

**2022年计算机系课程实践四**

期末作品报告

小组成员：　　王鹏博、吴雨娟

作品名称：

填写日期：　　2022.6.18

**目 录**

[第1章 作品概述 1](#_Toc106482729)

[1.1 技术路线 1](#_Toc106482730)

[1.2 创新点 1](#_Toc106482731)

[1.3 预期测试效果 2](#_Toc106482732)

[第2章 问题分析 3](#_Toc106482733)

[2.1 关键性问题 3](#_Toc106482734)

[2.2 相关已有的研究 3](#_Toc106482735)

[第3章 技术方案 3](#_Toc106482736)

[3.1 总体介绍 4](#_Toc106482737)

[3.2 人脸检测和数据收集 4](#_Toc106482738)

[3.3 训练识别器 5](#_Toc106482739)

[3.4 人脸识别 5](#_Toc106482740)

[3.5 相关信息传输 5](#_Toc106482741)

[第4章 已完成的工作 5](#_Toc106482742)

[第5章 预期效果 6](#_Toc106482743)

[第6章 课程心得 6](#_Toc106482744)

[参考文献 6](#_Toc106482745)

# 作品概述

## 技术路线

使用OpenCV和Python完成人脸识别工作，主要分为三个阶段：①人脸检测和数据收集；②训练识别器；③人脸识别。使用OneNET平台，将实时监控拍到的图片以及识别结果发送到手机APP上，提示用户当前监控到的环境状态。

## 创新点

1. 将树莓派板连接电源，并进行封装，应用于实际场景。



图 1 封装后产品实拍图

1. 使用物联网平台，实现手机实时查看监控的环境状态。



图 2 手机APP实时监控功能图

## 预期测试效果

在手机APP可以实时查看监控拍到的画面，并且收到监控发送的相关提示信息。

# 问题分析

## 关键性问题

1. 如何检测出人脸。
2. 如何将人脸录入到数据库中。
3. 如何训练分类器。
4. 如何将识别出的人脸匹配到数据库中的人脸信息，即输出人脸姓名或输出陌生人。
5. 如何将监控拍到的照片和鉴别信息发送到手机用户端，实现用户实时查看相关信息的功能。

## 相关已有的研究

1. 使用OpenCV完成的开源计算机视觉库。
2. 基于Haar级联分类器方法的目标检测。
3. 视频流传输，推流到云端。

# 技术方案

## 总体介绍

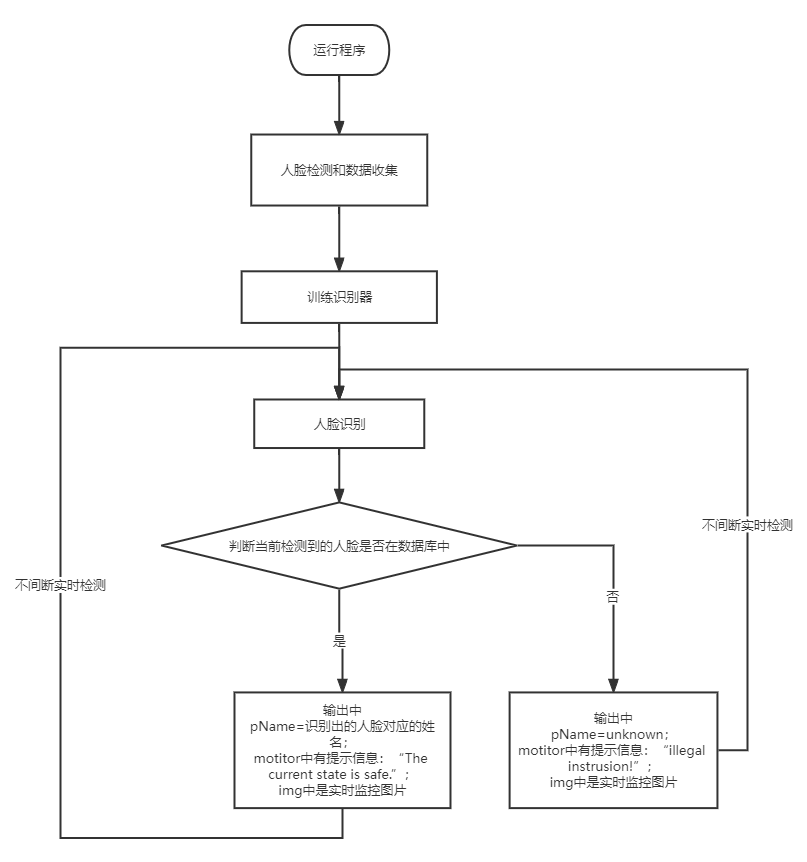


图 3 总体技术路线框架图

## 人脸检测和数据收集

1. 首先进行人脸检测。我们使用的是Haar级联分类器[1]，这种方法基于大量正面、负面图像训练级联函数，然后用于检测其他图像中的对象。然后用 OpenCV 创建人脸检测器, 在循环内部调用摄像头，并以grayscale模式加载输入视频, 接着调用分类器函数，向其输入一些参数，如比例因子、邻近数和人脸检测的最小尺寸。

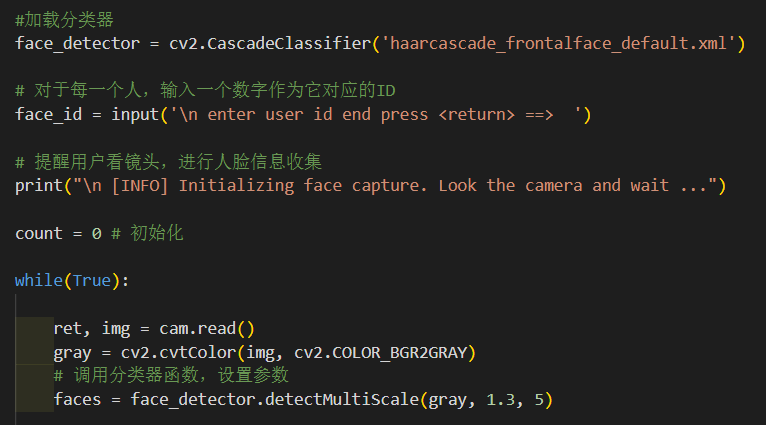


图 4 分类器相关代码

1. 然后要标记图像中的人脸，得到位置信息后绘制矩形，用imshow() 函数呈现结果。

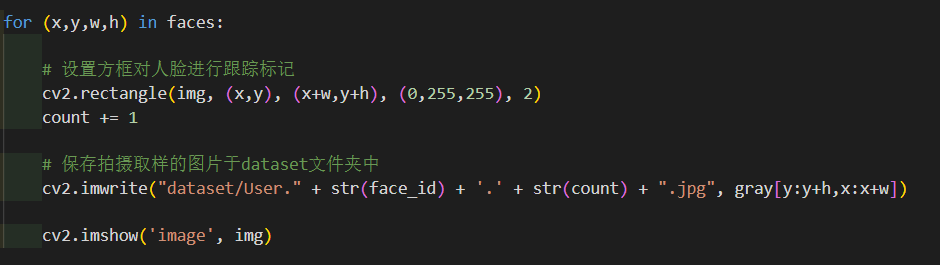


图 5 绘制矩形并显示相关代码

1. 为了进行数据收集，我们创建了一个简单的数据集，该数据集将储存每张人脸的 ID 和一组用于人脸检测的灰度图。

## 训练识别器

1. 从数据集中抽取所有的用户数据，并训练 OpenCV 识别器。使用OpenCV 提供的LBPH人脸识别器，对人脸照片数据进行处理。



图 6 人脸数据处理相关代码

1. 抽取dataset目录中的照片并返回两个数组：ids和faces。通过将这些数组作为输入，我们就可以训练识别器。在训练过后，文件trainer.yml将保存在trainer目录下。



图 7 训练并保存模型相关代码

## 人脸识别

1. 首先将人的ID与名字进行关联，并设置人脸规格。

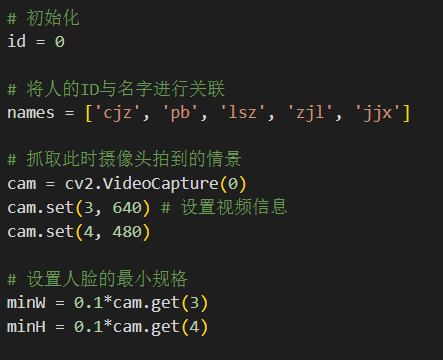


图 8 初始化相关代码

1. 通过摄像头捕捉一个新人脸，如果这个人的面孔之前被捕捉和训练过，我们的识别器将会返回其预测的id和索引，通过判断其置信度，来输出我们的预测结果。

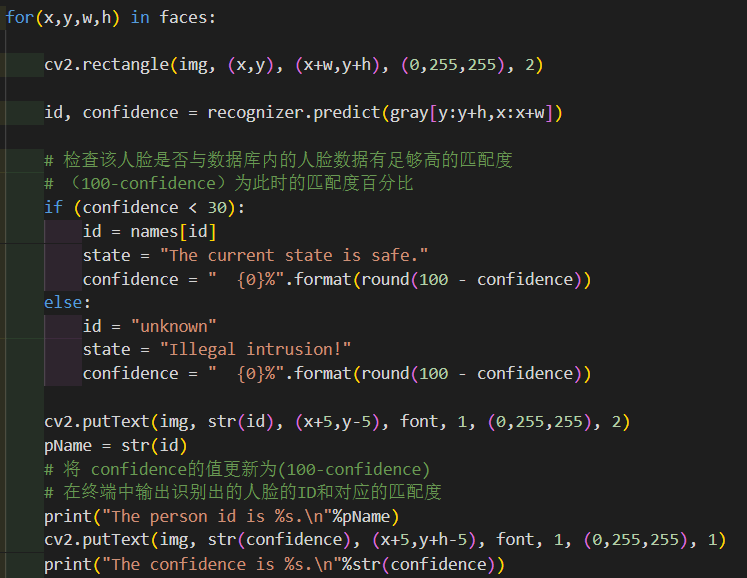


图 9 人脸识别相关代码

## 相关信息传输

1. 从摄像头捕捉视频流，并通过置信度判断得到人脸id信息，根据id信息填写要发送的提示信息。

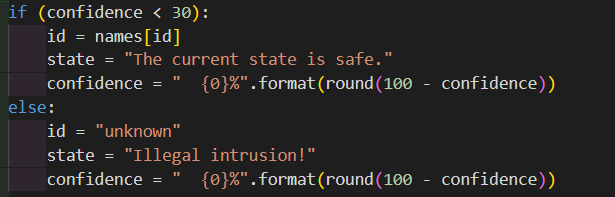


图 10 根据id填写state相关代码

1. 向OneNET平台已经创建的数据流发送数据点，实现实时监控的效果。

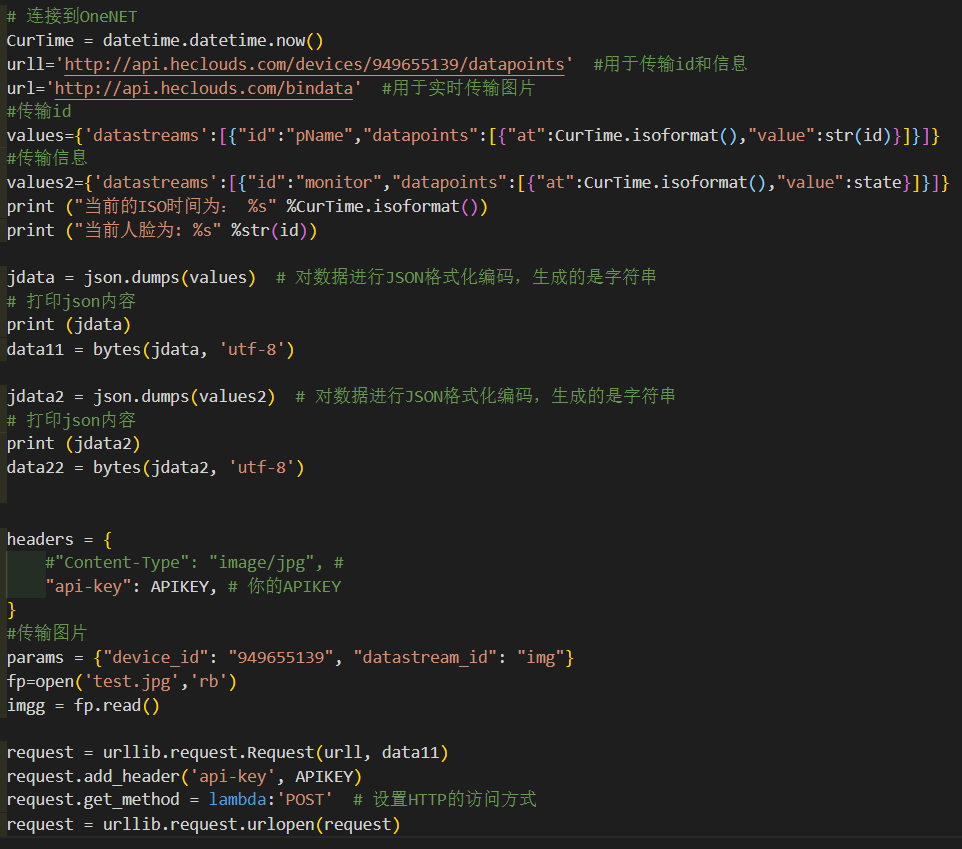


图 11 连接到OneNET相关代码

# 已完成的工作

1. 捕捉人脸
2. 标记图像中的人脸
3. 存储人脸数据集
4. 从数据集中抽取所有的用户数据，并训练 OpenCV 识别器
5. 通过摄像头捕捉一个新人脸，如果这个人的面孔之前被捕捉和训练过，识别器将会返回其预测的 id
6. 将实时拍摄的图片和相关预测结果发送至手机APP，供用户实时查看

# 预期效果

在宿舍或者需要进行实时监控的环境中，放置封装好的实时监控设备。用户使用电脑连接树莓派，启动程序，开启摄像头进行实时监控。用户离开该环境，之后可以在手机APP上实时查看环境中的实时监控图片，以及相应的检测结果和提示信息，得到用户远程实时监控的目的。

# 课程心得

在这门课上学习了树莓派的许多应用知识，将课堂上学习到的知识应用到实际的硬件上去。在知识落地的过程中，遇到过各种各样的问题，老师在课堂上讲授的都是示例代码，要完成作业，需要我们在课堂示例代码的基础上，结合自己的理解和网络上的参考资料，进行进一步的代码完善。在拍摄演示视频的过程中，也遇到过一些困难，比如碰撞检测的视频，还有棋盘识别的视频，需要反复拍摄，调整拍摄角度，才能达到预期的效果。

在最后的大作业项目，我们选择了做一个实时监控设备，也是对这学期工作的一个整合和总结，运用OpenCV和OneNET平台，实现物联网的效果，实现了我们把需求落地应用的愿望。

# 参考文献

* 1. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990517.