

南 京 师 范 大 学

毕 业 设 计（论 文）

（ 2018 届）



题 目： 在线学习行为多模态检测技术的研究与应用

学 院： 教育科学学院

专 业： 教育技术（信息技术）

姓 名： 何云帆

学 号： 04140214

指导教师： 王丽英

摘 要

近几年，大学生在线学习的过程设计已经比较成熟，但是整个学习过程中缺乏对于学生学习行为的跟踪和分析，而对于学习状态监测系统的实现就更加的寥寥无几。

为了更好地促进和提升在线学习者的学习效率及学习效果，本文基于旷视表情识别技术、物联网生理状态检测技术、视频图像分析技术和 Arduino 数据通信原理等，从底层嵌入式开发、数据分析软件开发到 Django 服务器后端开发等多个技术角度，设计开发实现一套对在线学习行为进行多模态数据采集、分析和结果呈现的系统。此系统从学习过程中的心理、生理反馈角度判断学生的学习状态，并给予学习者状态报告，为形成性评价或过程性评价提供了新的评价依据。

通过设计测试方案，对本系统对实际的在线学习者进行测试验证。测试结果表明，本软件对在线学习者具有较低的侵入和干扰，能够较全面的感知反馈在线学习状态。

关键词：在线学习行为；多模态检测；表情识别；物联网技术；Django 架构

Abstract

In recent years, the process of online learning for college students is becoming more and more popular. However, the entire learning process still lacks tracking and analyzing students' learning behaviors. the related Learning status monitoring system is few and urgent on the demand. In order to promote and enhance the online learning efficiency and learning effect, this paper is based on Face++ expression recognition technology, IOT physiological state detection technology, video image analysis technology and Arduino data communication principles. From the bottom embedded software, data analysis software to back-end Django server software, these technologies are studied. One multi-modal detection system ,which contains multimodal data collection, analysis, and presentation of online learning behavior, has been designed and developed. This system judges the student's learning status from the psychological and physiological feedback points in the learning process and gives the learner status report, which provides a new evaluation basis for formative evaluation or process evaluation.

Through the design of the experimental program, the system is tested and verified by the actual online learners . The experimental results show that the software has low intrusion and interference for online learners, and can be more comprehensively aware and feedback online learning status.

Keywords: Online Learning Behavior; Multi-modal Detection; Facial Expression Recognition; Internet of Things Technology; Django Architecture

目 录

摘 要.....	i
Abstract.....	ii
第 1 章 绪论.....	1
1.1 研究意义.....	1
1.2 研究内容.....	2
1.3 国内外研究现状.....	2
1.3.1 学习监控的相关研究.....	2
1.3.2 情感识别和分类算法的相关研究.....	3
1.3.3 监测生理状态的相关研究.....	4
1.3.4 结合情感识别和生理信号的相关研究.....	4
1.3.5 其他数据处理技术的相关研究.....	5
第 2 章 在线学习行为多模态检测系统架构设计	6
2.1 系统架构设计.....	6
2.1.1 数据采集层.....	6
2.1.2 数据分析层.....	6
2.1.3 呈现反馈层.....	8
2.2 系统流程设计.....	8
2.2.1 多模态数据采集.....	8
2.2.2 学习行为分析.....	8
2.2.3 学习行为建议.....	9
2.3 相关专业技术术语解释.....	9
第 3 章 在线学习行为多模态检测技术设计和实现	11
3.1 数据采集层.....	11
3.1.1 心率数据采集器.....	11
3.1.2 视频采集器.....	13
3.1.3 鼠标键盘监测组件.....	15
3.1.4 后台进程检测组件.....	15
3.2 数据分析层.....	15
3.2.1 视频的处理分析.....	16
3.2.2 表情识别器的开发.....	16
3.2.3 进程处理器的开发.....	17
3.2.4 鼠标键盘的监测数据处理.....	18
3.3 呈现反馈层.....	18
3.3.1 后端数据存储架构.....	19
3.3.2 数据的序列化与存储.....	26

3.3.3 数据前端网页呈现.....26

第 4 章 在线学习行为多模态检测测试29

4.1 整体测试流程.....29

4.1.1 测试准备.....29

4.1.2 测试期间.....29

4.1.3 测试后.....29

4.2 测试结果案例.....30

4.2.1 分析层的数据结果.....30

4.2.2 反馈层的数据结果.....31

第 5 章 总结和展望34

5.1 成果总结.....34

5.2 反思与展望.....35

参考文献.....36

致 谢.....38

第 1 章 绪论

经过了几年的发展，大学生在线学习的过程设计已经比较成熟，但是整个学习过程中缺乏对于学生学习行为的分析，包括学习过程的深度跟踪，基于大数据的诊断与预测，学习过程的心理活动分析和学习过程中情感状态的监控。此课题利用现有的技术条件，设计开发实现一套用于学生在线学习状态跟踪的系统。当学生结束上课时，系统反馈其本次上课的学习状态，从而提高学生对于整个学习过程的自我了解，提升学习成绩。老师可以轻松地掌握学生的学习专注程度，达到对学生学习状态的精准调整的程度。

1.1 研究意义

近几年，由于 HTML5 等新 WEB 设计规范的出台，越来越多交互性强，功能逻辑强的网页应用应运而生，在这一批兴起的 WEB 应用中，在线学习平台成为网上应用耀眼的分支。加上手机，平板等电子设备的普及，“随时学习，随时暂停”的学习模式已经被许多大学生吸收运用。大学生在网络平台上在线学习的时间在整体学习时间上占比日渐提高。在线学习是相对于课堂学习形式的学习方式，学习资料的载体是英特网，即由英特网提供学习的入口、学习资料，并由其主导拟定学习流程。在线学习在 MOOC 的推动下逐渐形成高潮。美国教育部 1996-2008 年间 50 项在线学习效果的数据表明，在线学习与传统学习的学习效果没有显著差异，而混合学习的效果明显优于面对面的传统学习。而在面向未来的学习型社会中，在线学习差不多成了主要的学习方式，再加上“数字土著”对在线学习的天然亲近，还可以有效降低人才培养成本，在线学习培养显得尤其重要。要促进在线学习的形成，首先要找出在线学习的发展动力。

大学生在线学习日益发展，使得有关学习过程、学习工具、学习资源、学习效果的分析很多。高等教育成为在线开放课程开发与应用的前沿。基于统一的 MOOCs 资源体系，可以实现多层次、多模式、多样化的应用，应用可以是自主学习、交互式学习、混合式学习，以及翻转课堂等等。

不过，很多 MOOC 平台注意的关键点在于，如何获取更为优质的教学资源，

如何与国内外的高校建立长远的合作关系,在这种教学资源驱动下的 MOOC 平台的发展,在教学内容的安排、教学过程的设计以及教学资源的提供与共享方面都做得十分到位,但是学习者的学习质量和学习效果往往达不到预期的目标,原因在于淡化了学习监控部分的重要作用。经过现在对以往经验的反思,人们已经逐步意识到了网络监控的重要性并且正在不断尝试将学习监控应用到网络课程的学习当中。此次的研究就是将学习行为跟踪加入到实际应用当中的一次尝试。

1.2 研究内容

本课题基于表情识别技术、生理状态检测技术、视频图像分析技术和数据通信原理等,设计开发一套对学生学习行为进行多模态采集、分析的系统,采集包括心率、面部视频、鼠标键盘操作、进程操作等多模态的学习数据并对其进行分析,最终系统将基于反馈的当前学习状态报告来判断学习质量的高低。

1.3 国内外研究现状

在线学习是指通过因特网进行学习与教学的活动,而研究远程学习者的在线学习行为特点,并发现影响这些行为的关键因素,将有助于远程教师及管理人员了解当前学习者的总体学习状况,为他们开展相关决策,如优化教学平台和教学资源、改进学习支持服务提供有力参考^[23]。国内相关专家提出了培育在线学习方式已经成为应对学习型社会建设的客观需要^[1]。但是目前现在网络学习存在学习平台使用率低、计划性差、教师讨论参与度不高、易用性差等问题^[2]。不过有文章就提出了包含自我监控、外部监控的监控实施策略来解决这样的问题。当然,外部监控的实施策略主要集中在对于学生的学习状态的检查和督促。

1.3.1 学习监控的相关研究

学习监控最直观的对于面部表情的观测。面部表情作为一种外显的生理状态可以反应情绪。早在 90 年代,孟昭兰就从心理学角度阐述了面部表情与情绪研究的关系,依据四个方面:面部表情的程序化的先天性,面部表情全人类的普遍性,婴儿的面部表情适应性,表情的发生原理。依据上述几个观点,揭示了面部表情和个人情绪的相关性,提出了相应表情的测量标准^[3]。之后,人脸的表情识

别、表情监控、情绪识别、情绪监控就成为了学习监控上探究的方向之一。

1.3.2 情感识别和分类算法的相关研究

有关情感识别和分类方面的算法和方案层出不穷。娄颜超在自己的论文中讨论了部分感情识别方法研究描述了智能化教学的现状,从普遍化情感理论出发分析教学中情感缺失的原因。基于面部表情特征提取和支持向量机理论,提出了基于改进的 ASM 方法的人脸特征点提取的方法,并且利用了这种方法讨论了人脸情感识别的相关算法和分类^[4]。冯满堂认为在传统智能教学系统中,学生在学习过程中缺少必要的情感状态描述,使得整个学习过程中的情绪变化是一种不可见的过程。据此,作者提出一种基于人脸表情识别的智能网络教学下系统模型。依据教育心理学为理论基础,以人脸面部识别为关键技术,通过捕捉和识别学习者的表情,判断和输出学习者的情绪状态。同时参照学习者的情感状态,对于学习者给予情感上的情绪补偿,用于填补智能网络教学中的情感缺失^[5]。孙波等人也认为,情感状态对于学习效果有一定的影响,而在学习环境中实现学习者分析有利于促进智慧学习的发生,依据心理学家 Ekman 的面部表情编码系统构建情感分析框架,通过特征分解将个体特征和表情特征分解到不同的子空间,排除个体特征对表情识别的干扰,并且已经在现实的教学环境中实现了基于面部表情的学习者情感识别和情感干预^[13]。Hwang K A 认为远程学习在于学生可以随时随地学习,但是很大程度上依赖于学生的自我特质。在不同的环境和条件下,不是所有的学生都可以细心。在这个研究中,作者设计了一个自动检测系统,这个系统使用图像处理和识别技术,依据学习者的面部表情和行为特征定义了注意力不集中的状态。通过贝叶斯网络模型来确定是否集中。在实施本研究体系并进行实践检验后发现,识别不良心态和注意缺陷行为特征的准确性较高。依据贝叶斯网络的评估和推理可以精确确定学生的学习注意力状态,得以让教师明确地控制学生的学习情况^[7]。Shan C, Gong S, McOwan P W 等人认为从原始人脸图像中提取有效的人脸表情是成功识别人脸的关键一步,本文基于统计局部特征局部二值模式对人脸无关表情识别进行了实证评估。对不同的机器学习方法在多个数据库上进行了系统的过程检验。大量的实验表明, LBP 特征对于面部表情是有效和高效的。该文章进一步研究了 LBP 特征在低分辨率人脸表情识别中的应用,得到了 LBP 特征提取方法的最佳识别性能^[8]。穆静等人按照子窗口表情采样、余弦

变换提取特征向量、面部隐马尔科夫建模、运用特征向量训练模型、测试准确率的步骤进行设计训练，最终达到了 87.3% 的正确识别率，识别率整体来说可行，但准确率略有差异^[14]。Eisenbarth H 认为特定的脸部区域（如眼睛）与情感表达的解码特别相关，但尚未研究观察者的扫描路径对于情感内容不同的脸部表情是否有所不同。在他的研究中，使用眼动追踪来监测健康参与者的扫描行为，同时观察不同的面部表情。记录固定位置及其持续时间，并计算优势比（即眼睛和嘴巴相对于脸部其余部分）^[16]。

1.3.3 监测生理状态的相关研究

与此同时，用于监测生理状态的物联网技术也蓬勃发展。蔡睿妍在论文中解释道，Arduino 系统是建立在 c/c++ 基础上的，Arduino 将 AVR 单片机相关的一些寄存器参数设置函数化，即使不太了解 AVR 单片机的朋友也能轻松上手^[9]。陆佳斌等人使用 arduino 的单片机设计一套监测长跑运动者实时状态的心率监测，实现了准确的生理数据的采集（包括心率数据和步伐数据），文中归纳了三种脉搏测量的方法：从心电信号中提取；血压波动计算；光电容积法^[10]，同时采用 PulseSensor 传感器测得的稳定心率数据误差维持在 2BPM（每分钟节拍数）内^[22]。所以基于此研究之上，本次物联网可穿戴设备选用 Arduino+PulseSensor 作为基础开发心率识别套件。

1.3.4 结合情感识别和生理信号的相关研究

如何在生理信号和情感之间建立一条映射是很多人考虑的问题。陈月芬在论文中介绍了情感识别的重要性，从生理学实验的研究角度阐述了心率、脑电、肌电、心电、皮电、呼吸生理信号与人类情感表现的相关性，说明了建立生理信号与情感的关系模型由充分的实验依据；描述了如何利用各种信号分析和处理方法来提取生理信号特征，并通过优化合成可用于情感分类器的特征向量^[11]。詹泽慧认为目前对学习状态识别的方法主要有脸部表情的识别、人体姿态的识别、语音语调的识别以及人体生理信号的识别。该文从视觉方面采集学习者的眼动数据，将此眼动识别技术与表情识别技术结合构建的基于智能 Agent 的远程学习者情感与认知识别模型，以提高远程学习者状态的识别准确率^[6]。温万惠以发现情感生理信号重要特征及其组合，并建立情感的生理信息计算模型为研究目的，

首次应用随机矩阵理论对情感生理信号进行信号序列相关性分析,以揭示信号序列中确实存在的情感生理反应和特定的情感生理反应时间模式^[12]。Kim K H 描述了一种实现方法,这种方法基于心电图,皮肤温度变化和皮肤电活动,因为所有这些都是从身体表面反馈出来的所以没有多少不适,同时可以反映情感对自主神经系统的影响。该系统由预处理,特征提取和模式分类三个阶段组成。采用支持向量机作为模式分类器,解决了情感的分类问题^[15]。

1.3.5 其他数据处理技术的相关研究

针对视频这个数据源,实时性和速度成为一个研究重点,相关研究者也有一定的研究成果。Viola P 开发了一个人脸检测框架,能够非常迅速地处理图像,实现高检测率。第一个是引入一个新的图像表示,称为“积分图像”,允许使用我们的检测器的计算速度非常快。第二个是使用 AdaBoost 学习算法(Freund 和 Schapire, 1995)构建的一个简单高效的分类器,用于从很大的一组潜在特征中选择少量的关键视觉特征。第三个贡献是将级联中的分级结合在一起,使图像的背景区域被快速丢弃,同时将更多的计算花费在有前途的类人脸区域上。在传统的桌面上实现了每秒 15 帧的人脸检测程序^[17]。姜明星提出一种时空局部三值方向角模式应用到视频情感识别中。首先对视频进行处理得到,表情和姿态两种模态的序列,提取三值特征。本方法对噪声和光照具有很好的鲁棒性^[18]。郑健围绕着:视频监控技术在交通管理、公共安全、智慧城市有广泛的应用前景,并且想着只能识别、实时处理、大数据分析的方向发展这一话题展开。采用 spark Streaming+分布式框架+olap 框架实现多维分析和查询。按照 ffmpeg、opencv、其他图像处理库的流程^[19]是非常值得本文借鉴的。赵兴方也针对视频分析技术,将 GPU 优化技术引入行人重识别问题。通过将行人重识别问题结构为行人检测、特征提取和特征匹配三个子问题,设计行人重识别的 GPU 并行识别框架,能够取得对于运动目标的检测和跟踪,也值得思考和学习^[20]。

第2章 在线学习行为多模态检测系统架构设计

利用现有的技术条件（表情识别库、心率传感器）以及相关分析技术，设计开发一套用于学生在线学习学习状态跟踪的程序套件，采集包括心率、面部视频、鼠标键盘操作、进程操作等多模态的学习数据并对其进行分析，最终系统将基于反馈的当前学习状态报告来判断学习质量的高低。学习者使用这个系统之后能够更了解自己的整个学习过程，从而提升自己的学习效果。

2.1 系统架构设计

系统分为以下三个层次，分别是：数据采集层、数据分析层、呈现反馈层。

2.1.1 数据采集层

通过摄像头、Arduino 心率采样器、鼠标键盘记录组件、进程监控组件，基于 sitewhere 大数据采集平台收集学习者有关面部表情、心率变化、进程控制等在线学习中多模态的数据，保存成对应视频格式、json 数据格式和相关数据日志格式，用于下一层的分析和整合。

2.1.2 数据分析层

使用 ffmpeg 方法对数据采集层中的视频格式文件进行转码和切分关键帧，将关键帧作为表情识别组件的输入参数。表情识别组件基于旷视科技 face++ 人工智能识别平台开发，通过输入关键帧等显式参数，反馈出关键帧所在时刻的情感状态。对于 arduino 采集的心率数据，是通过网络编程的方式获得结果，结果以一种 sitewhere 自我定义的 json 的数据格式反馈给下一个呈现反馈层。

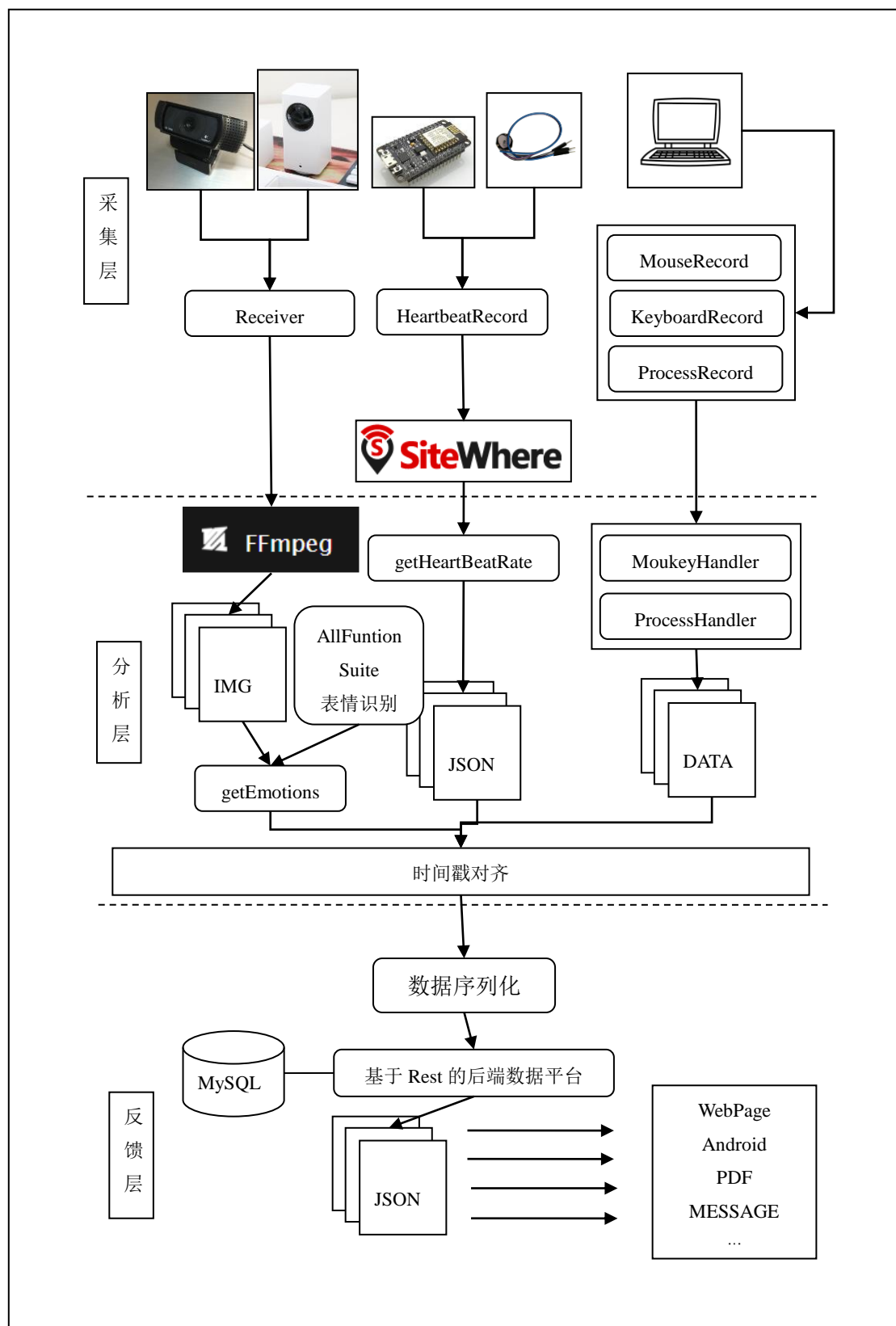


图 2-1 系统架构与流程图

2.1.3 呈现反馈层

创建后端服务器，基于 REST 服务器思路，将分析层的数据上传到云端服务器，使得所有报告中的数据从服务器上获取解析，而不是原始数据集。对于分析层经过了量化的心率数据，为了更好地呈现其变化程度，使用 echarts.js 绘制网页图表，并将图表嵌入到前端网页报告中，同时以数值方式显示分析层的处理数据。

2.2 系统流程设计

2.2.1 多模态数据采集

首先进行原始数据的采集。学生学习正脸视频由网络摄像头进行录制，用于后期分析学生面部表情。对于每一个进行在线学习的学生进行面部实时数据采集，标注好录制的关键时间点，以匹配心率时间轴。与之搭配的内容需要自己制作后台程序监测应用，10 秒钟返回一次后台应用进程列表。还需要自己制作鼠标键盘操作的相关监控应用，这三个应用整合在整个学习过程中电脑的操作行为集合中。学生学习心率状态表反馈出的是学生处于学习状态过程中的心率实时数据，在结果中作为报表反馈给学习参与者（学生）和教学过程监测者（如果有）。报表横坐标为学习时间/t，纵坐标为心率/次。心率状态表由专用心率光电传感器 pulsesensor 搭配 arduino 单片机制作而成，同时改进开发心率测量源码后将其烧录到开发板中，以测量更为密集，易理解的数据（BPM 数据）。心率的收集器选用物联网设备 sitewhere 开源程序，建立易用的传输路径，同时在获取数据渠道上，开发 android 应用（或者 weixinApp）使心率数据更加易于读取。

2.2.2 学习行为分析

其次进行学习行为分析。分析结果包括：心率量化状态表、情感状态时刻表、关键鼠标键盘操作、关键进程操作。心率量化状态表拟用 echarts 图表绘制，试用折线图做呈现。拟采用第三方情绪识别接口，开发情感检测套件，输入视频流中的帧以获得所在时刻的学习者情绪，集合成为情绪状态时刻表。同时对进程操作、鼠标键盘输入等变量进行筛选。

2.2.3 学习行为建议

最后,给出学习建议,用于提高学生对于整个学习过程的自我了解,提升学习成绩。系统预计给出整套的报告,其中包括学习过程中的心率图表,学习情感变化趋势图表,关键鼠标键盘操作、关键进程操作,学习建议等。

2.3 相关专业技术术语解释

PulseSensor: 用于脉搏心率测量的光电反射式模拟传感器,可以传递实时的心率数据,可以与 Arduino、树莓派等创客平台协同工作。

Arduino: 是一款电子原型平台开发板,包括硬件和软件(arduino IDE),通过 C/C++或汇编语言来对其进行开发。具有跨平台、简单、开放的特点,可以跨 Windows、OS X、Linux 主流操作系统运行。根据芯片不同分为多种型号,本次参与开发的主板型号是 Mode MCU。

Python: 一种高级编程语言,由于其支持动态类型、语法简单、可读性强、函数库丰富,被广泛运用在网络运维、数据分析、深度学习方面,本次的许多功能模块都是用这种类型的编程语言编写的。

Django: python web 的后端框架,支持快速建站,模块分工明确,此次将会作为 Restful API 的具体实现方法。

Restful API: 可重新表达的状态迁移(Representational State Transfer),一种万维网软件架构风格,目的是便于不同软件/程序在网络(例如互联网)中互相传递信息,也是本次后台与前端数据交互依赖的原则。

Echarts: 百度可视化数据分析图表开源库。

Sitewhere: 基于 JAVA 的开源物联网平台,用于采集实验中的心率数据。

Android: 常见的非官方中文名称为安卓,是一个基于 Linux 内核的开放源代码移动操作系统,由谷歌公司成立的开放手持设备联盟开发,主要设计触屏移动设备。

json: 轻量级的数据交换格式,被广泛应用在目前服务器数据交互上,具有易解析、易序列化的特征,是 javascript 语言的子集。

Postman: 用于后端服务器的接口测试,能很好地处理、呈现 JSON 数据格式。

Vultr: 全球云服务器提供商。

uWSGI: 旨在为构建托管服务开发一个完整的堆栈, uWSGI 通常用于与 Web 服务器 (如 Cherokee 和 Nginx) 一起提供 Python Web 应用程序, 它们直接支持 uWSGI 的本地 uwsgi 协议。

Nginx: 一种网络服务器, 其也可以被用作反向代理, 负载均衡器和 HTTP 缓存。

VPS: 虚拟私人服务器, 被云服务商作为商品销售给个人用户。

Linux: 服务器上常用的操作系统, 因其稳定性被广泛使用。

第3章 在线学习行为多模态检测技术设计和实现

3.1 数据采集层

3.1.1 心率数据采集器

用于采集在学习过程中的学习者的心率数据，采集器由 arduino 开发板、PulseSensor 心率传感器组成，以 sitewhere 物联网平台为数据转储的平台，是一套用于学习者实时心率监测的组件。

(1) arduino 组件

Arduino 是一种容易上手的开源开发板原型平台，其中包括两个部分：可以连接电路、传感器的开发板、计算机程序开发烧录环境 Arduino IDE，通俗说来，一个是硬件，一个是软件，两者相互配合，形成了一种广义上的软件控制硬件，难度较 c/c++ 更低，同时运用 lua 语言进行编写，代码量少，操作起来比 C 语言简单。同时由于 lua 的语言特性，可以作为 c 语言功能组件的插件，比较容易上手和编写。在本次的操作中运用的是 Node MCU 开发板，包含了型号为 ESP8266 的 wifi 芯片，用于连接云端 IoT 的 sitewhere 数据接收平台。通过 `Wifi.begin(ssid, password)` 的方法操作 ESP8266WiFi 芯片连接到 IoT 所在内网之中，建立数据传输的必备条件。

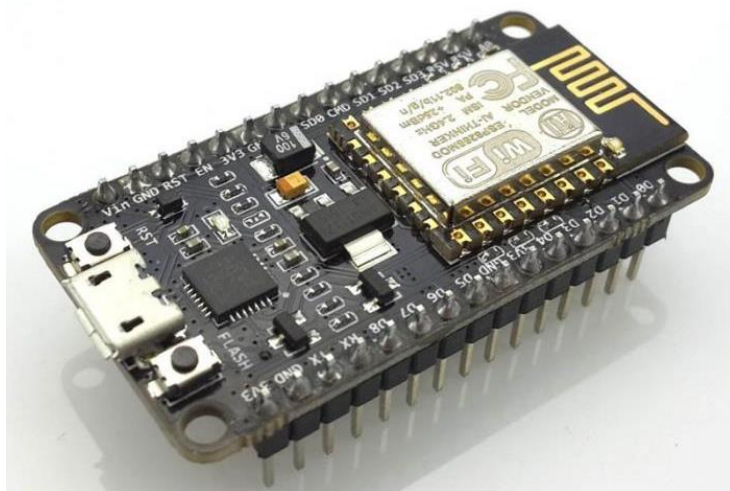


图 3-1 Arduino Node MCU

（2）PulseSensor 传感器

PulseSensor 传感器一种光电反射式心率测量传感器，只要将传感器接触处放置到手指指面、耳垂处就可以获得用于心率转换的信号。传统的心率测量有三种方式：从心电信号中提取出数据、从血压传感器的脉动提取出数据、光电容积法。光电容积法通过 LED 发射出绿光，之后光敏传感器接收到信号的变化，从中提取交流 AC 信号能反应心率特点，因为当光透过皮肤再反射到光敏传感器时有一定衰减，肌肉、静脉、骨骼和其他连接组织因为其静态的特性，对于光的吸收能力可以认为是不变的，但是动脉由于心脏的搏动，会导致吸收特性周期性变化，反馈出来的信号就是类心率的波形。光电容积法被广泛运用在可穿戴设备的心率测量功能上，例如知名的 iWatch、小米手环等，都是采用的基于这种测量原理的传感器。



图 3-2 PulseSensor

（3）Sitewhere 物联网平台

基于 JAVA 平台的开源的物联网平台，通过注册设备 token 来管理设备，通过多数据通信协议来传输数据，也提供底层的数据存储功能。以创建资产、创建设备规格、创建设备、记录 Token 的方式将先前的 Arduino 在数据库注册成为注册设备，通过 Rest API 进行上传数据和查看数据。

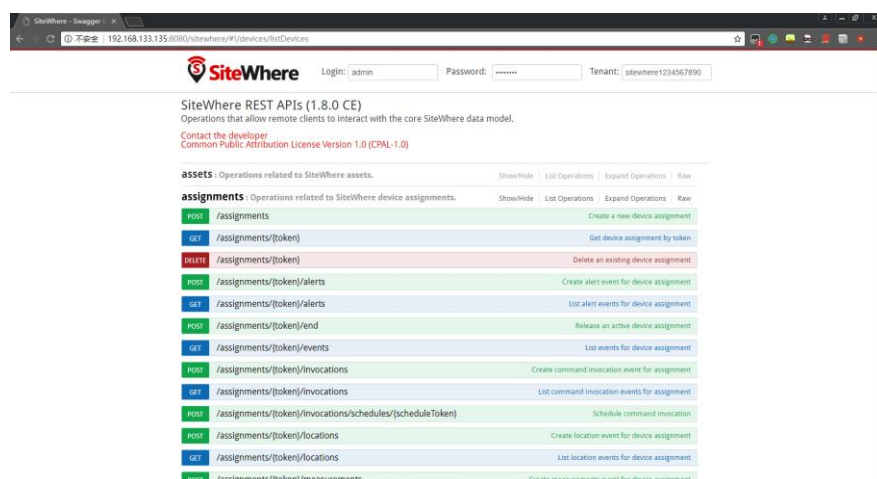


图 3-3 SiteWhere

(4) 使用流程

- ①测试人员将光敏电阻传感器面贴合在自己的任意一根手指的指面处。
- ②用 micro-USB 数据线给 Arduino 通电，Arduino 会执行烧录进开发板的代码。
- ③Arduino 通过 ESP8266 连接 SiteWhere 内网的无线信号，调试平台会显示已连接。
- ④建立连接后，心率传感器监测数据，将实时数据以 POST 方法，以 JSON 格式传递到连接到 sitewhere 数据测量的地址处，完成数据传输。
- ⑤停止传输之后，直接拔下 micro-USB 即可切断传输。
- ⑥可以通过浏览器预览数据平台上的已有数据。

3.1.2 视频采集器

尝试不同性质的采集器，选用小米大方家庭摄像机和罗技 HD1080p C922 PRO 的高清摄像头作为硬件采集设备。同样，视频采集器也有硬件和软件两个部分构成。

(1) 小米大方摄像头

优点在于支持局域网无线在线观看实时场景，操作简单，支持 SD 卡录制视频，录制的视频格式是 mp4，同时以时间为关键词命名每分钟的采集样本。



图 3-4 小米大方摄像头

(2) 罗技 HD1080p C922 PRO

像素相较于小米大方更高，不过录制的原视频需要调用 Windows 组件进行录制，或者使用第三方录制软件对摄像头进行系统调用，同时视频没有明确的时间戳，需要手工标注。



图 3-5 罗技 HD1080p C922 PRO

(3) 傲瑞视频录制

第三方视频录制，针对这个第三方软件，写了批处理程序用于在学习开始时自动开始录制视频（仅对于罗技摄像头）。

3.1.3 鼠标键盘监测组件

源码由 python 语言开发而成,使用两个源码文件,分别命名为 keyboard.py 和 mouse.py。针对于键盘监测 keyboard.py,使用 pynput.keyboard 获取相关键盘事件方法,使用 time 库定义操作时间戳,定义两个操作事件 on_press(key) 和 on_release(key),即当操作按下和松开键盘时,执行写入操作,将点按的字母的字符输入当前目录下的 Record.txt 文件中,同时加入时间戳;针对鼠标监测 mouse.py,使用 pynput.mouse 获得相关鼠标事件方法,定义两个操作事件 on_click(key) 和 on_scroll(key),即记录点按、滚动的鼠标位置和时间,执行写入操作,将数据写入当前目录下 Record.txt 文件,同时加入时间戳。原本准备对鼠标的运行轨迹进行记录,但实际考虑到这样的数据采集频率会占用太多的中央处理器性能,因此只记录点击和滚动操作。对于两份源码采用 pyinstaller 工具将 py 源码打包成 exe 文件 mouse.py 和 keyboard.exe,并人为设定了可执行文件的图标。

3.1.4 后台进程检测组件

源码同鼠标键盘监测操作一样由 python 写成,命名为 process.py。引用标准操作系统运维库 psutil,每隔 30 秒记录一下当前的进程数,打印以下关键数据到当前目录下 processInfo.txt 中:进程名,进程状态,创建进程的时间,线程数,在 processInfo.txt 的文件中形成块级数据,便于下一步的分析和操作。最后用 pyinstaller 打包成可执行文件。

对于整体的采集套件,当开始测量时,需要单独开启 arduino 心率测量器。对于两大录制操作和鼠标、键盘、进程的监控操作,提供两种解决方案:其一,以多线程的方式完成带有 UI 界面的开启界面,程序命名为 monitor.exe;二,批处理完成,最终成果 monitorStart.bat、monitorStop.bat。

3.2 数据分析层

对于数据采集层的数据进行拼接、整合、分析,得出量化的数据。

3.2.1 视频的处理分析

主要针对小米大方录制的视频进行整合分析。采集到的数据以分钟为单位存储成 MP4 文件，一个小时内的视频保存在以小时命名的文件夹中，简而言之，是一种碎片化的视频集。使用 ffmpeg 开源工具的基础功能将 MP4 视频转换成 ts 文件，即一个小时内的视频转换成 60 份 ts 文件（日本高清摄像机拍摄下进行的封装格式），原因是这样情况下再合并视频视频质量会比较高。之后对于一定数量的 ts 文件进行拼接。

当导出学习者完整在线的学习视频时，就可以对于学习视频进行截取关键帧操作。人为选定每隔五秒钟截取一张关键帧，即从第十分钟开始，每隔 5 秒钟执行一次截取操作，输出相应名称的 jpg 格式图片，这将作为输入表情识别器的原始图像。

3.2.2 表情识别器的开发

利用 face++ 平台人脸表情识别接口进行开发，新建名为 faceDetect 的类，公有成员变量两个，记录 api_key 和 api_secret；私有变量 10 个记录调用接口的 API 地址；12 个私有方法，底层的基础功能的搭建，有包括 _createFaceSet、_addFace2Set、_removeFaceFromSet 等底层方法和序列化；13 个公有方法，是可选择的功能项，可以初始化类使用其方法，相关使用方法如下表 3-1 所示：

表 3-1 表情识别操作类

方法	用途	输入参数	输出参数
createFaceset	创建一个面部集	outer_id	无
addFace2Set	添加 face 到面部集中	face_token, outer_id	无
removeFaceFromSet	在面部集中删除一个 face	face_token, outer_id	无
removeFaceSetAll	删除面部集中所有 face	outer_id	无
checkAllFacesets	查看账号下所有面部集	无	面部集
checkFacesets	查看指定面部集信息	outer_id	Face 的数量
searchInFacesetWithImage	检测指定照片中人是否在面部集中	image_file, outer_id	True False
searchInFacesetWithToken	检测指定照片中人是否在	face_token,	True

	面部集中	outer_id	False
outputEmotionAsFileWithDetect	通过探测方法识别人像数据	image_file	人像数据
outputEmotionAsFileWithDetectJustEmotion	只输出表情	image_file	{emotion:}
outputEmotionAsFileWithAnalyze	通过分析方法识别人像数据	image_file	参数对象
faceInfoInit	命名图片中的人脸并把他加入到某个 faceset 中	image_file, Name, outer_id	成功与否
checkWhoIsInPhoto	查看照片中有谁	image_file, outer_id	姓名

表情的判别，需要先对这个类进行初始化，运用循环的思想对其进行分析，输出 emotions 为键的字典对象列表。将每一张关键帧作为 outputEmotionAsFileWithDetectJustEmotion 方法输入参数，将输出每一张对应时间的关键帧的表情，也就是得到了分析之后的表情数据。

3.2.3 进程处理器的开发

针对数据采集层中 processInfo.txt 中存储的原数据进行整理和规范化输出，新建 processHandler 类，有以下基本方法：

表 3-2 进程处理类

方法	用途	输入参数	输出参数
file2List	将文件读取，并转换成数据结构为列表的数据流，每一个占位即是某一时刻电脑中的所有进程的字符串	文件路径	结果列表
getTwoList	获取一份列表备份用于错位比较	结果列表	结果列表（去尾） 结果列表（去头）
divideListIntoNameSection	将某一时刻的进程字符串转换为列表形式，每一个占位即是当前时刻某一个具体的进程片段	结果列表	进程列表 当前时刻时间戳
compareSections	比较两个进程列表中不同之处，留下有差别的部分	进程列表 1 进程列表 2	进程列表 3 进程列表 4
judgeInSectionList	给出判断	文件路径	{时刻}开启了{进程名} {时刻}关闭了{进

			程名} 没有变化
--	--	--	-------------

其中，进行比较的算法是通过复制一份数据拷贝，一个去掉头部，一个去掉尾部，错位比较得到的，这样游标第一位对应的是游标第二位，倒数第二位对应的就是倒数第一位，这样牺牲了一倍的内存空间，但是换取了一部分时间复杂度，不过，时间复杂度在本论文中不深入探讨。

要对元数据进行操作，只需要将 processInfo.txt 作为类的初始参数输入，再调用 judgeInSectionList() 方法就可以得到进程开启（关闭）的数据了，也就是对应时间戳的进程分析数据。

3.2.4 鼠标键盘的监测数据处理

建立名为 MouseKeyBoardMonitorHandler 的类，有以下基本方法：

表 3-3 鼠标键盘检测类

方法	用途	输入参数	输出参数
file2Str	读取文件并将文件转化为在内存中的字符串以供下一步操作	File	Str
file2List	读取文件并将文件转化为在内存中的列表以供下一步操作	File	[] (LIST)
collectNumbers	返回按键次数、鼠标次数、总操作数	无	keyPressedNum, mouseClickNum, allOperationNum
getContentTyped	返回学习者输入的内容	无	ContentList
getOperationTimestamp	返回每分钟学习者的操作数	无	Timestamp OperationNum

对于获取的元数据，只需要将 record.txt 作为类的初始参数，通过 collectNumbers(), getContentTyped(), getOperationTimestamp()操作对元数据进行分析，输出学生输入的次数、点击鼠标的次数、总体操作的次数、学生的输入内容以及做出输入操作时的时间戳，即获得处理后的鼠标键盘数据。

3.3 呈现反馈层

这一层的主要作用是对于数据反馈层的数据进行有序集合，按照一定的条件

进行输出，以及采用有艺术感的方式呈现数据采集成果、数据分析成果和建议。

3.3.1 后端数据存储架构

按照基本网络后端数据架构，建立学习状态检测平台，其中对于在线学习学生的包括测试集、状态、操作、心率、表情、进程的数据以一种对象化数据的思路进行存储；当外界有取数据的需求时，将存储的数据序列化成特定的数据格式输出。这一思路被广泛运用在各类网络应用中，被各大网络巨头采用，是目前最流行的网络架构。

(1) 数据字典

以下列出了基于对象的数据库的数据字典，涉及8个相关表：

datafeedback_clip 用于记录每次学习的学习数据集，每一次学习都有一个 clip_id 号；auth_user 用于记录学生的账号；datafeedback_emotion 用于记录每一个时刻的学生表情状态；datafeedback_heartbeat 用于记录每个时刻的学生心跳数；datafeedback_minutestimes 用于记录每个时间段学生的操作数；datafeedback_operation 用于记录学生每次学习的鼠标键盘操作总情况；datafeedback_process 用于记录学生每次进程操作情况；datafeedback_state 用于记录学生通过时间对其的心跳、操作、表情等数据集合。

表 3-4 数据字典

表名：datafeedback_clip					
字段名	数据类型	默认值	允许非空	自动递增	备注
id(pk)	INT(11)	无	否	是	主键
created	DATETIME(6)	无	否	否	创建时间
tested	DATETIME(6)	无	否	否	测试时间
owner_id	INT(11)	无	否	否	拥有人 id
clip_outer_id	VARCHAR(255)	无	否	否	此次昵称
表名：auth_user					
字段名	数据类型	默认值	允许非空	自动递增	备注
id(pk)	INT(11)	无	否	是	主键
password	VARCHAR(128)	无	否	否	密码
last_login	DATETIME(6)	无	否	否	登录时间
is_superuser	TINYINT(1)	无	否	否	管理员
username	VARCHAR(150)	无	否	否	名称
first_name	VARCHAR(30)	无	否	否	姓

last_name	VARCHAR (150)	无	否	否	名
email	VARCHAR (254)	无	否	否	
is_staff	TINYINT (1)	无	否	否	职员
is_active	TINYINT (1)	无	否	否	活跃
date_joined	DATETIME (6)	无	否	否	加入时间
表名: datafeedback_emotion					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键
state	VARCHAR (20)	无	否	否	表情状态
time	DATETIME (6)	无	否	否	时间
clip_id	INT (11)	无	否	否	组外键 id
owner_id	INT (11)	无	否	否	用户外键
表名: datafeedback_heartbeat					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键
beat_nums	INT (11)	无	否	是	心跳次数
time	DATETIME (6)	无	否	否	时间
clip_id	INT (11)	无	否	否	组外键 id
owner_id	INT (11)	无	否	否	用户外键
表名: datafeedback_minutestimes					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键
times	INT (11)	无	否	否	操作次数
time	DATETIME (6)	无	否	否	时间
operation_id	INT (11)	无	否	否	操作外键
表名: datafeedback_operation					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键
keypressed_num	INT (11)	无	否	否	按键次数
mouseclicked_num	INT (11)	无	否	否	鼠标次数
content	LONGTEXT	无	否	否	内容
alloperation_num	INT (11)	无	否	否	总操作
clip_id	INT (11)	无	否	否	Clip 外键
owner_id	INT (11)	无	否	否	用户外键
表名: datafeedback_process					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键

time	DATETIME (6)	无	否	否	时间
flag	VARCHAR (20)	无	否	否	开启关闭
process_name	VARCHAR (255)	无	否	否	进程名
clip_id	INT (11)	无	否	否	Clip 外键
owner_id	INT (11)	无	否	否	用户外键
表名：datafeedback_state					
字段名	数据类型	默认值	允许非空	自动递增	备注
id	INT (11)	无	否	是	主键
time	DATETIME (6)	无	否	否	时间
clip_id	INT (11)	无	否	否	Clip 外键
operation_num	INT (11)	0	否	否	操作数
heartbeats	INT (11)	无	是	否	心跳数
emotion	VARCHAR (20)	无	是	否	表情
process_flag	VARCHAR (20)	无	是	否	开启关闭
process_name	VARCHAR (255)	无	是	否	进程名

(2) 开发语言、框架

后端框架基于 RestApi 的开发原理，大部分业务逻辑由 python + Django + RestFramework 开发完成,为以后基于数据监测的相关应用建立强大的后端基础，设计具有人性化的接口调用方式，可以在 api 格式和 json 格式中选择返回格式，图 3-6 展示了一种查看数据的方式。在生产环境中，绝大多数情况下并不会访问这个页面，而是直接通过软件内部的 http 请求来取得这些数据，返回的大部分也是 json 数据格式，因为这样更加便于页面解析和渲染。文中使用 Postman 测试接口工具对后端开放框架进行测试，测试图如图 3-7 所示。

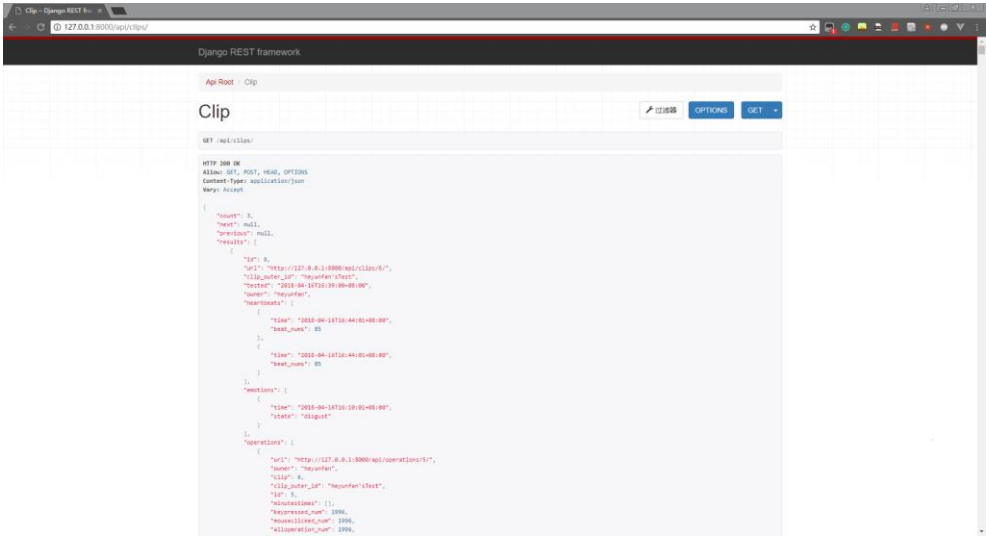


图 3-6 后端 api 调用模式图例

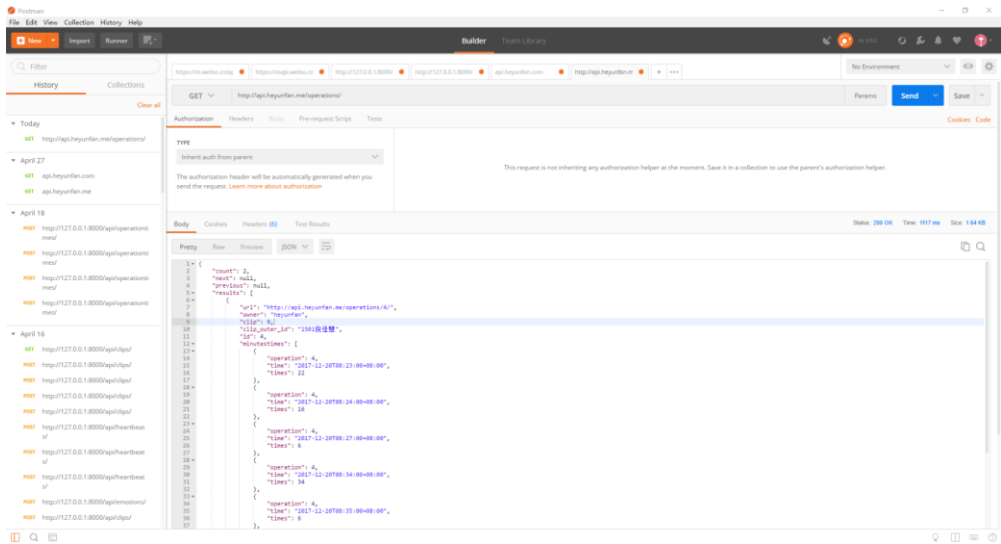


图 3-7 postman 模拟请求

同时，对于采集到的多模态的数据，定义了如表 3-5 查询方法和 API 文件说明。

表 3-5 API 文档说明

请求地址	描述	Post/Get 请求参数	方法	分页	排序
/api/	Api 根	无	Get	不支持	不支持
/admin/	管理员页面（可视化管理）	无	网页打开		
/api/users/	对用户将进行操作（有限权操作）	无	Get	不支持	不支持
/api/heartbeats/	获取/上传心跳数据	owner(option) clip(option) clip_outer_id(option)	Get/Post	支持	支持
/api/heartbeats/{id}	获取/上传/修改/删除心跳数据	clip beat_nums time	Get/Delete /Put	不支持	不支持
/api/emotions/	获取/上传表情数据	owner(option) state(option) clip_outer_id(option)	Get/Post	支持	支持
/api/emotions/{id}	获取/上	clip	Get/Delete	不支持	不支持

	传/修改/ 删除表情 数据	State time	/Put		
/api/operations /	获取/上 传操作数 据	owner(option) clip(option) clip_outer_id(option)	Get/Post	支持	支持
/api/operations/{id}	获取/上 传/修改/ 删除操作 数据	clip keypressed_num mouseclicked_num alloperation_num content	Get/Delete /Put	不支持	不支持
/api/operationtimes/	获取/上 传操作数 数据	operation(option)	Get/Post	支持	支持
/api/operationtimes/{id}	获取/上 传/修改/ 删除操作 数数据	operation time times	Get/Delete /Put	不支持	不支持
/api/processes/	获取/上 传进程数 据	owner(option) flag(option) clip_outer_id(option)	Get/Post	支持	支持
/api/processes/{id}	获取/上 传/修改/ 删除进程 数据	clip flag time process_name	Get/Delete /Put	不支持	不支持
/api/states/	获取/上 传状态数 据	emotion(option) process_flag(option) clip_outer_id(option) operation_num(option)	Get/Post	支持	支持
/api/states/{id}	获取/上 传/修改/ 删除状态 数据	clip time emotion heartbeats process_flag process_name operation_num	Get/Delete /Put	不支持	不支持

由于此次后端实现的方法比较复杂，在此并不展开叙述，操作参照官方文档

[21], 只描述如何简单使用。如果要获得 X 同学的相关数据, 可以执行如下操作: 通过向平台/api/heartbeats/发送 get 数据包, 可选的参数包括是哪一个数据组 (clip_outer_id), 是谁的数据(owner), 选择是哪一页的数据, 是否需要经过排序的数据(ordering=+/-), 是否需要使用 searching 字段(searching=?), 最终反馈一个 response.status_code 为 200 的数据包, 以 json 数据格式返回打印, 对数据进行解析即可; 如果要更新一条表情数据, 即向/api/emotions/发送 Post 数据包, 必须的参数包括, 属于哪一个数据组(clip_id), 处于什么表情状态(state=value), 是哪一刻时间发生的, 最终也会反馈一个 response.status_code 为 200 的数据包, 以 json 数据格式返回打印传输成功的 json 数据条目。

(3) 后端部署

采用 python + django + uwsgi + nginx 的基本思路将开发完成的后台部署到公网上。使用一台具有公网 IP 的 VPS 服务器, 主机位置坐落于美国迈阿密。服务器端操作系统为 Ubuntu Server 16.04。首先将本地的后端源文件打包上传到笔者自己的 github/datagridview 公共库中, 在通过 ssh 通过 22 端口远程登陆服务器, 使用 git clone 命令将公共库中的源码包下载到服务器存储空间中。

在服务器中没有自带的数据库, 为了与后端开发匹配, 手动安装 MySQL 5.7.22, 并且将开发机上的数据打包, 以 sql 文件格式导入远端服务器, 安装完成 MySQL 之后新建 learningstatedata 数据库, 并使用 mysqldump 命令将 sql 文件格式数据导入新建的数据库, 到此完成数据库的相关部署, 此时数据库拥有了一部分测试数据, 此时本地开发机可以通过 3306 端口登录到远端服务器查看、修改数据 (图 3-8)。

对于下载到服务器中的开发源码, 先安装一个纯净的 python 虚拟环境, 所有的依赖包都将安装到这个虚拟环境中。依次通过 pip 安装 django、django-rest-framework、django-filter、mysqlclient、django-cors-middleware 依赖包, 之后完成第一次测试, 使用自带的测试服务将业务逻辑映射到 ip 的 8000 端口上, 打开本地开发机, 通过 IP:PORT 方式访问测试网站, 能够成功打开网站后继续下一步操作; 在开发源码的根文件下创建 uwsgi.ini 文件, 其中设置几个 uwsgi 的选项: 绑定端口 80、log 日志位置、绑定 django-uwsgi 模块到 uwsgi 中、静态文件映射地址、线程数和用户执行句柄, 通过 uwsgi --ini uwsgi.ini 开启 uwsgi 网

络服务,通过本地浏览器访问 <http://IP> 可以看到成功完成;为了进一步提升性能和提高并发,在 `uwsgi` 外再加一层 `nginx` 服务程序处理并发提高反应速度:将 `nginx` 设置中 `uwsgi` 模块处替换成现有的端口即可将服务的跳转转接到 `nginx`,通过本地浏览器访问 <http://IP>,能够成功访问表示部署成功(图 3-9)。

部署远端服务器的最后一步就是将公网 IP 解析到域名上。笔者采用 `cloudflare` 域名解析服务提供商,将服务器的 IP 解析到 <http://api.heyunfan.me> 上,笔者已经实现了上述所说的部署。

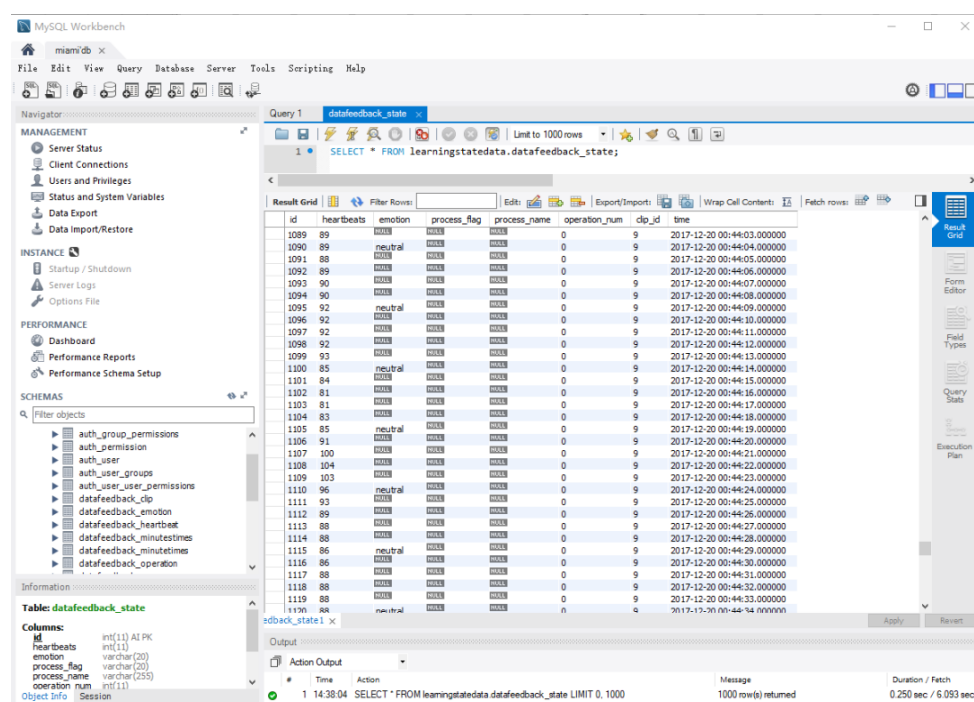


图 3-8 迈阿密远端数据库操作页

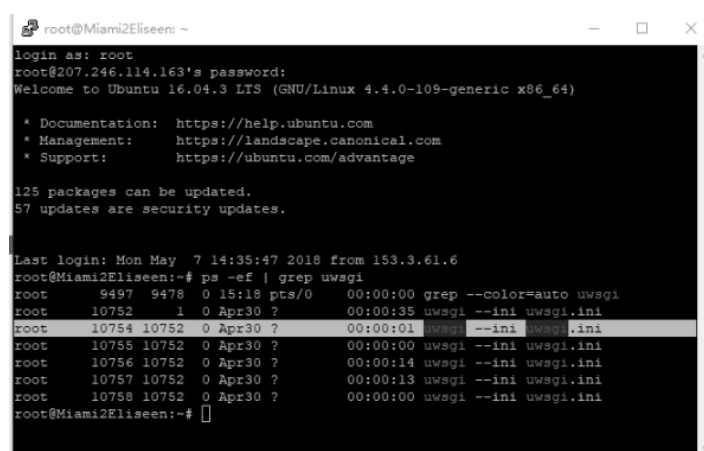


图 3-9 uwsgi 部署成功

3.3.2 数据的序列化与存储

为了对应图 2.1 反馈层数据序列化的过程,即将经过采集、分析之后的数据格式化后端能够接受的数据格式,特将数据切分修补功能(不改变具体数据值)和数据传输功能整合在数据处理工作组中。

(1) 数据的切分修补

其中包括两个使用程序: `timestamp2time` 组件和 `stateSerializer` 组件。前者用于将不规则的各类日期时间格式转换成后端接收的 `datetime(6)` 时间格式,基本思路是将各种时间时间种类转换成 Unix 时间戳,再将 Unix 时间戳转换为 `datetime(6)` 时间格式;后者用于将多模态的数据做成集合的形式,即某一个时刻学习状态分为心跳、操作、表情、进程四部分的数据,同时将序列化成容易解析的字典格式。经过这一部分的数据将会执行数据传输操作。

(2) 数据的传输

其中包括 6 个独立组件: `postCreteOperation`、`postEmotionData`、`postProcessData`、`postStateData`、`postOperationData`、`postHeartbeatData` 分别用于向后端传输操作数数据、表情数据、进程数据、状态数据、操作数据、心跳数据。采用 `requests` 依赖库发送 `http` 数据包。依据 API 数据格式 POST 出 HTTP 数据包,成功会打印 `Http` 状态码。

3.3.3 数据前端网页呈现

采用的前端开发基于 SPA(单页网页应用)的思想,所有的业务流程跳转通过 `javascript` 来实现。对于数据的呈现是通过 `jQuery` 原生库中 `Ajax` 来向上一部分中实现的后端发送 `http` 请求包,当成功取得数据之后,使用 `jQuery` 的各种方法来实现页面的事件、动画和数据绑定,整体的逻辑业务代码封装在 `index.js` 中。

前端的样式表文件以 `bootstrap.css` 为基础,按照整体的风格建立 `cover.css` 样式,形成一套风格简洁的 UI 界面(图 3-10),同时具有响应式的特性,随着设备大小的变化,与各种主流设备都可以比较好的适配。逻辑上分为两步跳转,主页和数据呈现页。

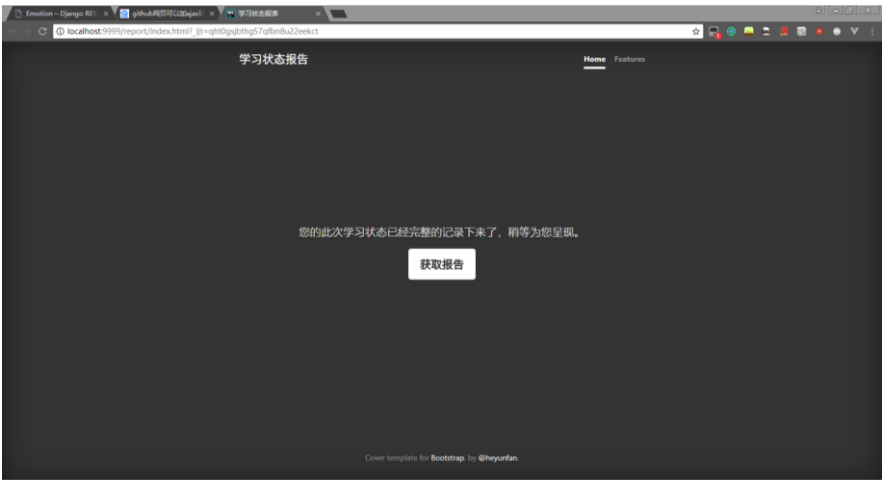


图 3-10 主页样式



图 3-11 数据呈现页

如图 3-10，当点击获取报告按钮时，会发送附带关于某次学习信息的 get 请求包到后端服务器，请求的地址是/api/state/，将返回该学生每一时刻的状态量数据，最终解析到对应的文本上去，即生成对于本次学习的数据总览表。对于表情相关、心率相关、操作相关和同轴图表部分，拥有相对应的触发事件，会获取对应 API 地址的学生数据，并通过 echarts.js 即时绘制基于 HTML5 的 canvas 的图表，使用 data 变量来绑定图表的数据变量，具有高可定义性和高扩展性，如图 3-11 所示。

(1) 表情相关分析

具体分析每一种表情的占比情况，所提及的表情包括：平静、喜悦、惊讶、恶心、伤心、生气和没有人脸的情况，为了更好的定义其二值性，将其认为的分为两个相对的方面：积极状态和消极状态。积极状态包括平静、喜悦、惊讶；消

极状态包括恶心、伤心、生气。将没有人脸单独列为极端情况。以饼图的形式绘制表示百分比，当积极情绪占比大于 75% 时，认定为认真状态。

（2）心率相关分析

在输入心率之初就剔除了部分明显极端数据，确保数据的可使用性。因为心率的数据比较多，因此采用大数据制图的方式进行绘制，图像 y 轴范围在[40,150]区间内，可具体查看某一时间段的心率变换，图标可以进行平移和缩放操作。从数据分类方面，对数据组取平均值操作。取平均值的 0.9 和 1.1 倍区间，当心率在(0,0.9)区间内，定义为心率过低；在(1.1,+∞)区间内，定义为心率过高。只要学习者处于的状态在[0.9,1.1]区间内超过 70%，认定为此次学习心态平稳。

（3）操作相关分析

用柱状图的形式表示每分钟的操作数，明显区分出数量值上的差异。同时将一部分不能转换为图表的东西渲染到下方的信息区中，包含键盘输入的内容，当中将一些特殊的操作符序列化成类似 c 语言的转义字符；包含进程操作的次数，将关闭的、开启的进程名输出到界面中，用于判断是否和本次学习相关。

（4）同轴图表分析

由于 state 状态量具有用时间来对应多模态的数据的特性，因此将后端数据库中 state 状态量与前端的同轴图标相联系，这样更加容易了解多种来源的数据随时间变化的特征。其中心率绘制与心率相关分析中的图标类似，为折线图；表情标识为坐标点的形式，由于数据量比较多，因此点集相当丰富；操作数也为点集，点的半径与操作数成正比，即该分钟操作次数越多，点越大。

当以上一些数据被获取后，整体会给出一个对此次学习状态的判断，分为优秀、良好和警告。当然，本次 demo 中的状态是以良好为示例进行阐述。

第4章 在线学习行为多模态检测测试

在此章中,选用一位实验者进行数据采集测试。测试之前已经与被测试者签订了数据采集同意书,出于保护用户个人隐私等原因,在此将用户名简略地称为X同学,取得的相应数据也分别称为某同学的数据报表。

4.1 整体测试流程

此小节对于整体测试流程做一个描述。

4.1.1 测试准备

准备连接了摄像头的实验机一台;将事先烧录好的带有指定 token 的 Arduino 心率测量套件准备好(安装流程在第三章有详细的描述),测试将以 USB 接口的方式给 arduino 开发板供电;将开发完成的软件集中到一个人命名为 LearningStateMonitor 的文件夹中,其中的软件包括: mouse.exe (记录鼠标操作)、keyboard.exe (记录键盘操作)、process.exe (记录进程变化)、oraycn.exe (录像操作)、monitorStart.bat (一键开启四个操作)、monitorStop.bat (一键关闭四个操作);开启 IoT 物联网 sitewhere 数据传送平台,测试是否正常工作;将小米或者罗技摄像头调整到最佳位置,便于采集到整个面部的最佳信息。

4.1.2 测试期间

被测试者到指定位置入座,佩戴上耳机,将光电心率传感器发光面对准指面,并固定这种姿势,将传感器用胶带固定在自己的手指上。点击 monitorStart.bat,开启监测的组件。将 Arduino 的 USB 连接线接到实验机的 USB 接口处。此后打开相关网络学习的课程,进行网络学习。

学习完毕,点击 MonitoStop.bat 执行关闭程序的批处理文件。

4.1.3 测试后

从 LearningStateMonitor 文件夹中获取视频文件、processInfo.txt、record.txt 文件,同时,相关心率数据记录在 sitewhere 物联网平台中,需要用相关的 HTTP 方法(GET)取得其数据。

对于心率数据,使用 `getHeartBeatRate.py` 程序执行对元数据进行请求、解析、输出的过程;对于视频数据,使用 `ffmpeg` 工具进行剪辑、拼合、截取操作(具体操作见 3.2.1);对截取下来的图片进行表情识别(具体操作见 3.2.2);对于进程文件 `processInfo.txt` 文件,使用进程处理器进行处理操作(具体操作见 3.2.3);对于鼠标键盘文件 `record.txt`,使用鼠标键盘的监测工具进行处理(具体操作见 3.2.4),经过这一层的分析操作,数据将被传送到呈现反馈层进行数据序列化操作。

在呈现反馈层进行数据序列化与存储(具体操作见 3.3.2),执行完毕表示已经将此学生的数据传输到云服务器平台 `http://api.heyunfan.me`,可以从中获取本学生本次的所有学习的数据;如要查看具体的报告,只需要打开前端网页报告页面即可查看本次的学习效果。

4.2 测试结果案例

4.2.1 分析层的数据结果

在这项研究数据的成果中,分析层得出的数据具有一定的研究意义。

表 4-1 X 同学的分析层操作数据表(部分)

时段	时刻	心率	情绪	进程开启	该时段内操作数
8:34	8:34:04	80.0	Sadness	无	10
	8:34:11	77.0	Disgust		
	8:34:16	81.0	无		
	8:34:21	84.0	Neutral		
	8:34:26	82.0	Sadness		
	8:34:31	80.0	Sadness		
	8:34:36	81.0	Sadness		
	8:34:41	85.0	Disgust		
	8:34:46	96.0	Neutral		
	8:34:51	90.0	Neutral		
	8:34:56	90.0	Disgust		

...	
8:50	8:50:17	80.0	neutral	无	2
	8:50:22	82.0	sadness		
	8:50:27	83.0	sadness		
	8:50:32	84.0	无		
	8:50:37	83.0	neutral		
	8:50:42	85.0	neutral		
	8:50:47	89.0	anger		
	8:50:52	86.0	surprise		
	8:50:57	89.0	无		
8:51	8:51:02	83.0	无	有	0
	8:51:07	83.0	无		
	8:51:12	83.0	无		
	8:51:17	83.0	neutral		
...	

总数据看出总学习时长 32 分钟，在这 32 分钟内未识别到人脸的次数为 6 次，总关键帧数为 389 帧，占比 30.38%；总体操作次数为 271 次，其中输入操作 110 次，鼠标操作 79 次，8:34 分操作数陡增；中途开启 taskeng.exe、taskhost.exe，之后关闭了这两个进程，判断为系统进程，与使用者无关，判定学生没有进行其他进程操作。

4.2.2 反馈层的数据结果

分析层的数据再次经过深加工之后，就会成为存储在云端，可以被共享、查看的数据集，也更加容易被获取、分析、反馈。此案例中，采用上一章中着重描述开发的数据前端反馈平台对此次学生学习状态进行评估，评估结果为良好。具体结果如图 4-1 至图 4-5 所示。

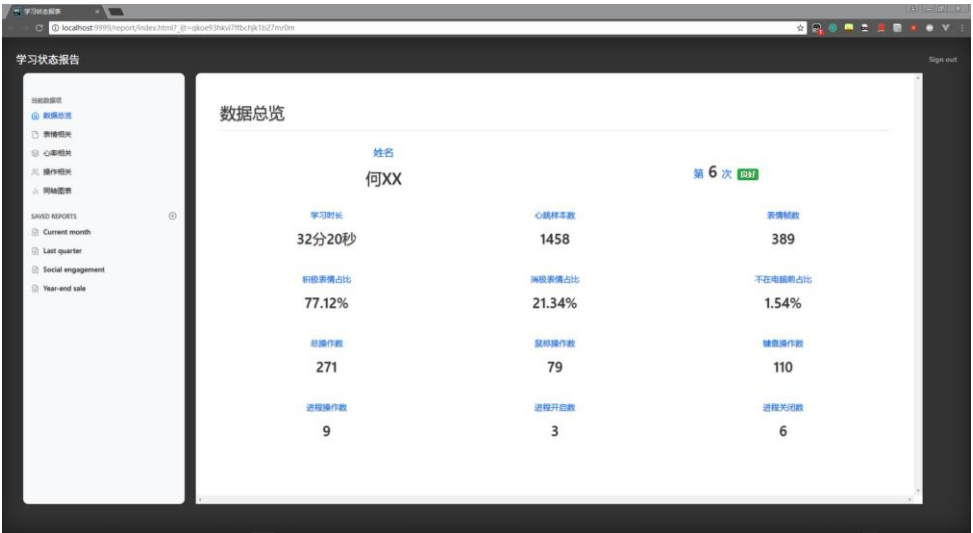


图 4-1 该同学的数据总览

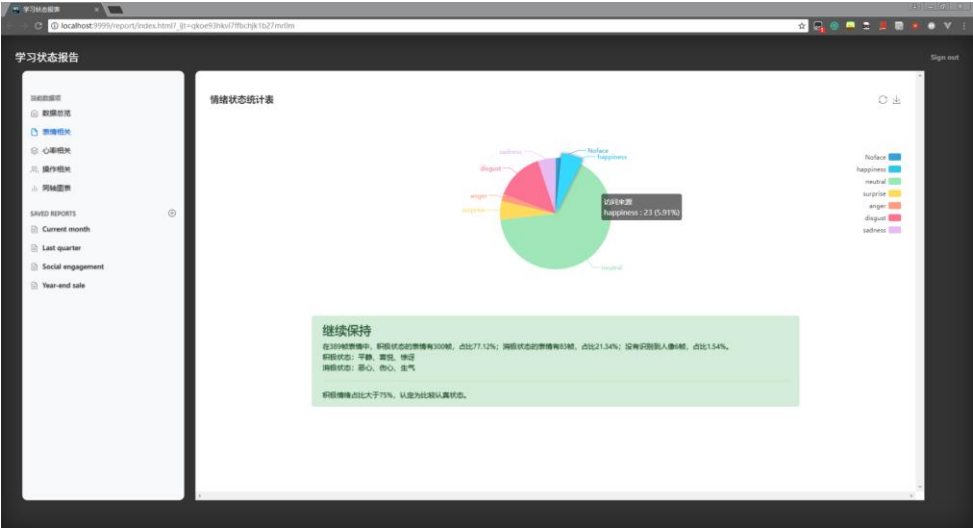


图 4-2 该同学的表情总览



图 4-3 该同学的心率数据图

第 5 章 总结和展望

本文通过研究多模态检测技术，完成了从数据采集、数据分析到数据呈现、状态判断的在线学习行为系统的开发与应用。测试阶段通过设计测试方案，对实际的在线学习者进行测试验证。测试结果表明，本系统对在线学习者具有较低的侵入和干扰，能够较全面的感知反馈在线学习状态，最终让学生对自己的整个学习过程更加的了解。

5.1 成果总结

相关的技术研究大部分停留在理论指导阶段，只是给出了应当如何去实现学习状态监测系统，而能够将其付诸实践的寥寥无几。在研究初期定下了整体系统的架构设计，并且对于每一步的实现方式都有一个预期的解决方案。从总体上来说，前期的理论基础研究比较充分，主要是从前人的研究成果中论证表情、心率等生理特征与心情、学习状态的关系。

研究中后期进行了相关系统的开发工作，该系统的实现具有以下的特点：

1、系统不依赖外部的数据源，所有的数据都是源自由自己组装、开发的硬件平台，尽管其拥有部分开源的代码，但是基于这种开源技术，我结合此研究面对的实际问题进行了二次开发，因此本文中的技术研究和实物开发都有其特性和专业性；

2、系统横跨硬件、软件、网络运维、平面设计，在任何一个方面的欠缺都会导致三层架构无法良好的实现；

3、系统涉及网络服务中的最前沿的思路，并且予以了实现，给未来的数据平台的多方实现提供了强大的后端支持，也给出了一种最常用的前端呈现方式，为未来的相关研究指明了道路。

该系统的应用流程主要包含以下步骤。第一步针对 Arduino 开发板、IoT 物联网平台进行心率监测组件的开发，并且成功在两者之间建立了联系，也开发了一个简单查看当前心率的 Android APP；第二步就是对于视频采集器、鼠标键盘检测器、进程检测器等测试时组件的开发；第三步在实地部署开发过的设备之后，

邀请学生进行在线学习，采集学习的数据，这里的数据就是元数据，是通过数据采集层获取的最初级的数据；第四步就是针对元数据开发特定的分析工具（见节 3.2）；第五步是对于后端的开发，历时 1 个月，实现了后端的大部分复杂的业务逻辑，并且成功在云端服务器部署，对外提供 <http://api.heyunfan.me/> 的数据接口（见节 3.3.1）；第六步就是向后端输入数据，其中涉及分析数据的序列化操作，对此实现了针对不同种类数据的序列化器（见节 3.3.2）；第六步实现了一个前端的呈现模型，从后端中获取数据，并绑定到单页应用中，真实反映学生的学习状态（见节 3.3.3）。

5.2 反思与展望

本系统还存在以下的不足之处：首先是远端服务器的反应速度问题，当在开发机本地服务器建立测试后端时，拿心跳数据的获取这个步骤来说，HTTP 处理时延高达 6 秒，当在远端服务器中部署之后，获取对应数据的时延高达 30 秒；其次就是原生 jQuery 框架的异步方式还有待优化，需要克服在处理请求时处于的假死状态。

不过，在可以想见的未来，上述提到的这些问题会被一一解决，会有反应速度更加快的后端服务器上线。对于网页呈现而言，与网页的交互也将更加的人性化、更加的优美；而对于其他多样化的衍生前端应用而言，教师短信的实时提醒、微信小程序的动态监测都是可以扩展的方面。同时，从多模态的数据源方面，提升学习状态检测的实时性也是未来要去实现的目标之一，因为本文是基于对采集数据的分析判断。如果将这个分析过程大大缩短，那么实时性也比较容易实现了，目前为止，后端的逻辑实现已经能支持这样的实时获取和分析。未来系统的功能将得到进一步完善，系统说明将逐步文档化，相信会有越来越多感兴趣的人参与到实现、优化的环节上来，让这个学习状态反馈系统更加的强大。

参考文献

- [1] 吴南中. 在线学习培育的顶层设计与推进机制研究[J]. 电化教育研究, 2016, 37(1): 45-50.
- [2] 胡明明, 隋毅. 基于网络课程的学习监控策略[J]. 科技资讯, 2017, 15(13): 143-144.
- [3] 孟昭兰. 为什么面部表情可以作为情绪研究的客观指标[J]. 心理学报, 1987 (2): 124-134.
- [4] 娄颜超. 智能化教学中的情感识别方法研究[J]. 硕士学位论文, 2011.
- [5] 冯满堂, 马青玉, 王瑞杰. 基于人脸表情识别的智能网络教学系统研究[J]. 计算机技术与发展, 2011, 21(6): 193-196.
- [6] 孙波, 刘永娜, 陈玖冰, 等. 智慧学习环境中基于面部表情的情感分析[J]. 现代远程教育研究, 2015, 2: 96-103.
- [7] Hwang K A, Yang C H. Learner attending auto-monitor in distance learning using image recognition and Bayesian Networks[J]. *Expert Systems with Applications*, 2009, 36(9): 11461-11469.
- [8] Shan C, Gong S, McOwan P W. Facial expression recognition based on local binary patterns: A comprehensive study[J]. *Image and Vision Computing*, 2009, 27(6): 803-816.
- [9] 穆静, 陈芳, 王长元. 人脸面部表情图像的隐马尔科夫建模及情感识别[J]. 西安工业大学学报, 2015 (2015 年 09): 705-709.
- [10] 蔡睿妍. Arduino 的原理及应用[J]. 电子设计工程, 2012, 20(16): 155-157.
- [11] 陆佳斌, 吴飞青, 吴成玉, 等. 基于单片机的运动者生理数据实时监控系统设计[J]. 安徽电子信息职业技术学院学报, 2017, 16(4): 17-21.
- [12] 陈月芬, 崔跃利, 王三秀. 基于生理信号的情感识别技术综述[J]. 系统仿真技术, 2017, 13(1): 1-5.
- [13] Eisenbarth H, Alpers G W. Happy mouth and sad eyes: scanning emotional facial expressions[J]. *Emotion*, 2011, 11(4): 860.
- [14] 詹泽慧. 基于智能 Agent 的远程学习者情感与认知识别模型——眼动追踪与表情识别技术支持下的耦合[J]. 现代远程教育研究, 2013 (5): 100-105.
- [15] 温万惠. 基于生理信号的情感识别方法研究[D]. 重庆: 西南大学, 2010.
- [16] Kim K H, Bang S W, Kim S R. Emotion recognition system using short-term monitoring of physiological signals[J]. *Medical and biological engineering and computing*, 2004, 42(3): 419-427.
- [17] Viola P, Jones M J. Robust real-time face detection[J]. *International journal of computer vision*, 2004, 57(2): 137-154.
- [18] 姜明星, 胡敏, 王晓华. 视频序列中表情和姿态的双模态情感识别[J]. 激光与光电子学进展, 2018
- [19] 郑健, 冯瑞. 基于 Spark 的实时视频分析系统[J]. 计算机系统应用, 2017, 26(12): 51-57.
- [20] 赵兴方. 基于 GPU 并行计算的智能视频分析技术[D]. 南京航空航天大学, 2016.

- [21] Tom Christie. *Djangorestframework* [OL]. <http://www.django-rest-framework.org>
- [22] 王宇. 基于 Pulse Sensor 的体征参数无线监控系统[J]. 数字技术与应用, 2016(10):90-92.
- [23] 魏顺平. 在线学习行为特点及其影响因素分析研究[J]. 开放教育研究, 2012, 18(4):81-90.

致 谢

论文完稿之际，在此想对帮助过我的人说一声谢谢。首先要感谢一下我的论文指导老师王丽英老师，王老师对我论文的研究方向做出了指导性的意见和推荐，我也在老师的循循善诱下逐渐找到了研究这个方向问题的思路。老师按时的提醒、以阶段性成果为流程的指导方式，使得整体的毕业设计的流程非常的顺畅和轻松，每次的汇报会老师都会针对当次的欠缺和下一步的任务做一定的总结和提示，完成这次论文，我不仅仅在论文书写、研究方法方面更为得心应手，而且在将复杂问题细分成每个任务逐个解决的这个思路下愈发熟练。由于这次的毕业设计涉及到相关程序开发、系统架构的思路，而王老师具备强大的理科知识基础，所以在整个软件工程的设计思路上，老师给予了很多的帮助。当整个系统开发出来之后，我将系统的整体的源码发送给老师，在和我对此进行细节上的对接之后，王老师百忙之中在实地部署了测试的程序，并给与了一定的意见，有这样认真负责的老师，是何等的幸运！

感谢我的爸爸妈妈，焉得谖草，言树之背，养育之恩，无以回报，在我倦怠时，是他们给予了我鼓励与督促，并且一整个学期都十分关心我论文的进展，现在对于他们的关心也算是有了交代。

开发的过程是艰苦的，但也是有趣的。在论文即将完成之际，我的情绪无法平静，从开始进入课题到论文的顺利完成，有多少可敬的师长、同学、朋友给了我无言的帮忙，在此表示我诚挚的谢意。

何云帆

2018年5月13日 南京师范大学西苑