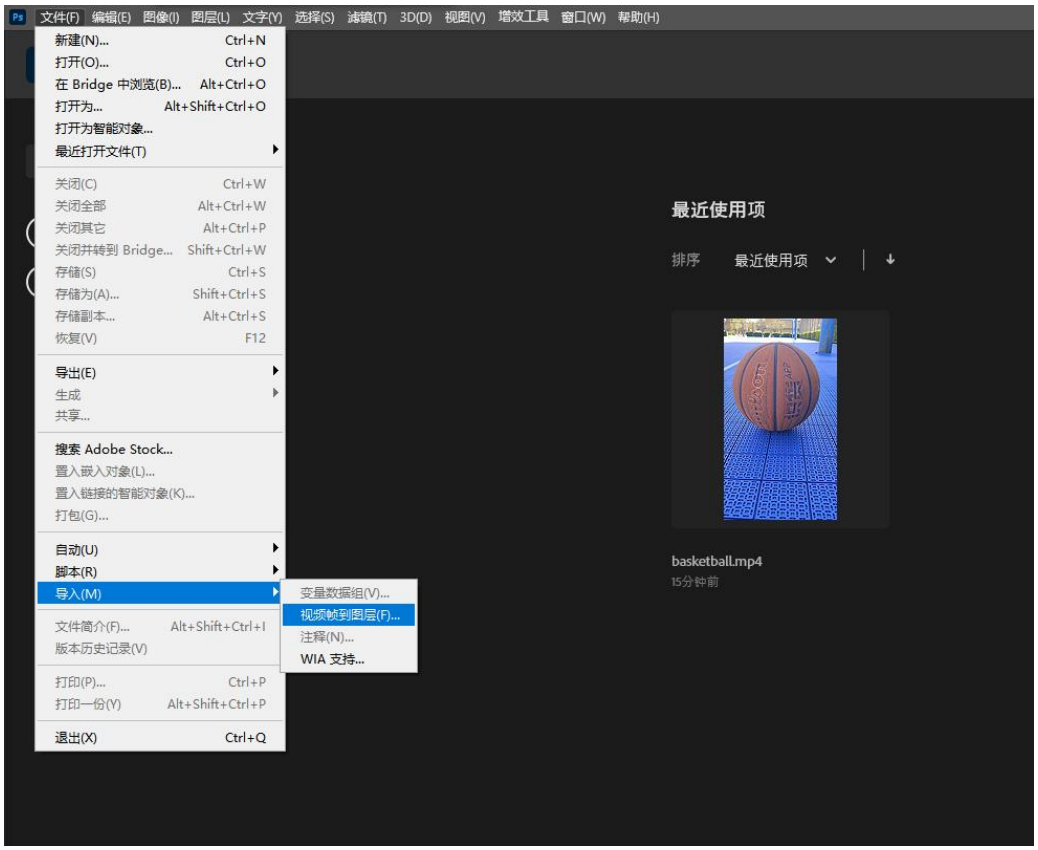
TASK3 使用具有泛化能力的 NeRF 模型，自己构建物体数据集（如手机拍摄），对物体进行三维重建

nerf_pytorch 训练自己的数据集

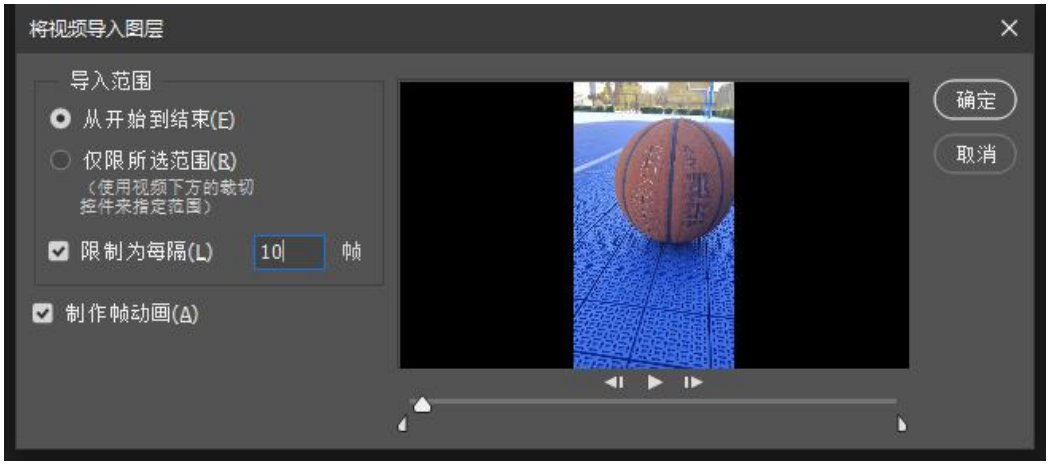
1、制作数据集

可以用手机拍摄一段视频，然后使用 PR 进行抽帧导出为图片。

把视频导入为视频帧到图层



我这里设置为每 10 帧抽取一张图片

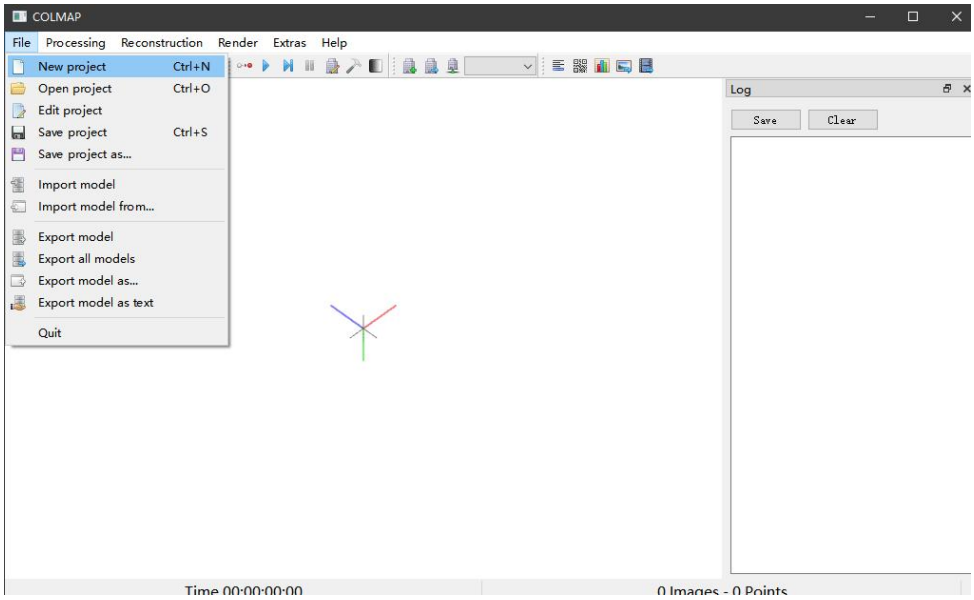


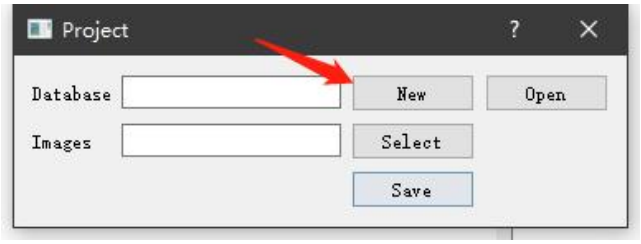
批量重命名文件（因为图片不能包括中文路径，如果包含在后续处理会有各种问题报错，而且为了美观，这里对图片重新命名）选择图片。以下展示重命名完成后的情况。

2、根据数据集使用 COLMAP 获取相机位姿

LLFF 格式数据可以将对应图片参数、相机位姿和相机参数简洁有效地存储在一个 npy 文件中，以方便 python 读取，且 NeRF 模型源码拥有直接对 LLFF 格式数据集进行训练的配置和模块，便于使用。

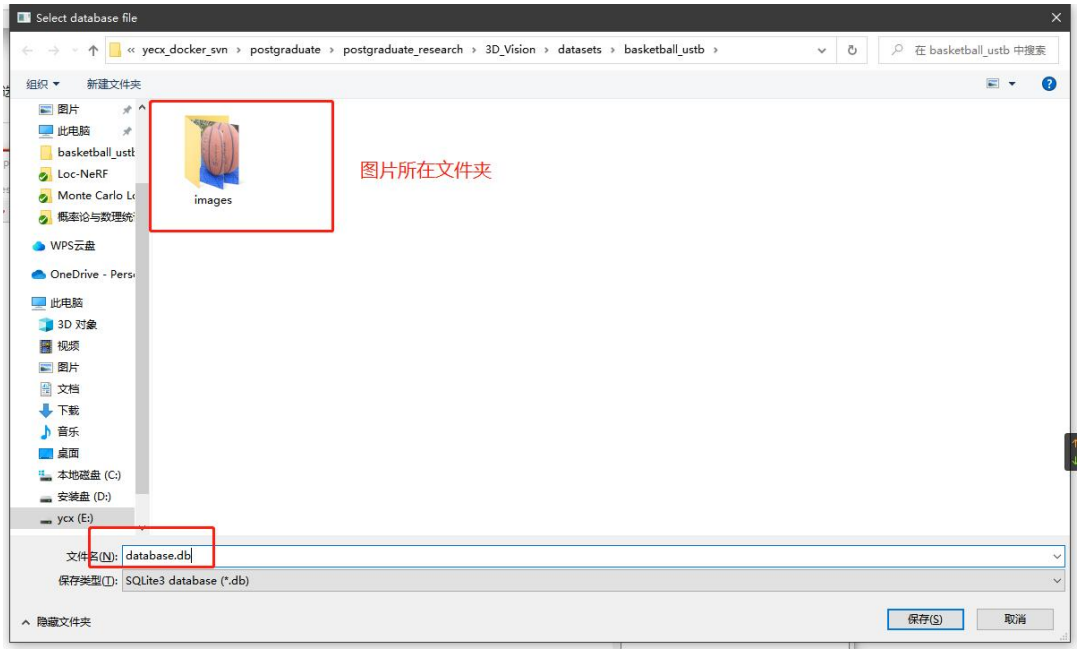
首先下载 COLMAP 软件，解压后，运行。



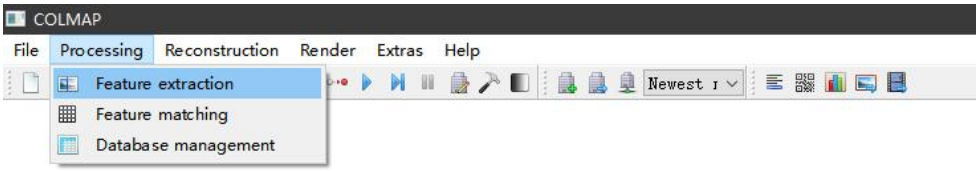


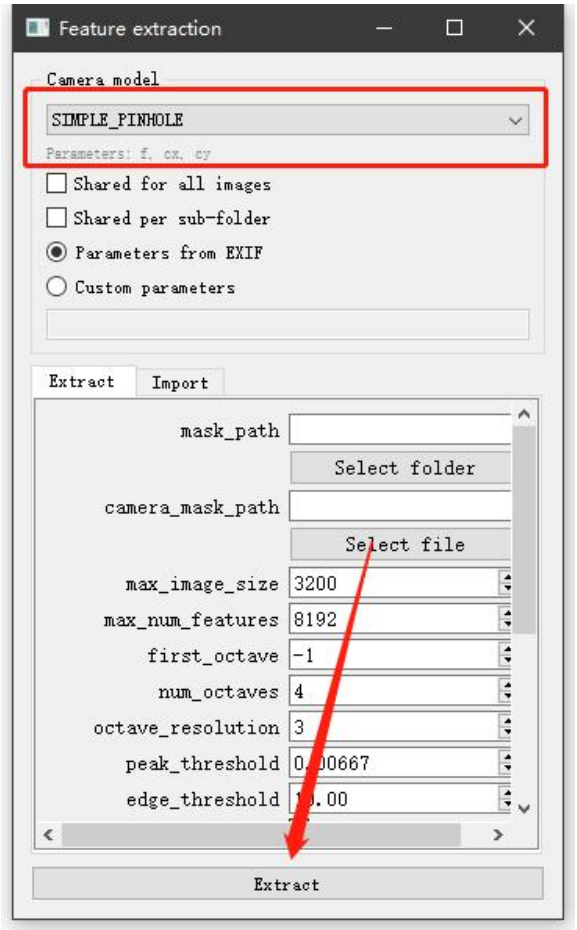
此时，需要将新建文件路径定位到我们的工作路径（新建一个文件夹并命名为自己数据集的名称，这里我们创建为/COLMAP_test/，注意，该文件夹的完整路径中不能有中文）；接着，在工作目录下创建/images/文件夹并将图片存放在这里；然后，在工作目录下创建 database.db 文件（需要手动输入）后点保存。

然后，选择图片路径，点击 Select 并选择刚才存放图片的/images/文件夹后保存，点击 Save

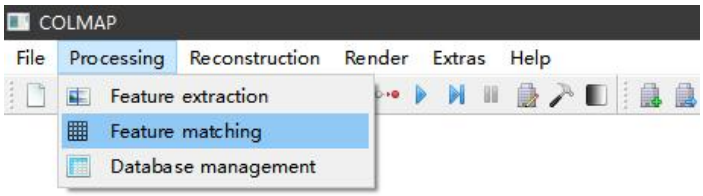


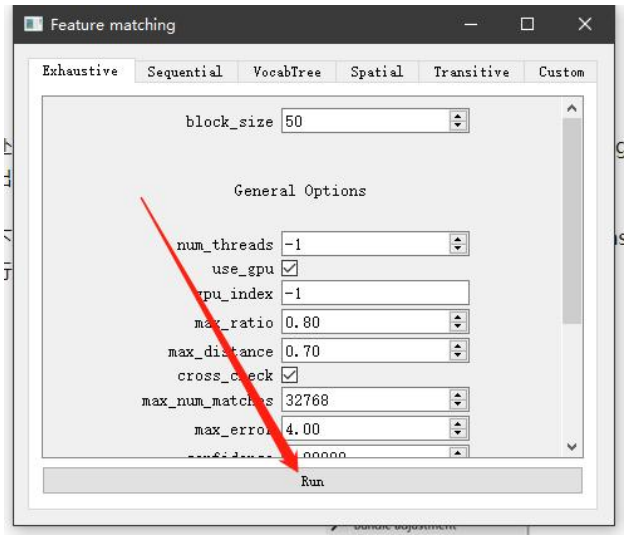
进行特征提取



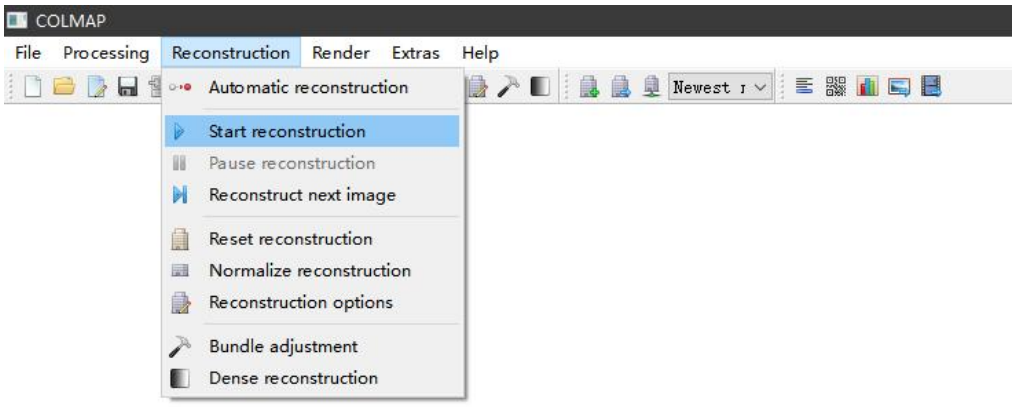


待特征提取完毕后，关闭 Feature extraction 窗口。并点击 Processing -> Feature matching 进行特征匹配

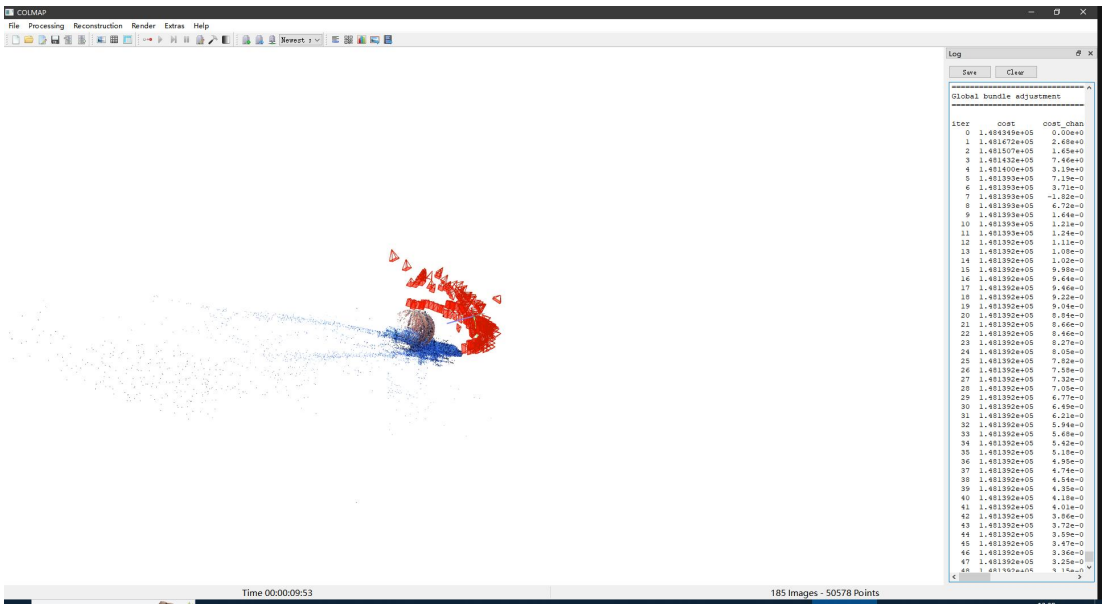




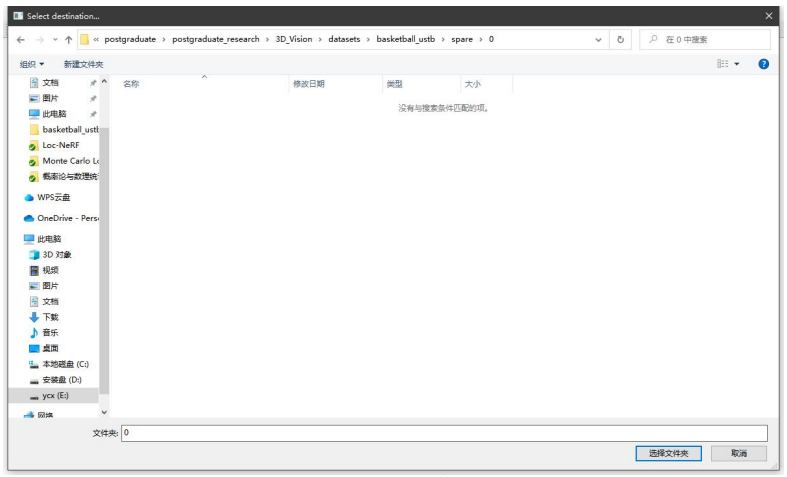
接下来对匹配到的点进行稀疏重建，点击 Reconstruction -> Start reconstruction 此时将会开始进行重建

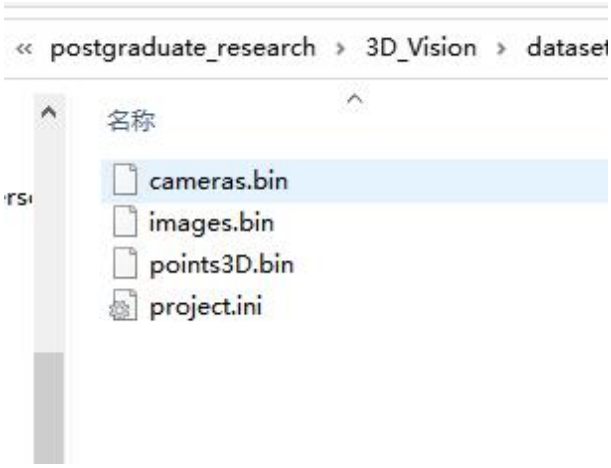


根据匹配到的特征点数目，此过程可能会持续一段时间，在窗口中可以看到重建过程。重建完毕后，得到如下图，可以通过右下角 Images 和 Points 来判断是否重建成功



最后，点击 File -> Export model 以导出模型，注意，请在工作目录下新建/sparse/0/文件夹，并将模型导入到该路径下例如（./sparse/0/），选择改文件夹。



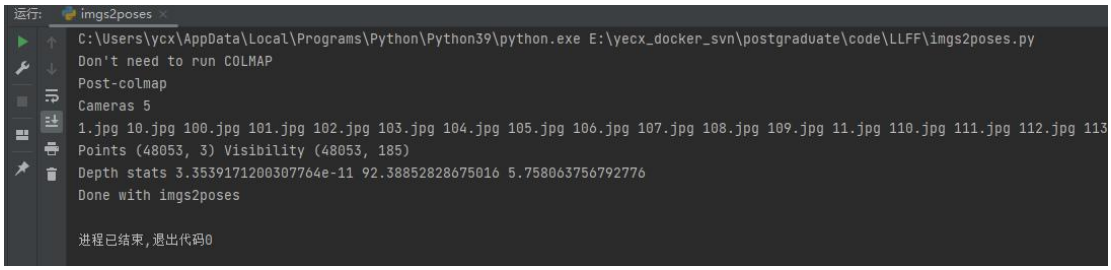


此时，将会在该目录下得到如下文件。至此，第一步位姿获取步骤完成

3、使用 LLFF 脚本对位姿数据进行格式转化，把数据集转换成 LLFF 格式数据集

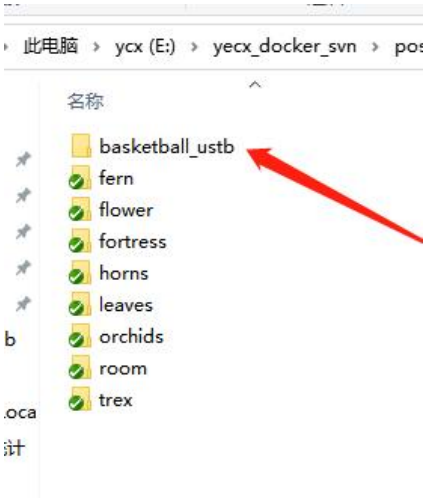
得到 COLMAP 位姿匹配数据后，我们要对每张图片的位姿信息进行格式转换，转换为 LLFF 格式方便 Nerf 模型读取。打开 LLFF 脚本 https://github.com/Fyusion/LLFF

打开 imgs2poses.py 文件，修改如下内容，改为刚才的工作目录，然后在终端运行该代码，例如：



4、把处理好的数据集配置到 Nerf 代码目录下

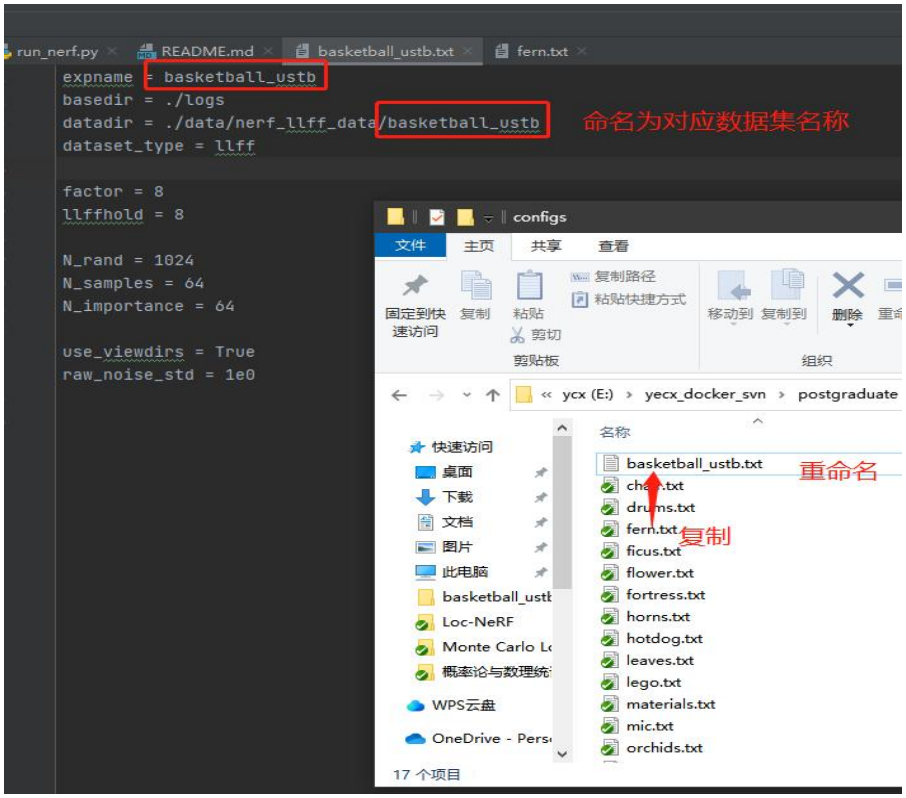
将整个工作目录的文件夹复制到 nerf 代码的/nerf-pytorch/data/nerf_llff_data/目录下



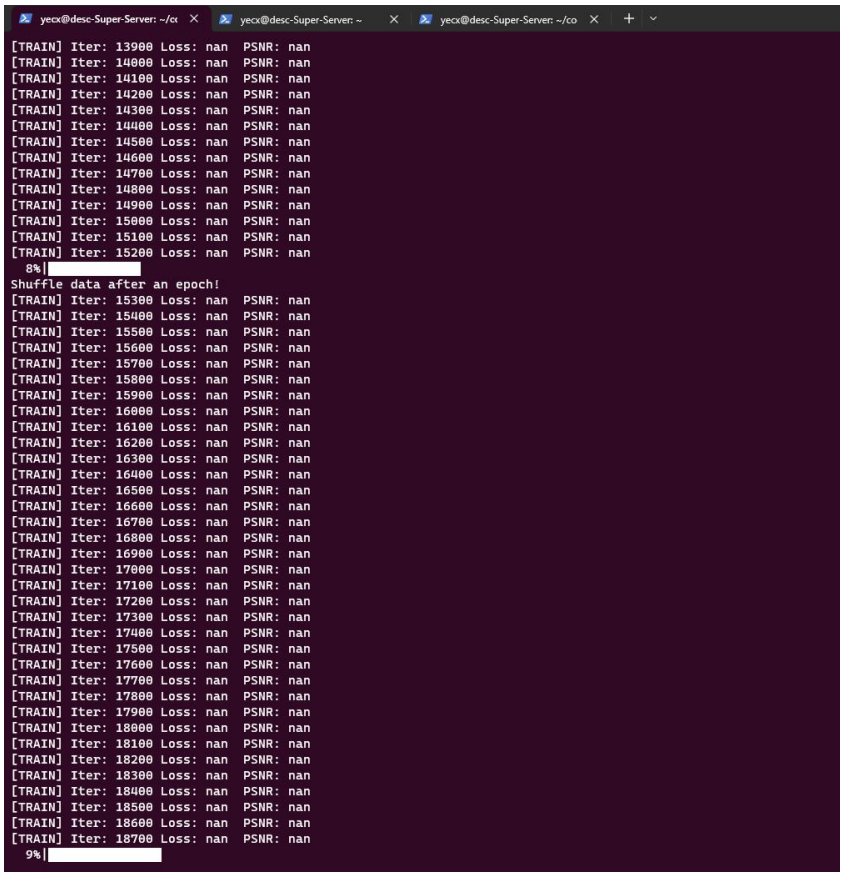
注意，之前/images/文件夹中的图片，只能保留已经匹配上的图片（匹配成功的图片名称及数目可以如步骤 2 最后所提到的方法来查看）

接着，需要设置配置文件。

在 NeRF 代码目录下，复制/nerf-pytorch/configs 目录下的 fern.txt 文件，并重命名为自己模型的名称（建议和工作目录名称一致），并修改如下内容：



最后就可以运行训练代码了



训练结果：但是毫无效果，一开始训练就 nan，试了调小学习率但没啥用。