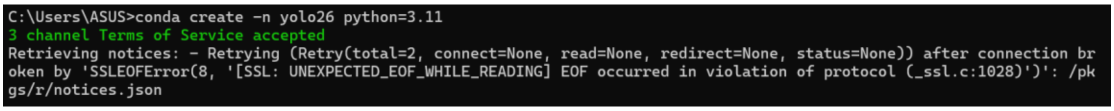
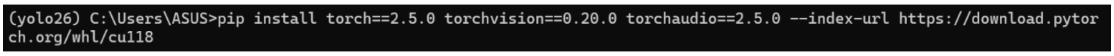
# 项目进程

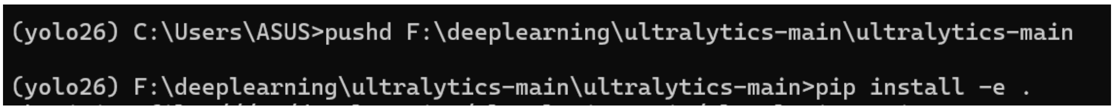
## 安装虚拟环境



```
C:\Users\ASUS>conda create -n yolo26 python=3.11
3 channel Terms of Service accepted
Retrieving notices: - Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection br
oken by 'SSLEOFError(8, '[SSL: UNEXPECTED_EOF_WHILE_READING] EOF occurred in violation of protocol (_ssl.c:1028)')': /pk
gs/r/notices.json
```



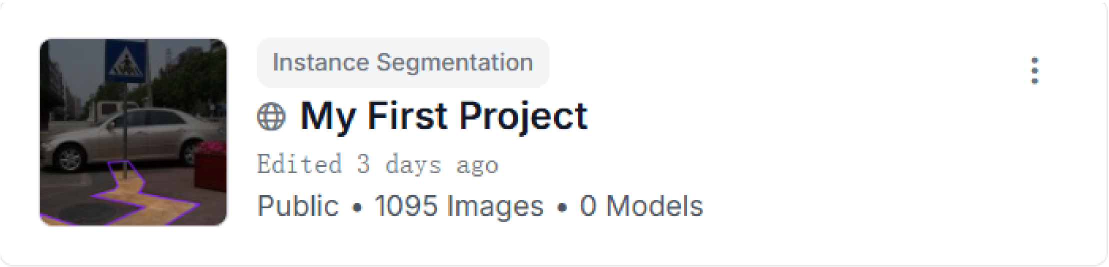
```
(yolo26) C:\Users\ASUS>pip install torch==2.5.0 torchvision==0.20.0 torchaudio==2.5.0 --index-url https://download.pytor
ch.org/whl/cu118
```



```
(yolo26) C:\Users\ASUS>pushd F:\deeplearning\ultralytics-main\ultralytics-main

(yolo26) F:\deeplearning\ultralytics-main\ultralytics-main>pip install -e .
```

## 数据集构建

基于选取的主题，我将需要识别的物体分为**障碍物、盲道和破损**三个种类，收集到1095 张图片。



Instance Segmentation

My First Project

Edited 3 days ago

Public • 1095 Images • 0 Models

## 标注软件选取

目前常用的标注软件有 LabelImg、Roboflow 等工具，经过学习与对比，决定使用Roboflow 进行标注

## Roboflow 的优点

1. 线上平台较为方便，无需下载
2. 可以进行数据增强并导出模型需要的格式
3. 社区中具有很多开源的数据库

## 标注难点

到了标注环节我发现，障碍物具有很多种类比如自行车、汽车、消防栓、石墩等，它们的视觉形状（轮廓、纹理）完全不同。这会增加模型收敛的难度，模型需要学会在高维

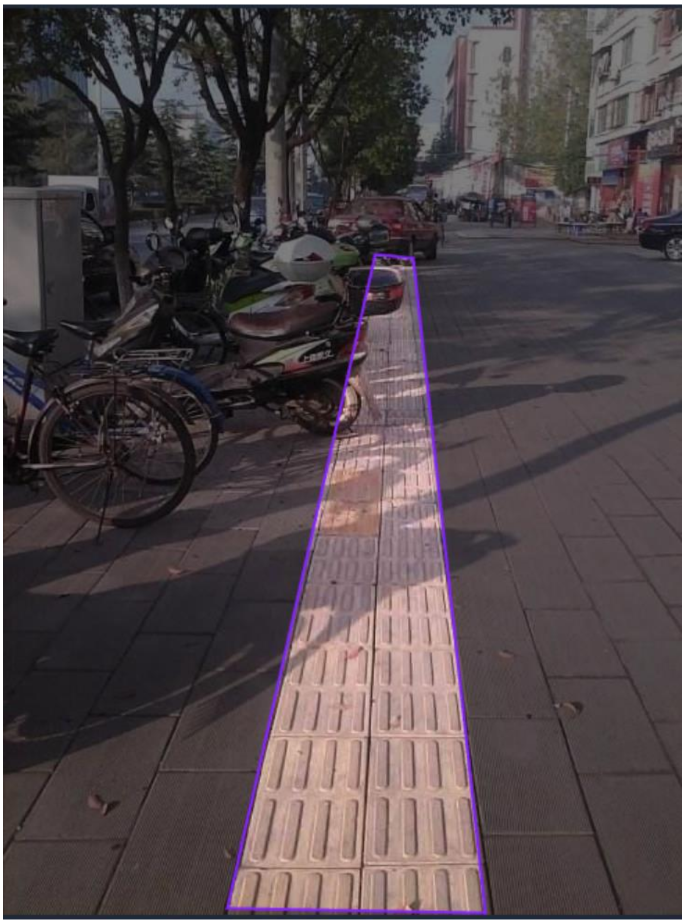
特征空间里把这几种完全不同的形状都映射到“障碍物”这个概念上。要想让模型彻底学会，对数据集和标注的要求非常严格，工作量也会大大增大，虽然多任务学习的模型性能会更好，但不得不退而求其次，选择双模型并行推理。

## 解决方法

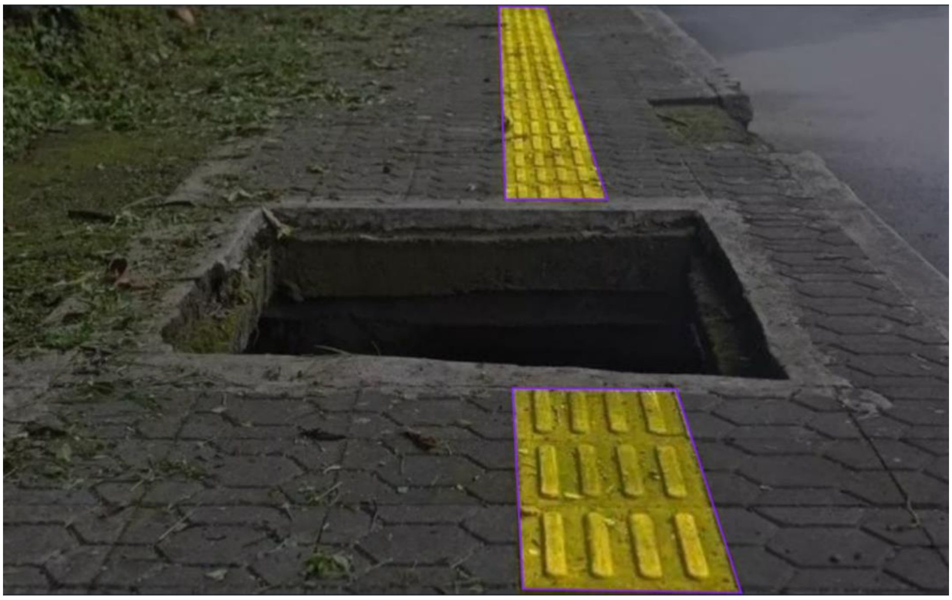
我的想法是使用官方预训练模型识别障碍物，而我只需要专注训练盲道和破损的专用模型，最后通过代码判断盲道上以及盲道周围的物体。

这样双模型的做法极大减少了我标注的工作量，同时，剔除障碍物后，盲道和破损都可以使用实例分割模型来识别，更符合我想要探索的主题。

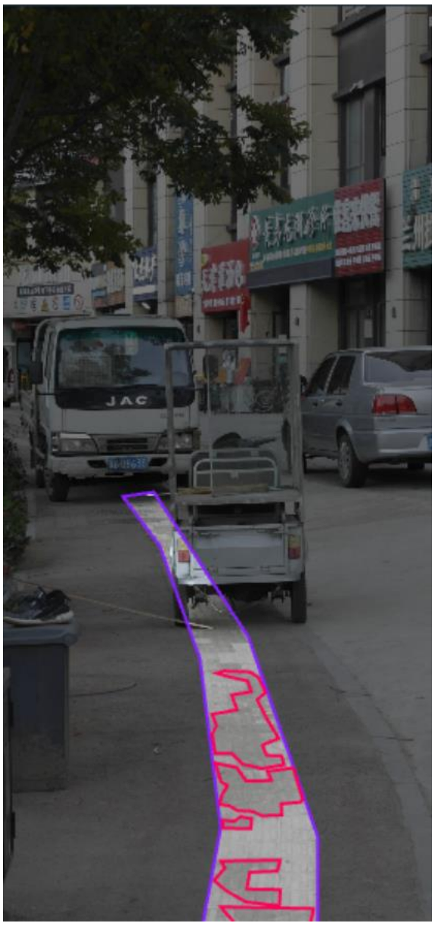
## 标注逻辑



临时性障碍选择补全盲道，，让模型学习连贯性，以便后续进行相对位置判断

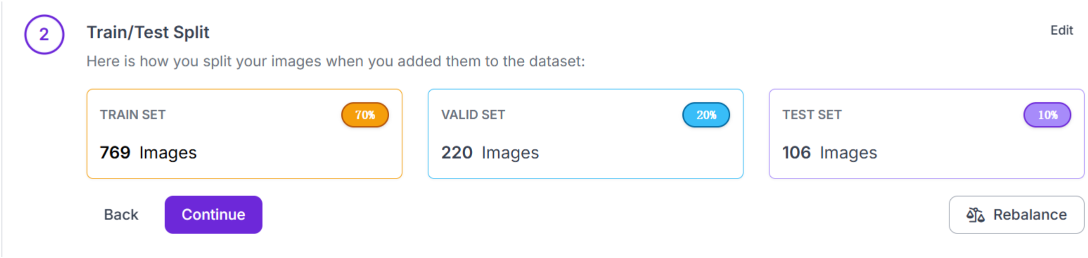


永久性障碍则选择断开



标注肉眼可见的破损

# 数据增强



2 Train/Test Split

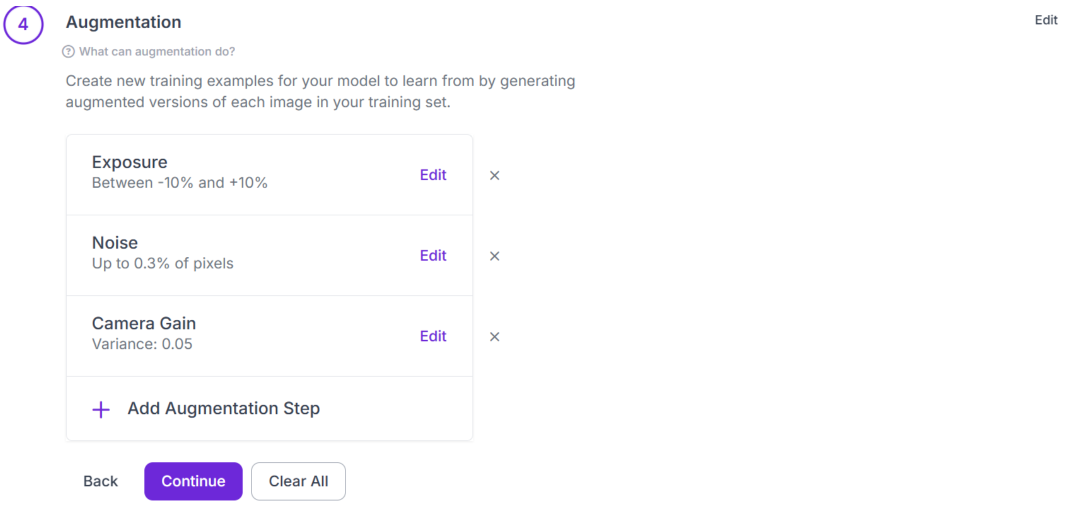
Here is how you split your images when you added them to the dataset:

TRAIN SET 70% 769 Images

VALID SET 20% 220 Images

TEST SET 10% 106 Images

数据平衡



4 Augmentation

What can augmentation do?

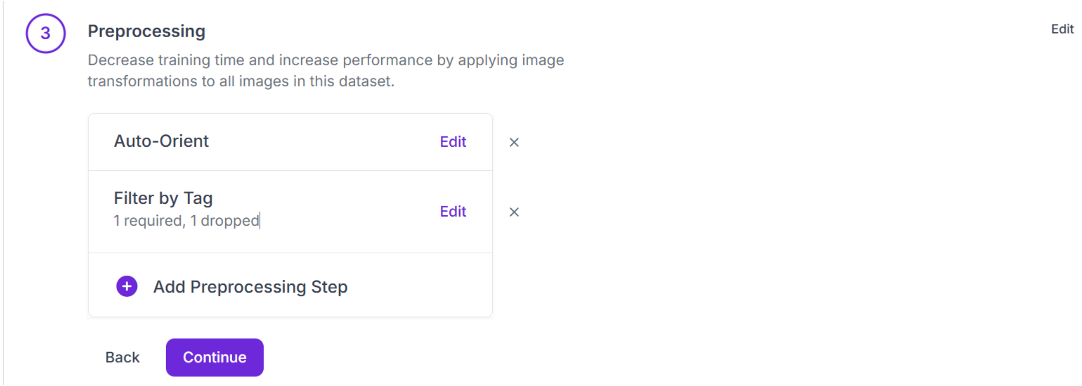
Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

Exposure
Between -10% and +10%

Noise
Up to 0.3% of pixels

Camera Gain
Variance: 0.05

总数据集增强（均为 yolo 无法实现的增强）



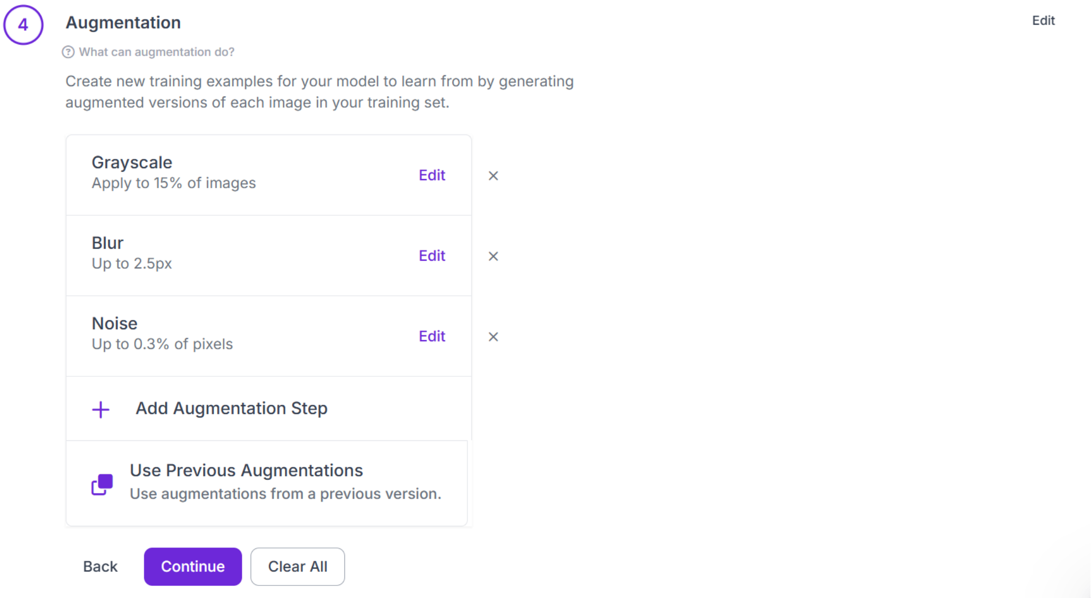
3 Preprocessing

Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient

Filter by Tag
1 required, 1 dropped

筛选有破损标注的数据



Augmentation

What can augmentation do?

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

Grayscale
Apply to 15% of images

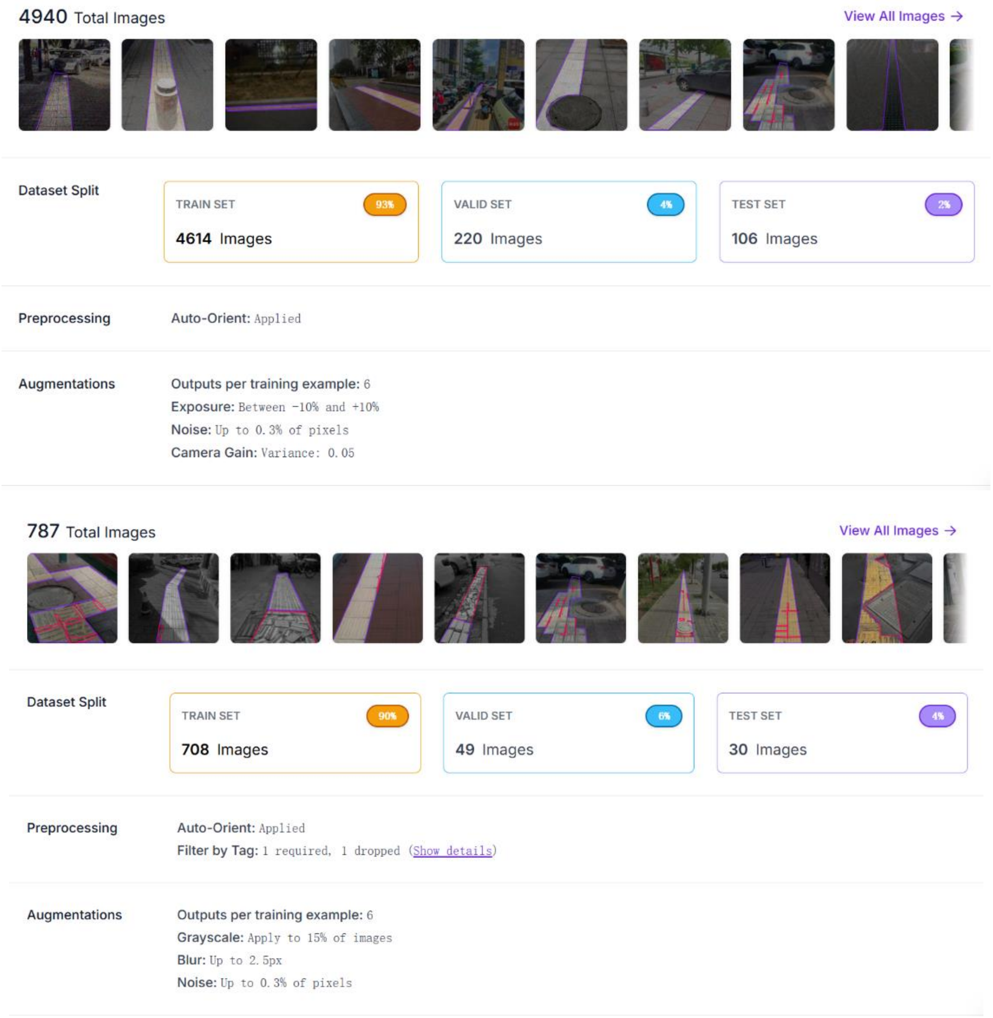
Blur
Up to 2.5px

Noise
Up to 0.3% of pixels

Add Augmentation Step

Use Previous Augmentations
Use augmentations from a previous version.

对有破损标注的数据集单独增强（提高破损数据的比例）



4940 Total Images

Dataset Split
TRAIN SET 4614 Images
VALID SET 220 Images
TEST SET 106 Images

Preprocessing
Auto-Orient: Applied

Augmentations
Outputs per training example: 6
Exposure: Between -10% and +10%
Noise: Up to 0.3% of pixels
Camera Gain: Variance: 0.05

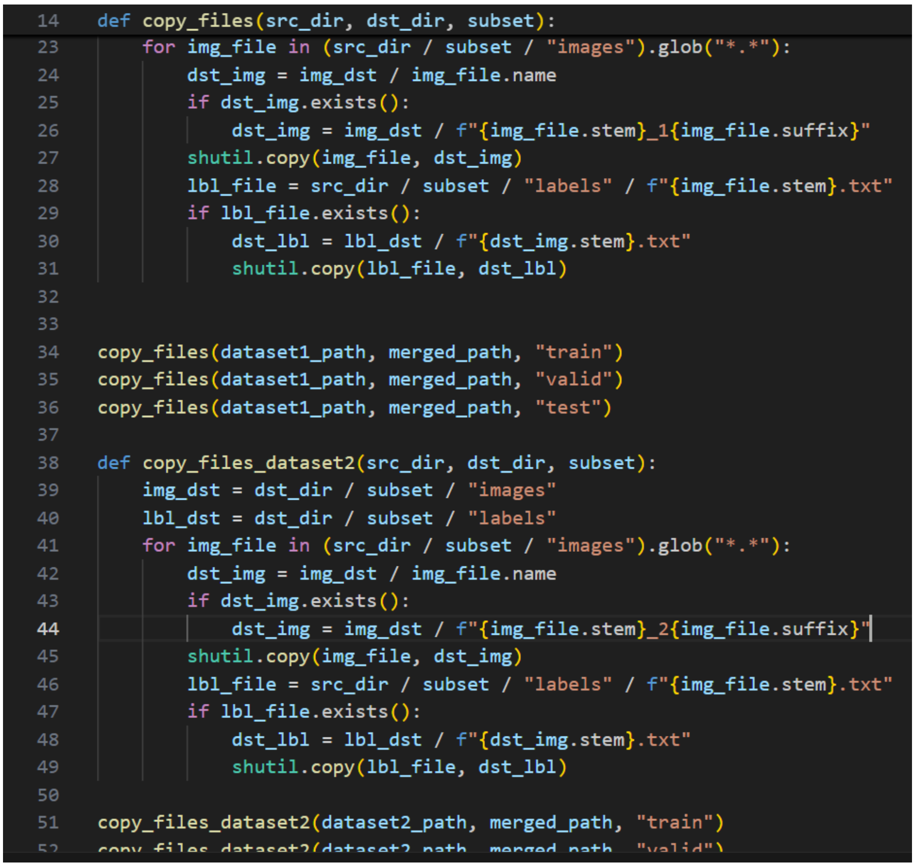
787 Total Images

Dataset Split
TRAIN SET 708 Images
VALID SET 49 Images
TEST SET 30 Images

Preprocessing
Auto-Orient: Applied
Filter by Tag: 1 required, 1 dropped (Show details)

Augmentations
Outputs per training example: 6
Grayscale: Apply to 15% of images
Blur: Up to 2.5px
Noise: Up to 0.3% of pixels

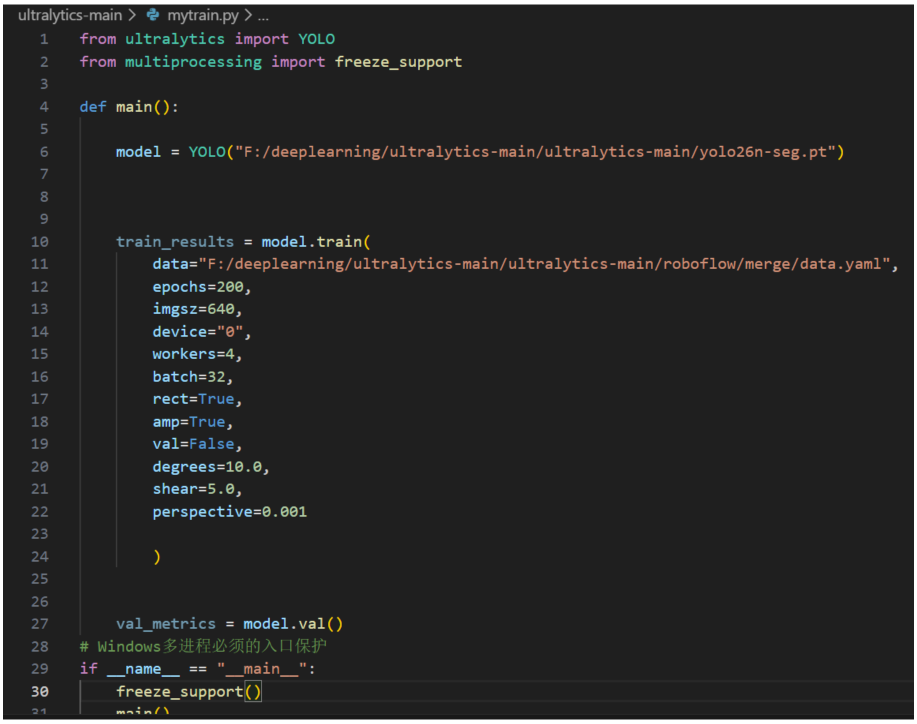
最后得到的两个数据集



```python
def copy_files(src_dir, dst_dir, subset):
    for img_file in (src_dir / subset / "images").glob("*.*"):
        dst_img = img_dst / img_file.name
        if dst_img.exists():
            dst_img = img_dst / f"{img_file.stem}_1{img_file.suffix}"
        shutil.copy(img_file, dst_img)
        lbl_file = src_dir / subset / "labels" / f"{img_file.stem}.txt"
        if lbl_file.exists():
            dst_lbl = lbl_dst / f"{dst_img.stem}.txt"
            shutil.copy(lbl_file, dst_lbl)


copy_files(dataset1_path, merged_path, "train")
copy_files(dataset1_path, merged_path, "valid")
copy_files(dataset1_path, merged_path, "test")

def copy_files_dataset2(src_dir, dst_dir, subset):
    img_dst = dst_dir / subset / "images"
    lbl_dst = dst_dir / subset / "labels"
    for img_file in (src_dir / subset / "images").glob("*.*"):
        dst_img = img_dst / img_file.name
        if dst_img.exists():
            dst_img = img_dst / f"{img_file.stem}_2{img_file.suffix}"
        shutil.copy(img_file, dst_img)
        lbl_file = src_dir / subset / "labels" / f"{img_file.stem}.txt"
        if lbl_file.exists():
            dst_lbl = lbl_dst / f"{dst_img.stem}.txt"
            shutil.copy(lbl_file, dst_lbl)

copy_files_dataset2(dataset2_path, merged_path, "train")
copy_files_dataset2(dataset2_path, merged_path, "valid")
```

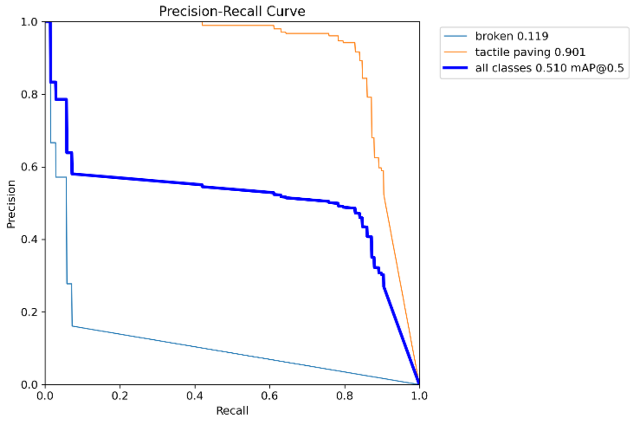
通过 merge.py 代码合并导出的两个数据集，最后得到 5322 张训练集，269 张验证集和 136 张测试集数据

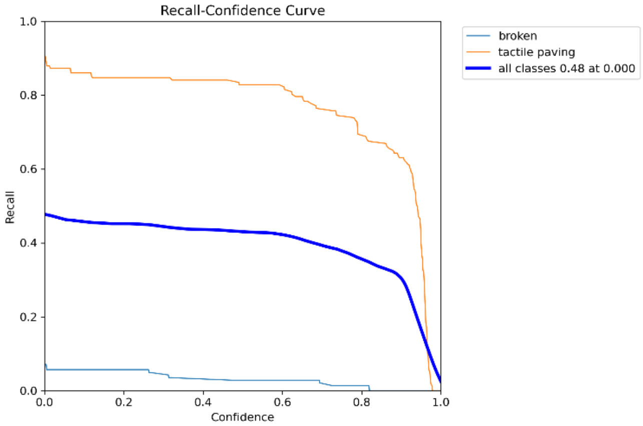
## 模型训练以及超参数

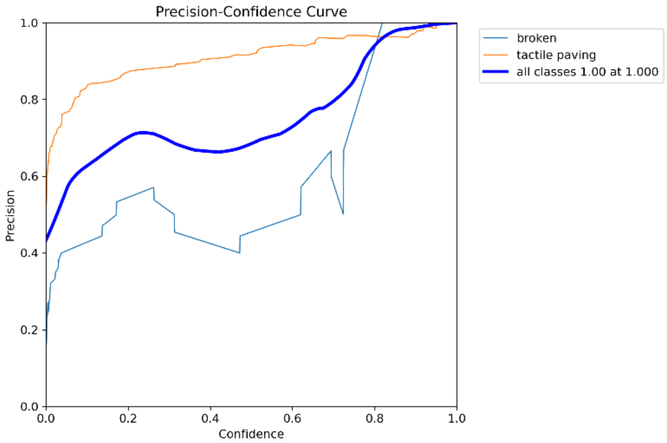


```python
from ultralytics import YOLO
from multiprocessing import freeze_support

def main():

    model = YOLO("F:/deeplearning/ultralytics-main/ultralytics-main/yolo26n-seg.pt")



    train_results = model.train(
        data="F:/deeplearning/ultralytics-main/ultralytics-main/roboflow/merge/data.yaml",
        epochs=200,
        imgsz=640,
        device="0",
        workers=4,
        batch=32,
        rect=True,
        amp=True,
        val=False,
        degrees=10.0,
        shear=5.0,
        perspective=0.001

        )


    val_metrics = model.val()
# Windows多进程必须的入口保护
if __name__ == "__main__":
    freeze_support()
    main()
```

使用 yolo26n-seg 模型训练 5322 张图片，经过调参与数据增强得到以上超参数，将 4060 8GB 实测一轮平均时间从 3 分 30 秒压缩为 1 分 10 秒

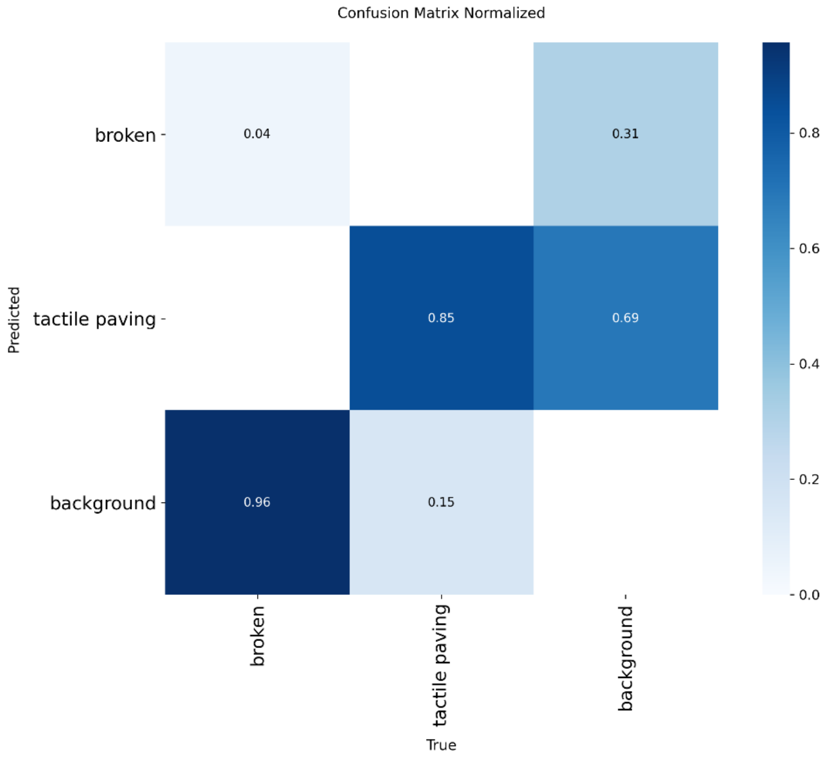


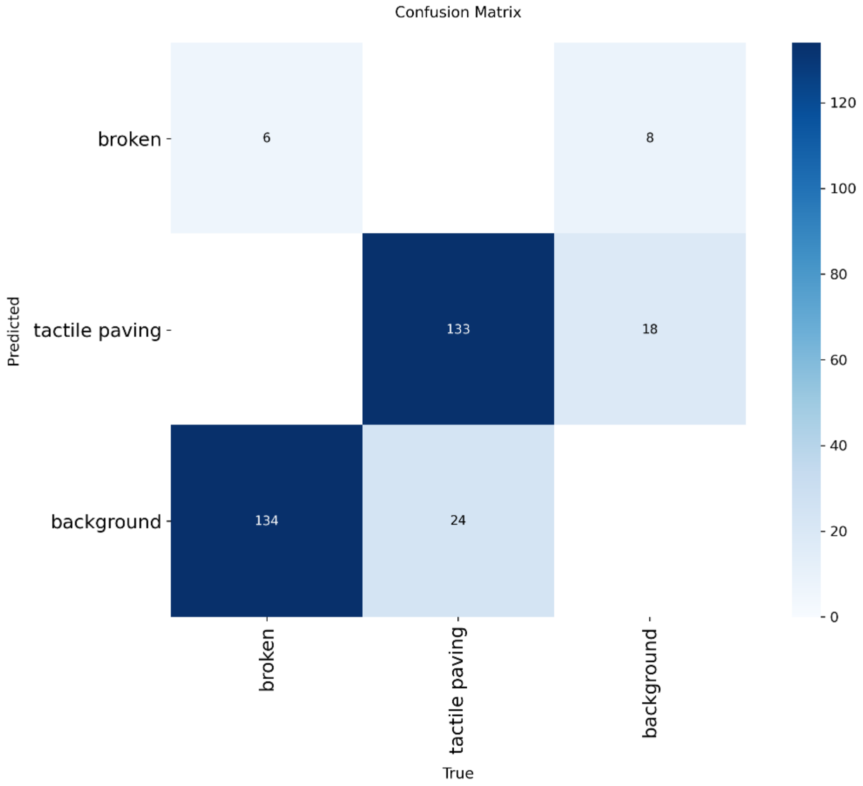




三表均为 best 权重对测试集的数据

由三表可知，盲道实例分割的 mAP 达 0.901，完全能够达到要求而破损的 mAP 很低，只有 0.119，而且具有 P 值高 R 值低的特点，可能是视觉特征微弱和样本不平衡的原因。

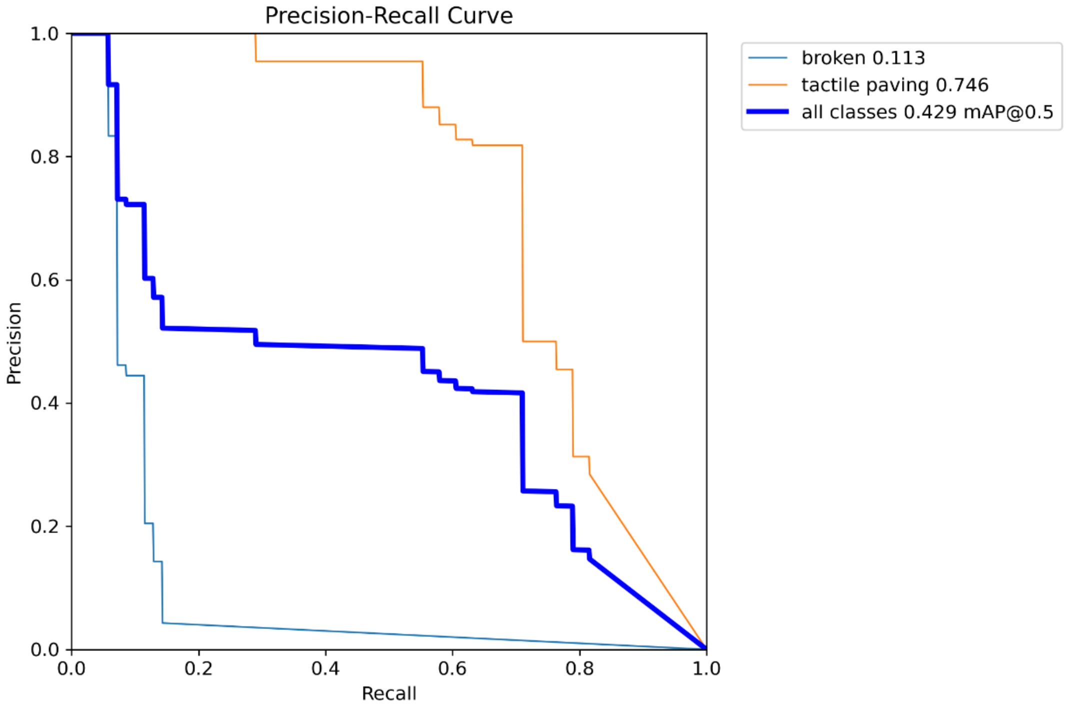




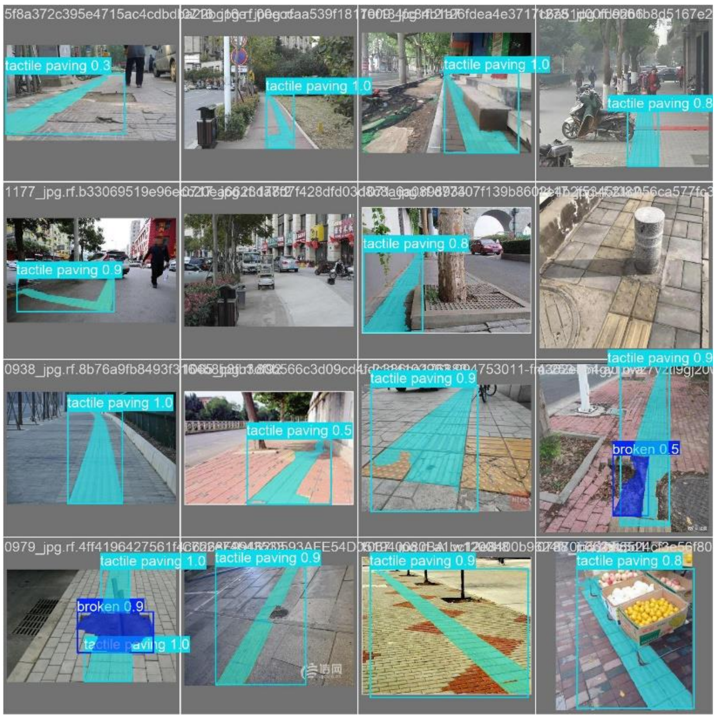
由矩阵图我们得知，大多数破损都被识别成背景，这也解释了前面为何 P 高 R 低，破损特征太弱，背景太强，模型极其“胆小”和“保守”。

虽然知道极有可能是数据集的问题，破损过小，分辨率过低，模型难以学习，但是还是想先通过调参来解决。

尝试只使用都是破损的数据集

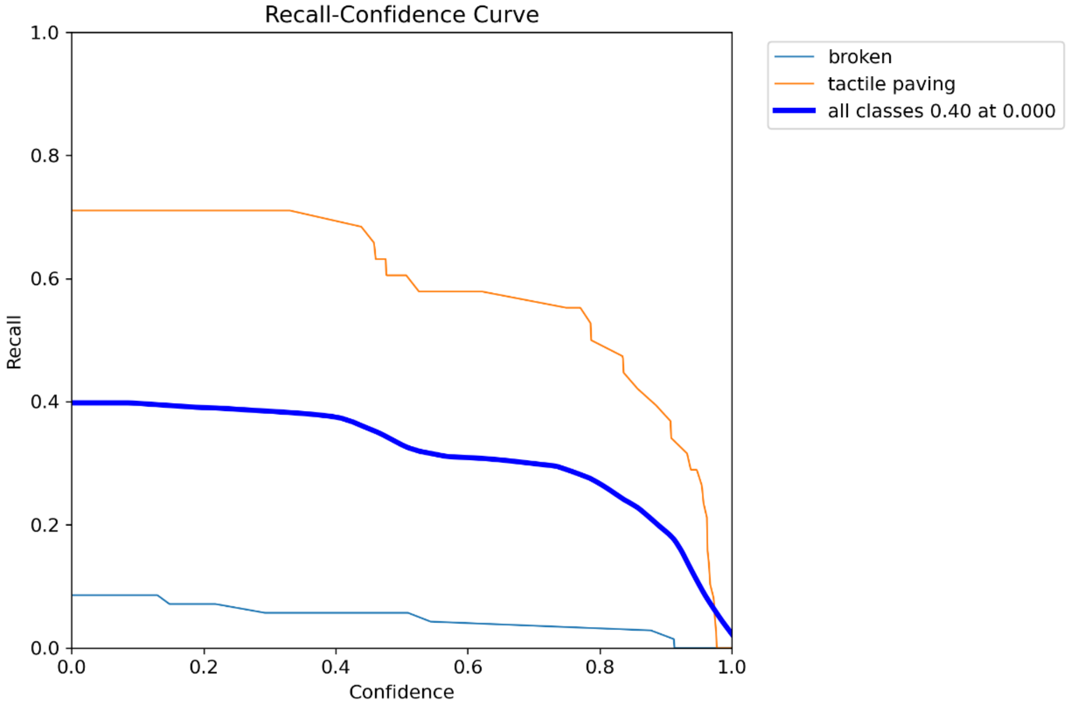


依旧是 p 高 r 低



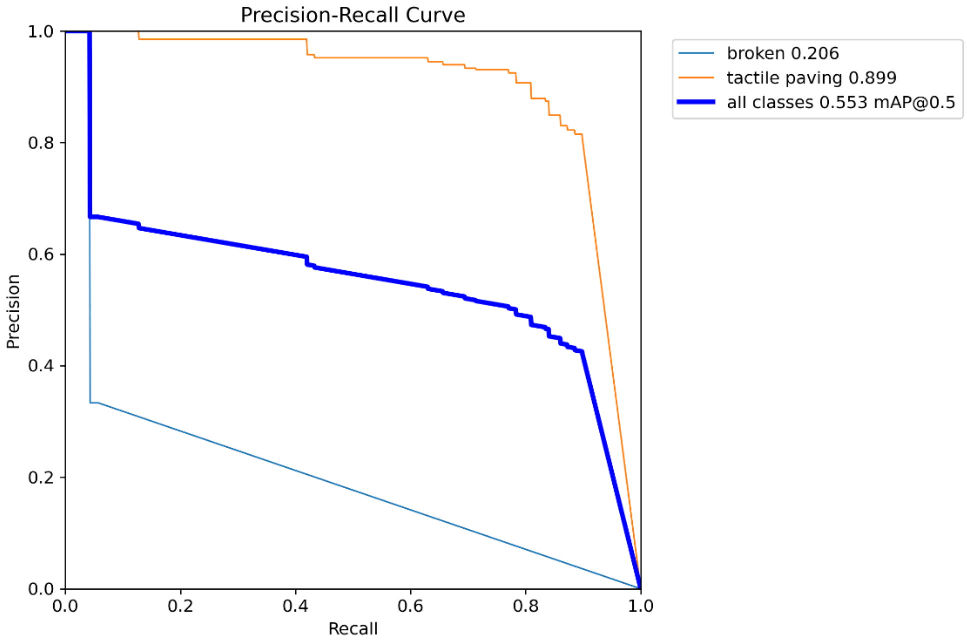
但是盲道的分割中预留了破损的位置，似乎可以从这里突破

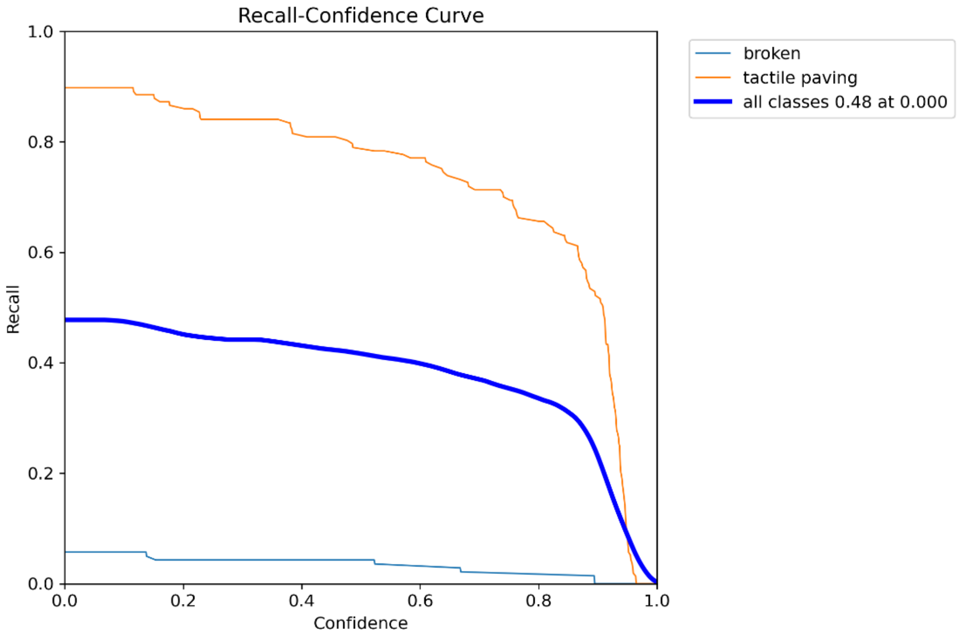
调整置信度 conf=0.1,但是 broken 的 r 值仍然没有上升



再尝试了更改权重损失、学习率、使用 s 模型 r 值都没有明显的提升

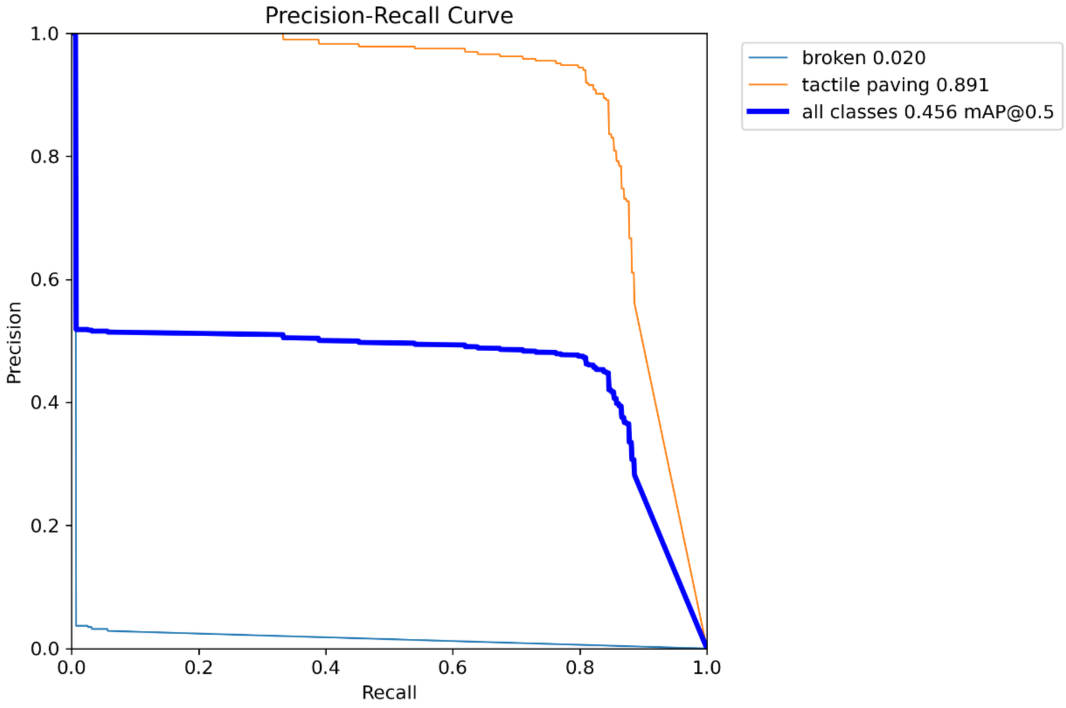
于是尝试调整数据集结构，在全是破损的数据集里加入两倍初始数据集，总数大概两千多张图片，此时有破损图片占数据集的 40%，看模型能否区分出破损





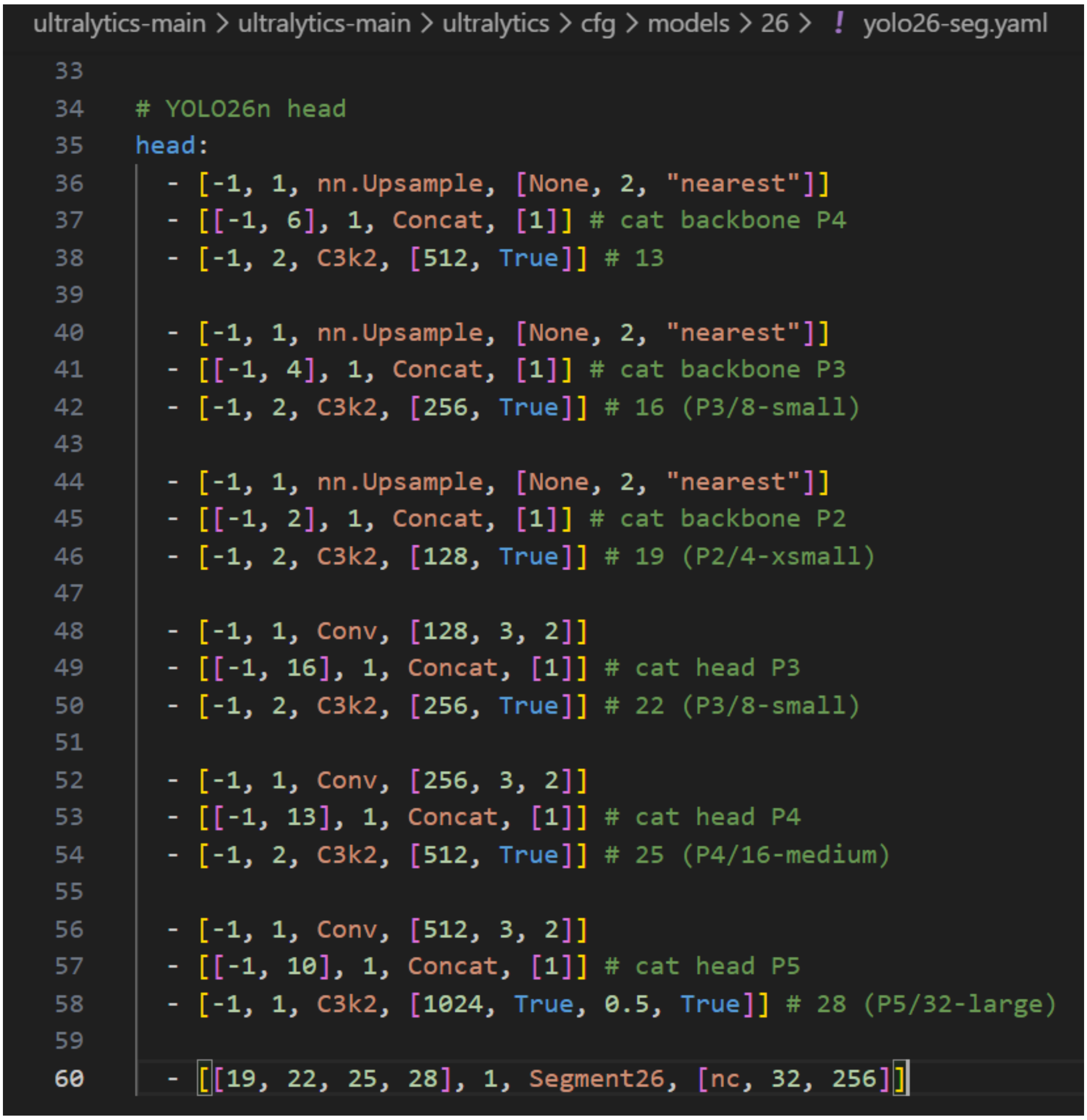
R 值仍然没有上升，而且 67 轮就早停了

尝试用最开始时的数据集，不进行 roboflow 的任意增强



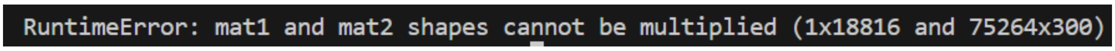
似乎更拉了

于是想要给模型加一个 p2 检测头，增加其小目标的检测能力


ultralytics-main > ultralytics-main > ultralytics > cfg > models > 26 > ! yolo26-seg.yaml

```yaml

# YOLO26n head
head:
  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 6], 1, Concat, [1]] # cat backbone P4
  - [-1, 2, C3k2, [512, True]] # 13

  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 4], 1, Concat, [1]] # cat backbone P3
  - [-1, 2, C3k2, [256, True]] # 16 (P3/8-small)

  - [-1, 1, nn.Upsample, [None, 2, "nearest"]]
  - [[-1, 2], 1, Concat, [1]] # cat backbone P2
  - [-1, 2, C3k2, [128, True]] # 19 (P2/4-xsmall)

  - [-1, 1, Conv, [128, 3, 2]]
  - [[-1, 16], 1, Concat, [1]] # cat head P3
  - [-1, 2, C3k2, [256, True]] # 22 (P3/8-small)

  - [-1, 1, Conv, [256, 3, 2]]
  - [[-1, 13], 1, Concat, [1]] # cat head P4
  - [-1, 2, C3k2, [512, True]] # 25 (P4/16-medium)

  - [-1, 1, Conv, [512, 3, 2]]
  - [[-1, 10], 1, Concat, [1]] # cat head P5
  - [-1, 1, C3k2, [1024, True, 0.5, True]] # 28 (P5/32-large)

  - [[19, 22, 25, 28], 1, Segment26, [nc, 32, 256]]
```

但是无论我怎么改，把 yaml 文件里的所有参数都改了一遍，都会报错

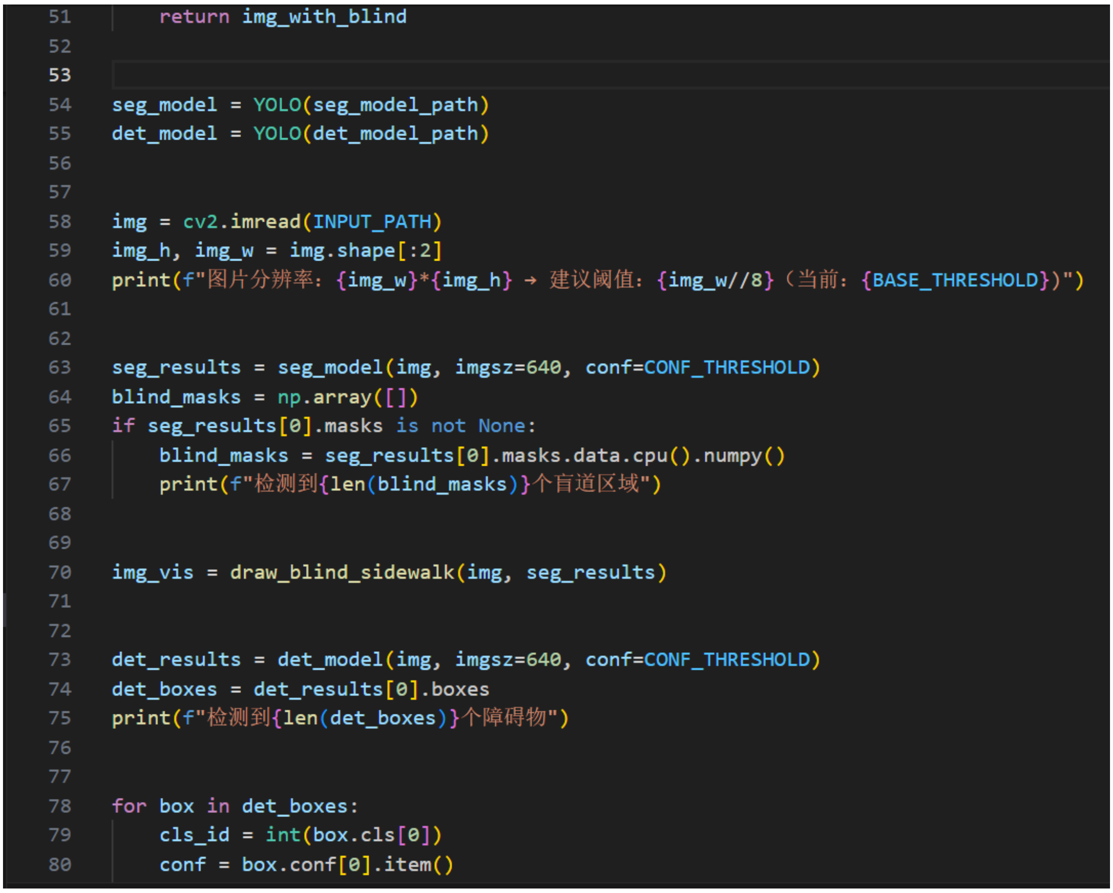


```
RuntimeError: mat1 and mat2 shapes cannot be multiplied (1x18816 and 75264x300)
```

矩阵乘维度不匹配（18816 vs 75264，后者 = 4×18816）

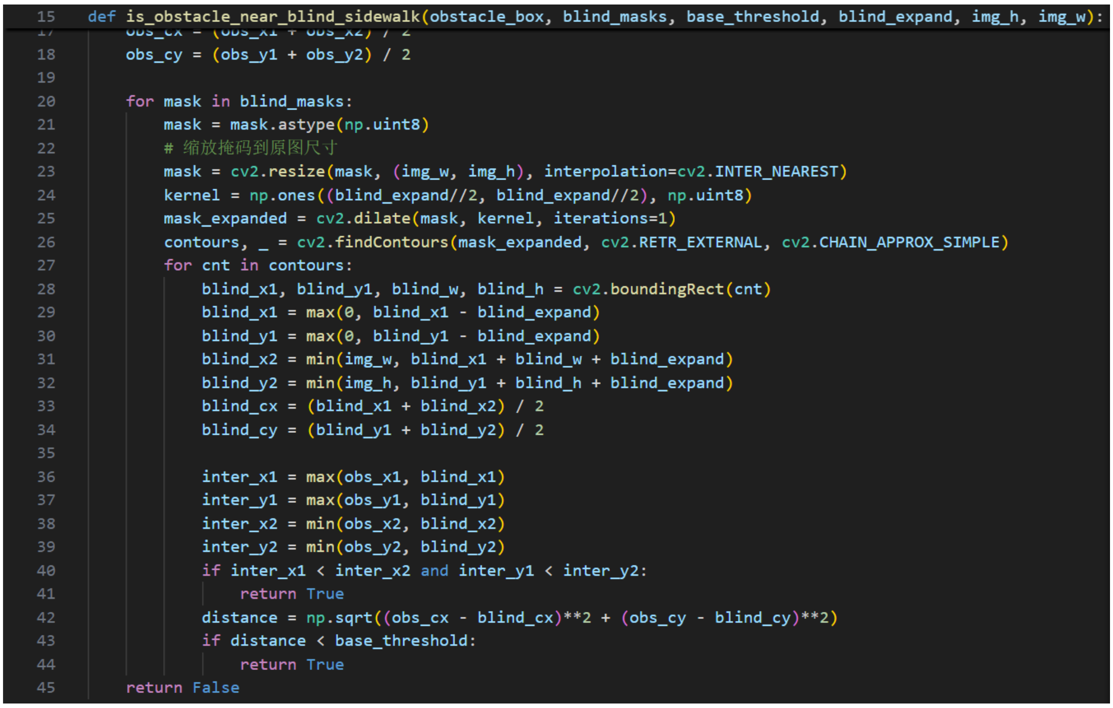
最后不得不承认是 segment26 的问题，segment26 内部就是只有 3 个特征图（P3,P4,P5），其实之前不是没把 segment26 换成 segment，但是尽管 segment26 继承 segment，但其使用 Proto26，并改变了 proto 的构建与 forward 流程，所以得出结论，这个检测头只有 yolo26 不能加！

由于时间问题就不能再继续往下尝试了



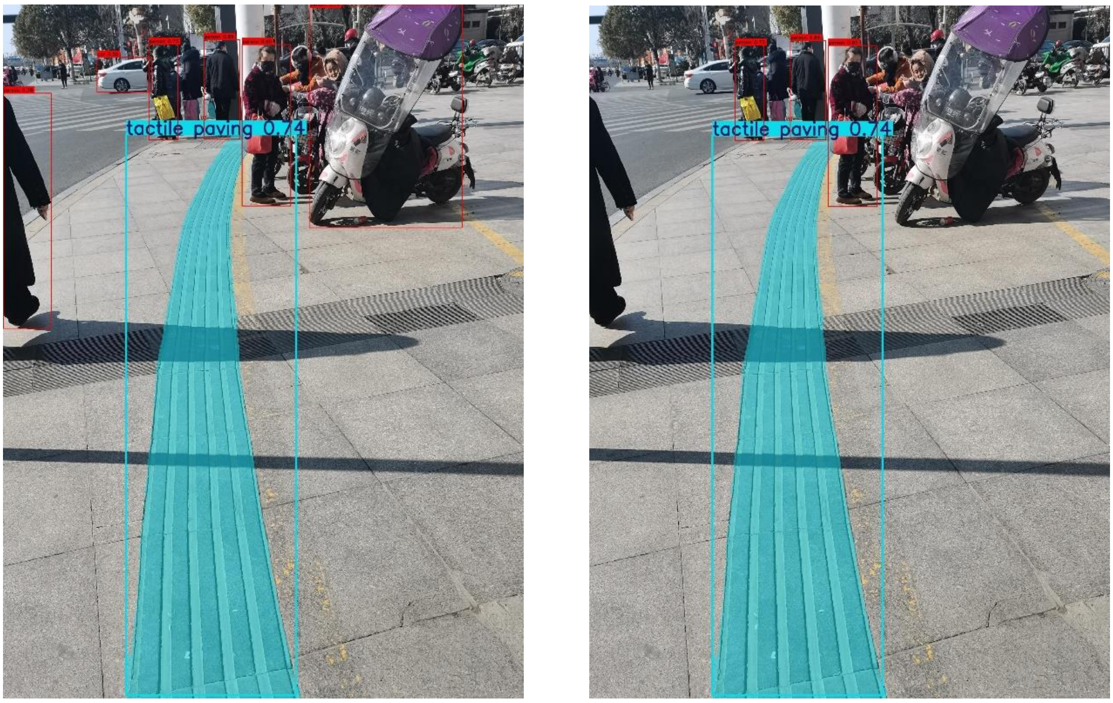
```python
    return img_with_blind


seg_model = YOLO(seg_model_path)
det_model = YOLO(det_model_path)


img = cv2.imread(INPUT_PATH)
img_h, img_w = img.shape[:2]
print(f"图片分辨率：{img_w}*{img_h} → 建议阈值：{img_w//8}（当前：{BASE_THRESHOLD}）")


seg_results = seg_model(img, imgsz=640, conf=CONF_THRESHOLD)
blind_masks = np.array([])
if seg_results[0].masks is not None:
    blind_masks = seg_results[0].masks.data.cpu().numpy()
    print(f"检测到{len(blind_masks)}个盲道区域")


img_vis = draw_blind_sidewalk(img, seg_results)


det_results = det_model(img, imgsz=640, conf=CONF_THRESHOLD)
det_boxes = det_results[0].boxes
print(f"检测到{len(det_boxes)}个障碍物")


for box in det_boxes:
    cls_id = int(box.cls[0])
    conf = box.conf[0].item()
```

双模型搭建

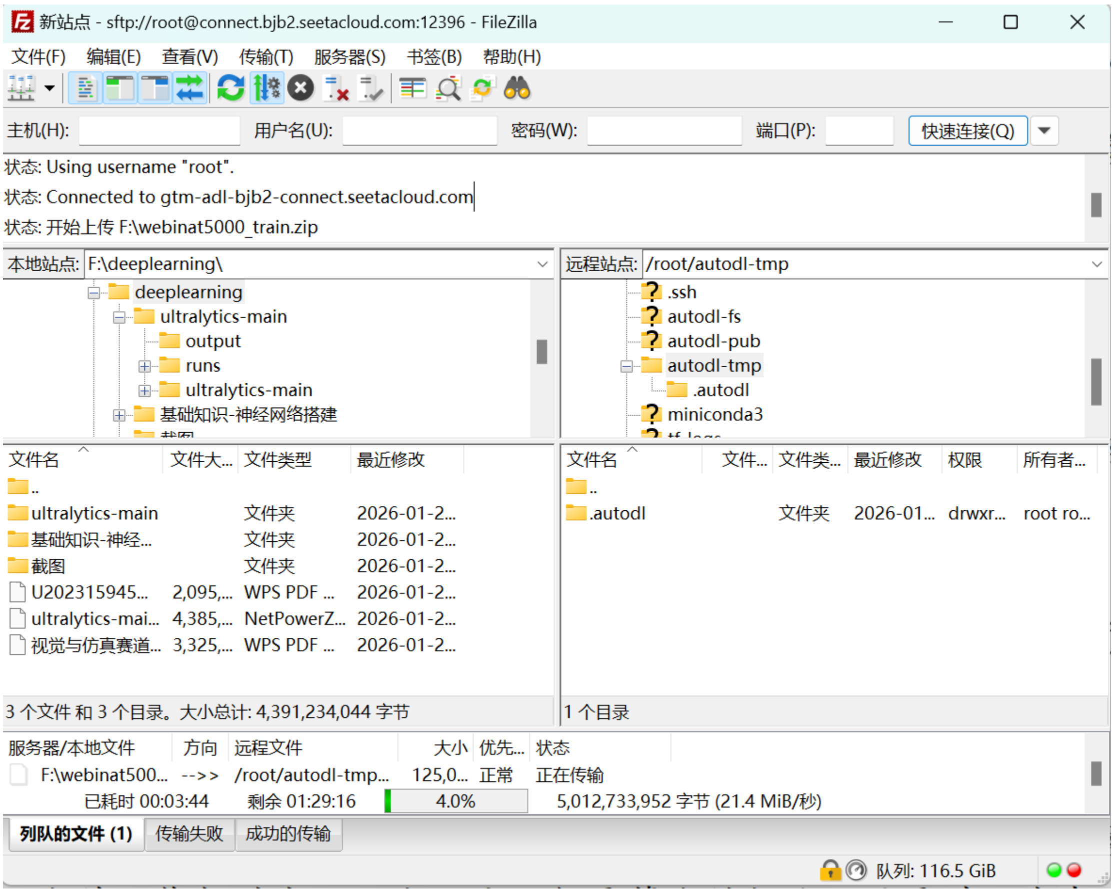


```python
def is_obstacle_near_blind_sidewalk(obstacle_box, blind_masks, base_threshold, blind_expand, img_h, img_w):
    obs_cx = (obs_x1 + obs_x2) / 2
    obs_cy = (obs_y1 + obs_y2) / 2

    for mask in blind_masks:
        mask = mask.astype(np.uint8)
        # 缩放掩码到原图尺寸
        mask = cv2.resize(mask, (img_w, img_h), interpolation=cv2.INTER_NEAREST)
        kernel = np.ones((blind_expand//2, blind_expand//2), np.uint8)
        mask_expanded = cv2.dilate(mask, kernel, iterations=1)
        contours, _ = cv2.findContours(mask_expanded, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        for cnt in contours:
            blind_x1, blind_y1, blind_w, blind_h = cv2.boundingRect(cnt)
            blind_x1 = max(0, blind_x1 - blind_expand)
            blind_y1 = max(0, blind_y1 - blind_expand)
            blind_x2 = min(img_w, blind_x1 + blind_w + blind_expand)
            blind_y2 = min(img_h, blind_y1 + blind_h + blind_expand)
            blind_cx = (blind_x1 + blind_x2) / 2
            blind_cy = (blind_y1 + blind_y2) / 2

            inter_x1 = max(obs_x1, blind_x1)
            inter_y1 = max(obs_y1, blind_y1)
            inter_x2 = min(obs_x2, blind_x2)
            inter_y2 = min(obs_y2, blind_y2)
            if inter_x1 < inter_x2 and inter_y1 < inter_y2:
                return True
            distance = np.sqrt((obs_cx - blind_cx)**2 + (obs_cy - blind_cy)**2)
            if distance < base_threshold:
                return True
    return False
```

相对位置判断



效果


新站点 - sftp://root@connect.bjb2.seetacloud.com:12396 - FileZilla
文件(F) 编辑(E) 查看(V) 传输(T) 服务器(S) 书签(B) 帮助(H)
主机(H): 用户名(U): 密码(W): 端口(P): 快速连接(Q)

```
状态: Using username "root".
状态: Connected to gtm-adl-bjb2-connect.seetacloud.com
状态: 开始上传 F:\webinat5000_train.zip
```

本地站点: F:\deeplearning\

- deeplearning
  - ultralytics-main
    - output
    - runs
    - ultralytics-main
  - 基础知识-神经网络搭建
  - 截图

远程站点: /root/autodl-tmp

- .ssh
- autodl-fs
- autodl-pub
- autodl-tmp
  - .autodl
- miniconda3

| 文件名 | 文件大... | 文件类型 | 最近修改 |
|---|---|---|---|
| .. | | | |
| ultralytics-main | | 文件夹 | 2026-01-2... |
| 基础知识-神经... | | 文件夹 | 2026-01-2... |
| 截图 | | 文件夹 | 2026-01-2... |
| U202315945... | 2,095,... | WPS PDF ... | 2026-01-2... |
| ultralytics-mai... | 4,385,... | NetPowerZ... | 2026-01-2... |
| 视觉与仿真赛道... | 3,325,... | WPS PDF ... | 2026-01-2... |

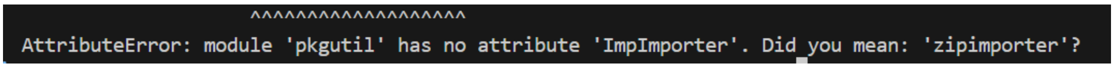
3 个文件 和 3 个目录。大小总计: 4,391,234,044 字节

| 文件名 | 文件... | 文件类... | 最近修改 | 权限 | 所有者... |
|---|---|---|---|---|---|
| .. | | | | | |
| .autodl | | 文件夹 | 2026-01... | drwxr... | root ro... |

1 个目录

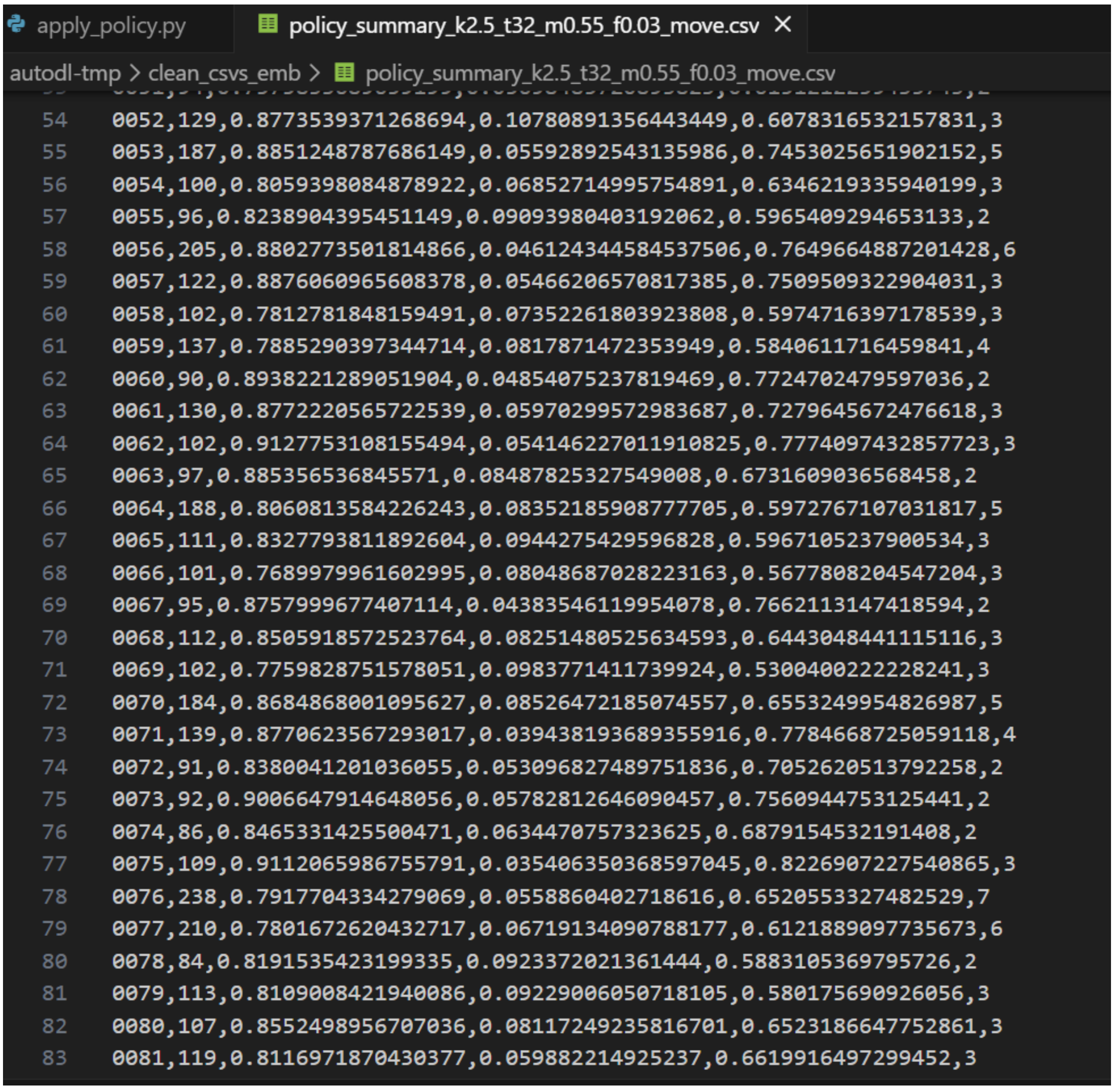
| 服务器/本地文件 | 方向 | 远程文件 | 大小 | 优先... | 状态 |
|---|---|---|---|---|---|
| F:\webinat500... | -->> | /root/autodl-tmp... | 125,0... | 正常 | 正在传输 |
| 已耗时 00:03:44 | | 剩余 01:29:16 | 4.0% | | 5,012,733,952 字节 (21.4 MiB/秒) |

上传数据



```
                ^^^^^^^^^^^^^^^^
AttributeError: module 'pkgutil' has no attribute 'ImpImporter'. Did you mean: 'zipimporter'?
```

模型与 py3.12 不兼容，重装 py3.10



apply_policy.py | policy_summary_k2.5_t32_m0.55_f0.03_move.csv

autodl-tmp > clean_csvs_emb > policy_summary_k2.5_t32_m0.55_f0.03_move.csv

```
0052,129,0.8773539371268694,0.10780891356443449,0.6078316532157831,3
0053,187,0.8851248787686149,0.05592892543135986,0.7453025651902152,5
0054,100,0.8059398084878922,0.06852714995754891,0.6346219335940199,3
0055,96,0.8238904395451149,0.09093980403192062,0.5965409294653133,2
0056,205,0.8802773501814866,0.046124344584537506,0.7649664887201428,6
0057,122,0.8876060965608378,0.05466206570817385,0.7509509322904031,3
0058,102,0.7812781848159491,0.07352261803923808,0.5974716397178539,3
0059,137,0.7885290397344714,0.0817871472353949,0.5840611716459841,4
0060,90,0.8938221289051904,0.04854075237819469,0.7724702479597036,2
0061,130,0.8772220565722539,0.05970299572983687,0.7279645672476618,3
0062,102,0.9127753108155494,0.054146227011910825,0.7774097432857723,3
0063,97,0.885356536845571,0.08487825327549008,0.6731609036568458,2
0064,188,0.8060813584226243,0.08352185908777705,0.5972767107031817,5
0065,111,0.8327793811892604,0.0944275429596828,0.5967105237900534,3
0066,101,0.7689979961602995,0.08048687028223163,0.5677808204547204,3
0067,95,0.8757999677407114,0.04383546119954078,0.7662113147418594,2
0068,112,0.8505918572523764,0.08251480525634593,0.6443048441115116,3
0069,102,0.7759828751578051,0.0983771411739924,0.5300400222228241,3
0070,184,0.8684868001095627,0.08526472185074557,0.6553249954826987,5
0071,139,0.8770623567293017,0.039438193689355916,0.7784668725059118,4
0072,91,0.8380041201036055,0.053096827489751836,0.7052620513792258,2
0073,92,0.9006647914648056,0.05782812646090457,0.7560944753125441,2
0074,86,0.8465331425500471,0.0634470757323625,0.6879154532191408,2
0075,109,0.9112065986755791,0.035406350368597045,0.8226907227540865,3
0076,238,0.7917704334279069,0.0558860402718616,0.6520553327482529,7
0077,210,0.7801672620432717,0.06719134090788177,0.6121889097735673,6
0078,84,0.8191535423199335,0.0923372021361444,0.5883105369795726,2
0079,113,0.8109008421940086,0.09229006050718105,0.580175690926056,3
0080,107,0.8552498956707036,0.08117249235816701,0.6523186647752861,3
0081,119,0.8116971870430377,0.059882214925237,0.6619916497299452,3
```

清洗数据

清洗数据这一步卡住了，总是移出一大半的数据

## 参考文献

[1]王新,谷亚东.基于双目立体视觉的盲人避障技术研究[J].电子测量技术,2025,48(07):98-106..

[2]岳剑峰,李伟明,宁黎华,等.基于 YOLO-DEFW 的焊缝缺陷实时检测算法研究[J].中国激光,2025,52(08):64-76.

[3]靳子越,李海涛,殷海晨,等.YOLO11n-seg-RF:一种改进 YOLO11n-seg 的轻量级岩石裂隙检测及分割算法[J/OL].北京大学学报(自然科学版),1-18[2026-01-30].

[4]马文杰,张轩雄.基于深度学习的盲道和盲道障碍物识别算法[J].电子科技,2024,37(03):75-83.

[5]Wang, W.; Jing, B.; Yu, X.; Sun, Y.; Yang, L.; Wang, C. YOLO-OD: Obstacle Detection for Visually Impaired Navigation Assistance. Sensors 2024, 24, 7621.

[6]李庆寒.基于深度学习的盲道路况检测技术的研究与应用[D].黑龙江大学,2024.

[7]唐武. 基于深度学习的户外盲道障碍目标检测与跟踪研究[D]. 江西:江西理工大学,2021.