

招商证券人工智能工程营课程报告

**项目名称：基于YOLO和DeepSort的人体识别与跟踪**

|  |  |
| --- | --- |
| 院系： | 电气与电子工程学院 |
| 小组成员： | 班婷婧 U202112342 |
|  | 陈颖林 U202112438 |
|  | 冯思媛 U202112255 |
|  | 张严匀 U202112430 |
|  |  |
| 指导教师： | 郑 玮 |
| 2023年9月16日 | |

目录

[1 研究背景和项目目标 1](#_Toc144545993)

[1.1选题依据 1](#_Toc144545994)

[1.1.1为什么选择该题目 1](#_Toc144545995)

[1.1.2该题目需要解决的问题 1](#_Toc144545996)

[1.1.3 该题目的意义 1](#_Toc144545997)

[1.2业界现状介绍 1](#_Toc144545998)

[1.2.1选题所要解决的问题 1](#_Toc144545999)

[1.2.2业界现有的解决方案 1](#_Toc144546000)

[1.2.3业界现有解决方案的不足 1](#_Toc144546001)

[1.3 本项目的目标 1](#_Toc144546002)

[2 项目总体设计 2](#_Toc144546003)

[2.1问题分解： 2](#_Toc144546004)

[2.2解决方案模块： 2](#_Toc144546005)

[2.2.1物体检测模块 2](#_Toc144546006)

[2.2.2图形用户界面模块： 2](#_Toc144546007)

[2.2.3图像处理模块： 2](#_Toc144546008)

[2.2.4线程管理模块： 2](#_Toc144546009)

[2.3整体组织： 2](#_Toc144546010)

[2.4设计复杂性和实施成本注意事项： 3](#_Toc144546011)

[3 项目关键技术 3](#_Toc144546012)

[4 项目实现 4](#_Toc144546013)

[4.1实现过程 4](#_Toc144546014)

[4.2不同模块是怎么实现的 4](#_Toc144546015)

[4.2.1视频捕获模块 4](#_Toc144546016)

[4.2.2YOLO 目标检测模块 4](#_Toc144546017)

[4.2.3深度排序跟踪模块：DeepSORT用于跟踪检测到的人体随时间的变化,关键步骤包括： 4](#_Toc144546018)

[4.2.4可视化模块 5](#_Toc144546019)

[4.2.5主循环 5](#_Toc144546020)

[4.2.6用户界面模块 5](#_Toc144546021)

[4.3最终实现的功能 5](#_Toc144546022)

[5 项目测试 5](#_Toc144546023)

[6 项目管理 6](#_Toc144546024)

[6.1 团队人员组成 6](#_Toc144546025)

[6.2 任务分工 6](#_Toc144546026)

[7. 总结与反思 6](#_Toc144546027)

# 1 研究背景和项目目标

## 1.1选题依据

### 1.1.1为什么选择该题目

该题目运用机器深度学习的方法训练模型实现人机交互，能够通过该题目熟悉机器学习的应用和提升通过GitHub等途径的自主学习。

### 1.1.2该题目需要解决的问题

训练一个良好的YOLO模型

通过DeepSORT实现目标跟踪

调用电脑摄像头

实现人机交互

### 1.1.3 该题目的意义

计算机视觉分析特定目标、了解实现人机交互、掌握机器学习的应用

## 1.2业界现状介绍

### 1.2.1选题所要解决的问题

实现对摄像头中出现的点击目标的跟踪

### 1.2.2业界现有的解决方案

YOLOv3检测+DeepSORT算法

### 1.2.3业界现有解决方案的不足

仅实现对摄像头全画面的追踪，无法实现对点击目标的特定跟踪

## 1.3 本项目的目标

在GUI界面显示通过opencv和YOLO实现的人体识别，并实现对点击目标的跟踪。

# 2 项目总体设计

## 2.1问题分解：

主要问题是对象检测，特别是使用 YOLO 模型检测对象。

子问题包括加载 YOLO 模型及其配置、读取视频帧、处理用于对象检测的帧、显示带有边界框和标签的帧、处理用户交互以及管理 GUI 界面。

## 2.2解决方案模块：

### 2.2.1物体检测模块

加载 YOLO 模型和配置。

使用 YOLO 模型处理用于对象检测的视频帧。

应用非最大抑制以删除重叠的边界框。

提取检测到的对象的类 ID、置信度和边界框坐标。

### 2.2.2图形用户界面模块：

使用 tkinter 创建 GUI 界面。

设置用于显示视频帧的画布。

处理用户交互，例如用于选择边界框的鼠标单击。

使用已处理的框架、边界框和标签更新画布。

提供用于启动、停止和取消对象检测的按钮。

### 2.2.3图像处理模块：

将视频帧转换为适当的格式，以便输入到 YOLO 模型。

将处理后的帧转换为适合在画布上显示的格式。

### 2.2.4线程管理模块：

创建和管理多个线程，以便并行执行对象检测和 GUI 更新。

使用队列进行线程间通信，存储已处理的帧和检测结果。

## 2.3整体组织：

对象检测模块处理加载 YOLO 模型、处理帧和检测对象。

GUI 模块管理 GUI 界面、用户交互和画布更新。

图像处理模块处理帧格式转换。

线程管理模块使用队列处理线程创建和通信。

线程用于启用对象检测和 GUI 更新的并行执行。

主程序协调模块，初始化 GUI，并启动线程。

## 2.4设计复杂性和实施成本注意事项：

通过将问题分解为更小的模块来管理设计复杂性，每个模块负责一项特定的任务。

使用现有的库和工具，如OpenCV，NumPy，tkinter和PIL，有助于降低实施成本。

多线程允许有效地并行执行对象检测和 GUI 更新。

仔细考虑资源使用情况（如内存和处理能力）有助于优化解决方案的性能。

定期测试和调试有助于识别和解决开发过程中的任何问题或复杂性。

从长远来看，文档和代码注释有助于理解和维护解决方案。

# 3 项目关键技术

语言：100%python语言

库：

OpenCV （cv2）：用于图像处理和对象检测。

NumPy：用于处理图像数据和数组操作。

tkinter：用于创建 GUI 界面。

PIL（pillow）：用于图像处理和显示。

threading：用于创建多个线程。

queue：用于线程间通信，存储处理后的图像和检测结果。

算法：

YOLO：用于对象检测。在代码中，加载了 YOLO 模型权重（yolov3.weights） 和配置文件 （yolov3.cfg），并使用 YOLO 模型进行对象检测。

NMS（非最大抑制）：用于删除重叠的边界框。在代码cv2.dnn.NMSBoxes 函数用于对检测到的框应用非最大抑制。

其他：

多线程：使用多个线程执行对象检测并并行更新 GUI 界面。

GUI 界面：使用 tkinter 库创建一个简单的 GUI 界面，包括用于显示图像的画布和用于控制对象检测的开始、停止和取消选择操作的按钮。

# 4 项目实现

## 4.1实现过程

设置环境：安装 Python，安装所需的库

获取 YOLO 预训练模型

加载 YOLO 模型

初始化深度排序

捕获视频流：从摄像机捕获视频源

对象检测和跟踪循环：处理视频源的每一帧，使用 YOLO 执行对象检测，并使用 DeepSORT 跟踪对象

后处理 YOLO 输出：从 YOLO 输出中提取边界框、类 ID 和置信度分数

显示跟踪结果：在视频帧上绘制边界框和标签以可视化跟踪。

测试和优化：通过调整参数（如 YOLO 置信度阈值或 DeepSORT 设置）来优化性能。

增加判断语句来添加操作键，用于跟踪目标对象

## 4.2不同模块是怎么实现的

### 4.2.1视频捕获模块

该模块负责从相机或视频文件捕获视频帧，使用 OpenCV 创建对象，并在循环中从中读取帧。

### 4.2.2YOLO 目标检测模块

YOLO 用于实时对象检测。加载预先训练的 YOLO 模型并处理每个帧以检测对象（在本例中为人体）。关键步骤如下：

* 1. 将每个帧转换为 blob。
  2. 将 Blob 设置为 YOLO 模型的输入。
  3. 通过 YOLO 网络转发以获取检测。
  4. 对 YOLO 的输出进行后处理，以提取边界框、类 ID 和置信度分数。

### 4.2.3深度排序跟踪模块：DeepSORT用于跟踪检测到的人体随时间的变化,关键步骤包括：

* 1. 创建一个 DeepSORT 跟踪器实例并使用预先训练的模型对其进行初始化。
  2. 为跟踪器提供每帧上来自 YOLO 的边界框。
  3. 更新跟踪器的内部状态并获取跟踪结果。

### 4.2.4可视化模块

该模块负责可视化跟踪结果，使用 OpenCV 在跟踪对象周围绘制边界框，并显示带有跟踪信息的视频。

### 4.2.5主循环

1. 从视频源捕获帧。
2. 将帧发送到 YOLO 模块以进行对象检测。
3. 获取检测到的边界框，并将其馈送到 DeepSORT 模块进行跟踪。
4. 从 DeepSORT 获取跟踪结果。
5. 结合对象检测和跟踪结果进行可视化。
6. 显示带有跟踪信息的帧。
7. 继续此循环，直到视频结束或用户决定退出。

### 4.2.6用户界面模块

添加用户界面以与系统进行交互，包括用于开始/停止跟踪和选择跟踪对象的功能

1. 创建图形用户界面模块：创建一个简单的 GUI 窗口，其中包含“开始跟踪”和“停止跟踪”按钮。当您单击这些按钮时，它将调用相应的函数
2. 将 GUI 与您的跟踪集成：修改start\_tracking和stop\_tracking与您的跟踪代码交互的功能。
3. 运行图形用户界面，同时运行 GUI 模块和跟踪系统，通常需要在主脚本中同时启动 GUI 和跟踪系统：
4. 获取界面上点击的坐标，调用相应函数以改变框的颜色

## 4.3最终实现的功能

* + 1. 基于OpenCV读取摄像头显示画面，利用YOLO实现摄像头人体识别
    2. 在GUI界面中实时显示识别的结果，点击某一个人，可以实现对这个人的跟踪。即点击这个人的bounding box，他的颜色和剩下的将不一样
    3. 有一个按钮可以取消跟踪

# 5 项目测试

1) 可以做到使用OpenCV读取摄像头显示画面，并且利用yolo进行人体识别

2) 将其集成到gui界面中，点击start键开始跟踪识别，识别框为绿色

3) 点击某一个人，该人的识别框颜色变为红色，完成跟踪识别

4) 点击cancel键，所有识别框变回绿色，完成取消跟踪功能

5) 点击stop键，gui界面关闭，程序结束

功能上完成基础功能；性能上图像刷新速度较慢，出现卡顿现象

# 6 项目管理

## 6.1 团队人员组成

班婷婧、陈颖林、冯思媛、张严匀

## 6.2 任务分工

张严匀负责实现用按钮取消跟踪的功能和完成ppt。作为队长，在整个团队工作过程过程中了解每位队员的意见并提出分工，积极组织团内讨论交流，是优秀的领头人，为团队的合作起到粘合剂作用，较好地监督促进了项目进展；同时也完成了分到的工作，完善了项目。

班婷婧负责实现基于OpenCV读取摄像头显示画面，利用YOLO实现摄像头人体识别功能，参考有关案例较快地成功实现了人体识别，为工作进一步推进开了个好头。在团队中，该同学积极配合，积极征求组员意见并提出新思路，主动参与团队交流，调节团队气氛，完成了分到的工作，促进了项目进展

陈颖林负责实现在GUI界面实时显示识别结果的功能，发现用原来的设想的模块完成其他功能和GUI界面不太适配的问题，促使团队及时修改了使用模块，让工作重新回到正轨。在团队中，该同学主动提出疑问、促进团内交流，共同促进项目进展

冯思媛负责实现对点击目标的跟踪功能，虽然参考案例较少，但另辟蹊径，以“覆写”的方式实现选中的效果。在团队中，她积极分析编程逻辑，热心帮助队员解决问题，极大地促进了项目的进展

# 7. 总结与反思

在本次学习中，队内几位成员从python初学者一步步探索“深度学习”这个新方向，最终合作完成了一个借助深度学习模型实现关键功能的项目，不得不说是一段艰难而有趣的经历。

为了储备相关知识，我们一方面从课堂上了解人工智能和深度学习的发展历程、了解深度学习的逻辑和主要方法、学习入门案例；同时也在csdn论坛等学习社区学习跟大作业选题相关的案例与算法，设计自己的程序。

在项目推进过程中，我们前期分工后各自开工，没有规定统一的时间地点，导致大家交流不及时，犯了很多重复的错误；选题答辩后参考别组的项目进展速度和工作方式，我们也及时调整，大家统一时间地点一起工作，及时向队友同步自己的进度，也能更快地通过讨论解决问题。这一工作方式使我们项目进度突飞猛进，也将应用于我们以后的项目合作中。

总而言之，本次课程在为我们学习“人工智能”和“机器学习”起到了启蒙和入门的效果，最后的大作业也让我们充分认识到团队合作的重要性和高效的合作方式。虽然因时间紧凑，我们在很多细节上没有打磨好，很多知识也没有融会贯通，但是仍旧受益匪浅。

**评分页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 成员姓名 |  |  |  |  |  |
| 预习报告  16%（百分制） |  |  |  |  |  |
| 平时成绩  20%（百分制） |  |  |  |  |  |
| 课程设计与答辩  64%（百分制） |  |  |  |  |  |
| 合计 |  |  |  |  |  |

**指导教师评语**

|  |  |
| --- | --- |
| 学生1 |  |
| 学生2 |  |
| 学生3 |  |
| 学生4 |  |
| 学生5 |  |
| 教师 | 郑 玮 |

2023年9月16日