# 一.项目简介

### 1.项目来源:

<https://github.com/xinntao/Real-RSRGAN>

### 2.环境安装



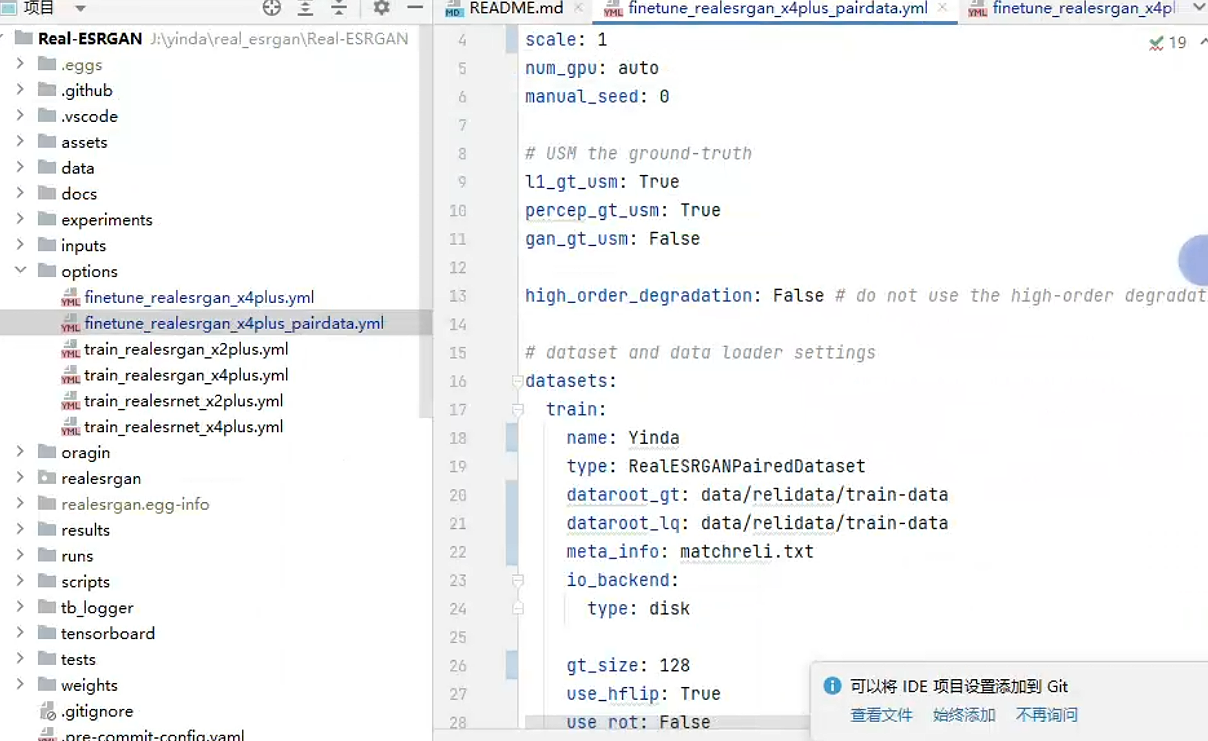
### 3.模型训练和推理

1.准备元信息 txt

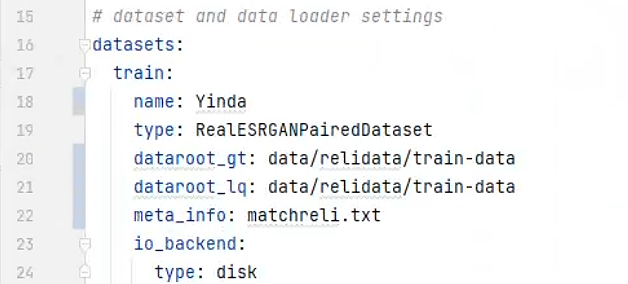
准备两个文件夹的数据，里面的数据要名字一一对应，数量也要对应

|  |
| --- |
| python scripts/generate\_meta\_info.py --input datasets/DF2K/DF2K\_HR, datasets/DF2K/DF2K\_multiscale --root datasets/DF2K, datasets/DF2K --meta\_info datasets/DF2K/meta\_info/meta\_info\_DF2Kmultiscale.txt |

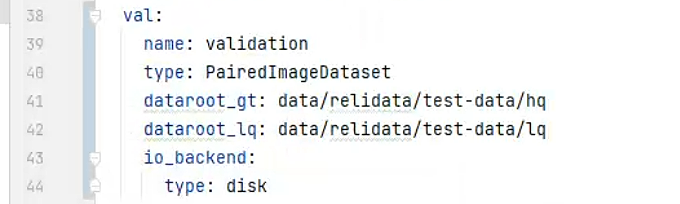
2.修改配置文件



gt为高质量的图片路径，lq为低质量的图片路径，meta\_info为上面得到的txt的路径（这个路径需要跟txt里面的路径拼接）



同理这个是为val但是不需要跟txt文件拼接



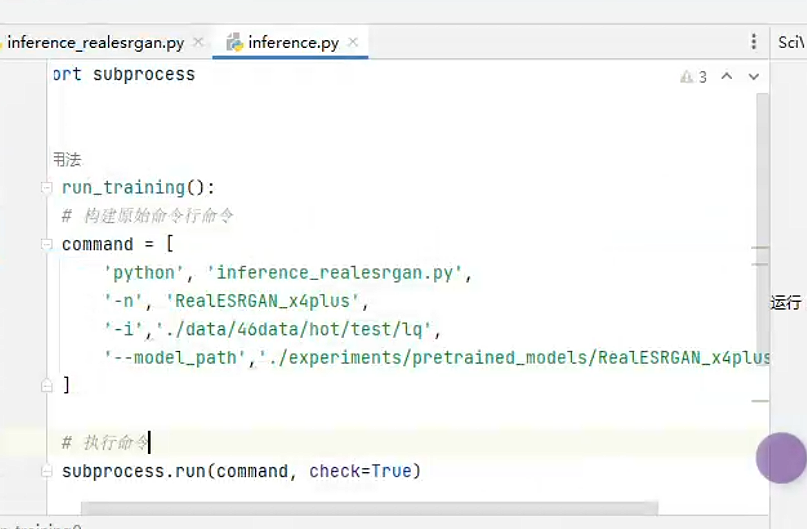
3.训练

运行start.py文件即可。

1. 验证

运行inference.py

如果在训练的代码中得到了新的生成器的model，那需要把--model\_path中修改路径



# 模型介绍

### 任务说明

我的任务是修复电子皮肤的热力图变成mask图，因此在这个对抗网络模型中，我主要使用其中的生成器模型，因此后面只介绍生成器模型。

### 超参数介绍

我使用的配置文件是这个finetune\_realesrgan\_x4plus\_pairdata.yml，

Batchsize为16，生成器的输入通道为3，num\_feat为64，num\_block为23，num\_grow\_ch为32，lr为1\*10-4，lr变化函数是MultiStepLR，优化器是Adam.

### 3.生成器loss

(1)l\_g\_gan生成器的gan损失，这个损失指的是生成器生成的图片在判别器中的判对损失。

(2)l\_g\_percep生成器的感知损失，这个损失指的是生成器生成的图片和原图经过一个特征提取器如VGG16得到的特征的损失。

(3)l\_g\_pix是生成器的像素损失，即生成器生成的图片跟原图的像素的绝对误差。

# nni实验

### 模型指标

我采用了两个指标，一是prsn另一个SSIM

PRSN 是一种通过计算原始图片跟生成器生成图片的像素绝对值差异得到。

SSIM它基于三个方面的图像比较：亮度、对比度和结构。SSIM 指数的取值范围为 [-1, 1]，其中 1 表示两幅图像完全相同，-1表示两幅图像完全不同。

### nni调参的内容

|  |
| --- |
| {  "model\_num\_block": {  "\_type": "choice",  "\_value": [17,23,29,31,37,41,47,53]  },  "model\_num\_grow\_ch": {  "\_type" : "choice",  "\_value" : [20,24,28,32,36,40,44,48]  },  "model\_num\_feat":  {  "\_type": "choice",  "\_value": [48,64,96,128,160]  } } |

遍历的有num\_block,grow\_ch,num\_feat,遍历列表的如上。