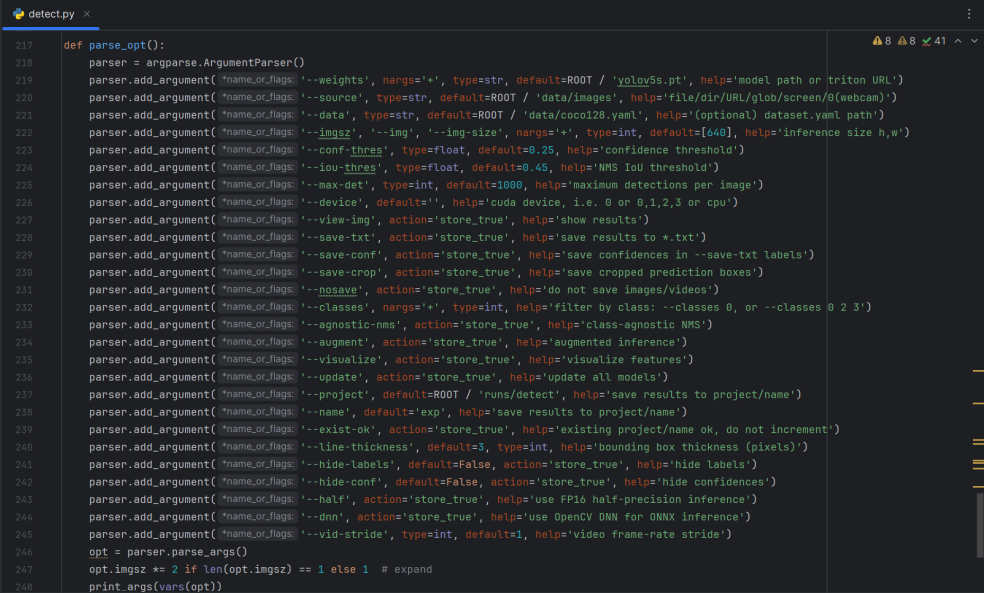
**场景人流识别**

1211002008 黄俊达

在本次的大作业中，我们小组使用了Yolov5+Deepsort相结合的模型进行场景人流识别检测。在本次大作业过程中，学习了如何搭建深度学习系统的虚拟环境，学习如何规范使用Yolov5模型，并正确配置其所需环境。下面将介绍我在本次大作业中所学习到相关知识。

# 首先是对Yolov5模型检测的相关知识。

Yolov5中的detect.py文件，是用来运行检测Yolov5模型是否安装正确的文件，也是用来检测我们主文件是否可以完整运行的文件。下图是detect.py文件中的parse\_opt函数。



**图 1 parse\_opt函数**

介绍其中几个比较重要的参数。

* weight:训练好的模型。

如未训练则可以使用Yolov5本身的四个模型之一，可以在parse\_opt函数中更改，也可以在终端使用命令“python detect.py --weights yolov5s.pt”进行更改。

* source:检测的目标，可以是单张图片，视频，屏幕等

可以在parse\_opt函数中修改，也可以在终端输入命令“python detect.py --weights yolov5s.pt --source data/images/bus.jpg”进行运行。

* conf-thres:置信度阈值，越低框越多，越高框越少

只显示预测概率超过conf\_thres的预测结果。想让YOLO只标记可能性高的地方，就把这个参数提高。在detect.py里的def parse-opt()函数部分更改。

也可通过命令“python detect.py --weights yolov5s.pt --conf-thres 0.8”进行更改。

* iou-thres:IOU阈值，越低框越少，越高框越多

IOU值：预测框大小∩真实框大小 / 预测框大小∪真实框大小。（预测框与真实框的交集与并集的取值。）

iou\_thres在detect.py中：越大，则容易将对于同一个物品的不同预测结果当成对多个物品的多个预测结果，导致一个物品出现了多个预测结果；越小，则容易将对于多个物品的不同预测结果当成对同一个物品的不同预测结果，导致多个物品只出现了一个预测结果。

* imgsz：输入图片的大小
* max-det:最多检测的数量
* device:设备 cuda或者cpu之类 不指定即去寻找做合适的设备
* view-img:检测过程中会出现弹出一个框出来
* classes:可以制定检测的类别

# Yolov5进行模型训练的配置

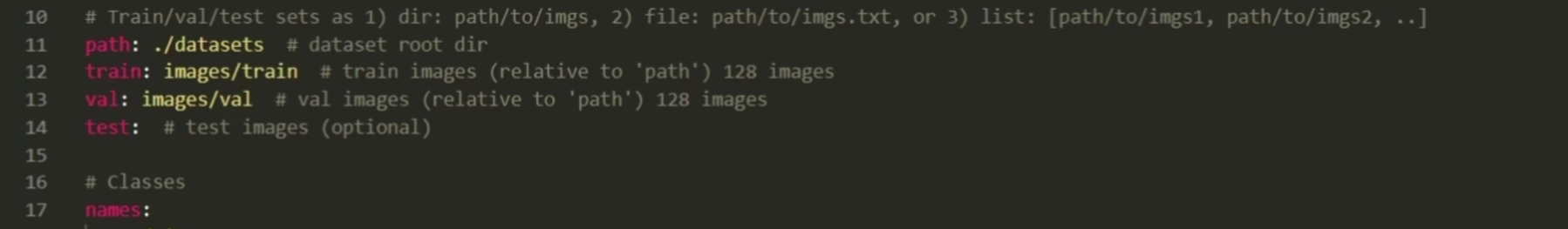
在总文档中，已经介绍了如何构建自己的训练集。现在介绍如何根据得到的训练集进行Yolov5模型的训练。

首先是将训练集分为images和labels两个文件夹，在两个文件夹下再分别设置train文件夹和val文件夹，images下的train文件夹用来存放训练集图片，val文件夹用来存放验证集图片；labels下的train文件夹用来存放训练集标签文件，要与训练集图片名称一一对应，val文件夹用来存放验证集标签文件，要与验证集图片一一对应。将images文件夹和labels文件夹存放在datasets文件夹下。将datasets文件夹拖入Yolov5文件夹下。



**图 2 训练集分类**

而后，前往yolov5文件夹下的data文件夹，复制一份coco128.yaml文件，并将其中的数据改成如下图所示。Classes类别可根据训练集中的classes调整。



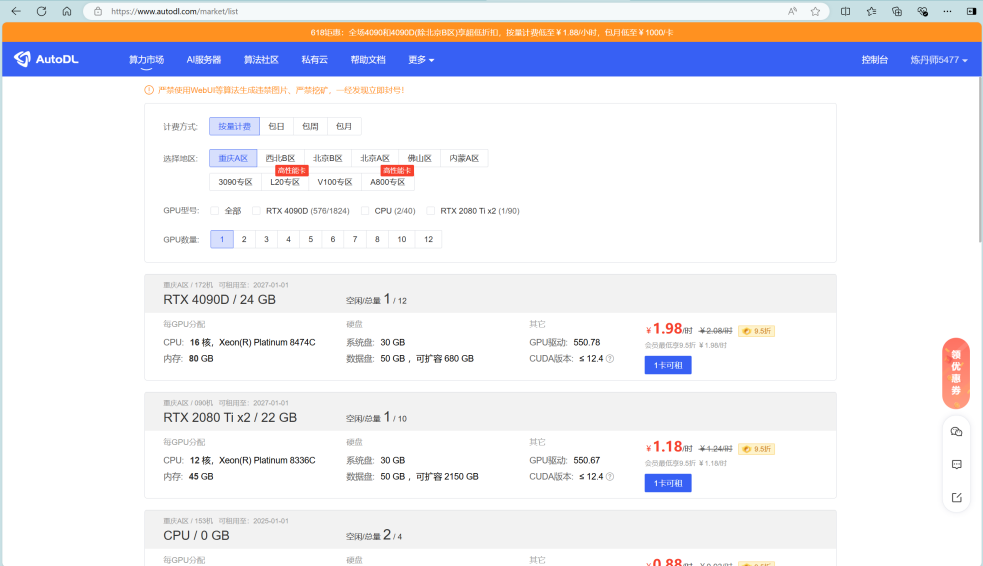
**图 3 修改.yaml文件**

并将该.yaml文件重命名成bvn.yaml文件。而后前往train.py文件，将其中parse\_opt函数中的data参数改成“data/bvn.yaml”。在命令行中输入“python train.py”，即可进行训练。

但是我的电脑没有NVDIA的芯片，使用的AMD芯片，故而运行效果不佳，下面将介绍使用AutoDL算力云平台进行模型训练以及相关训练后参数说明。

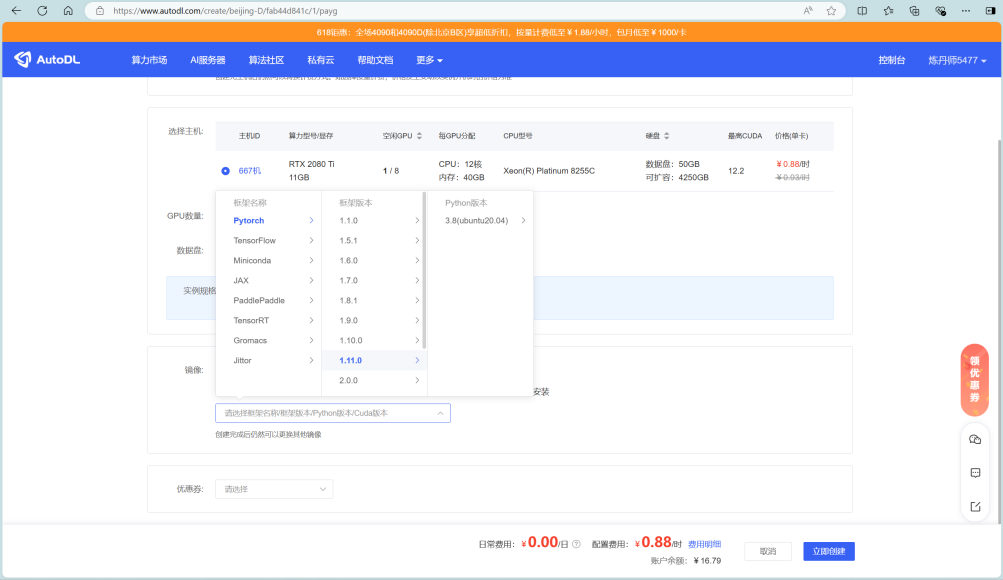
1. **AutoDL算力云平台模型训练**

可以直接搜索AutoDL或者前往网站“https://www.autodl.com/market/list”。



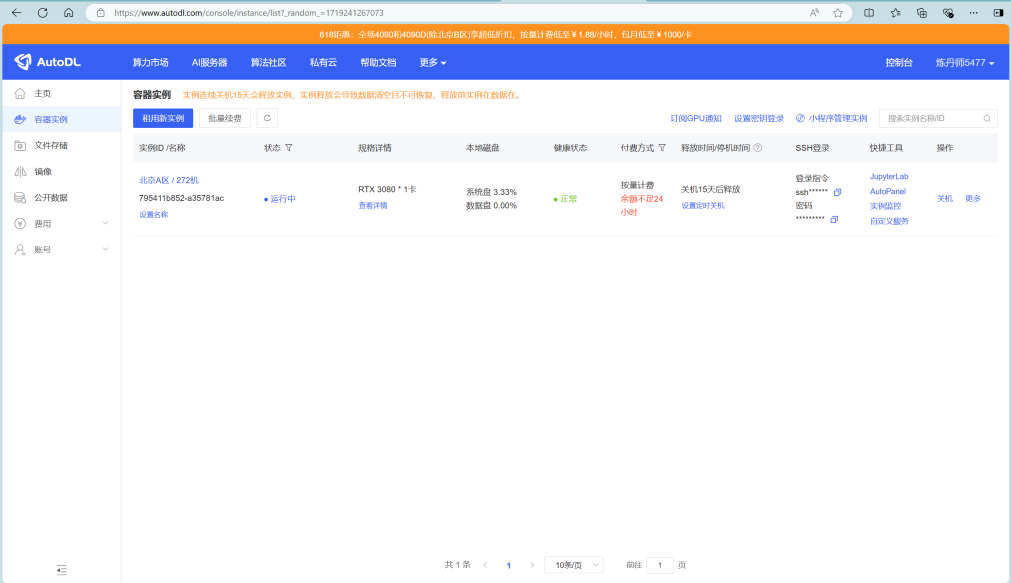
**图 4 AutoDL算力云平台首页**

这个缺点在于需要花钱，但是我总体使用下来其实也不是很贵，可能是我的训练时间比较短。直接选择一个比较合适的GPU即可，地区没有很大的关系，当然比较建议选择离自己地区比较近。如果训练集不是很大的话，建议选择不是太高的GPU型号，不然可能费用会比较高。选择好机子之后，就需要配置环境。比较建议的根据自己的Pytorch版本进行选择，不要使用模拟镜像，不然训练的结果都是零分。



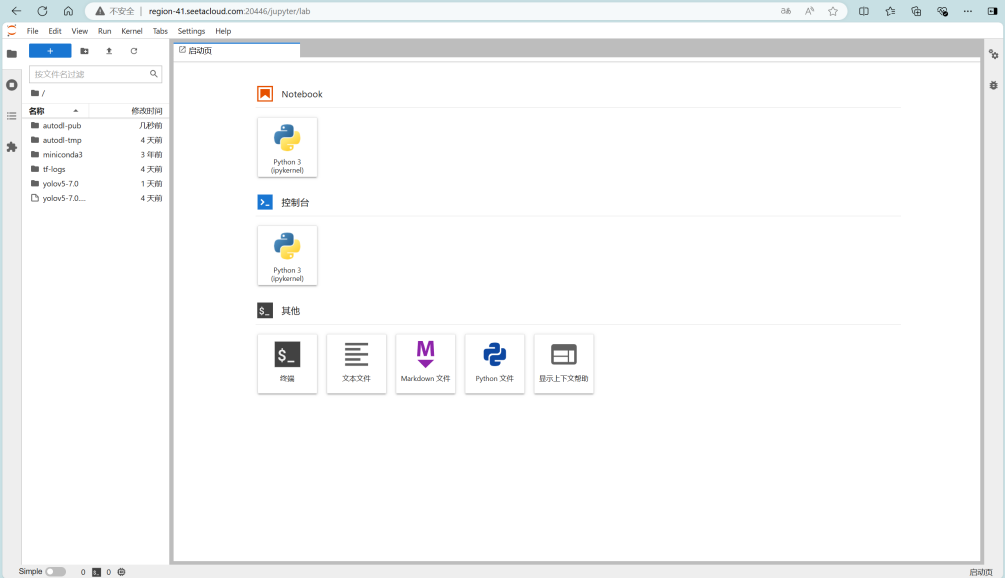
**图 5 配置环境**

租用之后，可以得到如下界面，选择开机。开机之后界面如下。



**图 6 开机之后的界面**

点击JupyterLab，即可前往如下界面。



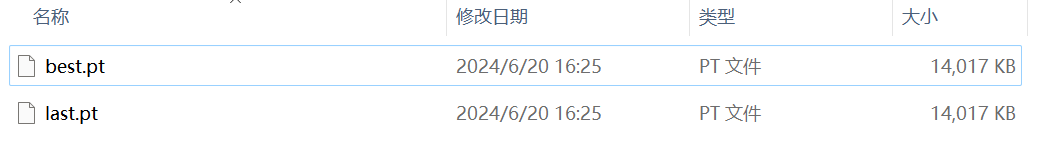
**图 7 JupyterLab界面**

上传yolov5压缩文件夹，打开终端，输入命令“unzip yolov5.zip”进行解压，而后“cd yolov5”。在终端中，需先输入命令“conda init”进行初始化，才可以进入虚拟环境。之后关闭终端，再次打开终端，即可进行训练。重复第二节的训练前准备，输入命令“python train.py”即可进行训练。得到的模型结果保存在runs/train/exp文件夹下，在终端输入命令“tar -cvf runs.tar.gz runs/”即可将模型文件夹打包，进行下载。

1. **训练所得数值分析**

在使用AutoDL平台进行模型训练后，得到多项数据，下面对这些数据进行解释说明。

1. **weights模型权重**



**图 8 weights模型权重**

以上两个文件就是我们训练好的权重，best.pt即训练过程在验证集上表现最好的模型权重，在训练过程中，每个epoch结束后都会对验证集进行一次评估，记录最好的模型权重。last.pt即训练模型训练最后一次迭代结束后的模型权重。

二者的区别是best.pt用于训练后对模型进行推理和部署；last.pt用于继续训练模型，包含最后一次训练迭代结束时的模型权重，可以从上一次训练结束的地方继续训练模型。

1. **confusion\_matrix（混淆矩阵）**

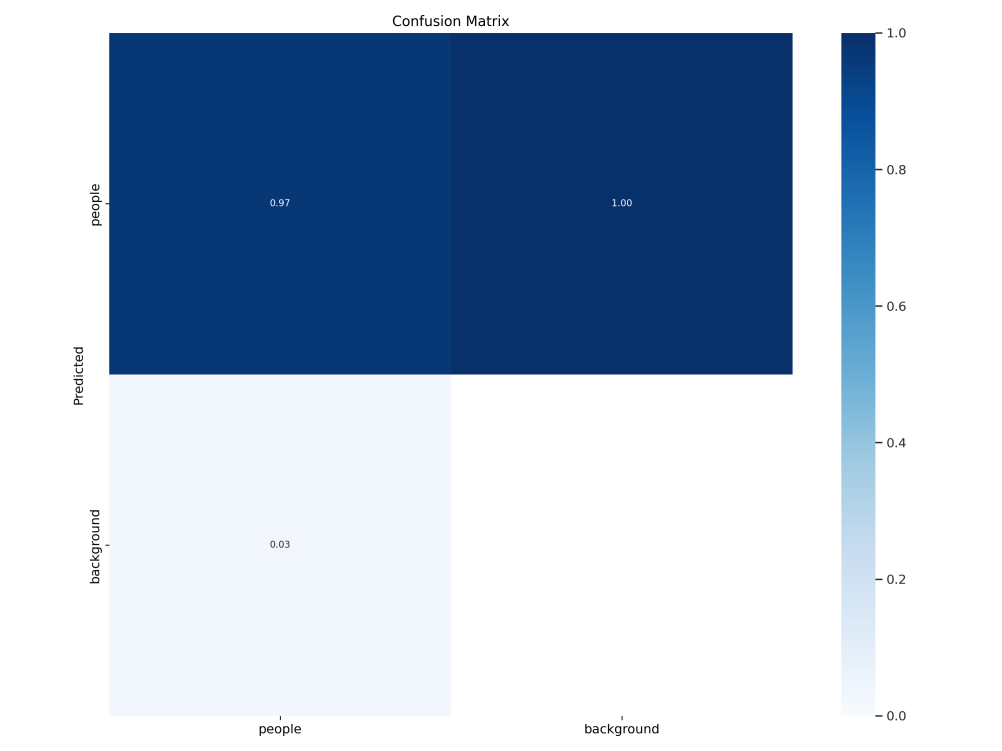
混淆矩阵，又称为可能性矩阵或错误矩阵。混淆矩阵是可视化工具，用于监督学习，在无监督学习一般叫做匹配矩阵。在图像精度评价中，主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个混淆矩阵里面。

**表 1 混淆矩阵结构**

|  |  |  |
| --- | --- | --- |
|  | 实际为正类别 | 实际为负类别 |
| 预测为正类别 | True Positive(TP) | False Positive(FP) |
| 预测为负类别 | False Negative(FN) | True Negative(TN) |

TP：模型将实际为正类别的样本正确预测为正类别。FP：模型将实际为负类别的样本错误预测为正类别。FN：模型将实际为正类别的样本错误预测为负类别。TN：模型将实际为负类别的样本正确预测为负类别。

混淆矩阵的每一行代表了预测类别，每一行的总数表示预测为该类别的数据的数目；每一列代表了数据的真实归属类别，每一列的数据总数表示该类别的数据实例的数目。



**图 9 confusion\_matrix**

从混淆矩阵得到分类指标。

精确率(Accuracy)：精确率用来表示模型的精度，即模型识别正确的个数/样本的总个数。Accuracy = (TP+TN)/(TP+FN+FP+TN)

正确率/准确率(Precision)：表示在模型识别为正类样本中，真正为正类的样本所占比例。Precision = TP/(TP+FP)

召回率/查全率(Recall)：表示模型正确识别为正类的样本的数量占总的正类样本数量的比值。Recall越高，说明有更多的正类样本被模型预测正确，模型的效果越好。Recall = TP/(TP+FN)

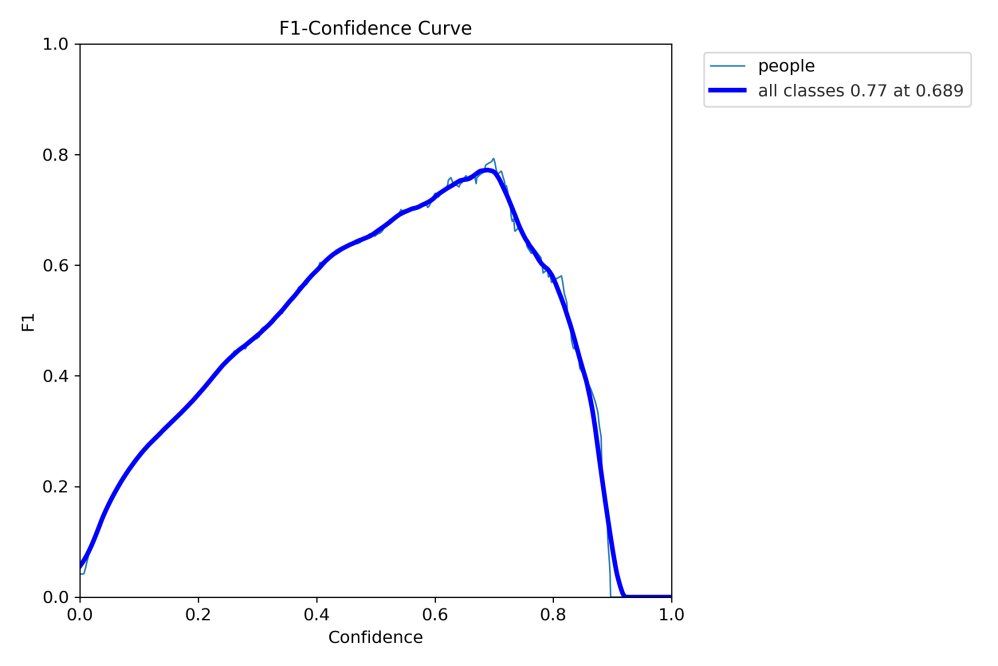
平均精度(Average Precision,AP)：AP衡量模型在每个类别的好坏。

均值平均精确度(Mean Average Precision,mAP)：mAP衡量在所有类型上的好坏，mAP取所有类别AP的平均值。mAP@0.5：当IoU为0.5时的mAP。mAP@0.5：0.95当IoU为0.5：0.95：0.05时的mAP平均数。

F1-score：分类问题的一个衡量指标，可以用于评估模型在检测出所有目标的情况下的精确性和完整性。F1-score综合考虑精确度与召回率的指标，最大值为1，最小值为0。F1 = 2 \* (Precision \* Recall) / (Precision + Recall)

1. **F1\_curve（F1曲线）**

F1-Confidence Curve就是F1-Score随着Confience逐渐增高而变化的曲线。



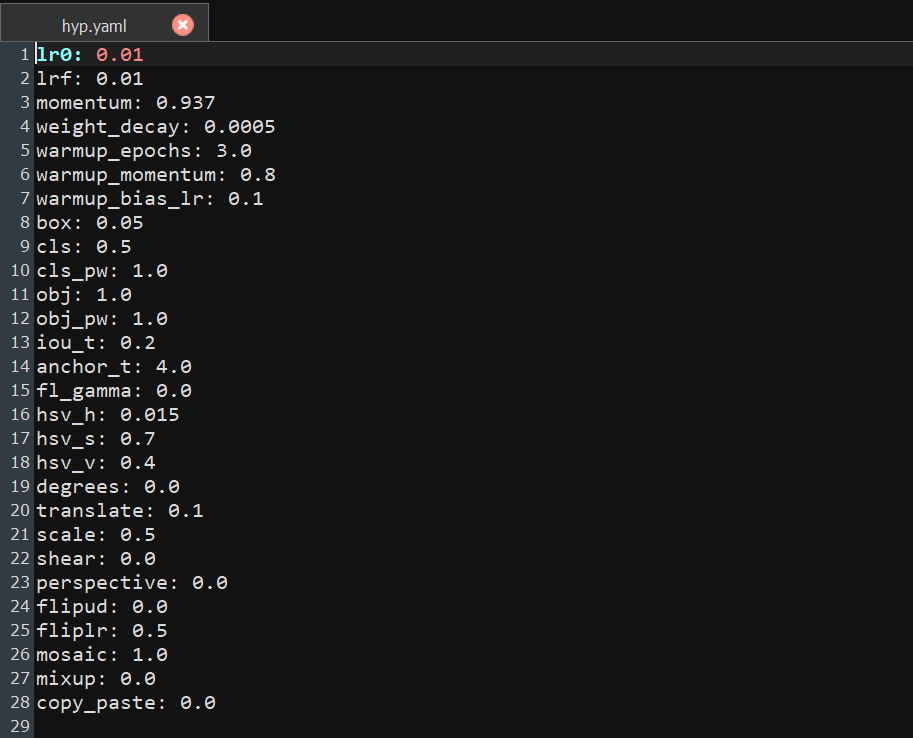
**图 10 F1\_curve曲线**

横坐标：Confidence（置信度）。纵坐标：F1-Score

置信度含义：模型判定一个物体并不是百分百判定它是属于某个分类，而是会给予它一个概率，Confidence就是设置的阈值，如果超过这个概率那么就确定为某个分类。

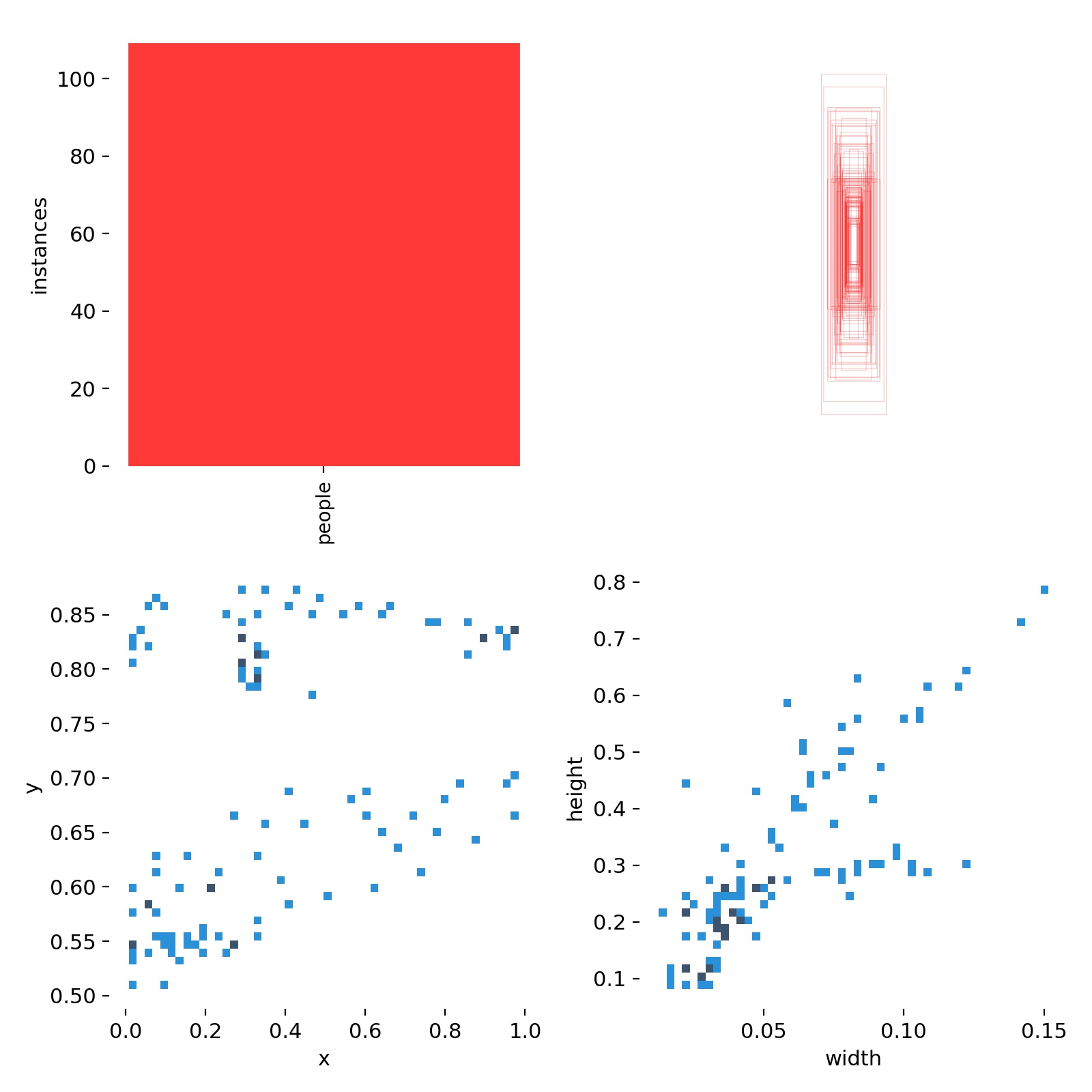
1. **hpy.yaml**

保存我们训练时指定的参数，内容如下：



**图 11 hpy.yaml文件**

1. **Labels**



**图 12 labels**

labels图片代表每个检测到的目标的类别和边界框信息。

1、第一个图是训练集的数据量，每个类别数；

2、第二个图是框的尺寸和数量；

3、第三个图是中心点相对于整幅图的位置；

4、第四个图是目标相对于整幅图的高宽比例；

1. **labels\_corrlogram**

含义：指的是一种图形，用于显示目标检测算法在训练过程中预测标签之间的相关性。

作用：体现中心点横纵坐标以及框的高度间的关系

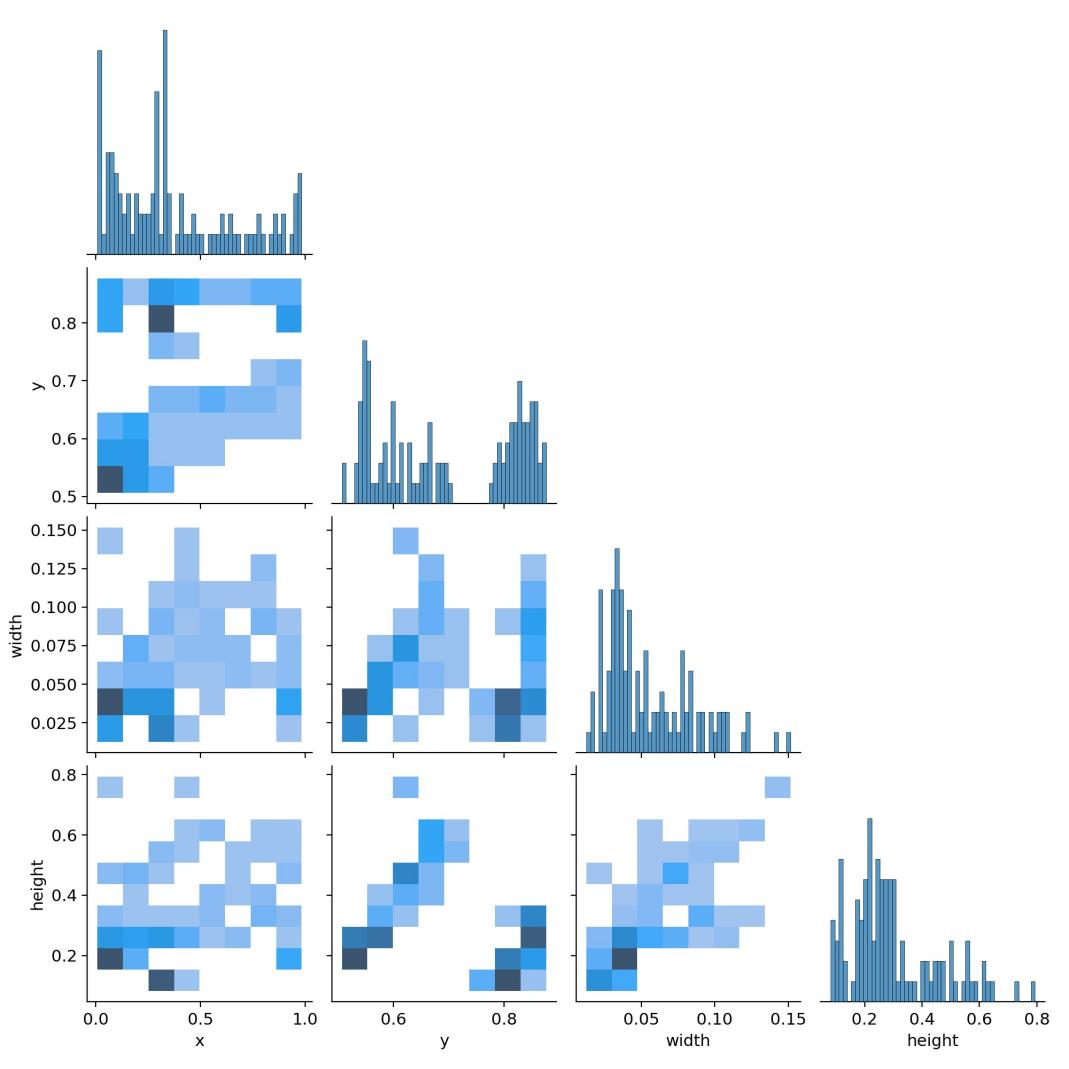
表示中心点坐标x、y，以及框的高宽之间的关系。每行的最后一幅图代表的是x，y，宽和高的分布情况；

1、第一行的图表明中心点横坐标x的分布情况，可以看到大部分集中在整幅图的中心位置；

2、第二行的图表明中心点纵坐标y与横坐标x，y的分布情况

3、第三行的图表明框的宽与横坐标x，y和款的宽的分布情况

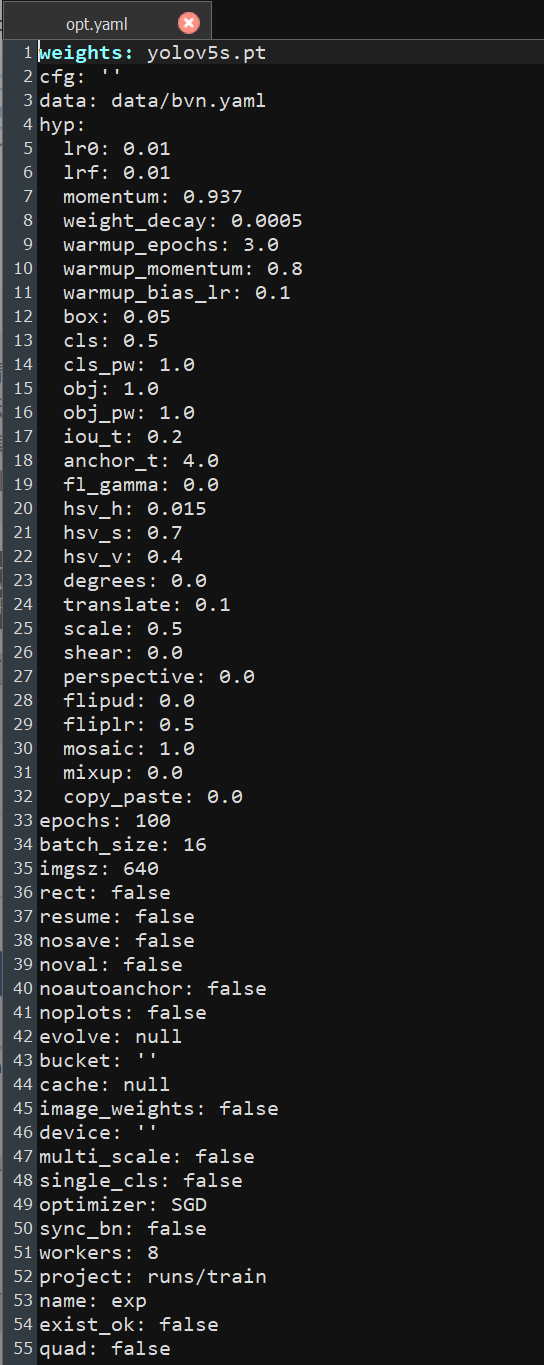
4、第四行的图表明框的高与横坐标x，y和框的宽、高的分布情况



**图 13 labels\_corrlogram**

1. **opt.yaml**

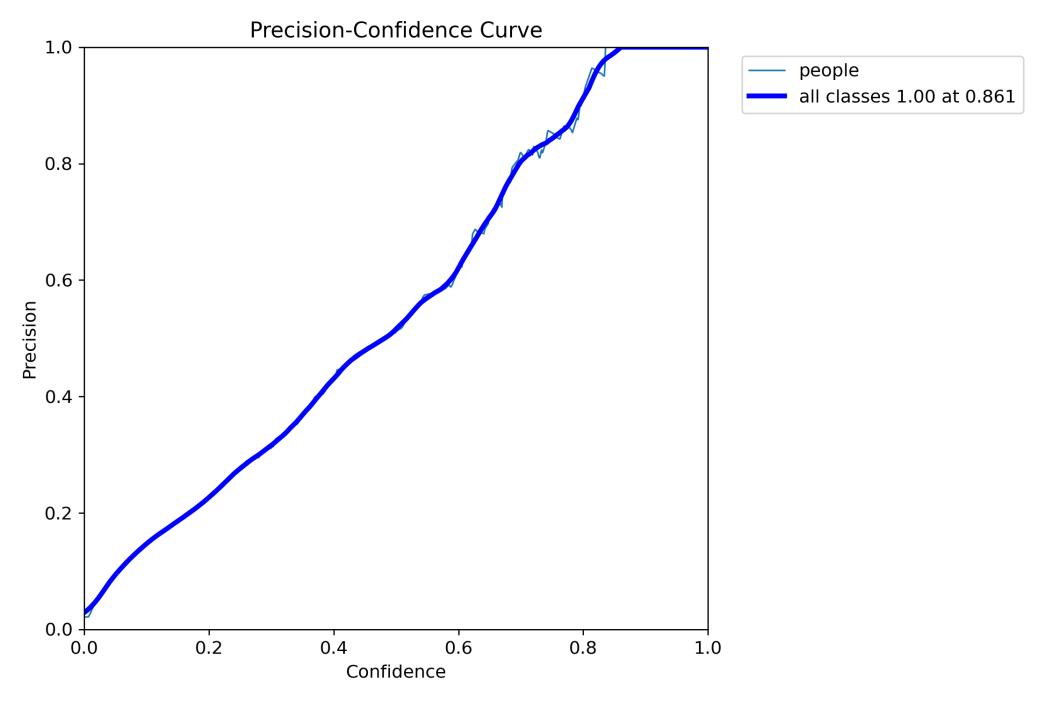
包含了数据集路径、权重路径，项目名称等一系列设置性参数。



**图 14 opt.yaml**

1. **P\_curve**

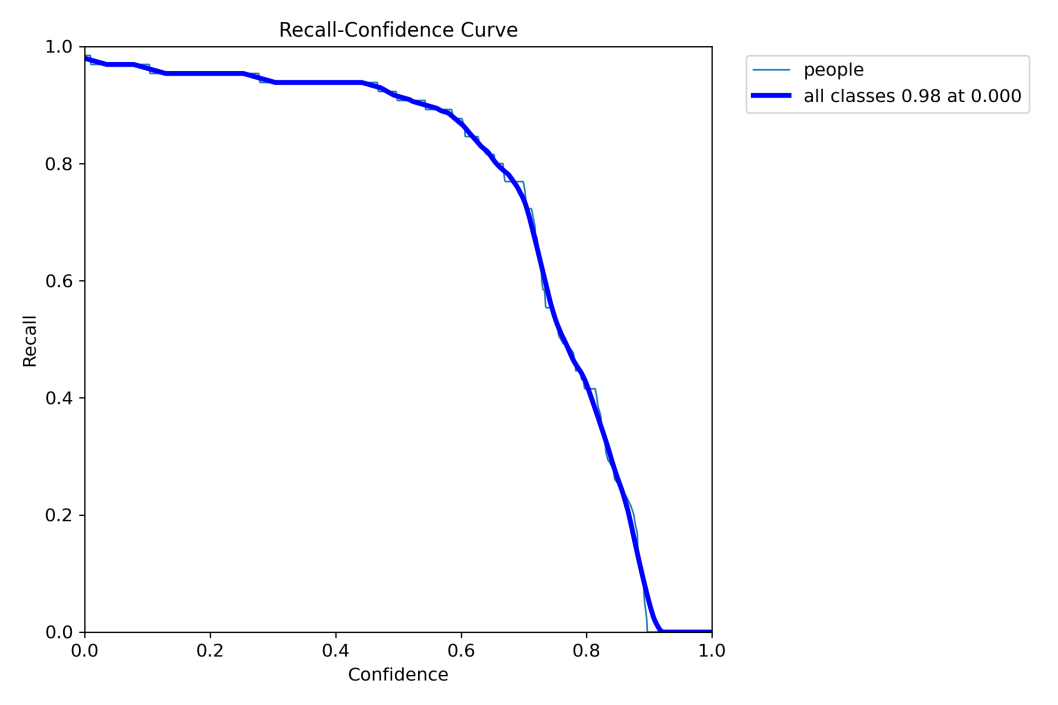
准确率（Precision）与置信度（confidence）之间的关系图。



**图 15 P\_curve**

1. **R\_curve**

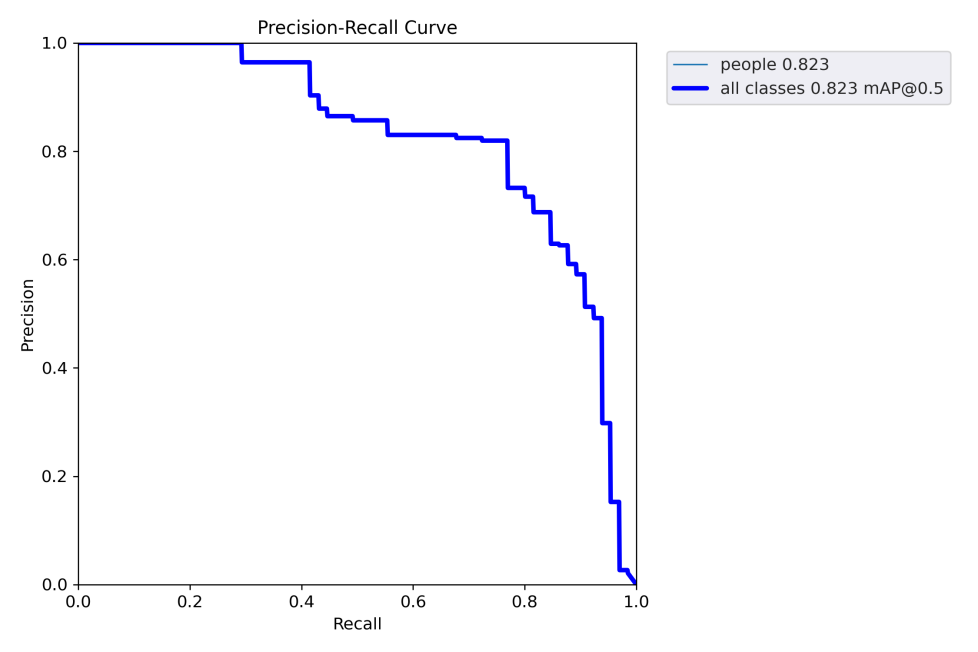
召回率（Recall）和置信度（Confidence）之间的关系。

****

**图 16 R\_curve**

1. **PR\_curve**

PR曲线中的P代表的是precision（精准率），R代表的是recall（召回率），其代表的是精准率与召回率的关系，一般情况下，将recall设置为横坐标，precision设置为纵坐标。PR曲线下围成的面积即AP，所有类别AP平均值即mAP。



**图 17 PR\_curve**

1. **result.csv**

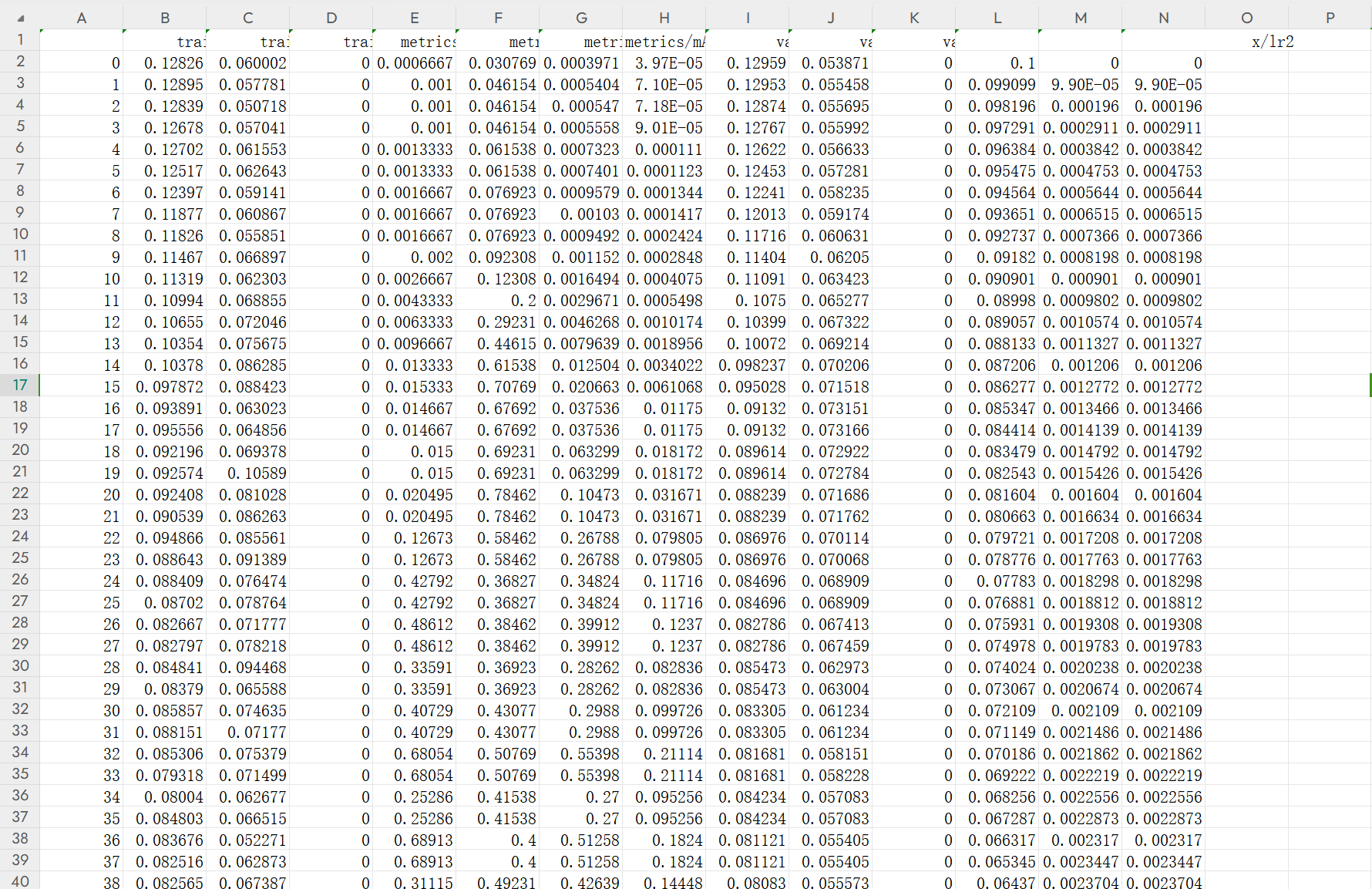
全部列从左往右依次为：训练轮数、训练集边界框损失、训练集置信度损失、训练集分类损失、精确率、召回率、均值平均精度mAP@0.5、均值平均精度mAP@0.5：0.95、验证集边界框损失、验证集目标检测损失、验证集分类损失、学习率。

结果loss：

1、定位损失box\_loss:预测框与标定框之间的误差（CIoU）,越小定位越准

2、置信度损失obj\_loss:计算网络的置信度，越小判定为目标的能力越准

3、分类损失cls\_loss:计算，锚框与对应的标定分类是否正确，越小分类越准



**图 18 result.csv**

1. **result.png**

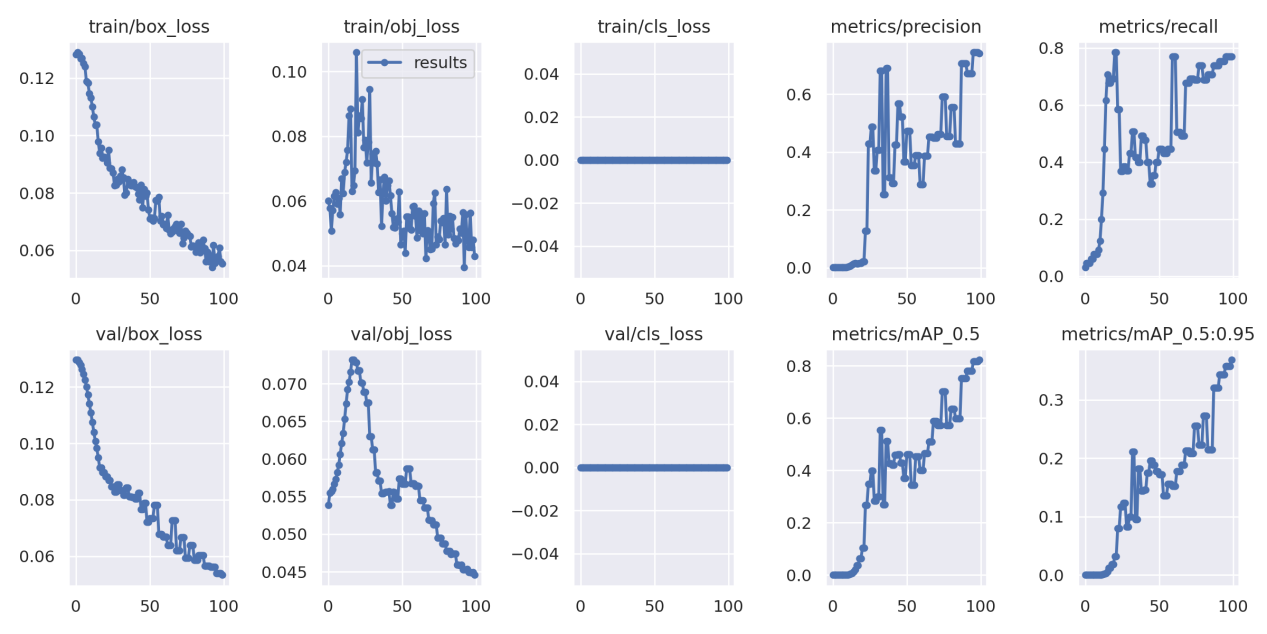
边界框损失（box\_loss）：预测框与标定框之间的误差，越小定位得越准；

置信度损失（obj\_loss）：目标检测loss均值，越小判定目标的能力越准；

分类损失（cls\_loss）：计算锚框与对应的标定分类是否正确，越小分类越准；

mAP@0.5:0.95（mAP@[0.5:0.95]）表示在不同IoU阈值（从0.5到0.95，步长0.05）（0.5、0.55、0.6、0.65、0.7、0.75、0.8、0.85、0.9、0.95）上的平均mAP；

mAP@0.5： 表示阈值大于0.5的平均mAP



**图 19 result.png**

1. **train\_batchx**

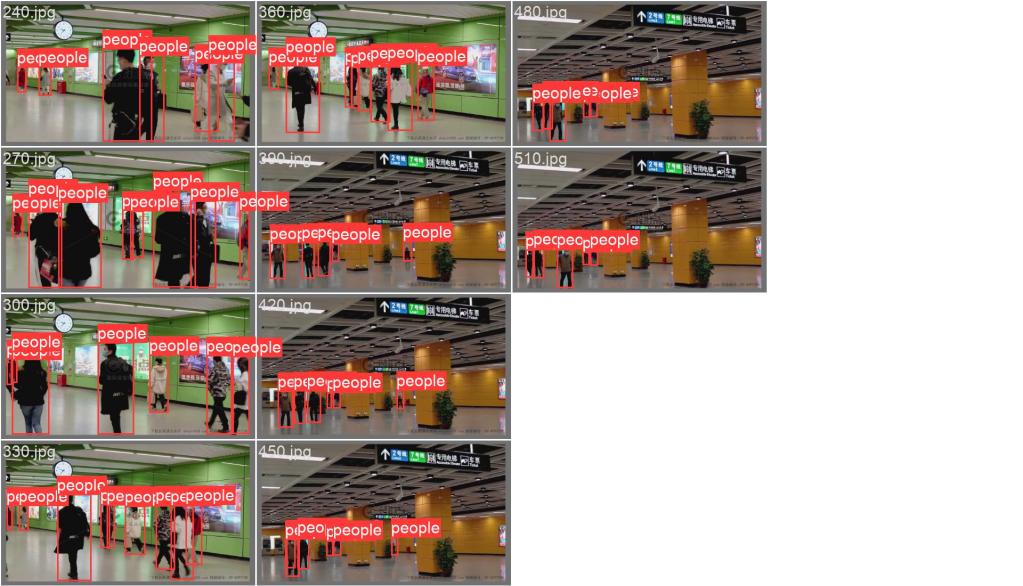
用于训练模型性能的标签：是YOLOv5模型训练集的图片，其中x表示训练集的批次编号。train\_batchx中每个图片都有一个对应的标签，用于描述图像中的目标的位置和类别信息。



**图 20 train\_batchx**

1. **val\_batchx\_labels**

验证集第x轮的实际标签：指验证集中一个batch的真实标签和框的信息，其中x为batch的编号。这些信息通常包括每个样本的分类标签和相应的边界框坐标。



**图 21 val\_batchx\_labels**

1. **val\_batchx\_pred**

验证集第x轮的预测标签：指验证集中一个batch的预测标签和框的信息，其中x为batch的编号。这些信息通常包括每个样本的分类标签和相应的边界框坐标。



**图 22 val\_batchx\_pred**