课程项目设计报告

# 研究背景和项目目标

## 选题依据

由于多目标跟踪任务的景像繁杂性，它的建模困难得多。单目标跟踪是对一个目标连续数据的滤波。多目标除了需要实现邻接两帧之间相同目标多个数据的联系之外，同时也存在着频繁遮挡、目标偏小、目标相似性明显、运动轨迹起止时间未知等问题。在设计多目标跟踪算法时主要从以下方面进行考虑:一是如何判断同一帧图像内不同跟踪对象的相似度,涉及到包括外观特征、运动特征等特征提取的建模;二是如何识别不同帧之间的相同目标，涉及到数据关联和目标检测。

视觉跟踪是目前CV领域中一个重要的数字图像处理方向，它可以为高级视觉任务提供数据保障和应用基础。作为-种先进的视觉技术，跟踪技术 是交通分析、场景理解和行人估计等领域进一步 展开研究的重要基础技术,同时在机器人导航、智能视频监控和航天航空等诸多领域已经展开应用部署。新兴的人机交互智慧生活、交通监控智慧城市和远程医疗智能产业都离不开目标跟踪的技术运用。作为计算机视觉科学的经典任务，跟踪同时也是其他视觉技术的算法补充，可以很好的应用在其他视觉任务中，与其他任务相互促进，例如可以提供在视频或语义连续的相关图像中目标的空间位置，以降低整个系统的复杂性。一旦跟踪算法的性能取得较大的突破，其在视频分析等方面的应用价值将呈井喷式突显出来。在技术日益成熟、企业商业化应用能力不断增强的情况下，未来跟踪在视觉市场中的应用规模将实现突破性的进展。

## 业界现状介绍

视觉目标跟踪的主要任务是在一组图像序列中寻找目标。与计算机视觉中的其他视觉任务相同，目标跟踪同样需要用相机摄像头来取代人眼进行目标搜索和观察，获取目标及其背景的数字图像信息，然后通过扮演人脑角色 的人工智能来解析和处理数码信息。跟踪并不局限于跟踪视频中某一物体,除了对连续视频图像序列中目标运动信息进行预测外，还包括上下文建模、时空信息等研究内容，以实现对目标运动状态的解析和了解，提供针对物体语义内容上进行分析理解的数据保障，从而为完成更高级的视觉任务提供数据支持。  
 起初跟踪聚焦于单一运动目标的视觉任务需求，即单目标跟踪。主流算法主要围绕相关滤波和孪生网络展开研究。最小输出误差平方和滤波器 (Minimum Output Sum of Squared Error Filter, MOSSE)将信号处理领域的相关滤波技术与跟踪领域问题结合起来,实现了超高速的跟踪性能。核函数 逐点循环跟踪(Circulant Structure with Kermnels, CSK)和核相关滤波器 (Kernelized Correlation Filters, KCF)都是基于MOSSE改进的算法。CSK提出结合核方法解决跟踪问题，KCF提出循环矩阵提高跟踪速度。基于相关滤波的算法相较于同期其他跟踪算法具有更高的鲁棒性,不仅可以实现超高的算法效率，而且具有良好的精度,但是在复杂场景中进行跟踪时，漂移现象时有发生，极大地制约了相关滤波跟踪的性能。尽管有各种改进的方法如 大边距循环跟踪(Large Margin with Circulant Feature, LMCF)、连续卷积算子 跟踪(Continuous Convolution Operators for Tracking, C-COT)等提高了精度,但是运行速度小于每秒1帧，以速度降低为代价提升了精确度,体现不出相关滤波的高速优势。

## 本项目的目标

### 基于YOLOv5和DeepSort实现视频/摄像头内的物体识别与跟踪

### 设计GUI界面，实时展示视频/摄像头中物体识别与跟踪的画面

# 项目总体设计

## pytorch部分

通过yolov5和deepsort技术对物体进行识别与跟踪。其中pytorch为整个环境的基础，对yolov5模型进行训练并得到准确度较高的模型。预期可以达到0.8以上的准确度，且能实现实时跟踪，避免出现异常情况。

## GUI部分

运用GUI技术，对项目的可视化提出更加高的要求。期望能达到具有更突出的特点，能够实现用户与程序的交互，制作成exe软件供运行。同时能够实现视频播放的暂停，快进，快慢等功能，做到一个功能齐全的程序

# 项目关键技术

## YOLOv5

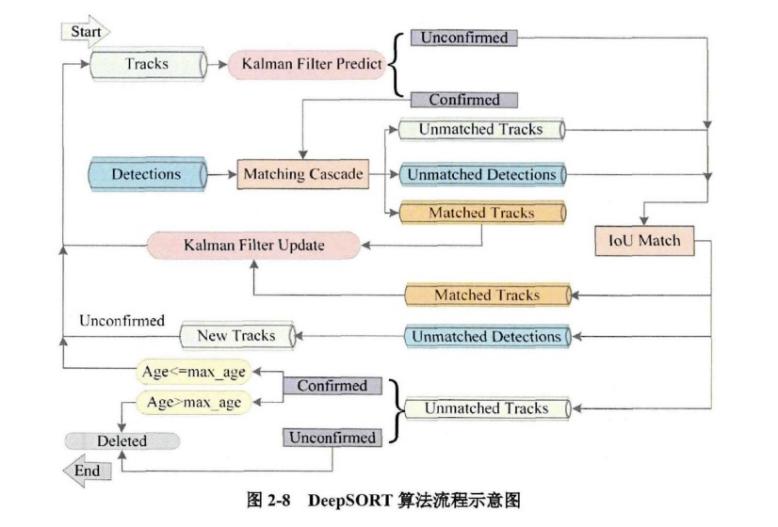
在图片尺寸处理的方式上，YOLO中常用的标准输入尺寸为416X416 和608X608。 在对输入图片尺寸进行裁剪统一时， YOLOv5 的训练阶段使用传统的填充方式，即将图片统一缩放到标准的尺寸，但在测试阶段使用缩减填充的方式进行填充以提高速度，该缩减填充的方法是基于减少冗余信息影响的目的提出的。YOLOv5在测试阶段自适应的补充最少的冗余信息，自适应的方法为:首先根据缩放尺寸计算原始图像的缩放比例，对于不同尺寸的原始图像会得到多种缩放系数,选择其中最小的缩放系数;其次将原始图片的长宽都乘以缩放系数，再将其填充至标准的缩放尺寸。

在网络结构上，YOLOv5 有四种量级的网络结构，分别是YOLOv5-s、 YOLOv5-m、YOLOv5-l和YOLOv5-x. YOLOvS-s是四种结构中网络深度最浅且训练速度最快的网络,其他结构的网络层数依次增加。YOLOv5的结构借鉴了CSPNetl47的设计思路，在主干网络中设计了CSP结构，并且在网络的不同部位使用不同结构的CSP,从而降低计算和内存成本。

## Deepsort

SORT根据检测结果来完成识别多个目标的任务，用检测的识别机制帮助预测轨迹的跟踪器实现相邻图像中目标的联系和区分功能。由于SORT基于检测算法进行多目标的识别，所以SORT的跟踪效果与目标检测的结果有关，不同的检测器可以达到不同的跟踪效果。在SORT算法中，用卡尔曼滤波(KalmanFilter){30]实现跟踪器对运动轨迹的预测功能，用匈牙利算法(HungarianAlgorithm)川实现检测结果和跟踪器预测结果的最优化分配功 能。

主要流程:  
(1)输入视频中某相邻两帧的画像;  
(2)利用检测器对图像进行目标检测，生成检测结果;  
(3)在检测到的对象中，裁剪提炼出所有目标的局部画像，并运行特征提取器提取包括外观特征或运动特征在内的特征向量;  
(4)求取目标特征之间的相似性，通过衡量相似度得到前后两帧目标之间的特征匹配程度;  
(5)进行数据关联，为下一帧中匹配度高的目标分配上一帧相应的目标的身份识别码ID。



### 卡尔曼滤波

卡尔曼滤波作为一种状态最优估计的方法，对于需进行预判的自动化控制系统起着不可取代的作用，一方面是因为卡尔曼滤波对含有噪声的系统参数和不准确的观测值有一定的容错能力，另一方面是因为卡尔曼滤波预测步骤的全面性能尽可能的实现对动态系统状态值的最优估计。跟踪基于目标当前运动轨迹预测未来状态，所以卡尔曼滤波是研究跟踪算法必不可少的组成部分。

### 匈牙利算法

为了解决SORT中的检测和跟踪结果的分配问题，使用匈牙利算法在检测器的检测结果和跟踪器的跟踪轨迹两者之间找到-一个代价最小的最优分配。SORT采用带权重的匈牙利算法逐帧进行跟踪目标之间的关联，用对象 重叠度(Intersection over Union, loU}距 离作为匈牙利算法的权重，同时通过给IoU设置-一个阈值来界定错误的匹配，屏蔽无效的关联结果。通过IoU阙值的设置还对短时遮蔽有一定的容忍性,但只能建立在障碍物比目标略大的情况下。当目标短时间被遮挡时，检测器容易发生将目标附近的障碍物错误识别为目标的情况。由于IoU在大小相似距离相近的物体中计算结果较大,所以通常不会得到一个低于IoU阈值的结果，这导致被遮挡目标的跟踪轨迹匹配中断，并与短时间内遮挡物体的障碍物进行匹配，因此当遮挡结束时真实目标可以根据障碍物的轨迹快速恢复正确的关联。SORT工作时，跟踪器首先通过卡尔曼滤波估计每个目标在下一副图像中的定位,然后根据每个检测器在下一幅图像中的识别结果计算与跟踪器的预测结果的IoU,将IoU作为匈牙利算法的代价矩阵，使用匈牙利算法优化分配各个目标的轨迹，当检测框与预测框重叠度小于IoU阈值时，拒绝将两者进行匹配。

# 项目实现

## 配环境

首先，我们对yolov5所需要的运行环境进行了安装，遇到一些问题如pytorch不同版本与不同电脑显卡配置的对应关系，tensorflow安装报错，pycocotools需要安装c语言程序等等。这些环境的配置问题我们在youtube和csdn上进行了大量的搜索，这也是对于新手入门来说最困难的一个环节，我们花费了大量的时间在代码环境的配置上，这些都是基于python的基础知识而产生的衍生知识，我们克服这些困难后之后的代码运行就能更加得心应手，得到更多收获。

## yolo模型训练

代码运行的第一步是选择一个yolo模型进行训练。Yolo模型根据复杂程度分为n，s，m，l，x等级别。为了加快模型运行速度，同时保证模型的训练效果，我们在网上搜集资料后发现，s模型的神经网络结构比较简单，可以适配于代码测试，于是我们选择s模型进行训练。我们将图片分为训练集和测试集，通过代码自动从网上下载图片。我们对运行结果的多个指标进行分析后得出结论，该模型的可信程度较高，对物体的检测达到平均0.8以上，但是对于重叠的物体，以及重复出现的物体，会有识别误差。

## 检测以及跟踪

这一部分与pytorch的功能息息相关。PyTorch是一个基于Torch库的开源机器学习框架，用于计算机视觉和自然语言处理等应用程序。Pytorch提供了两个高级功能：张量计算（如 NumPy），通过图形处理单元 （GPU） 实现强加速；基于自动微分系统构建的深度神经网络。在SORT算法中，用卡尔曼滤波(KalmanFilter)实现跟踪器对运动轨迹的预测功能，用匈牙利算法(HungarianAlgorithm)实现检测结果和跟踪器预测结果的最优化分配功能。

## GUI配置

Tkinter 是一个绑定到 Tk GUI 工具包的 Python文件。它是 Tk GUI 工具包的标准 Python 接口， 并且是 Python 的事实上的标准 GUI。我们利用tkintervideo，将视频在GUI窗口进行播放，以得到最佳的用户体验。同时我们设计了按钮的颜色，位置，将项目特色充分融入到代码当中，人性化的窗口可缩放设计，以及进度条的背景设置为了华科的南大门，充分体现了GUI界面的独特性与创意性，反映出我们对于代码的理解和学习。当然我们的GUI配置也有不足之处，其内部逻辑比较复杂，我们只是作为改参工程师，对代码的部分参数，超参数进行了修改，内部接口问题仍为完全解决，这也是我们要去改进的方向。

## 模块的连接

在不同模块的连接方面，我们首先采用了fileopen函数，以供用户手动选择视频，便于对用户自定义的素材进行分析跟踪。其次我们对代码进行了优化，采用了函数的调用方式，将打开的视频进行播放，并且可以随时暂停。当我们打开视频时，我们的程序已经开始对视频进行分析了，对不同对象进行识别跟踪，将跟踪解析之后的结果放到相应文件夹目录下面。

# 项目测试

## 模型准确性的实现

对于模型的准确性，我们在项目后期对yolov5的复杂度进行了变更，进行多次测量后我们可以直观得出随着模型复杂度的提升，准确度也在随之上升。

## 跟踪检测实现

我们从bilibili官方网站上下载了一段视频，在代码运行的过程中，我们可以很清楚的看到，虽然画质在一定程度上有降低，但是程序的主要功能还是达到了，能够对不同类型的对象进行检测和跟踪，同时还能显示每个对象的可信程度（可关闭），这些措施无疑印证了我们之前的设想，运用yolov5和deepsort进行物体识别和跟踪是可行的。

## 界面交互的实现

界面交互实现程度较高，我们将python程序，通过pyinstaller打包成一个exe文件，让用户可以随心所欲的运行改程序，同时对于界面的美观程度也在原先基础上得到较大的改善。界面总共有4个按键，一个进度条，多个显示框，我们对每个元件都进行了自定义的设置，达到了我们预期的目标。

# 项目管理

## 团队人员组成

电气2009班黄屹坚、电气2009班王树森、电气2010班吴裕亮

## 任务分工

黄屹坚 软件环境配置，负责写pytorch代码和GUI代码设计，报告编写

吴裕亮 解决pytorch代码等报错以及各类调参设计、ppt制作、报告编写

王树森 参与GUI代码设计、检查代码错漏、资料搜集整理、报告编写

# 总结与反思

一开始确定大作业题目后，三个人对此非常不熟悉，从软件下载、环境配置、各类报错查询解决、熟悉使用GitHub和CSDN平台到学习理解并运用代码，遇到了许多挫折并付出了努力，每一次被代码卡住的时候大家齐心协力去思考解决，往往最后都有一种柳暗花明又一村的感觉。

本项目提出的多目标跟踪算法在算法改进和算法应用方面都取得了一定成果，针对多目标跟踪任务的开展有更多的开发意义和研究价值。尽管目前的多目标跟踪算法已经有了不错的跟踪性能，但还能在模型体积上进一步优化，模型速度上进一步提升。此外，多目标跟踪的研究效果一定程度上取决于检测器的检测效果,检测算法的进一步优化也是一个可以改进的方向。本课题所用的算法基于一阶段和二阶段的两主流检测算法进行实验，如果能进一步改进检测器提升检测效果想必会有更好的跟踪性能。  
 目前的多目标跟踪效果还可以进一步提升精度、速度等性能指标，需要进一步加强检测器的研究和跟踪策略的研究。