# 基于Yolov8的实时检测系统

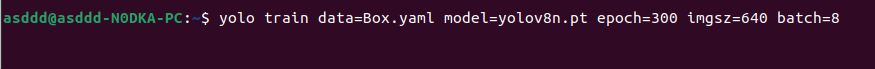
## 1.模型训练

基础模型采用官方的Yolov8模型，利用异常检测数据集对其进行再次训练。

根据提供的5种不同程度的盒体图像构建数据集，为了平衡正常和异常图像数量，同时避免模型误判，经过筛选后剔除部分模糊图像以及肉眼难以分辨的“轻微”类别图像。构建完成后的原始数据集中包含468张图片，214张异常图像，254张正常图像。随机划分出训练集378张（正常202张，异常176张），验证集42张（正常22，异常20），测试集47张（正常29，异常18）。

Yolo训练参数：epochs=300 imgsz=640 batch=8 ，利用Nvidia A10（24GB）训练，大约耗时1小时。模型大小约为6MB。

图 1 利用命令行输入参数对模型进行训练（配置文件的地址和其中数据集地址按照实际情况改写）



best.pt模型在测试集上取得良好结果，选择作为最终采用的模型。

## 2.界面设计

运行前请确认已部署好python环境并下载ultralytics和文本

AI 生成的内容可能不正确。Pyqt库。环境配置完成后，运行yolo\_UI.py进入界面。

图 2 运行py程序

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图 3 程序初始界面

首先选择进行检测的输入源，支持摄像头输入视频流检测（实际帧率受到部署设备算力影响），对视频每一帧进行检测，对单张图像进行检测。默认使用同级目录下的best.pt模型。

电脑软件截图

AI 生成的内容可能不正确。

图 4 对视频进行推理的运行示例

## 3.异常检测

模型会对每一视频帧或图像进行检测，具体实现上，是对每一帧调用check\_abnormal()方法，该函数会解析模型推理后输出的result对象，如果其中的conf置信度属性高于设置的阈值就会在结果栏打印一行警告文本，后续可在此处完善向主控报警等逻辑。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图 5 check\_abnormal()方法的实现