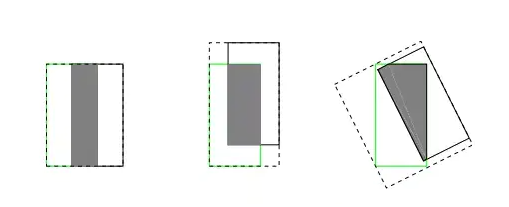
**YoloV5代码修改日志**

**2020年11月25日星期三　centermask阅读**

launch可用于单节点的分布式训练，支持CPU和GPU。对于GPU而言，若每个进程对应一个GPU，则训练将取得最大性能。可通过指定参数（nproc\_per\_node），让launch在单节点上创建指定数目的进程（不可大于该节点对应的GPU数目）。

2020年11月26日星期四 IOU

1. IOU
2. GIOU

用IOU作为损失函数时有如下两个问题：  
1.但检测框与gt之间没有重合时，IoU为0。而在优化损失函数时，梯度为0，意味着无法优化。  
2.在检测框与gt之间IoU相同时，检测的效果也具有较大差异，如下图所示：

下图所示为GIou和Iou之间的计算方式的区别，以及对应的代码。且GIou有如下优势。  
1.GIoU具有作为一个度量标准s的优良性质。包括非负性，同一性，对称性，以及三角不等式的性质  
2.与IoU相似，具有尺度不变性  
3.GIoU的值总是小于IoU的值  
4.对于两个矩形框A和B，0≤IoU(A,B)≤1，而-1≤GIoU≤1  
5.在A，B没有良好对齐时，会导致C的面积增大，从而使GIoU的值变小，而两个矩形框不重合时，依然可以计算GIoU，一定程度上解决了IoU不适合作为损失函数的原因

1. DIOU

**YOLO v5版本更新**

1. 新版本
2. 创建新文件夹/data/lyh/detection/YOLOv5
3. 利用git下载代码，方便以后代码的更新

①当前文件夹路径下：git init

②将yolov5最新代码下载到当前路径：

git clone <https://github.com/ultralytics/yolov5.git>

1. 搭建环境（参照github的环境要求）
2. 创建虚拟环境lzzyolov5：conda create -n lzzyolov5 python=3.8
3. 激活虚拟环境：conda activate lzzyolov5
4. 安装指定包

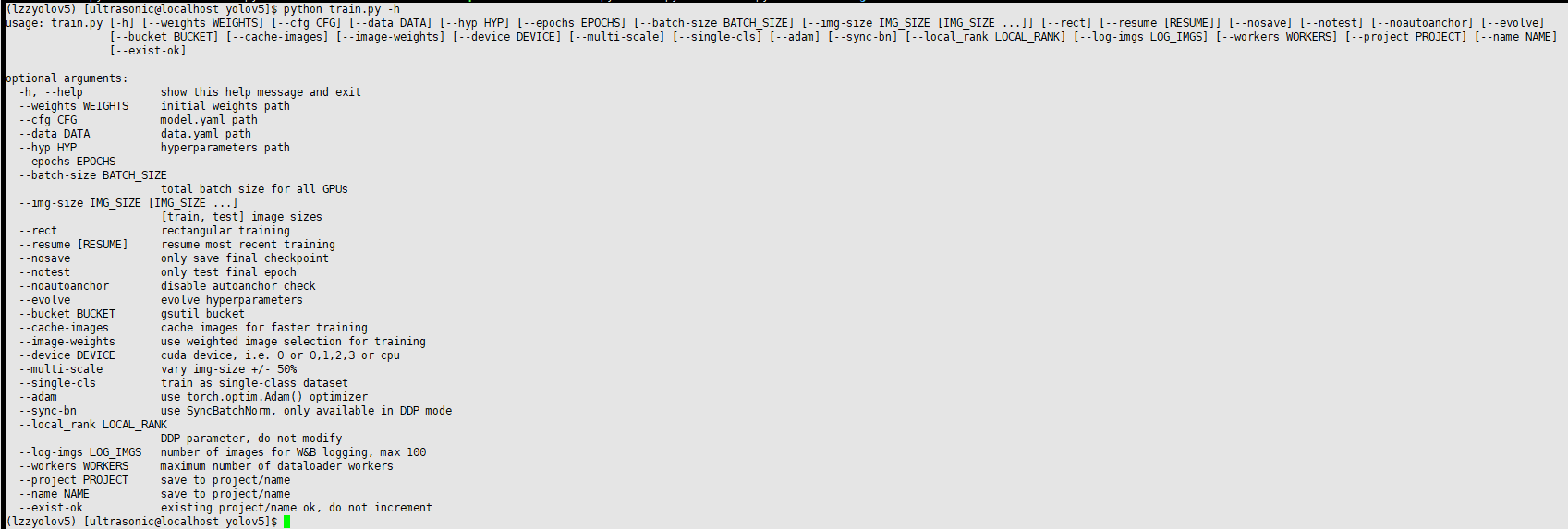
①安装pytorch:

pip install torch==1.7.0+cu92 torchvision==0.8.1+cu92 torchaudio==0.7.0 -f <https://download.pytorch.org/whl/torch_stable.html>

②安装其他包：pip install -r requirements.txt

1. 测试环境

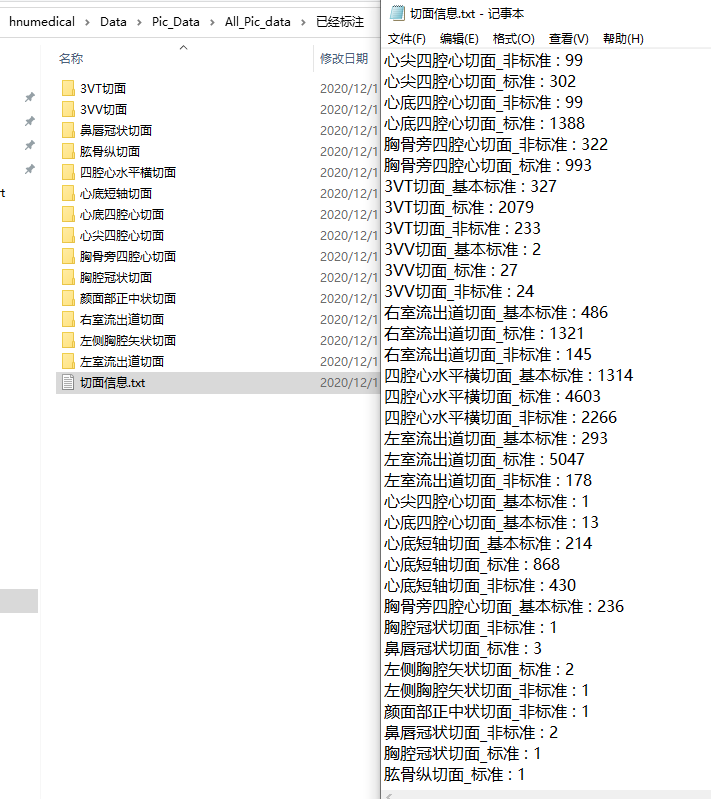
python train.py -h



**标准切面检测**

1. 数据整理

1、将本地的pic数据进行整理，主要分为未标注、已经标注；已经标注中分别按照切面、标准与否进行划分



2、对所有数据求包围盒（除去肋骨、脊柱）：

E:\hnumedical\Heart\_detection\标准切面检测\code\GetHeartBox.py

1. 训练YOLOv5(检测标准切面)
2. 将数据整理成所需格式

参照博客：

<https://blog.csdn.net/ai_faker/article/details/107099907>

＃train和Val的数据集（图像目录或\* .txt与图像路径文件）  
train： …/coco128/images/train2017/

val： …/coco128/images/train2017/

＃图像

数据集/图像/ train2017 / 000000109622.jpg

＃标签  
数据集/标签/ train2017 /000000109622.txt

需要训练的数据：

# 四腔心水平横切面（心尖、心底、胸骨旁）：  
# 左心房、左心室、右心房、右心室、房室间隔十字交叉、室间隔、降主动脉、脊柱、肋骨  
# 右室流出道切面:  
# 右心室、右室流出道及主肺动脉、主动脉弓、升主动脉、降主动脉、上腔静脉、脊柱  
# 3VT切面：  
# 主动脉弓、主肺动脉及动脉导管、气管、上腔静脉、降主动脉、脊柱  
# 左室流出道切面  
# 左心室、右心室、室间隔、左室流出道及主动脉、脊柱  
# 心底短轴切面 Short-axis  
# 右心室、右心房、主肺动脉及动脉导管、升主动脉、降主动脉、脊柱、右肺动脉

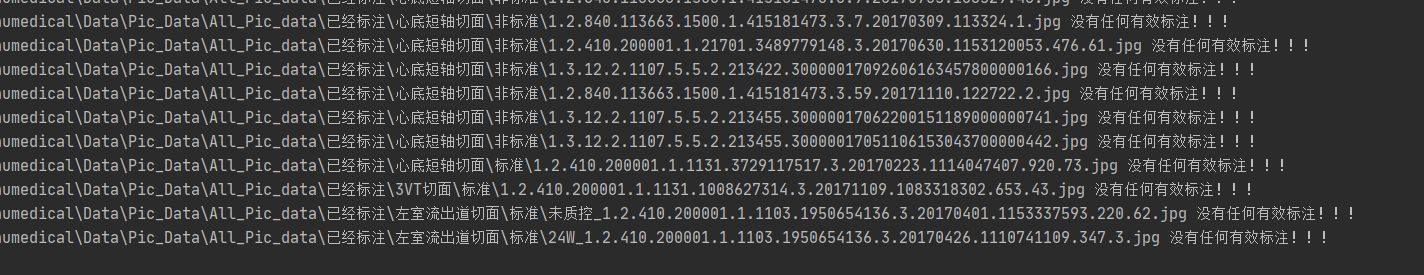
统计现有的结构信息：

|  |  |
| --- | --- |
| 切面 | 结构信息 |
| 3VT切面 : 2639 | 主肺动脉及动脉导管 : 2556 主动脉弓 : 2553 降主动脉 : 1724 上腔静脉 : 2395 气管 : 2182 脊柱 : 2425 心脏 : 2398 肺动脉及肺动脉导管 : 11主肺动脉和动脉导管 : 1 主肺动脉合并动脉导管 : 8 主肺动脉 : 1 左肺动脉 : 1 胸骨 : 1 升主动脉 : 1 |
| 心底短轴切面 : 1512 | 右心室 : 1435 主肺动脉及动脉导管 : 921 右肺动脉 : 750 升主动脉 : 1429 心脏 : 1081 右心房 : 1137 脊柱 : 1111 降主动脉 : 593 上腔静脉 : 2 主肺动脉及分叉 : 469 右上腔 : 3 主动脉弓 : 10 左肺动脉 : 20 右室及其流出道 : 7 主肺动脉 : 44 动脉导管 : 6主肺动脉合并动脉导管 : 4 肋骨 : 3 |
| 右室流出道切面 : 1952 | 右心室 : 1837 右室流出道及主肺动脉 : 664 脊柱 : 1418 心脏 : 1795  主肺动脉及动脉导管 : 1162 升主动脉 : 563 降主动脉 : 922 上腔静脉 : 956 主动脉弓 : 868 气管 : 571 动脉导管 : 6 主肺动脉 : 80  主肺动脉合并动脉导管 : 8 右心房 : 27 右上腔 : 5 右室及其流出道 : 4肋骨 : 17 右室流出道 : 8 左心室腔 : 5 室间隔 : 43 右心室腔 : 2  左室 : 51 左肺动脉 : 5 右肺动脉 : 12 右室 : 30 右心室壁 : 1 |
| 左室流出道切面 : 5518 | 左心室 : 5251 左室流出道及主动脉 : 5249 心脏 : 5215 右心房 : 84右心室 : 4433 室间隔 : 4367 脊柱 : 3073 降主动脉 : 1077 左室及左室流出道 : 13 左心房 : 246 肋骨 : 25 主动脉瓣 : 2 右心室壁 : 3 升主动脉 : 1 X: 3 二尖瓣 : 1 左室流出道 : 1 |
| 4C 四腔心水平横切面 : 8183 心底四腔心切面 : 1500 心尖四腔心切面 : 402 胸骨旁四腔心切面 : 1551 | 左心室 : 6594 降主动脉 : 7037 右心房 : 6745 右心室 : 6559 左心房 : 6454 脊柱 : 8074 室间隔 : 6885 房室间隔十字交叉 : 6670 心脏 : 5862 肋骨 : 13739 X: 1 |
| 脊柱 : 1387 降主动脉 : 1369 室间隔 : 1446 房室间隔十字交叉 : 1419  肋骨 : 2695 左心室 : 1108 右心房 : 1108 左心房 : 2493 右心室 : 1108 心脏 : 1401 右心室壁 : 34 |
| 心脏面积 : 5 脊柱 : 369 左心室 : 6 右心室 : 6 右心房 : 6 左心房 : 1404 降主动脉 : 355 心脏 : 302 室间隔 : 365 房室间隔十字交叉 : 350  肋骨 : 454 三尖瓣开放 : 1 X: 1 |
| 脊柱 : 1503 室间隔 : 1481 降主动脉 : 1427 房室间隔十字交叉 : 1421  肋骨 : 2662 右心室 : 1073 左心室 : 1086 右心房 : 1055 左心房 : 2405 心脏 : 1229 三尖瓣关闭 : 4 心脏面积 : 3 二尖瓣关闭 : 1 |

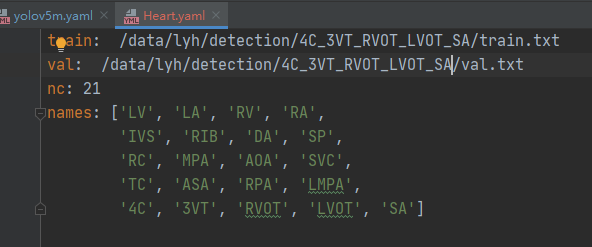
整理成txt格式

数据中存在的问题：

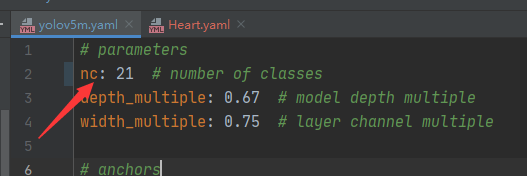
1. 标注的结构没有正确的结构名字 修改即可
2. 有一些图片是完全没有标注的(大部分是非标准图片)



1. 训练yolov5
2. 定义自己的数据集：heart.yaml



1. 修改要训练的cfg中的nc数目：yolov5m.yaml



注意图片和label的组织，文件夹名字要与datasets中一致



1. 训练指令

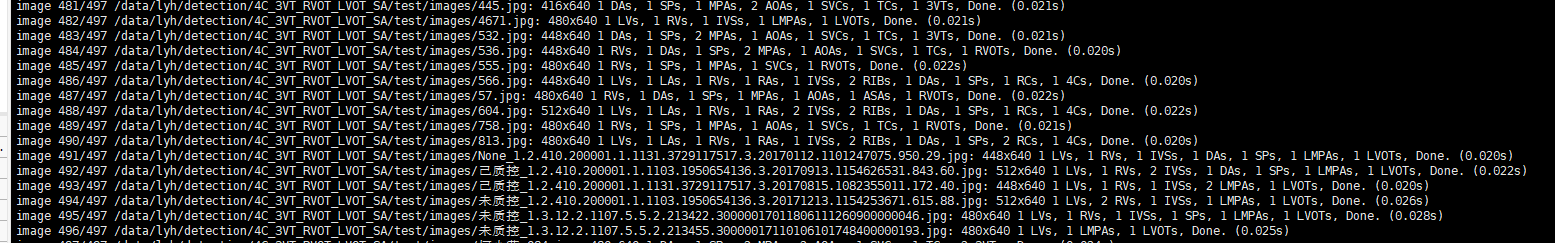
nohup python train.py --img 640 --epochs 50 --data ./data/Heart.yaml --cfg ./models/yolov5m.yaml --weights ./preweights/yolov5m.pt --epochs 50 --device 1,2 --batch-size 80 > train1204.log 2>&1 &

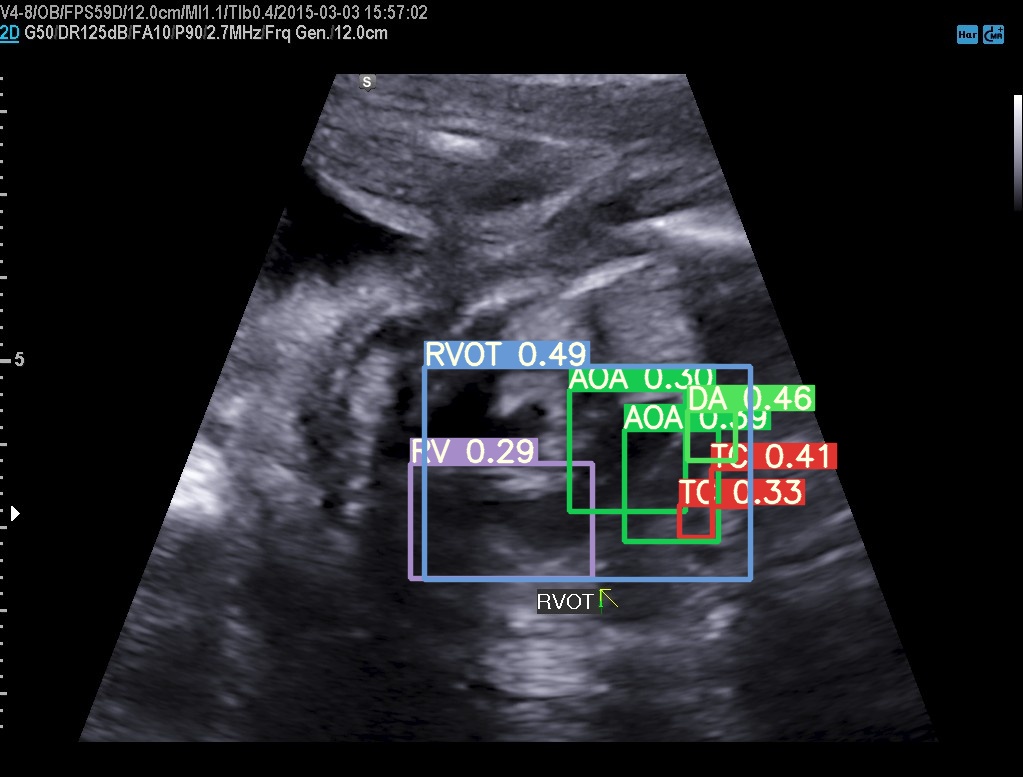


1. 测试指令

先测试一下看看训练的对不对

python detect.py --weights runs/train/exp10/weights/best.pt --source /data/lyh/detection/4C\_3VT\_RVOT\_LVOT\_SA/test/images/ --device 0,2



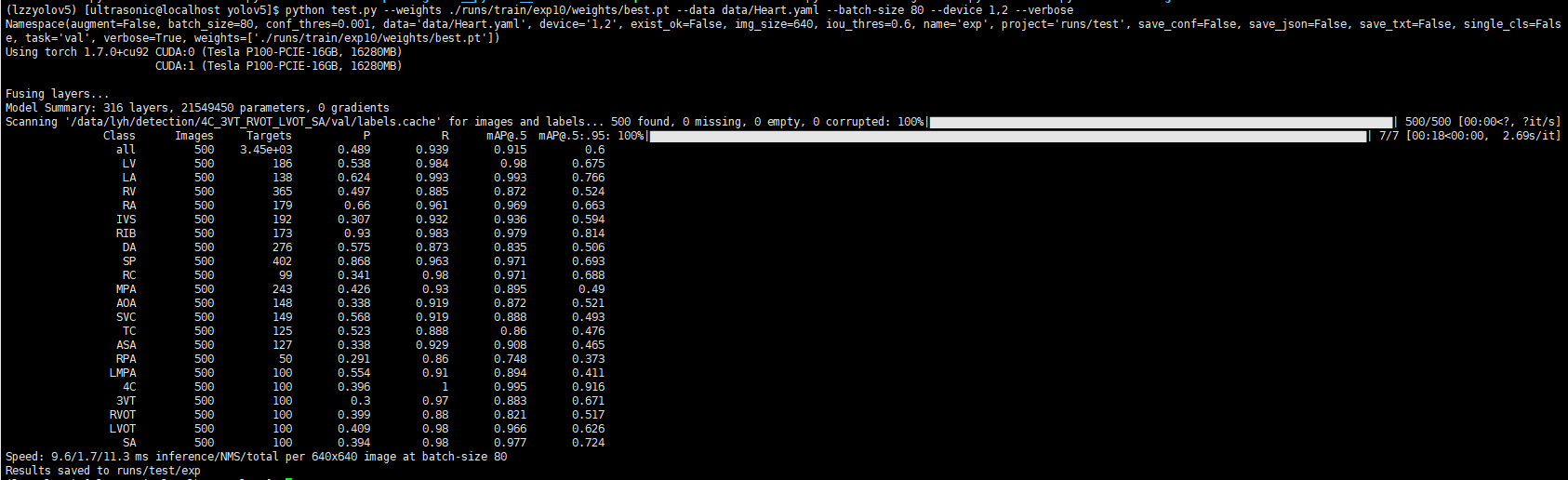


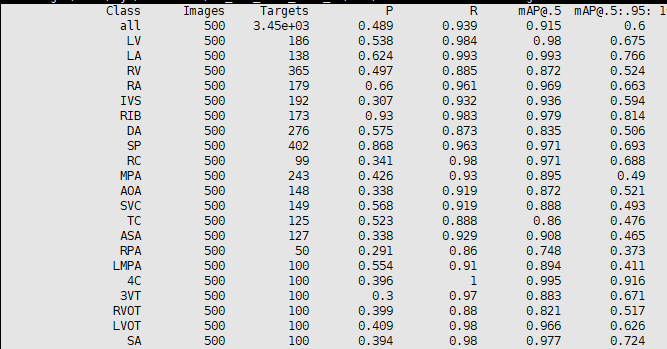
看来训练的没有问题

1. 验证指令

对最终训练好的best.pt测试验证

python test.py --weights ./runs/train/exp10/weights/best.pt --data data/Heart.yaml --batch-size 80 --device 1,2 --verbose





1. 更新测试文件 画框框写标签

①加入判断类别、打分

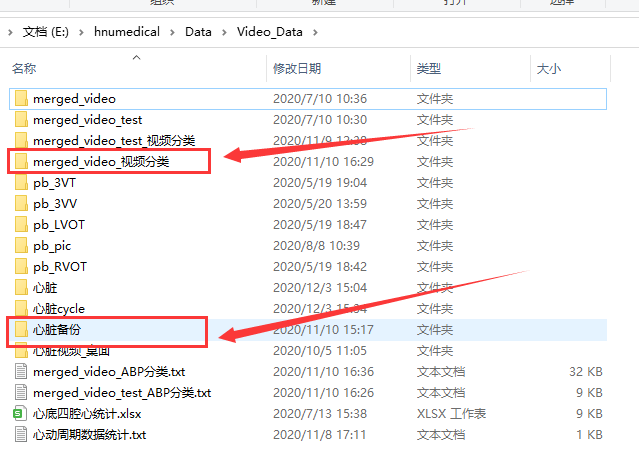
进行测试

1. 收到

**心动周期检测**

1. 数据整理

1、当前的数据来源【merge\_video】、【心脏】

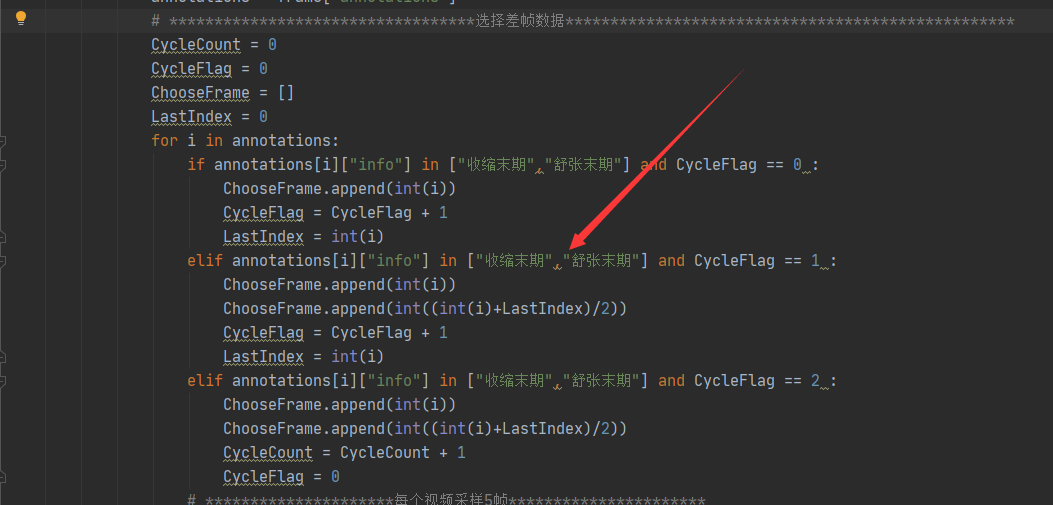


1. 训练需求

Video → 标准 & 周期 box，前后差帧

Picture → 非标准 box，自己和自己差帧

1. 整理数据为指定格式



原先的采样代码可能采样到连续的收缩(舒张)末期

video处理：

①先将心尖四腔心视频中的目标图片，原图片和差帧图片都保存下来

②用yolov5原先的模型检测原始图片的小结构

③通过求包围盒得到std、cycle的目标框

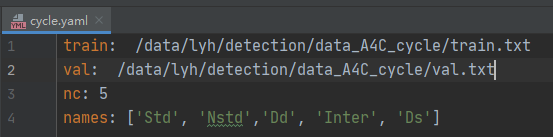
pic处理：

①将心尖四腔心非标准的图片挑选出来

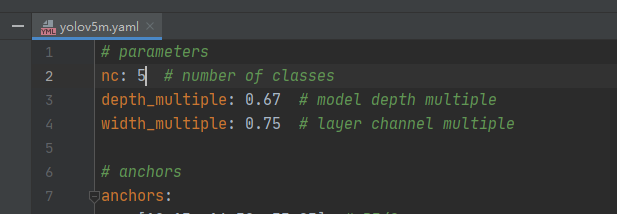
②自己和自己差帧、只保留nstd的框

1. 训练yolov5

定义自己的数据集：



修改yolov5m.yaml中的类别：



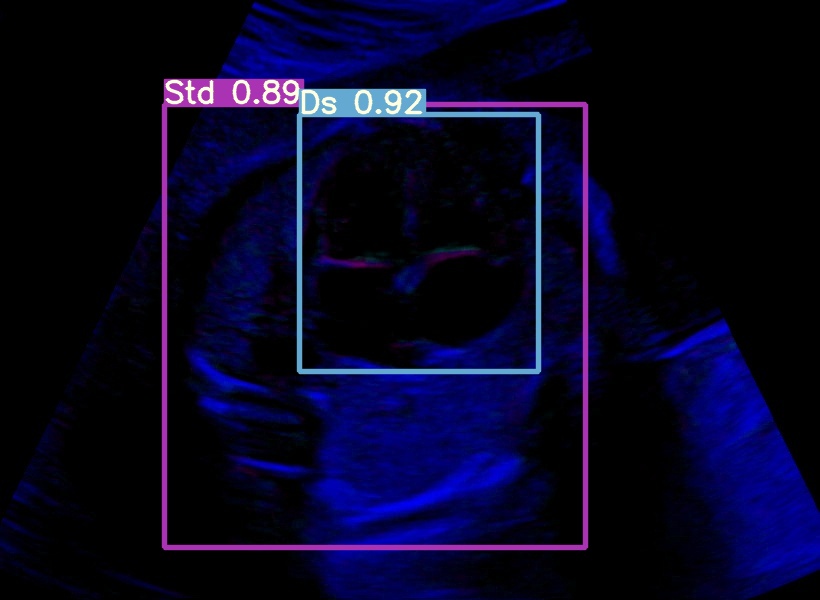
训练指令：

nohup python train.py --img 640 --data ./data/cycle.yaml --cfg ./models/yolov5m.yaml --weights ./preweights/yolov5m.pt --epochs 50 --device 1,2 --batch-size 80 > train1205.log 2>&1 &



测试

python detect.py --weights runs/train/exp2/weights/best.pt --source /data/lyh/detection/data\_A4C\_cycle/test/images/ --device 1,2

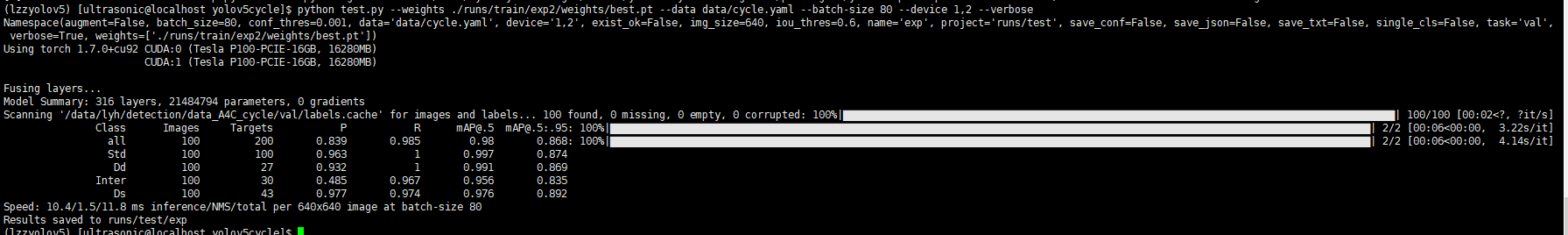


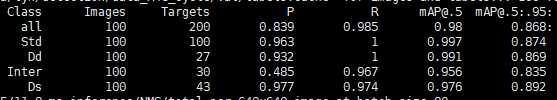
测试时候发现测试文件与之前的不共用了，model加载时候会报不匹配错，所以还需要把之前的后处理整理到新的detect文件中

验证

对最终训练好的best.pt测试验证

python test.py --weights ./runs/train/exp2/weights/best.pt --data data/cycle.yaml --batch-size 80 --device 1,2 --verbose





1. Dbd

**更新前的yolov5训练标准切面检测**

1. 数据组织

和之前训练新的yolov5共用一份数据

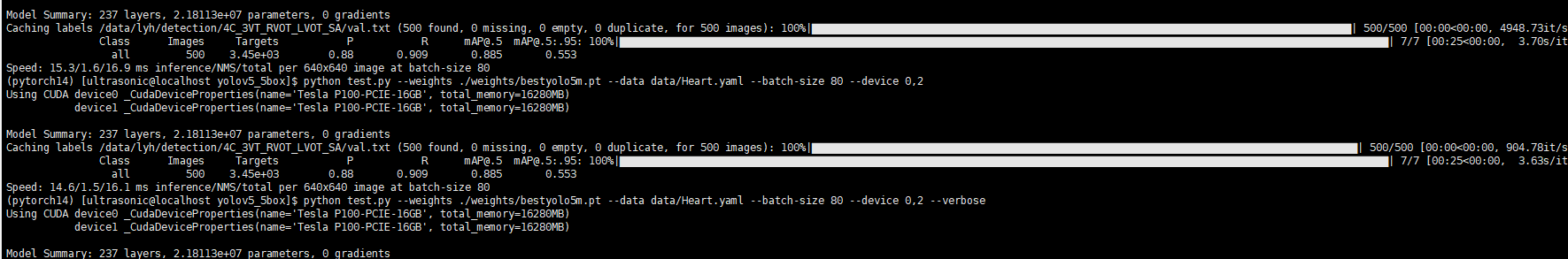
1. 训练
2. 注册自己的数据
3. 修改yolov5.yaml中nc数
4. 将datasets中图片文件夹名和标签文件夹名统一为images、labels
5. 训练

nohup python train.py --img 640 --data ./data/Heart.yaml --cfg ./models/yolov5m.yaml --weights ./weights/yolov5m.pt --epochs 50 --device 0,2 --batch-size 80 > train1205.log 2>&1 &



1. 验证

python test.py --weights ./weights/bestyolo5m.pt --data data/Heart.yaml --batch-size 80 --device 1,2 --verbose



但是之前的测试文件一直只输出总的ap值，不输出各个类别的ap值

重新训练



还是没有解决 网上搜索没有特别相关的解决办法 否则就是得更新升级版本