一、项目概述

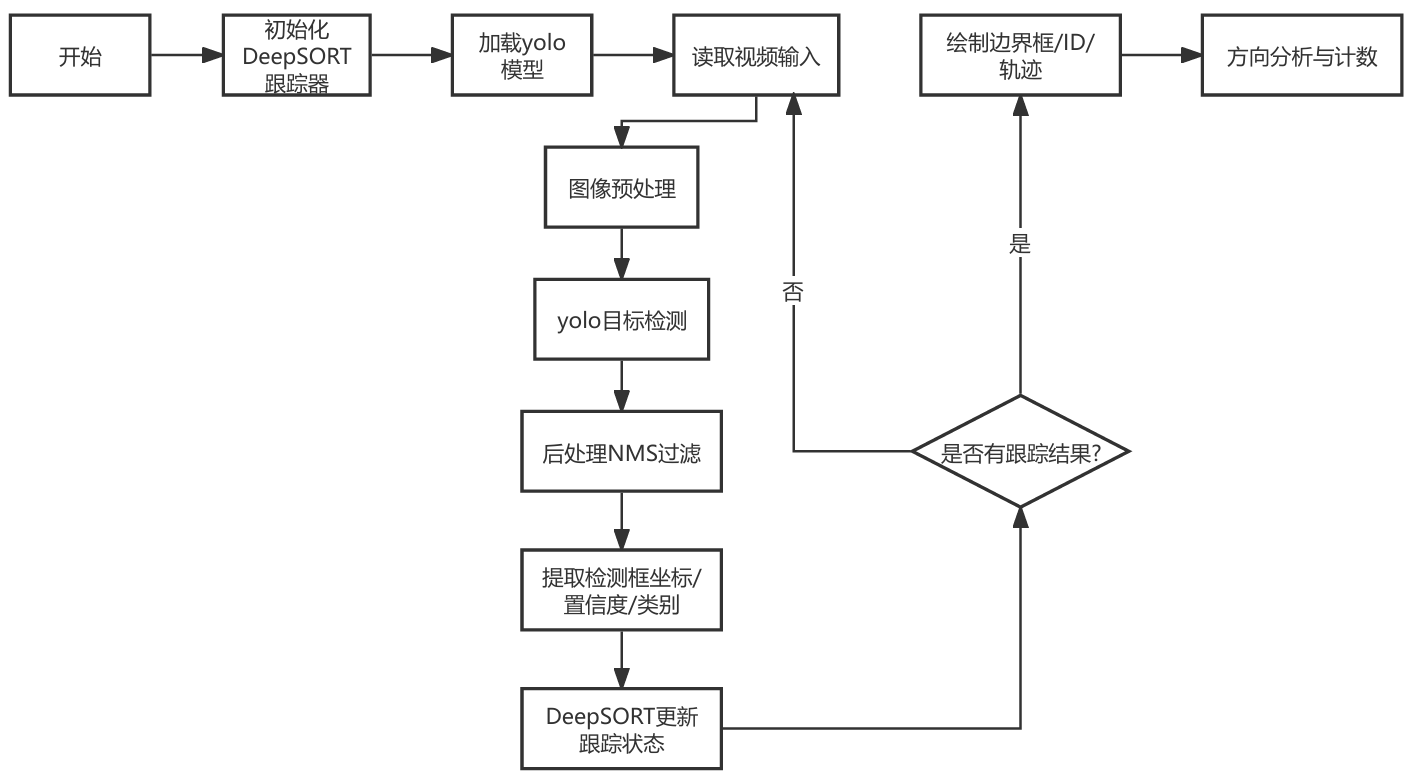
本项目基于实际视频监控场景，利用 YOLO 系列模型（YOLOv5 和 YOLOv8）与 DeepSORT 跟踪算法，实现对人流的检测与计数。

二、项目来源

本项目参考自GitHub 开源仓库：

git@github.com:dyh/unbox\_yolov5\_deepsort\_counting.git

三、整体思路与核心算法



目标检测

1. 使用 YOLOv5或YOLOv8 进行目标检测。

2. 支持检测类别：人、自行车、汽车、摩托车等。

3. 对检测结果执行非极大值抑制（NMS），去除重叠冗余框。

目标跟踪

1. 使用 DeepSORT 算法对检测目标进行跨帧跟踪，生成唯一的跟踪 ID。

2. 每个ID与类别标签、检测框绑定，可追踪个体的跨帧轨迹。

进出区域统计

1. 在画面中自定义两个多边形区域（如上下行通道），如黄色区域表示下行，蓝色区域表示上行。

2. 实时判断目标 ID 是否进入/离开指定区域。

3. 若目标从一个区域到另一个区域，计数器加一。

四、代码结构与功能说明

Main.py

核心入口，加载检测器与跟踪器，使用 OpenCV 处理视频流，叠加检测框、跟踪ID与计数结果。

Detector.py

加载YOLO模型权重文件，对输入帧进行预处理，执行推理，输出检测框、类别、置信度。

Tracker.py

调用DeepSORT，分配ID，并提供ID与标签的可视化接口。

五、实验结果

YOLOv5和YOLOv8在相同场景下的检测表现对比：

- YOLOv8 速度更快，检测更灵活。

- YOLOv8 在部分小目标或遮挡条件下，容易将人误识别为防火栓、自行车等。

- YOLOv5 表现相对稳定，但对小目标检测灵敏度较低。

六、问题分析

|  |  |
| --- | --- |
| 问题 | 分析 |
| YOLOv8 将人识别为非人物体 | YOLOv8 放弃 Anchor-Based 方法，直接预测中心点和宽高，对形状和上下文依赖更高，形变、遮挡时易受相似物体干扰 |
| YOLOv5 偶尔漏检 | Anchor-Based 方法对小目标相对不敏感，需更高分辨率或数据增强优化 |

七、改进方法与新思路

1. 数据集增强：增加易混淆物体数据，微调 YOLO 权重。

2. 结合时序上下文：利用相邻帧运动状态，过滤误识别。

3. 多模态方法：引入光流与背景建模，验证运动一致性。

4. 多区域自适应计数：根据实时人流密度自动调整检测区域，提升计数准确性。

八、总结

本大作业通过 YOLO + DeepSORT 实现了一个可用于实际场景的人流检测与计数系统，涵盖检测、跟踪、统计、可视化、分析与改进。

未来可继续尝试轻量化模型用于边缘部署、结合自适应阈值减少误报、引入更多传感器提升鲁棒性。