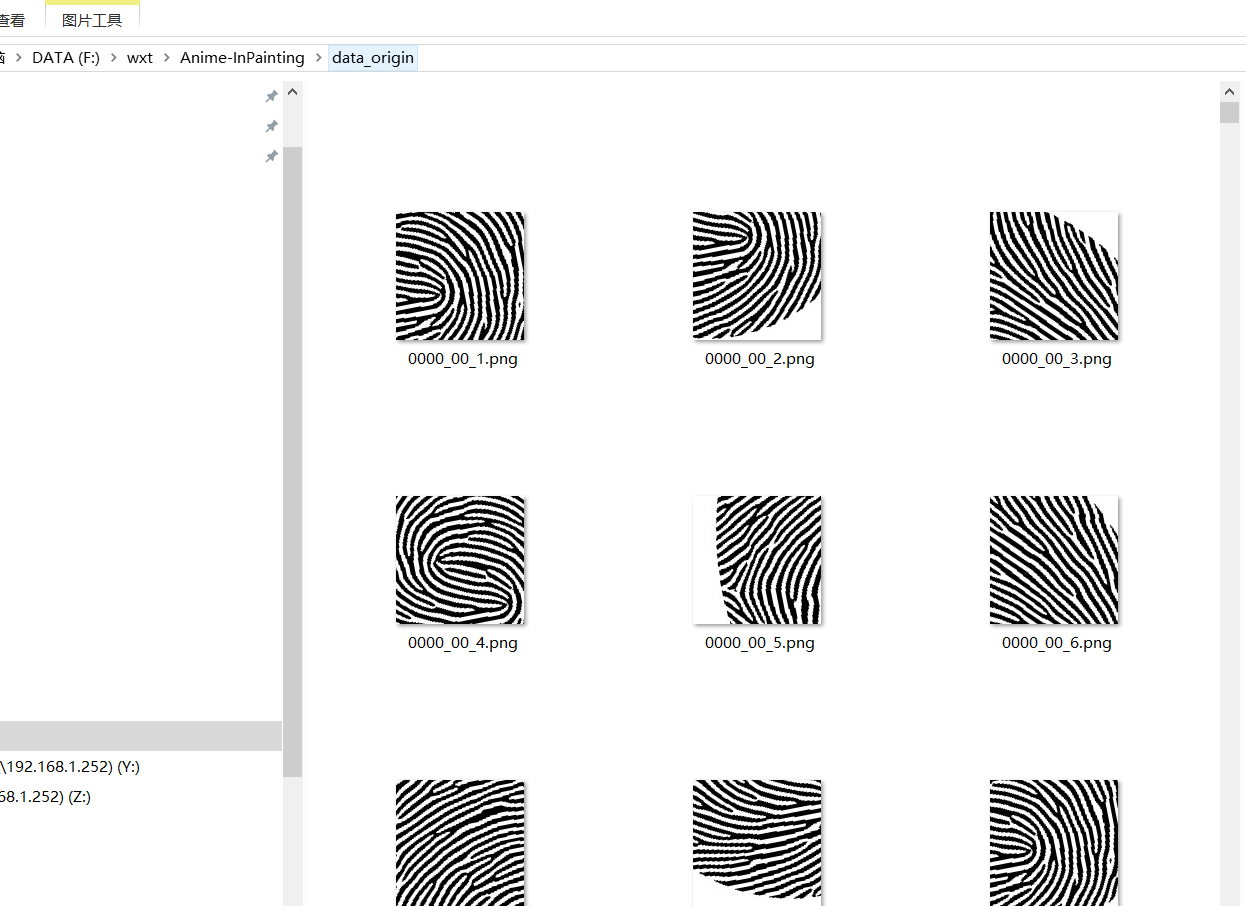
其他的包按照他的requirement.txt安装

# 数据集配置

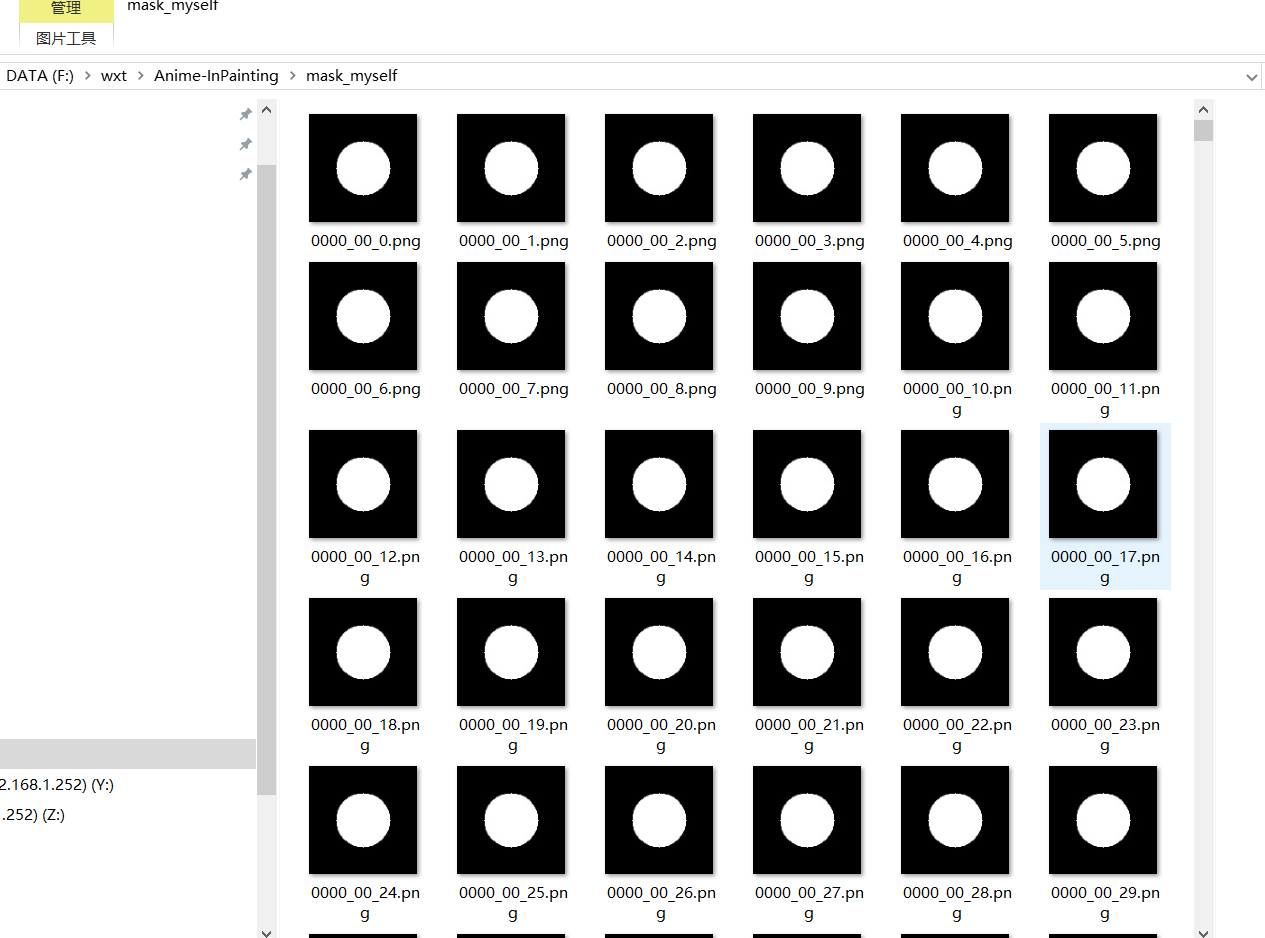
要求的是原数据图片，没有污损的。

和mask图片库，可以自己生成，或下载。

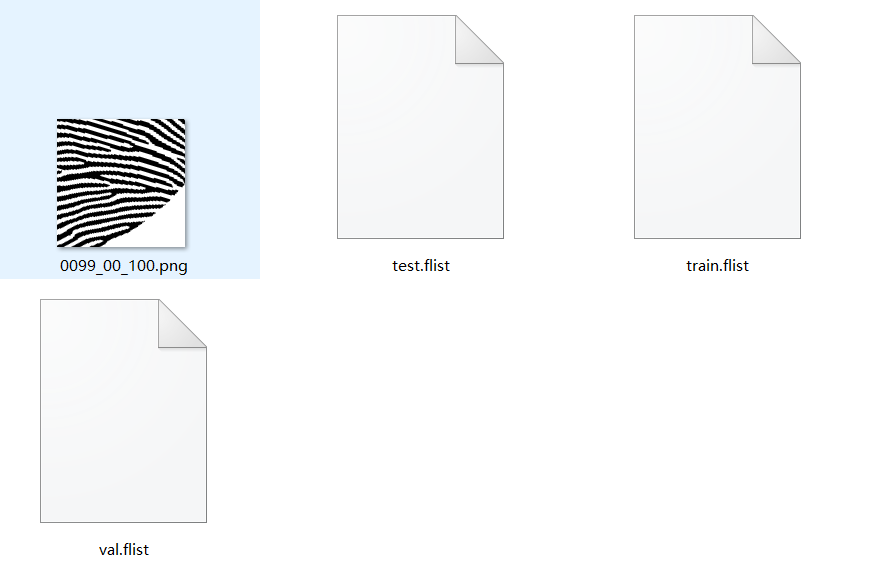
原数据集是：1w张



Mask集是：1w张



然后生成这三个列表



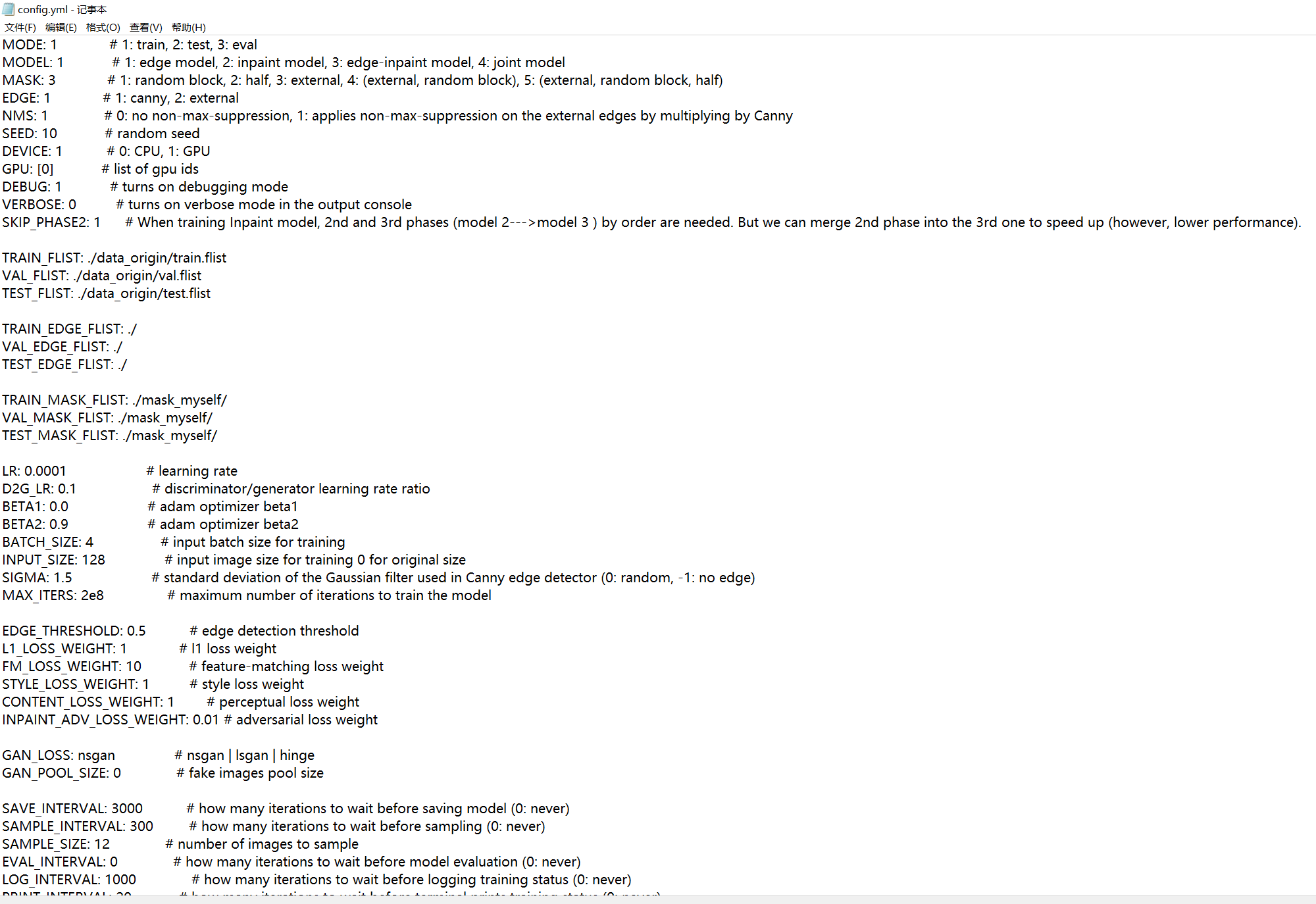
Console上运行

python scripts/flist\_train\_split.py --path <your dataset directory> --output <output path> --train 28 --val 1 --test 1

自动按28：1：1分配数据

然后修改config.yml

我的长这个样子



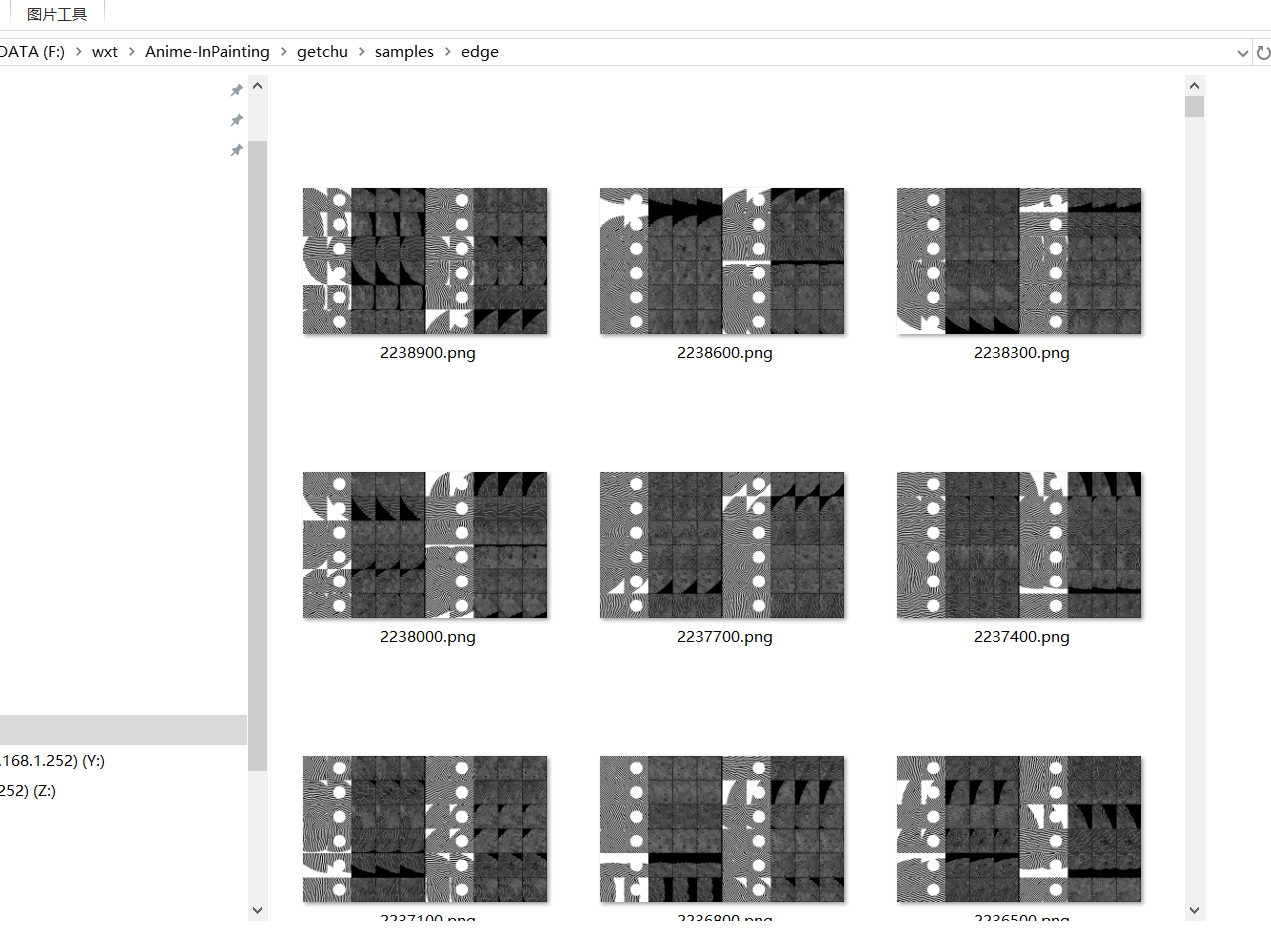
Mask那个配置写到flist的地方就行了，自动给你匹配一个个小mask文件。

训练的话是

python train.py --model 1 --path ./getchu

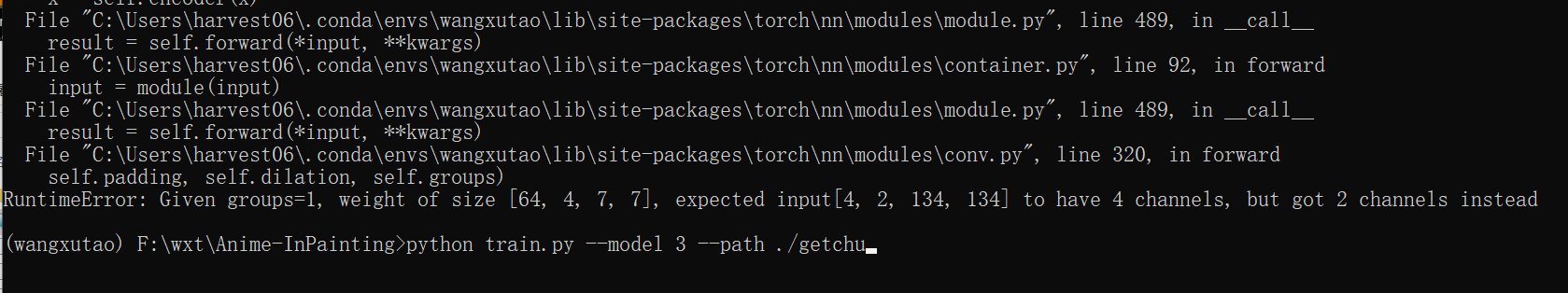
训练第一个阶段

阶段图在文件夹下有生成



训练阶段三有报错：

python train.py --model 3 --path ./getchu



4 与2的区别在于

原模型用的是彩色图，我的是黑白图，所以要修改。

1. 修改我的数据为彩色图（我在试这个）（所用的脚本是scripts文件夹下的转换脚本）（效果还行）
2. 修改模型的参数为2（这个，我怕牵一发动全身。。。）

# 测试数据

To test the model, create a config.yaml file similar to the example config file and copy it under your checkpoints directory. Read the configuration guide for more information on model configuration.

You can test the model on all three stages: 1) edge model, 2) inpaint model and 3) joint model. In each case, you need to provide an input image (image with a mask) and a grayscale mask file. Please make sure that the mask file covers the entire mask region in the input image. To test the model:

python test.py \

--model [stage] \

--checkpoints [path to checkpoints] \

--input [path to input directory or file] \

--mask [path to masks directory or mask file] \

--output [path to the output directory]

按他的要求还是输入那些彩色图，具体命令如下（试了几个）

python test.py --model 1 --checkpoint getchu\ --input data\_origin\rgb\_origin --mask mask\_myself\ --output ./result/Edge

python test.py --model 2 --checkpoint getchu\ --input data\_origin\rgb\_origin --mask mask\_myself\ --output ./result/Inpaint

python test.py --model 3 --checkpoint getchu\ --input data\_origin\rgb\_origin --mask mask\_myself\ --output ./result/Joint

python test.py --model 3 --checkpoint getchu\ --input test\test\finger\_image\_patch\_test\rgb\_origin --mask mask\_myself\ --output ./result/20200916

python test.py --model 1 --checkpoint getchu\ --input data\_origin\原训练图rgb --mask mask\_myself\ --output ./result/20200919

（这个的测试将edge-gen 过程的residual 的dilation 因子改为1，以前是2.就是空洞卷积的因子

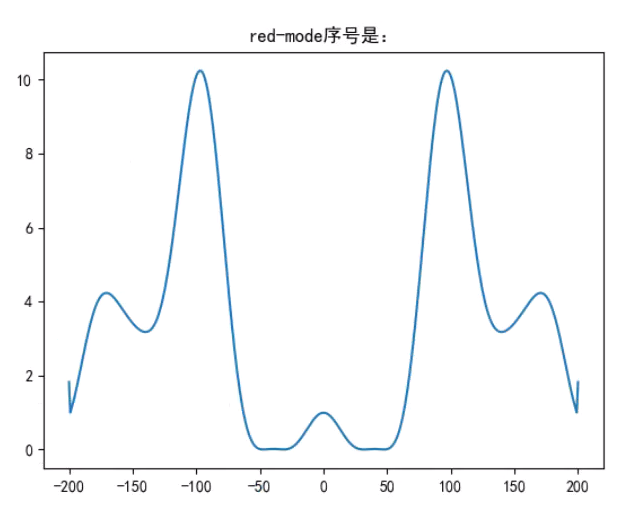
效果不知道算不算好，感觉没有很大的提高。）

# 后面的idea

9-8 idea

指纹的处理？

先对指纹提取特征点，对特征点的种类，位置进行训练



1999的传统算法

算法思想

<http://graphics.cs.cmu.edu/people/efros/research/NPS/alg.html>

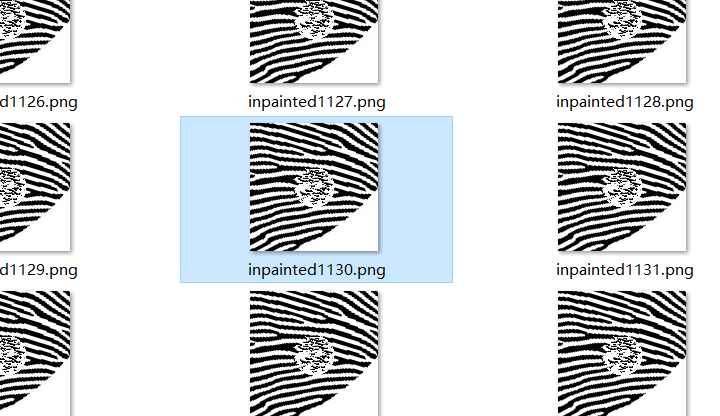
算法代码

<https://blog.csdn.net/a_rose_for_tang/article/details/108624598>

效果可太差了

Hhhhha//

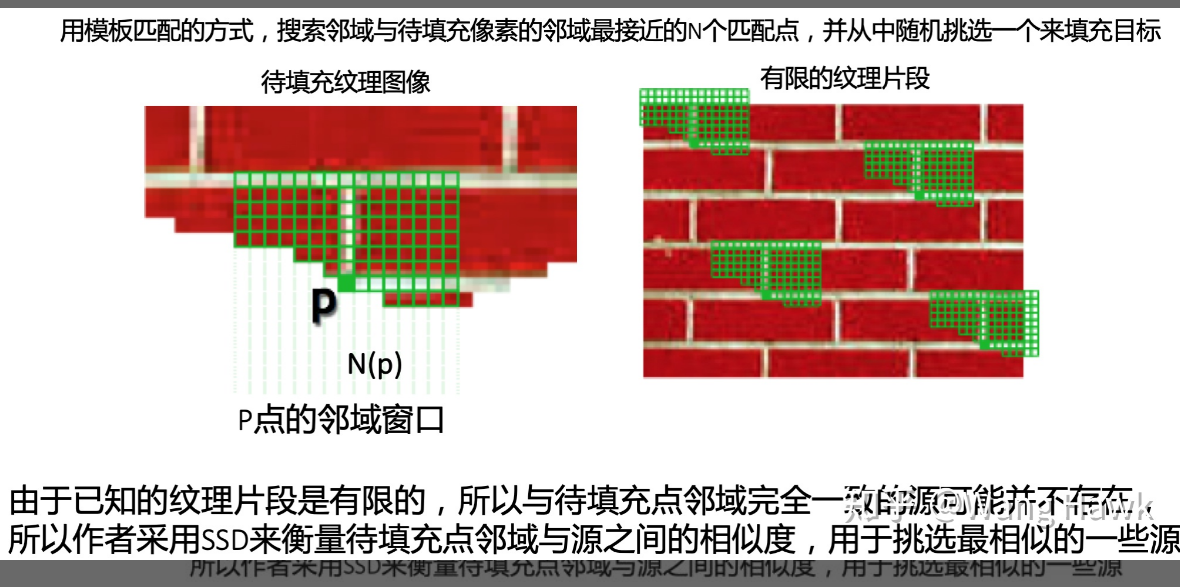
（对细化后的线条可能有好的效果修补）



？？？？

**Texture Synthesis by Non-parametric Sampling**

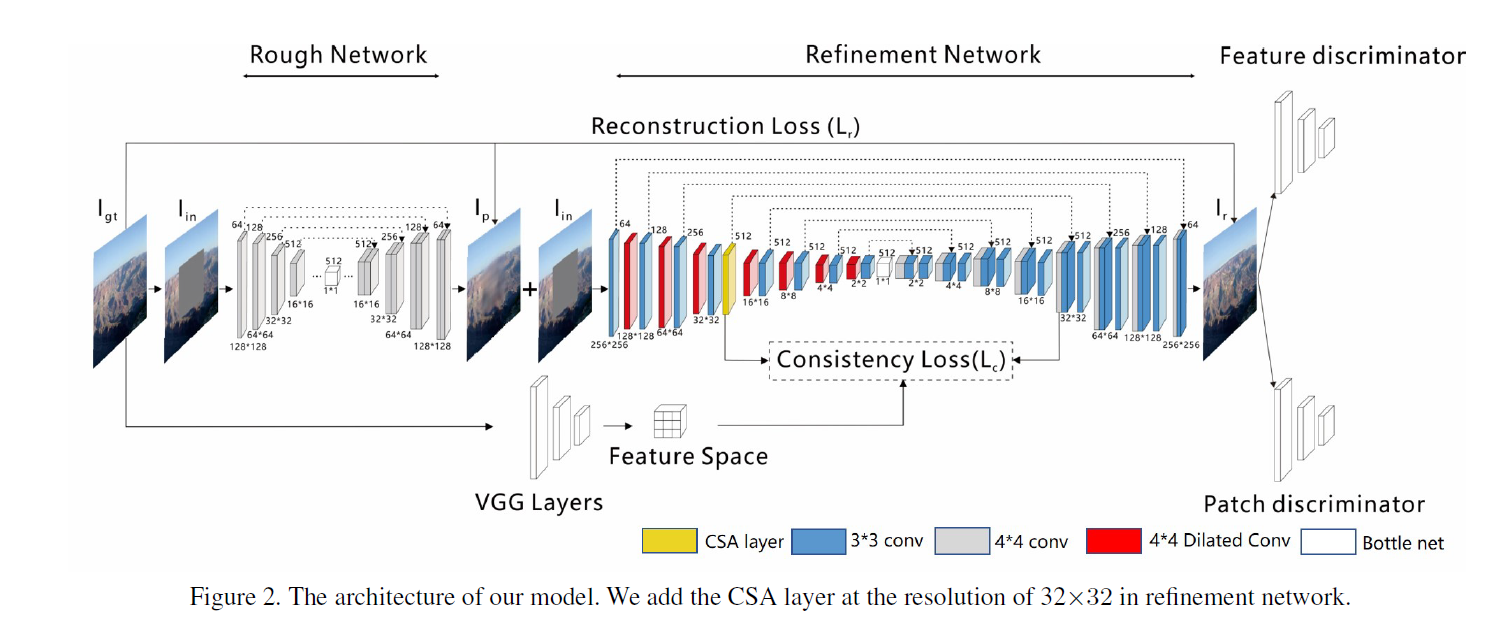
**模板匹配**





2019 Coherent Semantic Attention for Image Inpainting

粗化



思考：

不借助神经网络是否能够修复指纹纹路？这个问题需要嘉豪那边进行思考，我目前的看法是既然能通过播撒随机种子生成方向场信息然后生成指纹纹路，那么是否可以通过周围有指纹区域的方向场预测无指纹区域的方向场，然后生成纹路，达到修复的结果。

基于特征点的图像拼接方法捏？