# 课程项目设计报告

## 1 研究背景和项目目标

### 1.1选题依据

现如今随着计算机技术的高速发展，运动物体的检测和跟踪在图像处理、计算机视觉、模式识别、人工智能、多媒体技术等领域越来越受到人们的关注。运动跟踪和检测的应用广泛，在智能监控和人机交互中，如：银行、交通、超市等场合常常使用运动跟踪分析技术，通过定位物体并对其行为进行分析，一具发现物体有异常行为，监控系统就发出警报，提醒人们注意并即时的处理，改善了人工监督注意力分散、反应时间较慢、人力资源浪费等问题。运动目标的跟踪在虚拟现实、工业控制、军事设备、医学研究、视频监控、交通流量观测监控等很多领域也有重要的实用价值。尤其是在新冠病毒肆虐的今天，各地都能看见的体温监控装置便充分运用了该项技术。但由于物体比例、遮掩物、行为等方面都会影响检测的准确性，因此研究物体检测变成计算机视觉领域和人工智能方面极富挑战的热门课题。

### 1.2业界现状介绍

目前物体检测已广泛应用于汽车自动驾驶、视频图像检索、智能视频监控、医学图像分析、工业检测等领域。传统的人工提取特征检测算法主要包括预处理、窗口滑动、特征提取、特征选择、特征分类和后处理六个步骤，一般基于图像上的行人识别和跟踪。它的缺点主要是数据量小，可移植性差，没有针对性，时间复杂度高，窗口冗余，对多样性没有鲁棒性变化，只包含空间特征，缺少时序信息，在复杂条件下的精度不高，而视频序列均含有空间特征和时序信息，因此物体检测在视频识别的研究中具有重大意义。

### 1.3 本项目的目标

本项目主要实现的目标是(1)基于YOLOv5和DeepSort实现视频或摄像头内的物体识别与跟踪；(2)设计GUI界面，实时展示视频或摄像头中物体识别与跟踪的画面。

## 2 项目总体设计

该项目总的来说较为复杂，想要实现，需要发布进行，大致分为以下几步：

### 2.1前期调研与准备

通过前期的调研，大致了解目前物体识别与跟踪的发展水平，明确实现该项目所需的电脑配置、时间成本以及其他人在完成类似项目时遇到的困难，做好前期准备工作。

### 2.2项目实际分解

2.2.1实现物体检测功能

该部分主要运用YOLOv5进行实现。YOLOv5是一个单阶段探测器和基于区域的目标检测网络模型，由三个主要组件构成：BackBone、Neck和Head，具有极好的灵活性和速度，在模型的快速部署上具有极强优势。

2.2.2实现对物体的跟踪

该部分通过Deepsort多目标跟踪算法来进行实现。Deepsort算法可对检测到的物体进行轨迹跟踪，并为其分配特定的ID号，具有深度关联的特征。

2.2.3GUI界面的设计

在物体检测与跟踪部分结束后，通过设计GUI界面，实时展示视频中物体的识别与跟踪画面，以此来简化操作流程，验证项目是否成功实现。

### 2.3项目总结

项目成功实现目标功能之后，对完成项目过程中所遇到的问题进行总结思考，结合一定专业指导，对项目进行一定改进。

## 3 项目关键技术

### 3.1YOLOv5实现物体识别

运用YOLOv5进行物体识别之前，首先需要搭建Pytorch深度学习环境。这一步是整个项目当中最初始，也是最复杂的一步。代码能不能跑起来，能不能实现目标环境非常重要。在搭建环境时，首先需要安装anaconda与pycharm。anaconda是一个管理用于python开发的包含不同库的虚拟环境的平台，可以高效的管理和创建适用于多个不同项目的project interpreter。之后需要安装cuda和cudnn。通过查询计算机配置，选择安装cuda11.3和cudnn8.5.0并添加环境变量。之后通过Anaconda Prompt创建深度学习虚拟环境（命名为yolov550）。在该虚拟环境下下载yolov5-5.0所需库和模型。最终yolov5-5.0所用到的库包含为Pillow-8.3.2、PyYAML-5.4.1、absl-py-0.13.0、attr-0.3.1、attrs-21.2.0、cachetools-4.2.2、charset-normalizer-2.0.4、colorama-0.4.4、coremltools-4.1、cycler-0.10.0、cython-0.29.24、google-auth-1.35.0、google-auth-oauthlib-0.4.6、grpcio-1.39.0、idna-3.2、importlib-metadata-4.8.1、kiwisolver-1.3.2、markdown-3.3.4、matplotlib-3.4.3、mpmath-1.2.1、numpy-1.19.5、oauthlib-3.1.1、onnx-1.10.1、opencv-python-4.5.3.56、packaging-21.0、pandas-1.3.2、protobuf-3.17.3、pyasn1-0.4.8、pyasn1-modules-0.2.8、pycocotools-2.0.2、pyparsing-2.4.7、python-dateutil-2.8.2、pytz-2021.1、requests-2.26.0、requests-oauthlib-1.3.0 rsa-4.7.2、scikit-learn-0.19.2、scipy-1.7.1、seaborn-0.11.2、six-1.16.0、sympy-1.8、tensorboard-2.6.0、tensorboard-data-server-0.6.1、tensorboard-plugin-wit-1.8.0、thop-0.0.31、post2005241907、torch-1.9.0、torchvision-0.10.0、tqdm-4.62.2、typing-extensions-3.10.0.2、urllib3-1.26.6、werkzeug-2.0.1、zipp-3.5.0。

YOLOv5大致分为输入端、主干网络、Neck网络、输出端四部分。其中输入端包含Mosaic数据增强、自适应锚框计算，以及自适应图片缩放。主干网络分为Focus结构和CSP结构。Neck网络首先使用自顶向下的FPN层可以使语义特征顺利传达下去，通过PAN结构可以有效定位特征，使每一个主干层中的检测层完成参数聚合。输出端的损失函数由分类损失函数和回归损失函数组成。

### 3.2Deepsort实现物体跟踪

Deepsort的前身是sort算法，sort算法的核心是卡尔曼滤波算法和匈牙利算法。卡尔曼滤波算法作用：该算法的主要作用就是当前的一系列运动变量去预测下一时刻的运动变量，但是第一次的检测结果用来初始化卡尔曼滤波的运动变量。匈牙利算法的作用：简单来讲就是解决分配问题，就是把一群检测框和卡尔曼预测的框做分配，让卡尔曼预测的框找到和自己最匹配的检测框，达到追踪的效果。Deepsort算法在sort算法基础上增加了级联匹配（Matching Cascade）和新轨迹的确认（confirmed）。Tracks分为确认态（confirmed），和不确认态（unconfirmed），新产生的Tracks是不确认态的；不确认态的Tracks必须要和Detections连续匹配一定的次数（默认是3）才可以转化成确认态。确认态的Tracks必须和Detections连续失配一定次数（默认30次），才会被删除。

Deepsort在YOLOv5的环境下不需要再下载多余的库和函数，可直接运行。通过DeepSort实现的物体跟踪模块完成流程：

（1）将第一帧次检测到的结果创建其对应的Tracks。将卡尔曼滤波的运动变量初始化，通过卡尔曼滤波预测其对应的框框。这时候的Tracks一定是unconfirmed的。

（2）将该帧目标检测的框框和第上一帧通过Tracks预测的框框一一进行IOU匹配，再通过IOU匹配的结果计算其代价矩阵（cost matrix，其计算方式是1-IOU）。

（3）将（2）中得到的所有的代价矩阵作为匈牙利算法的输入，得到线性的匹配的结果，这时候我们得到的结果有三种，第一种是Tracks失配（Unmatched Tracks），我们直接将失配的Tracks（因为这个Tracks是不确定态了，如果是确定态的话则要连续达到一定的次数（默认30次）才可以删除）删除；第二种是Detections失配（Unmatched Detections），我们将这样的Detections初始化为一个新的Tracks（new Tracks）；第三种是检测框和预测的框框配对成功，这说明我们前一帧和后一帧追踪成功，将其对应的Detections通过卡尔曼滤波更新其对应的Tracks变量。

（4）反复循环（2）-（3）步骤，直到出现确认态（confirmed）的Tracks或者视频帧结束。

（5）通过卡尔曼滤波预测其确认态的Tracks和不确认态的Tracks对应的框框。将确认态的Tracks的框框和是Detections进行级联匹配（之前每次只要Tracks匹配上都会保存Detections其的外观特征和运动信息，默认保存前100帧，利用外观特征和运动信息和Detections进行级联匹配,这么做是因为确认态（confirmed）的Tracks和Detections匹配的可能性更大）。

（6）进行级联匹配后有三种可能的结果。第一种，Tracks匹配，这样的Tracks通过卡尔曼滤波更新其对应的Tracks变量。第二第三种是Detections和Tracks失配，这时将之前的不确认状态的Tracks和失配的Tracks一起和Unmatched Detections一一进行IOU匹配，再通过IOU匹配的结果计算其代价矩阵（cost matrix，其计算方式是1-IOU）。

（7）将（6）中得到的所有的代价矩阵作为匈牙利算法的输入，得到线性的匹配的结果，这时候我们得到的结果有三种，第一种是Tracks失配（Unmatched Tracks），我们直接将失配的Tracks（因为这个Tracks是不确定态了，如果是确定态的话则要连续达到一定的次数（默认30次）才可以删除）删除；第二种是Detections失配（Unmatched Detections），我们将这样的Detections初始化为一个新的Tracks（new Tracks）；第三种是检测框和预测的框框配对成功，这说明我们前一帧和后一帧追踪成功，将其对应的Detections通过卡尔曼滤波更新其对应的Tracks变量。

（8）反复循环（5）-（7）步骤，直到视频帧结束。

通过DeepSort算法，在行人特征提取后得到一帧一帧的图像，完成对行人的跟踪。此方法可以有效改善遮挡问题。

### 3.3GUI界面设计

GUI界面设计主要使用PyQt5框架。PyQt5是由一系列Python模块组成。超过620个类，6000种函数和方法。能在诸如Unix、Windows和MacOS等主流操作系统上运行。通过PyQt5设计出的GUI界面比较美观，同时利用Qt Designer可以拖动一些控件，实现GUI可视化的快速开发。

本项目运用的PyQt5主要模块如下：

QtCore：包含了核心的非 GUI 的功能。主要和时间、文件与文件夹、各种数据、流、URLs、mime 类文件、进程与线程一起使用。

QtGui：包含了窗口系统、事件处理、2D 图像、基本绘画、字体和文字类。

QtWidgets：包含了一系列创建桌面应用的UI元素。

QtTest：提供了测试 PyQt5 应用的工具。

## 4 项目实现

### 4.1物体识别模块的实现

YOLOv5拥有YOLOv5n、YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x等多个模型版本，这些模型的参数量依次增大，其相应的检测精度也随之提升。

本项目采用YOLOv5s权重文件，有效降低了采集成本，且识别精度能够得到一定保证。最开始采用已经训练好的coco训练集进行检测，以此实现物体检测模块。

在能实现基本检测功能后，我们决定自己训练训练集。首先在网络寻找自己想要识别的照片，然后利用labelimg标注数据，我们只选择了人这一类图片。打完标签后将标注完的数据集划分为训练集和验证集，这样更加方便模型的训练和测试。选用的训练参数设置为：输入图像尺寸为640像素\*640像素，训练轮数300epoches，共选用75张照片进行训练，其中训练集68张，验证集7张。最终训练精度达到99.6%，然后用生成的best文件可作为后续检测的权重文件。但由于训练所需时间过长，该训练集不得不放弃了增加后续分类的可能。

### 4.2物体跟踪模块的实现

DeepSort算法通过将使用目标跟踪检测的算法（即YOLOv5）计算得到的目标检测框与之前预测的目标跟踪框的iou(称为交并比)进行输入组合。输入到匈牙利这个算法中进行线性分配来直接关联这个帧中的ID,目标的物体外观位置信息可以加入连接到帧间匹配的过程计算中，这样在目标被物体遮挡但后续目标再次出现的实际情况下，还可以正确的来匹配这个帧间ID,在进行实时检测目标追踪过程中，可以改善在有遮挡目标情况下的实时目标自动追踪检测效果。同时，也大大减少了目标id之间跳变频繁的问题，达到持续跟踪的目的。

其代码主要分为三个部分：目标追踪的相关代码和权重；目标检测相关代码和权重，该部分代码可直接使用YOLOv5目标检测算法；以及调用检测和追踪代码相关py文件。加入该模块后可以改善有遮挡情况下的目标追踪效果，减少了目标ID跳变的问题。

### 4.3界面设计的实现

UI界面设计使用pyqt5自带的图形化设计工具designer,设计完成后生成ui文件，并且可将ui文件轻松转化成python代码。在该代码基础上加入yolov5检测代码，即将yolov5中的detect.py中相应的代码移植过来。

## 5 项目测试

本项目能在设计的GUI界面中实时展示图片检测、视频检测和摄像头检测画面等界面。采用coco训练集进行物体检测，最终实现可识别人、自行车、摩托车等八十钟不同类型的物品，训练精度与拍摄物体清晰度有关，当物体离电脑摄像头较近且清晰时，检测精度在百分之八十甚至百分之九十左右，已达到所需要求。

但是在加入Deepsort算法后，与先前使用的YOLOv5版本无法匹配，YOLOv5中utils文件内容不同无法调用，无法在物体检测的同时实现跟踪功能。在重新下载匹配版本后的Deepsort和yolov5，能够成功实现调用摄像头进行物体跟踪和识别。

## 6 项目管理

### 6.1 团队人员组成

本团队共有三位人员。组长：电气2003班赵子阳，学号U202012286;组员：电气2003班张凯轩，学号U202012265，电气2003班俆展鹏，学号U202012271.

### 6.2 任务分工

赵子阳同学负责查找相关文献，收集数据集，YOLOv5环境搭建，编写和修改Deepsort部分代码，完成总结报告等任务。该同学担任该项目小组的组长，能够认真负责的对待本次项目，能够对项目进行分工并且实时关心项目进展。在编写代码的同时，能够完成其他如查找文献、编写报告等琐事，对整个团队至关重要。

张凯轩同学负责YOLOv5整体代码的编写，对数据集进行训练，编写和修改Deepsort部分代码，整合GUI部分代码，对整个项目进行调试修改，是整个团队中的攻坚手，承担了项目中绝大部分的困难任务。该同学热衷于编写代码，对该项目十分了解，是整个团队不可或缺的一部分。

俆展鹏同学主要负责GUI界面的设计，查找YOLOv5和Deepsort相关资料，同时负责团队的答辩工作。该同学需要对项目进行收尾工作和一些细节上的调整，需要跟进整个项目，对整个团队贡献突出。

## 7. 总结与反思

本次项目对我们来说是一次全新的尝试，通过结合课堂讲授的关于python和人工智能的基础知识、卷积神经网络的学习等内容，我们从最开始的不清楚YOLOv5、Deepsort以及GUI到底是什么，到现在成功基本实现预期目标，可以说是一次质的提升，对上课所学内容有了更深刻的认识。在项目推进过程中，我们遇到了很多难题，比如YOLOv5所需环境始终搭建不成功、YOLOv5训练数据集十分缓慢、GUI运行闪退等等。我们小组成员通过查找资料、询问学长，一点点克服了这些问题，实现了预期功能，积累了十分丰富的工程经验。从满屏报错到完美实现物体检测与跟踪，我们能够做到吸取他人成功的经验，不断完善自身，这是我们做的比较满意的地方。但是，由于工程经验不足，我们最终项目完成的并不完美，而且该项目所花费的时间有些超出预期，没有一个固定的工作时间点，并且项目还有些粗糙，功能不够完善，这些是我们未来需要改进的地方。

该项目可以说是我们小组做的第一个有关Python的具有一定功能的项目，我们会继续完善该项目，在原有功能的基础上继续加以改进，让其在识别与跟踪目标的同时，能够实现按用户想法选择所要跟踪的目标，能够对部分画面进行放大等等功能，让该项目趋于完美。