# 技术说明文档

1. **引言**

1.1编写目的

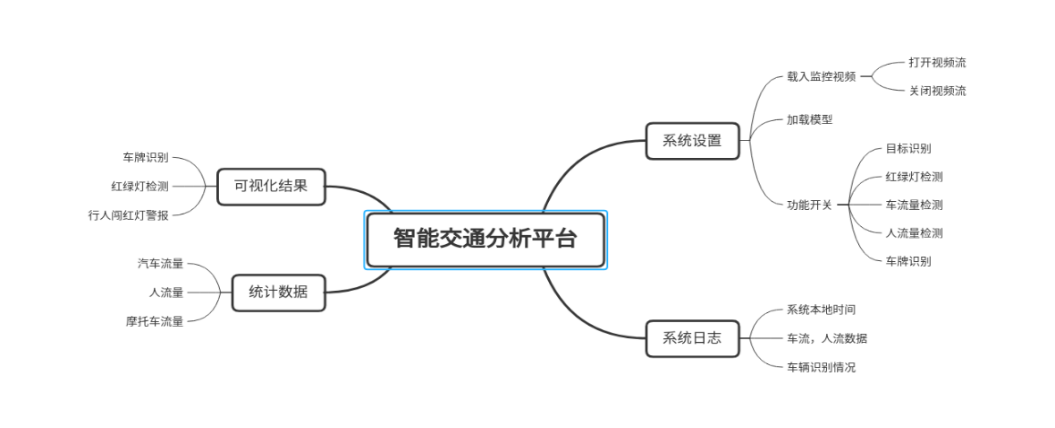
本文档主要用于阐述“智能交通分析平台”桌面应用所开发的各个模块的详细介绍及所用到算法的详细介绍。

1.2项目背景

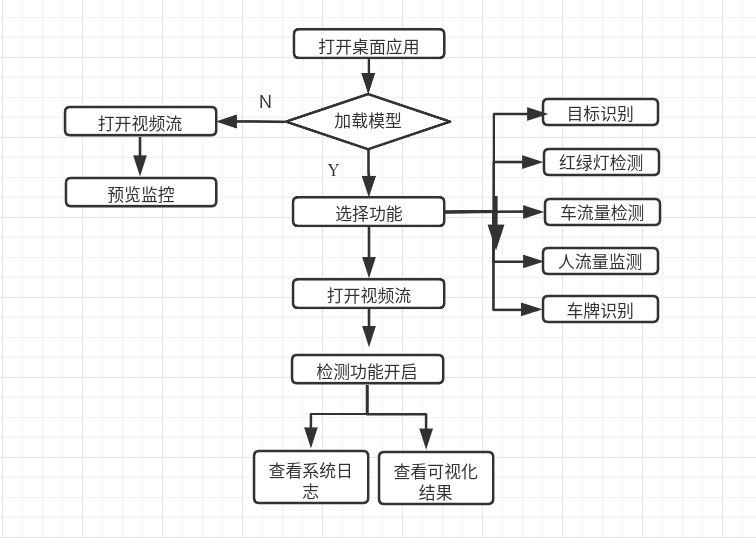
随着我国社会经济的迅速发展以及城市化进程的加快,我国机动车数量不断增长,导致交通问题日益严重,实施有效的交通监控对于解决日益增长的交通问题具有积极意义。伴随着人工智能进入国家战略层面的议程，以及人工智能技术的不断革新突破，智能交通系统在未来会成为必然的发展趋势,要实现交通系统的直观认识和方便手段,就要将大量的交通信息利用计算机的视觉技术进行处理。在这种技术应用上,不仅降低了交通的拥堵现象,实现交通的运输期间的畅通性,也减少了大量的交通事故,加强了交通的监管及安全性。

**2、总体设计**

2.1软件结构图



2.2软件流程图



**3、程序描述**

3.1各模块详细说明

**3.1.1 加载模型模块**

本模块主要用于提前加载多目标识别算法和车牌识别算法所用到的神经网络模型。

**3**.**1.2 载入视频流模块**

（1）功能介绍：本模块用于打开存储在本地文件系统上的交通监控视频并将视频流实时显示在软件界面视频播放模块中。

（2）详细实现：由于整个应用程序是基于pyqt5开发的，因此为了不影响整个软件其他功能模块的使用，我们使用opencv读入视频流并给这一模块加入单独的线程，将opencv视频流读取的图像帧显示在之前使用qt设计的视频播放模块内。

**3.1.3目标识别模块**

（1）功能介绍：本模块通过使用YOLOv3多尺度目标识别技术对视频流的图像帧进行识别，并将事先在训练类别中的目标在视频画面中实时标注出来，并将识别的信息提供给其他模块使用，比如红绿灯检测模块，车流量检测模块等。

（2）详细实现：本模块我们使用著名的YOLO(You Only Look Once)模型对图像中的物体进行识别。首先在模型构建方面，我们读取darknet框架使用的yolov3.cfg 卷积神经网络模型文件，对网络的结构进行分析然后转换为相应的pytorch模型，使用已经加载的yolov3.weight权重模型，至此我们的CNN识别模块已经搭建完成。然后将opencv读取的图像帧丢入已经定义的卷积神经网络模型对结果进行预测，得到输出信息（包括置信度、位置坐标），使用opencv作图模块将这些输出信息实时标注在对应的图像帧上。最后将处理过的图像帧显示在界面上即可。

**3.1.4红绿灯检测模块**

（1）功能介绍：本模块使用目标识别模块的输出信息得到交通灯的位置坐标，使用颜色识别算法对交通灯状态进行分析。

（2）详细实现：首先对pytorch预测输出的信息进行分析，得到交通灯对应的位置信息。接着，调用opencv模块将交通灯对应的ROI区域截取出来，由于HSV空间能够非常直观的表达色彩的明暗，色调，以及鲜艳程度，方便进行颜色之间的对比，因此将截取出区域的RGB图像先转换为HSV图像。最后将每种颜色对应的HSV阈值与截取区域中的HSV颜色进行比对，重合率最高的即为识别到的颜色种类。

**3.1.5车流量检测模块**

（1）功能介绍：本模块对图像帧中出现的车辆数目进行统计，实时分析交通路口的车流量，提供参考数据给使用人员。

（2）详细实现：首先对pytorch预测输出的信息进行分析，得到图像帧中所有识别到的车辆信息，对车辆信息进行统计，然后实时在界面上显示。

**3.1.6人流量检测模块**

（1）功能介绍：本模块对图像帧中出现的行人数目进行统计，实时分析交通路口的人流量，提供参考数据给使用人员。

（2）详细实现：首先对pytorch预测输出的信息进行分析，得到图像帧中所有识别到的行人信息，对行人信息进行统计，然后实时在界面上显示。

**3.1.7车牌识别模块**

（1）功能介绍：本模块对图像帧中出现的车辆车牌号进行识别，并将识别到车牌的图像区域和OCR识别结果实时显示在界面上。

（2）详细实现：使用HyperLPR 车牌识别开源项目，首先利用利用cascade进行车牌定位，其次利用左右边界回归模型，预测出车牌的左右边框，进一步裁剪，进行精定位，最后利用CRNN进行车牌字符识别。

**3.1.8系统日志模块**

（1）功能介绍：本模块对视频分析中得到的信息进行汇总，并实时在在画面上进行刷新，使用者可以方便地查看当前以及之前的分析结果；

（2）详细实现：首先定义一个日志消息队列，在视频分析开始运行之后将得到的信息使用put函数加载到队列当中，然后在日志显示模块使用get函数得到最新的消息并实时显示在画面上。