边界分割总结：

1. 获得图像数据集：

手动挑选所需要的图像

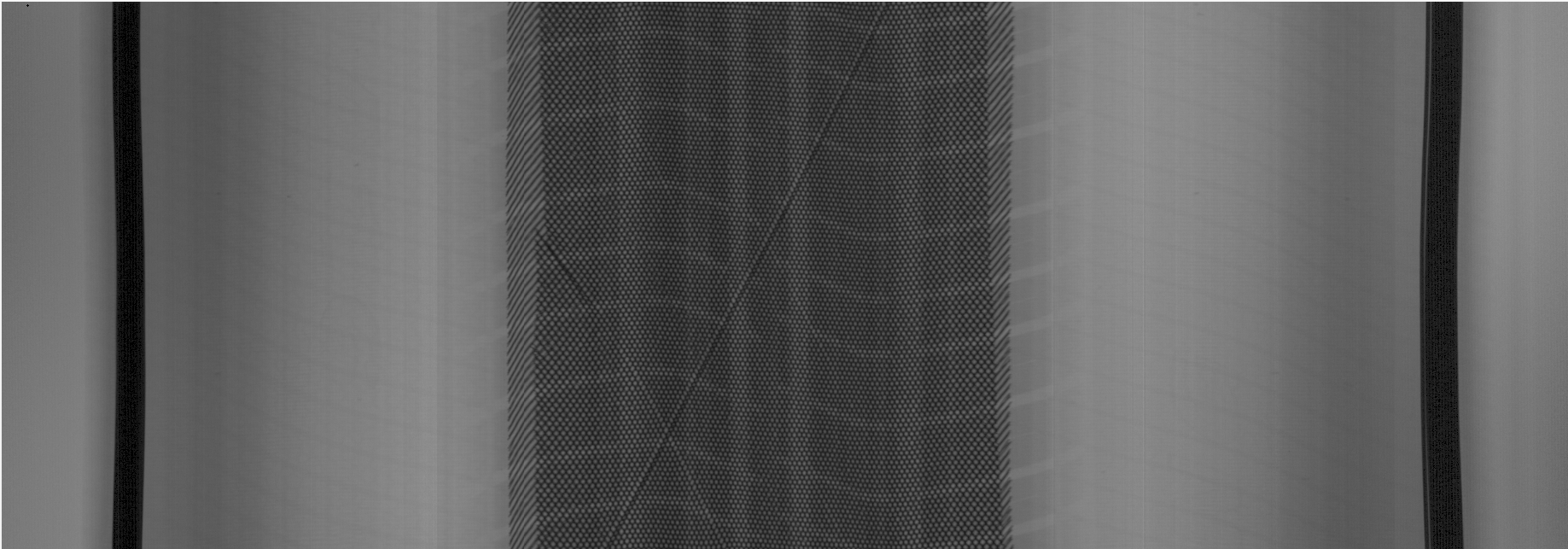
注意：全程路径禁止中文

注意：图像名称中严禁有空格，否则程序崩溃

F:\modle\_data\split\image\6thFactory#23-20200528

1-5：代表之前的数据集

6\_big：初步挑选的图像（10）张；



2right

2left

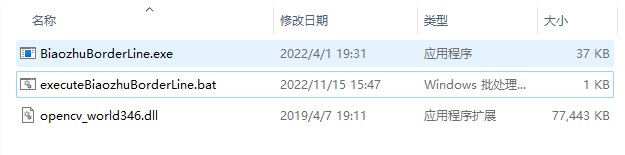
1right

1left

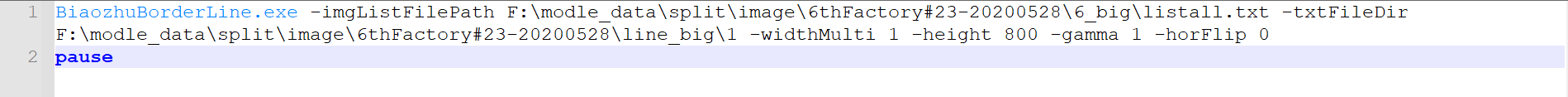
1. 使用标注程序【人工为所需要的边界进行标注】

上图为例需要标注四条边界从左到右：1left、2left、2right、1right

标注工具：F:\modle\_data\Line\_PCR\标注边界程序



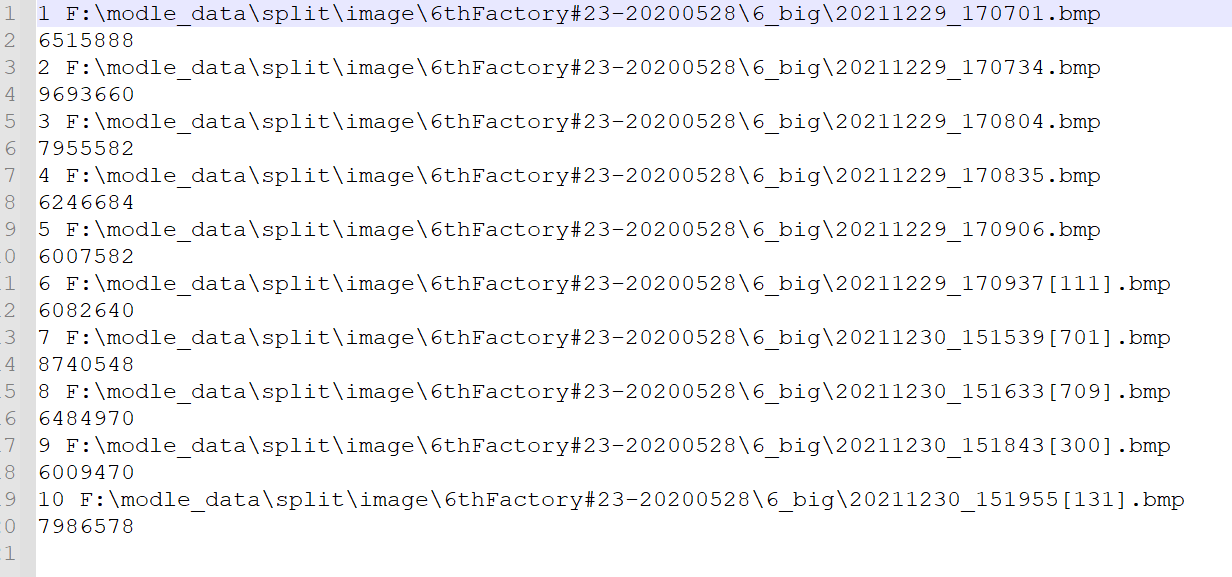
编辑选择路径：

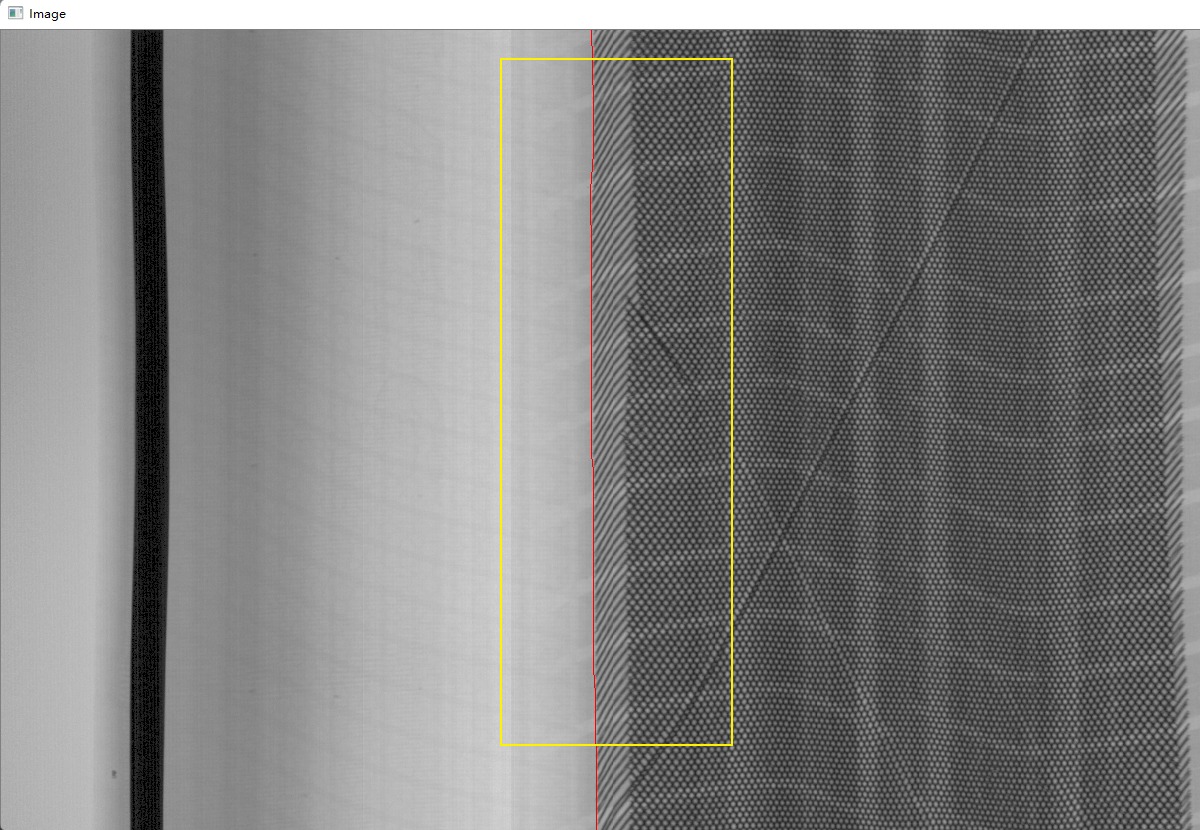


ImgListFilePath：表示原图目录，listall为该路径下图像的统计

TztFileDir：表示标签目录

Listall生成方法：使用testall2软件





标注的边界：代表从左到右1left、2left、2right、1right

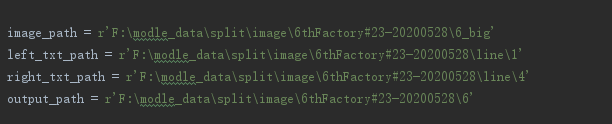


1. 根据标签边界进行图像切割（只留下带束层区域）：

使用程序：F:\modle\_data\Line\_PCR

CutRoughAreaWithBoundaryTxt.py

参数说明：



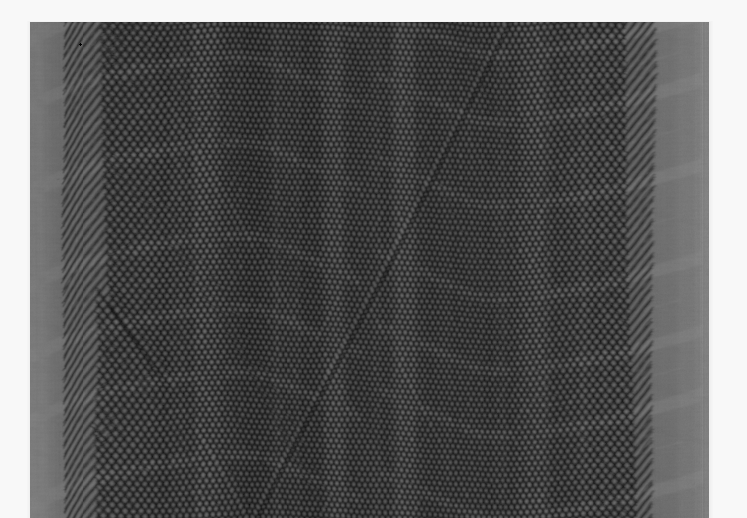
Image\_path：原图路径

Left\_txt\_path：第（2）步标注好的左边界，最外侧

Right\_txt\_path：第（2）步标注好的右边界，最外侧

Output\_path：输出切割好的边界

切割后的图像：只剩下中间区域

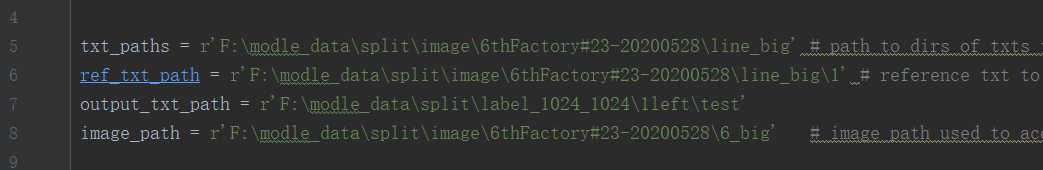


1. 根据标签边界进行边界切割（之前是整张图像上的边界，现在为切割后的图像边界）：

使用程序：F:\modle\_data\Line\_PCR

OffsetBoundariesWithRoughBoundary.py

参数说明：同时对所有标签进行切割



txt\_path：原标签路径：注意此处为标签的上一级路径，也就是该路径下有1,2,3,4,四个文件夹

ref\_txt\_path：最左侧的边界

out\_txt\_path：输出的边界值

image\_path：之前的图像路径（未切割之前）



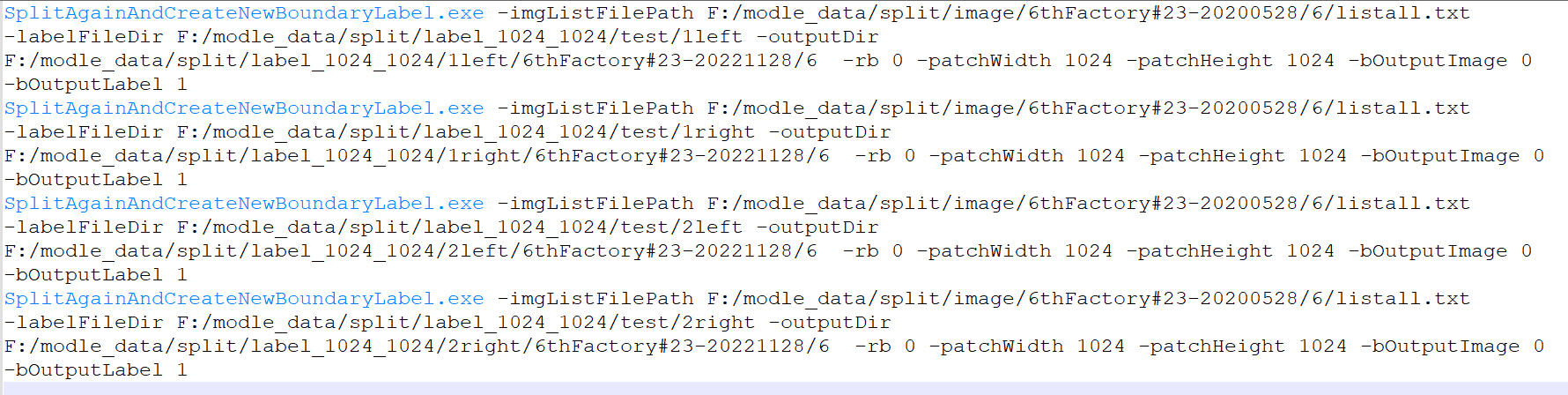
名字是自己改的，之前是1,2,3,4

1. 竖直方向切割小pach（图和标签）：

使用脚本：F:\modle\_data\split

executeSplitAgainAndCreateNewBoundaryLabel-to1024-1024.bat（使用前编辑参数）

参数说明：



ImgListFilePath：第（3）步粗切割后的图像

LableFiledir：第（4）步粗切割后的边界

OutPutDir：输出路径，

切割标签时：依然是按照四条边界分为四个文件夹输出的

切割图像时：使用一个输出目录即可

OutPutImage：是否切割图像（1是，0否）

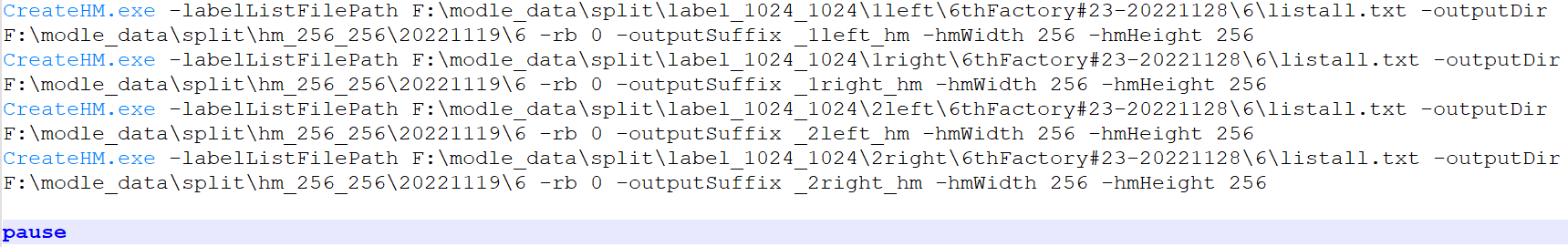
OutPutLable：是否切割标签（1是，0否）

1. 根据lable标签生成热图（热图：训练时所用的标签）：

使用脚本：F:\modle\_data\split

executeCreateHM.bat（使用前编辑参数）

参数说明：



LableLixtFilePath：第（5）步切割后的标签

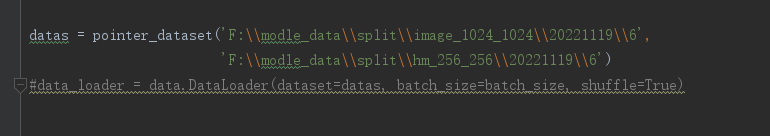
OutPutDir：输出的热图路径（热图只需要放在一个文件下即可）

1. 训练模型：

所用程序：F:\modle\_data\Line\_PCR

train.py

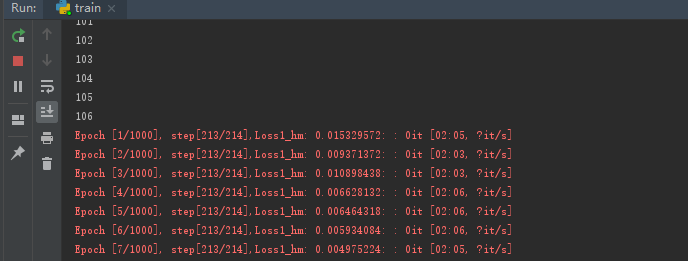
参数说明：



第一个路径参数表示第（5）步切割后的图像路径（输入）

第一个路径参数表示第（6）生成的热图路径（标签）

运行，开始训练：



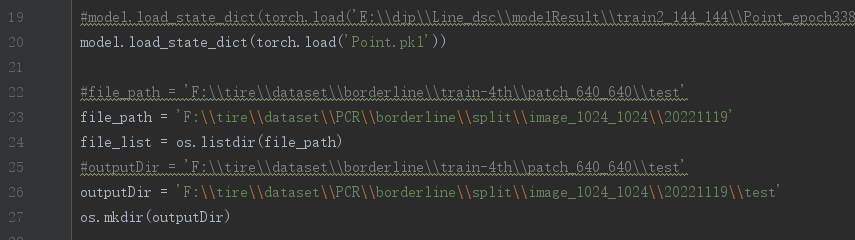
生成模型文件： Point.pkl

1. 转换为系统可以调用的模型文件参数

所用程序：F:\modle\_data\Line\_PCR

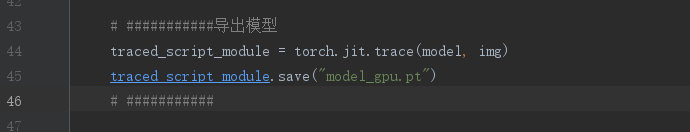
test\_split.py

参数说明：



Model：步骤（7）生成的模型文件

File\_path：测试集位置

OutputDir：输出测试热图  


导出可运行模型：model\_gpu.pt