

更新记录

2023.2.23

json转yolo

背景置黑裁剪

2023.2.21

命名纠正

小图分类重新生成json

2023.2.2

绘制标注

拆分数据集

分割类别小图

2023.1.31

读取json

按比率随机切分数据集

按工位-形态切分数据集

修复推理json绘制问题

2023.2.23

json转yolo

更新记录：1. 添加json转yolo 分割模式

代码接口：

```
jf = r"D:\PAD\noline\images\val\val.json"
img_path = os.path.dirname(jf)
dst = os.path.dirname(img_path)+"\labels"
api.json_to_yolo_api(img_path, jf, dst, seg=True)
```

背景置黑裁剪

更新记录：1. 缺陷以外所有背景置黑
2. 置黑小图裁剪并填充

代码接口：

```
jf = r"D:\PAD\line\images\val3\ng.json"
img_path = os.path.dirname(jf)
dst = img_path+"_fill"
api.fill_crop_via_api(img_path,jf,dst,crop=True,pad=0.5)
```

2023.2.21

命名纠正

更新记录：1. 现场命名中的中文删除
2. - 和 _ 命名不规则修复

代码接口：(full_dpp.ipynb)

```
img_path = r""
jf = None
dst = r""
start = ""
api.check_filename_api(img_path,jf,dst,start):
```

小图分类重新生成json

更新记录：1. 添加分类过程中某些缺陷需要重新标注或者删除存放指定文件夹后筛选中原图及json；
固定名称：mark del 文件

代码接口：

```
img_path = r"D:\PAD\noline\images\val_classify"
jf = r"D:\PAD\noline\images\val\val.json"
dst = os.path.dirname(jf)
api.classify_json_api(img_path,jf,dst)
```

2023.2.2

绘制标注

更新记录：1. 修复绘制多个JSON绘制少图问题,以原始标注为基准；支持自定义类别

代码接口：

```
import api,os
jf = r"C:\Users\lubin\Desktop\pad\NG\ng.json"
inf_jf = r"C:\Users\lubin\Desktop\pad\NG\model_0489999.json"
img_path = os.path.dirname(jf)
dst = img_path+"_rect"
draw={"shape":"RECT","label":True,"score":False,"area":False,"color":False,"font_size":1,"classid":False}
api.draw_mask_api(img_path, jfs=[jf,inf_jf], dst=dst, spotcheck=False,draw=draw)
```

拆分数据集

更新记录：1. 添加随机种子seed

代码接口：

```
jf = r"D:\PAD\images\images\data_merge.json"
img_path = os.path.dirname(jf)
repeat = 2
ratio = 1/4 # (训练集3/测试1)
api.dataset_partition_api(img_path,jf,repeat,ratio,seed=88)
```

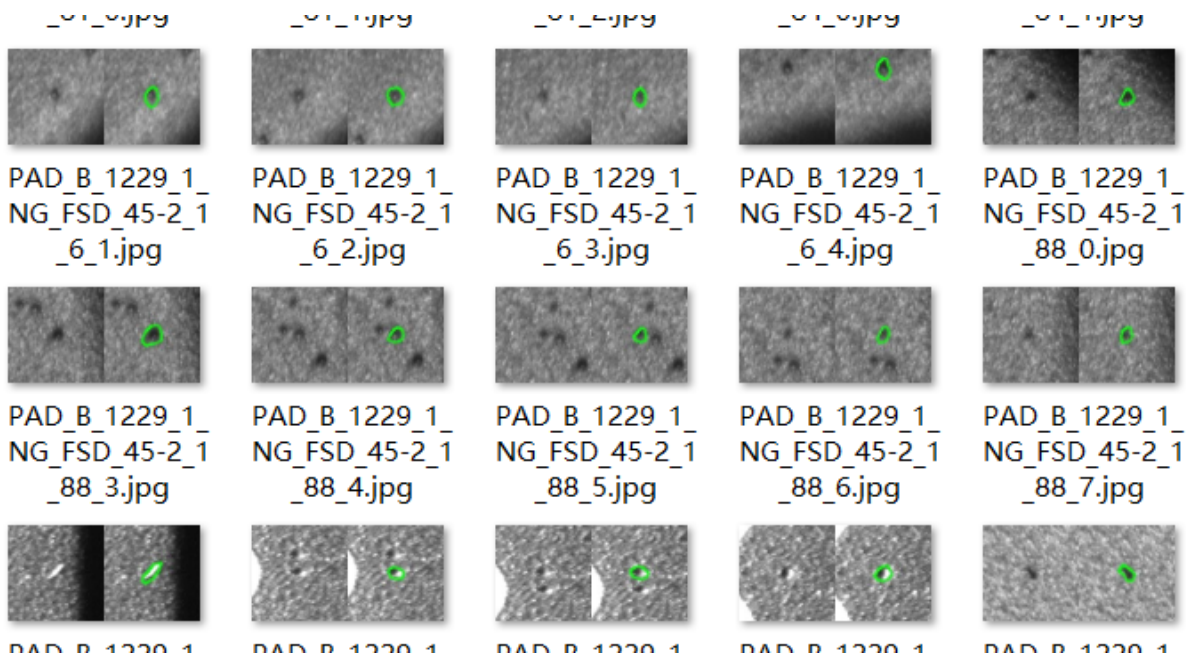
分割类别小图

更新记录：1. 修复一张缺陷小图中显示多个标注

代码接口：

```
jf = r"D:\PAD\ng\origin_mask.json"
img_path = os.path.dirname(jf)
dst = img_path+"_classify"
api.crop_small_img_api(img_path, jf, dst, scale=True,offset=50,crop_size=120)
```

效果：



2023.1.31

读取json

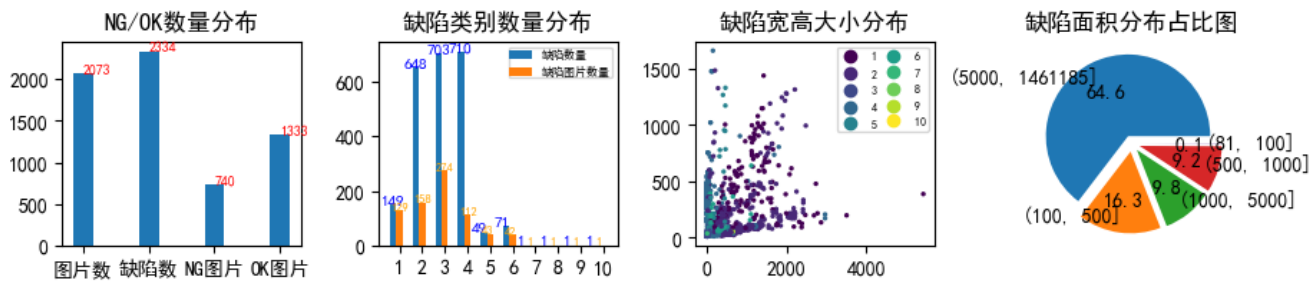
- 更新记录：1. 添加类别对应图片数量
2. 修复缺陷宽高大小分布图legend显示问题
3. 验证多浏览器支持

代码接口：

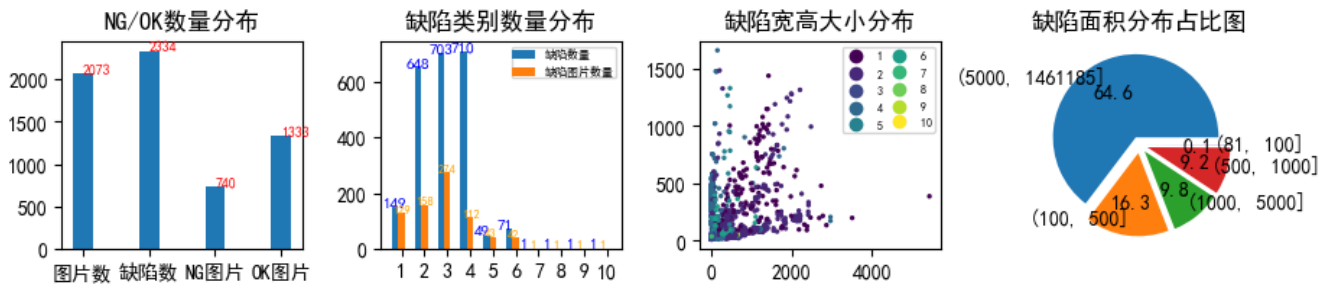
```
jf = r"D:\PAD\data_merge.json"
api.read_json_api(jf,mini = False)
```

效果：

谷歌



edge



按比率随机切分数据集

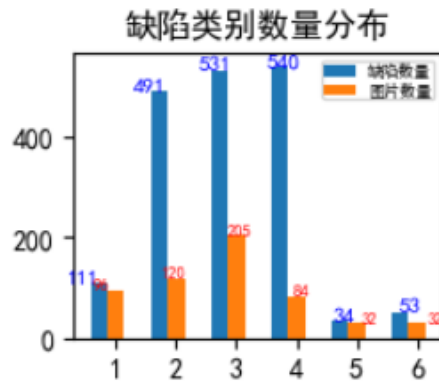
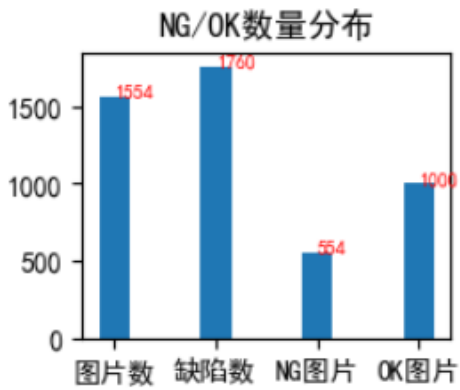
- 更新记录：
1. 缺陷按比值切分数据
 2. 修复一张图片缺陷过多时样本划分比例失调
 3. 修复切分小图后显示多个缺陷

代码接口：

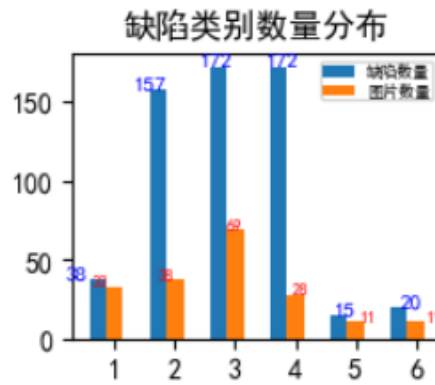
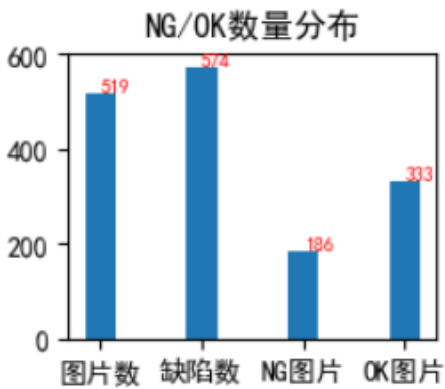
repeat	多缺陷降维
ratio	测试集/总数据集比值

```
jf = r"D:\PAD\ng\origin_mask.json"
img_path = os.path.dirname(jf)
repeat = 2
ratio = 1/4 # (训练集3/测试1)
api.dataset_partition_api(img_path,jf,repeat,ratio)
```

效果：



```
jf = r"D:\PAD\images\test\filter_json.json"
api.read_json_api(jf,mini = True)
```

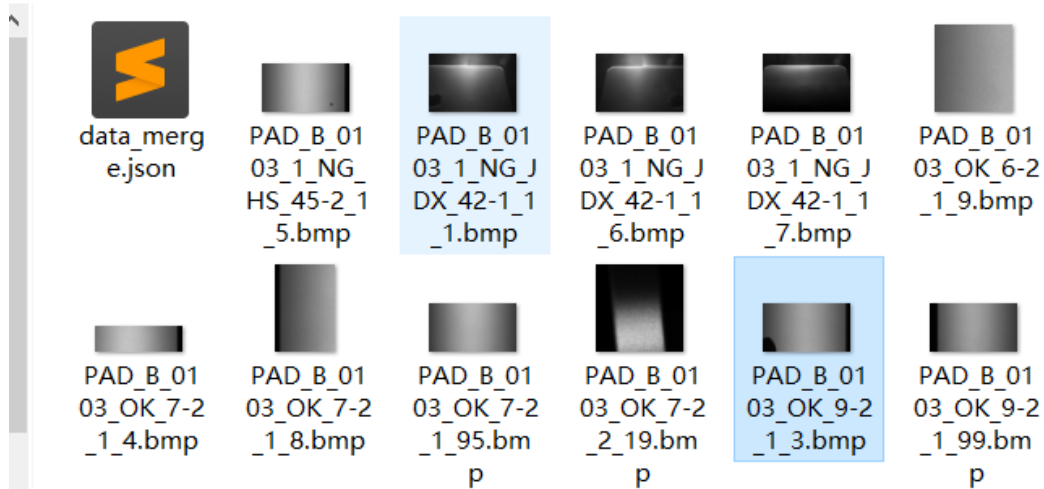


按工位-形态切分数据集

- 更新记录：
1. 按工位切分后，再按形态分类，添加人工操作能够使数据分类更精准
 2. 类别数量较少缺陷占比相对提高

代码接口：

i:) > PAD > images

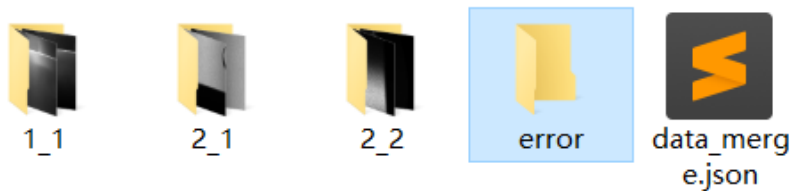


1. 图片按工位切分,无点位信息图片自动移至error文件夹, 每个文件夹中有n张图片;

手动调整各文件夹之间不合理点位文件分布继续第二步

```
img_path = r"D:\PAD\images"
api.split_img_by_station_api(img_path)
```

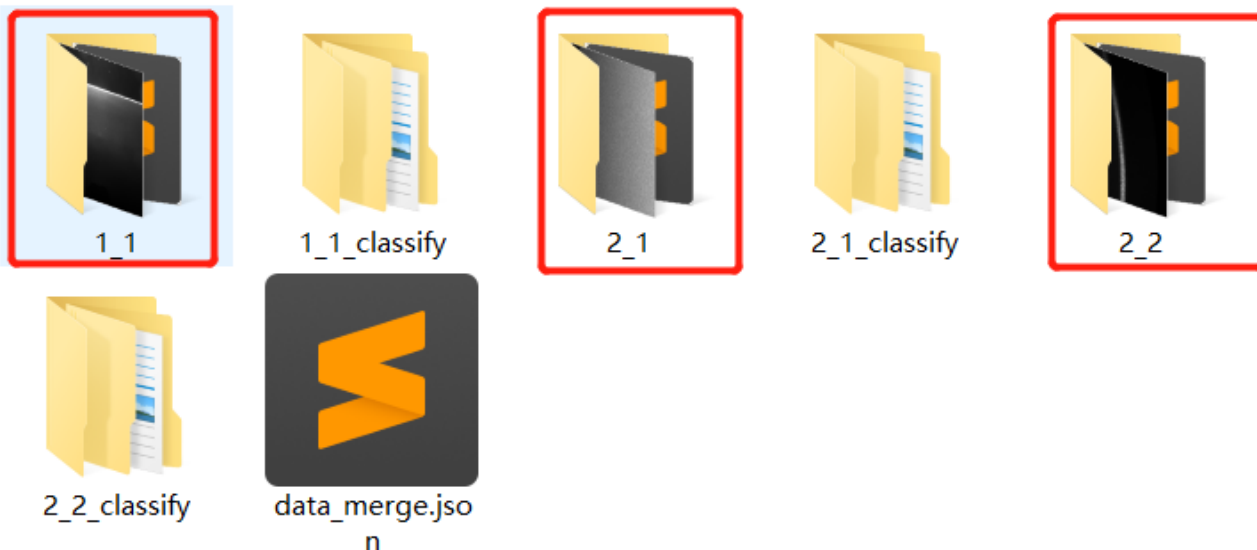
> PAD > images >



2. 按点位分布拆分json,每个点位文件夹都切分出json; 再根据json截取各点位类别小图, 以_classify结尾;

手动调整各文件夹之间不合适的分类后继续第三步

```
jf = r"D:\PAD\images\data_merge.json"
img_path = os.path.dirname(jf)
api.label_partition_by_station_api(img_path, jf, scale=True, offset=50, crop_size=120)
```



> PAD > images > 1_1_classify >



🔍 在 1_1_c

名称	修改日期	类型
1	2023/1/31 16:32	文件夹
2	2023/1/31 16:32	文件夹

3. 根据手动调整后的形态分布重新生成json,保存在分类文件夹下；拆分数据集，每个点位文件夹都生成train&test文件夹，其中包括n张图片及1个json，最终将所有点位中的train&test数据合并整理至主目录。

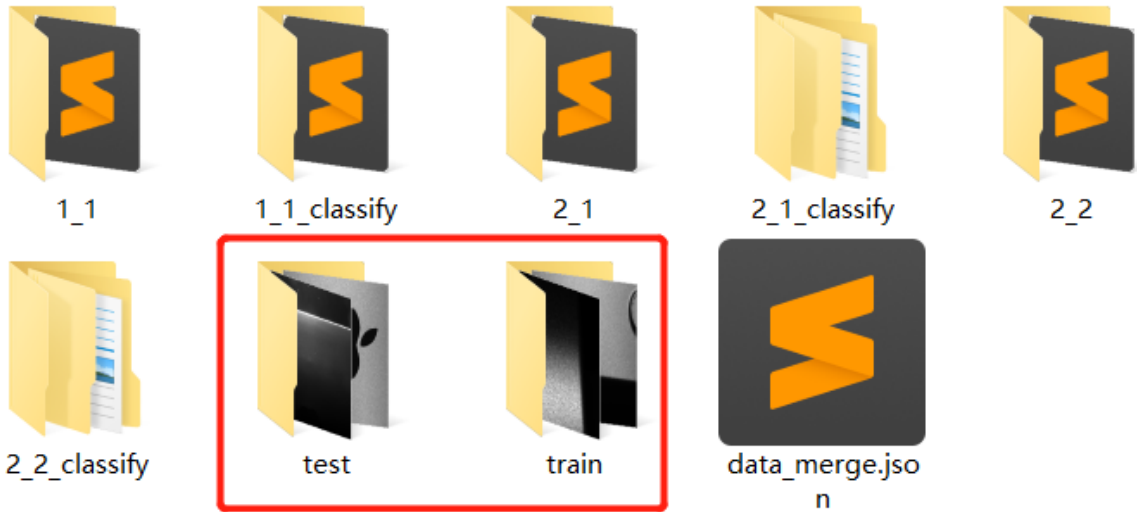
```
folder_path = r"D:\PAD\images"
api.dataset_partition_by_station_api(folder_path,repeat=2,ratio=1/4)
```

) > PAD > images > 1_1_classify

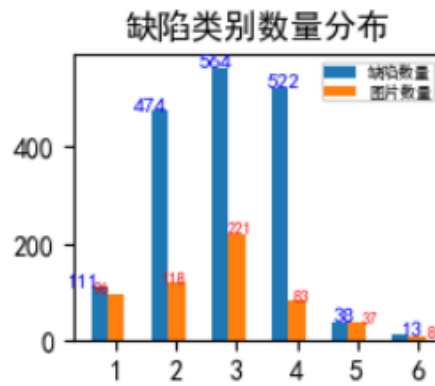
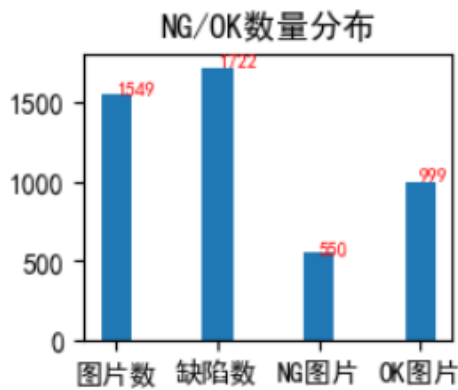


🔍 在 1_1_classi

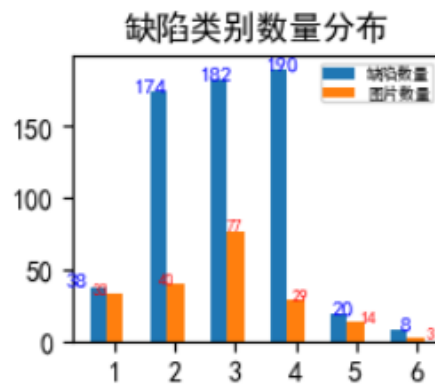
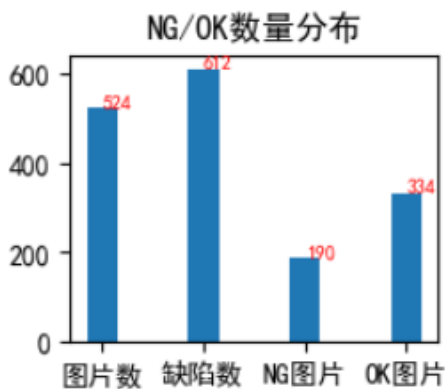
名称	修改日期	类型
1	2023/1/31 16:32	文件夹
2	2023/1/31 16:32	文件夹
1_1_classify.json	2023/1/31 16:49	JSON 文件



效果:



```
[3]: jf = r"D:\PAD\images\test\test.json"
      api.read_json_api(jf, mini = True)
```



修复推理json绘制问题

更新记录：1. 添加多JSON绘制（推理：多边形标签得分+标注：矩形）

2. 推理JSON一个实例可能或有多段，尤其是线段或大面积缺陷；标注直接连起来会有错乱

```
jf = r"D:\val\hp\ng.json"
inf_jf = r"D:\val\hp\model_0019999.json"
img_path = os.path.dirname(jf)
dst = img_path+"_rect"
draw={"shape":"RECT","label":True,"score":False,"area":False,"color":False,"font_size":1,"classid":False}
api.draw_mask_api(img_path, jfs=[jf,inf_jf], dst=dst, spotcheck=False, draw=draw)
```

