# 大型车辆转弯报警器项目研究报告

### 项目成果简介

本项目已取得研究报告一份，包含大型车辆转弯报警器可行性分析，分模块设计以及部分代码实现。另外也得到一份实物装置设计方案，包括电路图设计和代码实现。

### 2，项目研究的目的、意义

内轮差，顾名思义，是车辆转弯时，前轮的转弯半径与后轮的转弯半径之差。由于内轮差的存在，车辆转弯时，前、后车轮的运动轨迹不会重合。在行车中如果只注意到前轮的通过效果，而忽略了内轮差对行车轨迹的影响，就有可能造成后轮驶离路面或与其他物体相刮撞，引发交通事故。例如，当汽车右转时，驾驶员通常会借助右侧观后镜观察后方情况。但由于视野范围有限，有时会产生一定视觉盲区，尤其是车身高大、较长的货车，其产生的视觉盲区范围更大，由于私家车、摩托车、自行车、行人体积较小，一旦进入视觉盲区，极易发生危险。

避免此类惨剧的发生的一个构想是在车流量较大转弯路口设置大型车辆转弯报警器，当有大型车辆转弯时预测转弯轨迹，绘制内轮差区域，若检验到大型车辆转弯路径上是否存在障碍物，则对司机和行人进行报警。

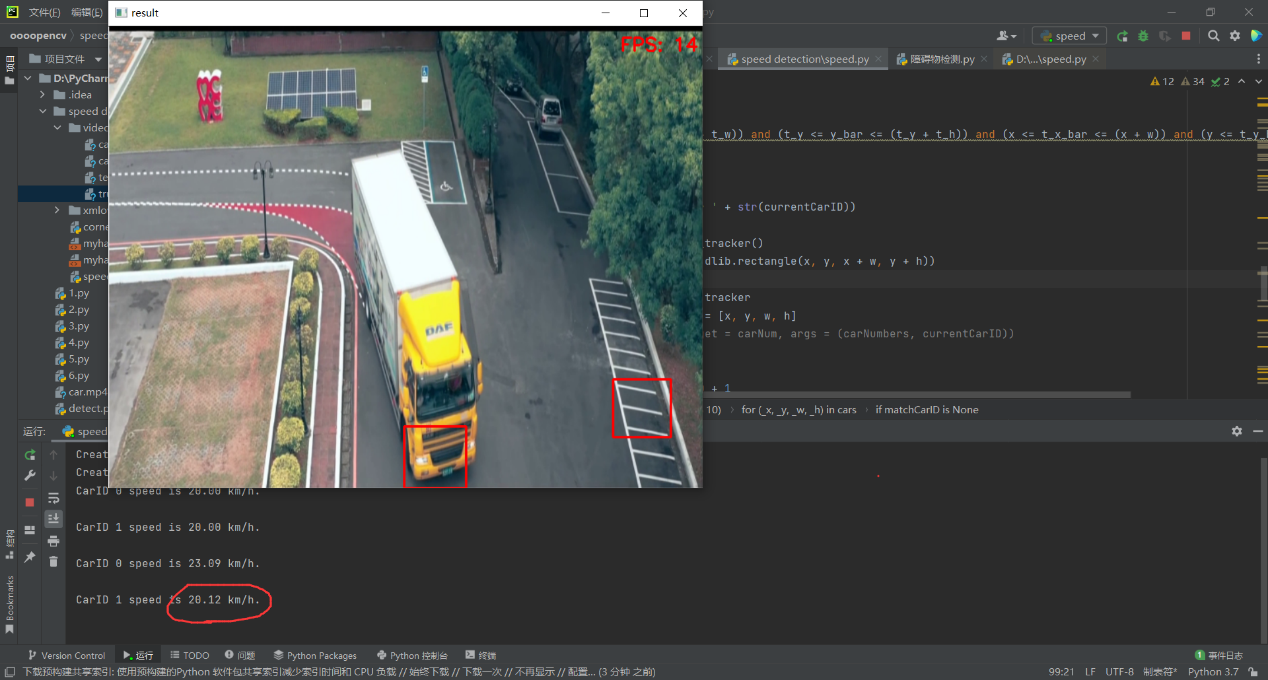
本项目拟为城市交通系统设计一个检测大型车辆转弯报警器，对其理论研究和可行性进行初步探索，为更深入的研究做出铺垫。

### 3，项目研究的主要内容

#### 3.1，大型车辆测速

为了判断大型车辆转弯时与内轮差区域内的行人以及障碍物之间是否会发生碰撞，我们需要计算大型车辆的速度与行人速度之间的关系，由此得出行人是否会在转弯时间内经过内轮差区域而导致发生交通事故。我们基于opencv提供了一种测量视频内车辆速度的方法：运用图像处理技术进行分析，并通过摄像机的标定建立图像像素坐标与空间中对应点的世界坐标的映射关系，获取车辆在一段时间内的实际位移，从而求出车辆速度。

通过一定的方法获取车辆在一段时间内的像素位移，通过摄像机的标定来建立图像像素坐标与空间中对应点的世界坐标的映射关系，从而根据车辆的像素位移求出车辆的实际位移，进而计算车辆速度。



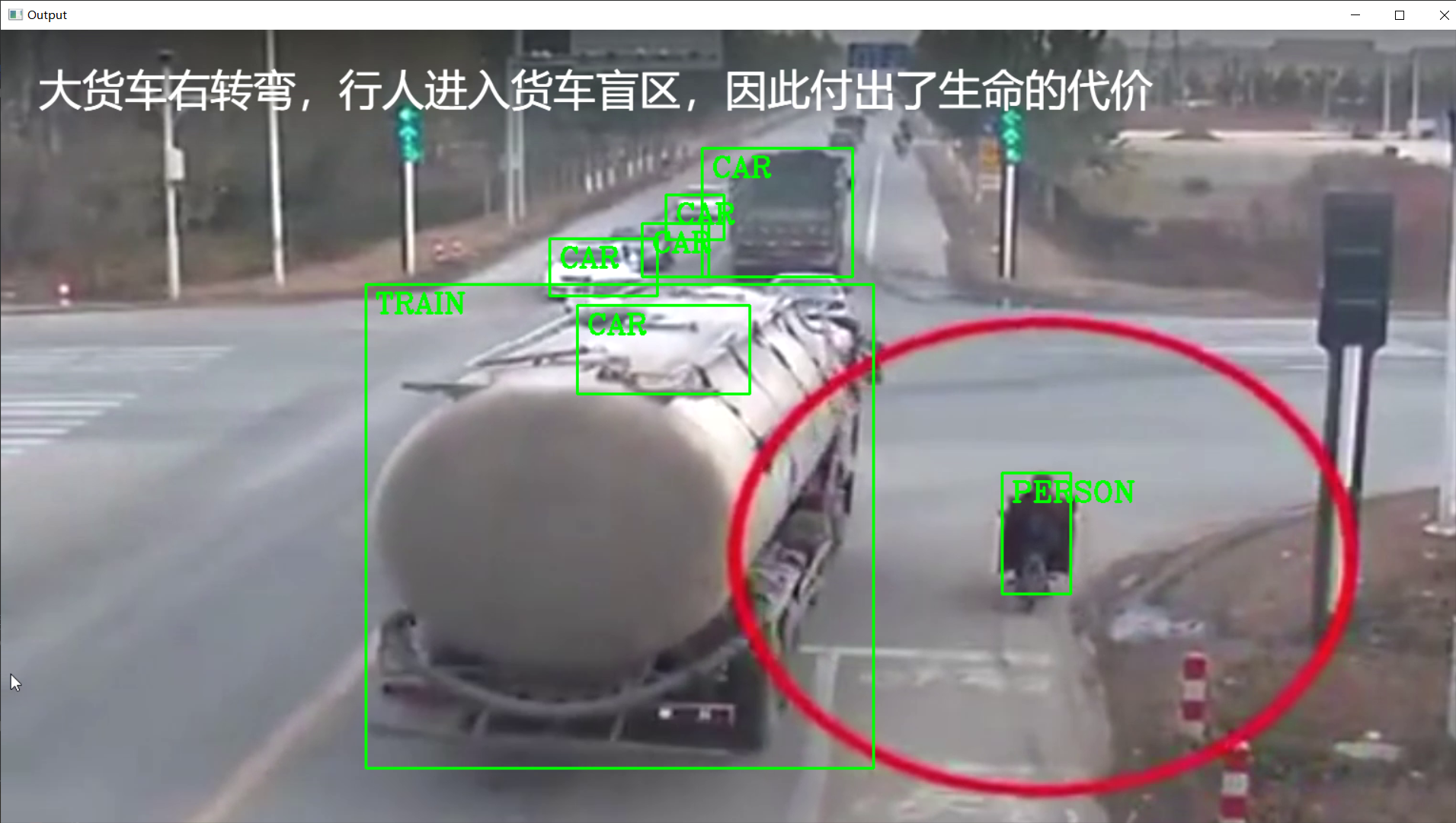
最终实现对车辆的速度进行测定，得到视频内货车的速度为20.12km/h

部分重要代码如下：

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **车辆测速代码** | | **def trackMultipleObjects():  while True:  start\_time = time.time()  # 读取帧  rc, image = video.read()  # frameCounter为10的倍数的时候，执行  if not (frameCounter % 10):  # 将图像转换成灰度图像  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  cars = carCascade.detectMultiScale(gray, 1.1, 13, 0, (24, 24))  # 车辆检测  print('Creating new tracker ' + str(currentCarID))  # 构造追踪器  tracker = dlib.correlation\_tracker()  # 设置追踪器的初始位置  # 如果识别出车辆，会以Rect(x,y,w,h)的形式返回车辆的位置，然后我们可以用一个矩形网格标识车辆  # 获得追踪器的当前位置  trackedPosition = carTracker[carID].get\_position()** | |
| |  | | --- | | **车辆测速代码** | | **def trackMultipleObjects():  while True:  start\_time = time.time()  # 读取帧  rc, image = video.read()  # frameCounter为10的倍数的时候，执行  if not (frameCounter % 10):  # 将图像转换成灰度图像  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  cars = carCascade.detectMultiScale(gray, 1.1, 13, 0, (24, 24))  # 车辆检测  print('Creating new tracker ' + str(currentCarID))  # 构造追踪器  tracker = dlib.correlation\_tracker()  # 设置追踪器的初始位置  # 如果识别出车辆，会以Rect(x,y,w,h)的形式返回车辆的位置，然后我们可以用一个矩形网格标识车辆  # 获得追踪器的当前位置  trackedPosition = carTracker[carID].get\_position()** | |

#### 3.2，障碍物检测

大型车辆转弯轨迹预测模块提取大型车辆转弯初始几帧画面中车轮坐标，利用模型预测轨迹坐标，得到内轮差区域范围，将其输入到障碍物检测模块。通过我们前面所提到的目标识别技术，识别出画面中的行人和障碍物。



障碍物检测模块读取坐标点，利用opencv的追踪器模块，跟踪行人和障碍物运动轨迹，得到行人的坐标轨迹，再利用LSTM多维时间序列预测出行人的未来行驶轨迹（图中红色线条），并提取出内轮差区域的坐标范围，判断与内轮差区域的预测轨迹是否重合，如果重合则认为存在障碍物，触发报警模块。



|  |
| --- |
| **障碍物检测报警代码** |
| **import cv2**  **# 选择一个视频并读取第一帧**  **cap = cv2.VideoCapture('行人.mp4')**  **ret, frame = cap.read()**  **# 选择一个物体作为追踪目标，并初始化追踪器**  **bbox = cv2.selectROI(frame, False)**  **tracker = cv2.TrackerCSRT\_create()**  **tracker.init(frame, bbox)**  **# 保存目标物体的坐标**  **points = []**  **# 遍历视频的每一帧**  **while cap.isOpened():**  **ret, frame = cap.read()**  **if not ret:**  **break**  **# 更新追踪器的位置**  **success, bbox = tracker.update(frame)**  **# 计算目标物体的坐标并保存**  **if success:**  **x, y, w, h = [int(i) for i in bbox]**  **x\_center = x + w // 2**  **y\_center = y + h // 2**  **points.append((x\_center, y\_center))**  **# 绘制轨迹线**  **for i in range(len(points) - 1):**  **cv2.line(frame, points[i], points[i + 1], (0, 0, 255), 2)**  **# 显示结果**  **cv2.imshow('result', frame)**  **if cv2.waitKey(1) == ord('q'):**  **break**  **cap.release()**  **cv2.destroyAllWindows()** |

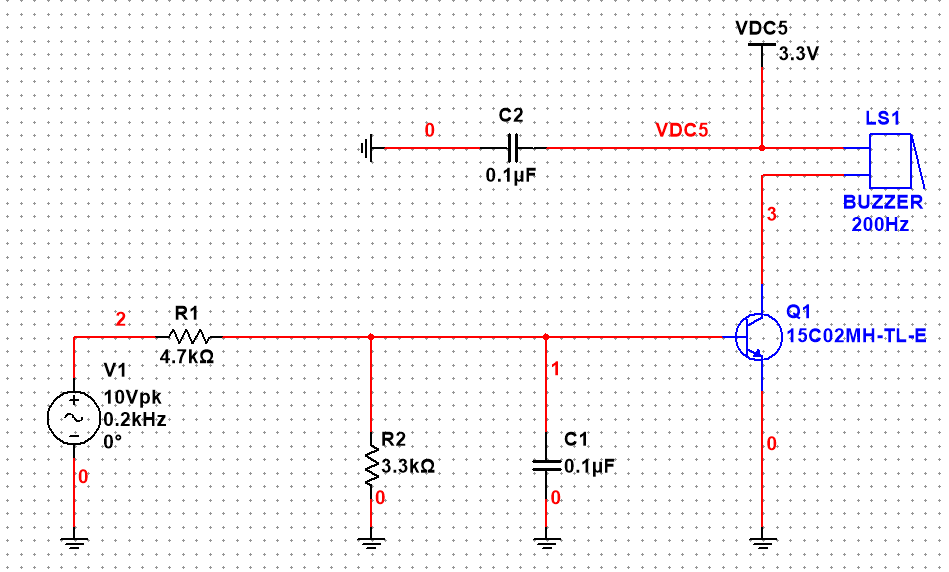
#### 3.3，蜂鸣报警器的设计：

* 报警器需求分析：

基于大型车辆转弯车辆过程中，可能伴随有大量的杂音，以及环境较为嘈杂，我们的蜂鸣报警器应具有警示性高，声音响亮等特点，才可以使处于危险地带的人与司机收到警报。

基于这些需求，我们自行设计了一款蜂鸣报警器。

* 二、报警器的设计：



图表 1蜂鸣器的设计示意图

上图为通用有源蜂鸣器的驱动电路。电阻R1为限流电阻，防止流过基极电流过大损坏三极管。电阻R2有着重要的作用，第一个作用：R2 相当于基极的下拉电阻。如果A端被悬空则由于R2的存在能够使三极管保持在可靠的关断状态，如果删除R2则当BUZZER输入端悬空时则易受到干扰而可能导致三极管状态发生意外翻转或进入不期望的放大状态，造成蜂鸣器意外发声。第二个作用：R2可提升高电平的门槛电压。如果删除R2，则三极管的高电平门槛电压就只有0.7V，即A端输入电压只要超过0.7V 就有可能导通，添加R2的情况就不同了，当从A端输入电压达到约2.2V 时三极管才会饱和导通，具体计算过程如下：

假定β =120为晶体管参数的最小值，蜂鸣器导通电流是15mA。那么集电极电流IC=15mA。则三极管刚刚达到饱和导通时的基极电流是 IB=15mA/120=0.125mA。流经R2的电流是0.7V/3.3kΩ=0.212mA，流经R1的电流 IR1=0.212mA +0.125mA=0.337 mA。最后算出BUZZER端的门槛电压是0.7V+0.337mA× 4.7kΩ=2.2839V≈2.3V。

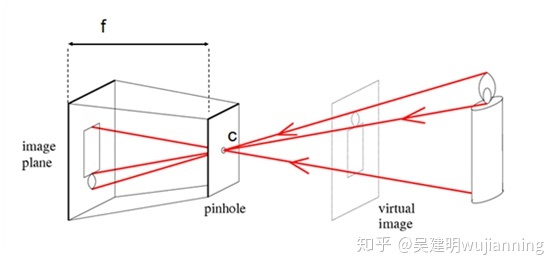
图中的C2为电源滤波电容，滤除电源高频杂波。C1可以在有强干扰环境下，有效的滤除干扰信号，避免蜂鸣器变音和意外发声，在 RFID射频通讯、Mifare卡的应用时，这里初步选用0.1uF 的电容，具体可以根据实际情况选择。

### 4，项目研究过程与方法

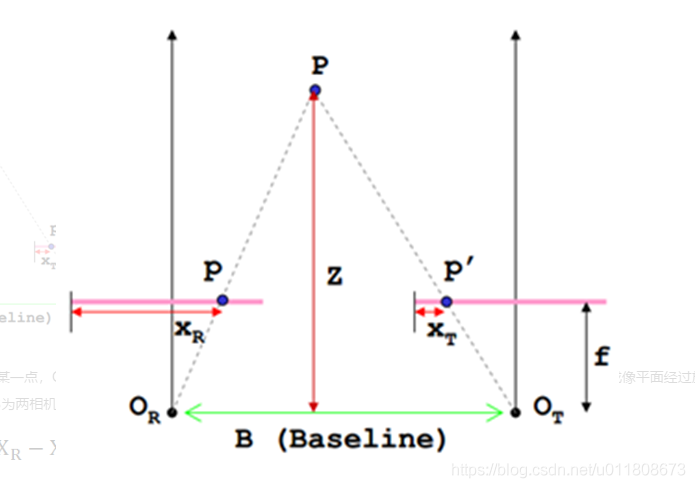
整体上看，项目首先采用建立数学模型的方法拟定关于计算内轮差的范围的方法，通过算法测量车身长度，即为前轮和后轮的轴距车身的宽度，以及最小转弯半径分别为四个轮子的转弯半径，最终得到内轮差范围计算公式。

我们通过基于SSD网络的目标识别，成功实现对画面中物体和行人的识别，并将其运用在后续的轨迹预测和障碍物检测模块中。

对比单目与双目摄像头的优缺点，以及结合本次项目的经费情况，我们最终选择使用单目摄像头来完成项目实验。单目的优势在于成本较低，对计算资源的要求不高，系统结构相对简单。其缺点在于必须不断更新和维护一个庞大的样本数据库，才能保证系统达到较高的识别率；无法对非标准障碍物进行判断；距离并非真正意义上的测量，准确度较低。单目摄像头的大致测距原理，是先通过图像匹配进行目标识别（各种车型、行人、物体等），再通过目标在图像中的大小去估算目标距离。



双目测距的优点一是激光雷达等方案相比成本较低；二是没有识别率的限制，因为从原理上无需先进行识别再进行测算，而是对所有障碍物直接进行测量；三是精度比单目高，直接利用视差计算距离；四是无需维护样本数据库，因为对于双目没有样本的概念。原理双目检测的方式就是通过对两幅图像视差的计算，直接对前方景物（图像所拍摄到的范围）进行距离测量，而无需判断前方出现的是什么类型的障碍物。



除了这些原理学习之外，我们还对摄像头市场进行了一个调研，并且询问了许多已经有过使用摄像头经验的前辈，发现双目测距的摄像头一般价格在2k-8k不等，而单目测距多在1k以内。

鉴于此次设备的使用对象为大货车与行人，均为比较好识别的物体，又考虑到双目摄像头的价格，所以先采用单目摄像头进行理论模型的核验。

在测量车辆速度部分，通过一定的方法获取车辆在一段时间内的像素位移，通过摄像机的标定来建立图像像素坐标与空间中对应点的世界坐标的映射关系，从而根据车辆的像素位移求出车辆的实际位移，进而计算车辆速度。

在障碍物检测方面，我们选择一种巧妙的方法来达到识别障碍物的目的，经团队商讨，我们已经得到预测出的内轮差范围，利用轨迹在图像中的坐标范围，结合行人速度和车辆速度，判断行人的行动轨迹是否会在转弯时间内与内轮差轨迹范围重合，若重合我们将启动报警模块。

内轮差轨迹预测部分：

在实验过程中，我们需要选择一个视频作为样本，我们选择的视频为一个已经在视频中标定理想情况的内轮差轨迹，这样可以作为实验提取数据的一个参考。



图表 1项目所用视频展示

实验过程：

首先，我们先利用opencv创造一个CSRT追踪器，方便后期追踪，再利用手动选择方框对车轮进行标定，由于标定的范围为矩形（有可能为正方形）我们需要提取其中心部分作为我们的坐标点，并将他绘制在canvas画布上

最后，我们设计了一款蜂鸣报警器，检测当内轮差区域有障碍物时，发出报警

### 5，项目创新点

本项目的特色和创新之处在于本预警装置采用在路边安装的方式，区别于传统的将预警器安装在大型车辆车身上，可节约成本，并对各种类型的大型车辆都进行有效识别。也可以同时对行人和司机进行预警，安全系数提升。

### 6，项目研究存在的问题、建议

整体项目开发较为顺利，但过程中仍存在不同方面的一些问题，并将针对这些部分提出一些建议。

首先，团队在起步阶段由于均为大一新生，在对于计算机视觉这部分以及python语法处理上表现的并不是很好，但在团队成员积极的学习态度下，以及网络学习资源的普及下，问题基本上都可以解决，所以对此的建议是可以在计算机视觉这边参阅一些专业书籍。

其次，鉴于许多精度较高的单目摄像头价格都比较高，团队的科研经费难以负担，所以我们采用的分析办法都是通过现有视频进行轨迹追踪与预测，虽然说理论上模型从预测到发出警报的时间是比较短的，但是还是需要实地进行验证。

最后，对于轨迹的预测，如果仅仅用前面的数据预测后面的轨迹，可能在轨迹曲线的最后部分贴合程度并不是很好，其实可以考虑在车子拐弯过程再提取中间10-20组数据对曲线进行再次修正，从而达到更高的拟合度。

### 7，项目完成预期目标情况

1），预期完成目标

* 完成整体方案论证和详细设计
* 完成大货车数学模型建立和OpenCV的双目视觉系统目标定位与测距方法
* 完成利用计算机技术预测内轮差区域及轨迹的方法学习
* 完成基于Faster RCNN和YOLO特征的目标识别与声音示警模块
* 撰写技术报告、设计文档、撰写论文；项目总结、验收

2），结题时已完成目标

* 完成整体方案论证和详细设计
* 完成大货车数学模型建立和OpenCV的单目视觉系统目标定位与测距方法
* 完成利用计算机技术预测内轮差区域及轨迹的方法学习
* 完成声音示警模块设计
* 撰写技术报告、设计文档、撰写论文；项目总结、验收

### 8，项目研究工作的分析与总结

经过一年多的研发过程，我们成功开发出大货车转弯报警器的基本模型，该模型主要由以下五个部分组成：识别货车-读出轨迹-预测轨迹-识别障碍物-报警。

在识别货车部分，我们使用RNN网络，对目标画面内的一切进行特征识别，从而寻找画面中出现目标对象——货车，并将此信号传递给下个模块。

当货车出现在画面中，我们通过特征提取的办法提取出货车靠近摄像头这端的前后轮，并利用opencv中的CRST追踪器对其进行追踪，提取目标框内的中心作为追踪点，从而实现货车转弯过程中坐标的样本提取。

我们通过提取到的样本，建立最小二乘法轨迹分析模型，对货车轨迹进行预测，若预测曲线与实际情况差距较小，则预测情况可以近似作为实际的转弯情况，在通过不同的尝试下，我们选择了三次项拟合作为轨迹预测的模型，同时发现在曲线的后半部分拟合效果并不是特别理想，但是基本上实现了双轮轨迹的预测。

最后，我们将两次预测的轨迹通过画布叠加于视频画面上，再通过追踪器提取障碍物坐标，并判断其是否处于我们预测的两条曲线内，若是，则输出高电平给蜂鸣器，蜂鸣器报警；若不是，则输出低电平给蜂鸣器，蜂鸣器不工作。

以上就是我们对本项目的分析，在整个开发过程，还是存在很多知识欠缺，相关知识储备不足等问题，这需要我们在日后的学习生活中重视实践，这样才可以真正在科研训练中，作出自己的贡献