



基于云边端的智能安防综合实战项目的设计与实现

李美蓉, 刘嘉旭, 陆永胜, 李松睿

(电子科技大学 信息与软件工程学院, 成都 610054)

摘要: 该文探讨了面向大二、大三本科生的综合项目课程, 以校园安防为背景, 引导学生将云计算、物联网和移动应用技术融入安防领域, 构建一套基于云边端的安防系统。项目涵盖了云端平台、边缘设备和移动端的设计与实现。云端平台负责数据存储、分析和管理工作, 包括用户管理、设备管理、视频监控等功能; 边缘设备使用树莓派进行实时数据处理和决策, 并连接摄像头、温湿度传感器、舵机等终端设备; 此外, 借助微信小程序打造了一款移动端应用, 为用户提供便捷的使用体验。项目实施过程中解决了多端系统数据交互、人脸识别驱动舵机开关门等复杂工程问题, 并从数据加密、权限管理等多个维度探讨了系统安全性与隐私保护等关键问题。通过这一综合项目课程的设计与实施, 学生的实践能力得到了提升, 对云边端技术有了更为深入的理解和应用。

关键词: 云边端; 智能安防; 人脸识别; RESTful

中图分类号: TP391.9

文献标志码: A

DOI: 10.12179/1672-4550.20240094

Design and Implementation of Intelligent Security Comprehensive Practical Project Based on Cloud Edge and Terminal

LI Meirong, LIU Jiayu, LU Yongsheng, LI Songrui

(School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: The article discusses a comprehensive project course for sophomore and junior undergraduate students. This course takes campus security as the background and guides students to integrate cloud computing, Internet of Things and mobile application technologies into the security field to build a set of cloud-edge-based security system. The project covers the design and implementation of cloud platforms, edge devices and mobile terminals. The cloud platform is responsible for data storage, analysis and management, including user management, device management, video monitoring and other functions. Edge devices use Raspberry Pi for real-time data processing and decision-making, and connect to terminal devices such as cameras, temperature and humidity sensors, and servos. In addition, a mobile application is developed using WeChat Mini Program to provide users with a convenient usage experience. During the implementation of the project, complex engineering issues such as multi-terminal system data interaction and face recognition driving the steering gear to open and close doors are solved. Furthermore, the article discusses key issues such as data encryption and access control from multiple dimensions, ensuring system security and privacy protection. Through the design and implementation of this comprehensive project course, students' practical abilities have been enhanced, leading to a deeper understanding and application of cloud-edge technologies.

Key words: cloud edge and terminal; intelligent security; face recognition; RESTful

自教育部提出“新工科”建设行动计划以来, 该计划在全国范围内得到了积极的响应。特别是在“新工科”建设与改革方面, 已经取得了显著的进展^[1]。在推动这一改革过程中, 一种“始

于新生、逐级递进、贯通四年的新工科项目式课程体系”建设理念被广泛采纳^[2]。与此同时, 随着“卓越工程师教育培养计划”持续升级与完善, “兴趣牵引、项目驱动、立足系统”^[3]的教

收稿日期: 2024-03-01

基金项目: 教育部产学育人项目 (231006375162922)。

作者简介: 李美蓉, 硕士, 实验师, 主要从事软件工程专业实验、实践教学和研究工作。E-mail: limeirong@

uestc.edu.cn

学思想也被融入整个综合项目的实践教学, 以配合新工科项目式课程体系的实施。

“基于云边端的智能安防综合实战项目的设计与实现”就是在此背景下应运而生的挑战性项目课程, 该课程主要针对软件工程专业大二、大三的学生开设。

1 课程背景及意义

随着社会的不断进步和城市化的快速发展, 安全问题一直是人们关注的焦点之一。传统的安防系统依赖于监控摄像头和人工监控, 然而, 这些系统往往存在一些局限性, 如监控盲区、人力资源的依赖性以及系统本身的安全性较低等问题。

该综合项目旨在培养学生的实践能力, 学习云边端的技术架构, 充分利用云计算和边缘计算的优势^[4], 实现人脸识别、实时监测和多源数据处理, 从而提高安防水平。该综合项目的设计不仅包括硬件与软件层面的结合, 还关注了隐私保护和权限认证等关键问题。

2 系统的总体设计

基于云边端的智能安防项目是一个集“云端平台+边缘设备+终端设备”于一体的软硬件综合实战项目, 如图 1 所示。该项目充分体现了新工科所强调的跨学科知识融合与综合应用。在该项目中, 云端平台负责数据存储、分析和决策; 边缘设备完成实时数据处理和决策; 终端设备涵盖了各类传感器和摄像头, 用于监测环境、采集数据以及实现门禁功能。学生需要综合运用多学科知识, 将“云-边-端”进行紧密结合, 最终构建一套智能安防系统。

该项目的主要功能模块如图 2 所示。其中云端平台作为项目的核心, 主要包括用户管理、设备管理、视频监控、通知预警、可视化展示模块。边缘设备使用树莓派进行模拟, 用于实时数据处理和决策。树莓派上连接了温湿度传感器和摄像头等终端设备, 用于采集温度、湿度和视频数据; 树莓派上还部署了人脸识别和智能门禁模块, 通过使用基于深度学习的人脸识别算法完成人脸识别后驱动舵机开关门模拟门禁系统。另外该项目还需要使用微信小程序开发移动端应用, 用户可以使用小程序完成用户注册、人脸采集以及其他服务。

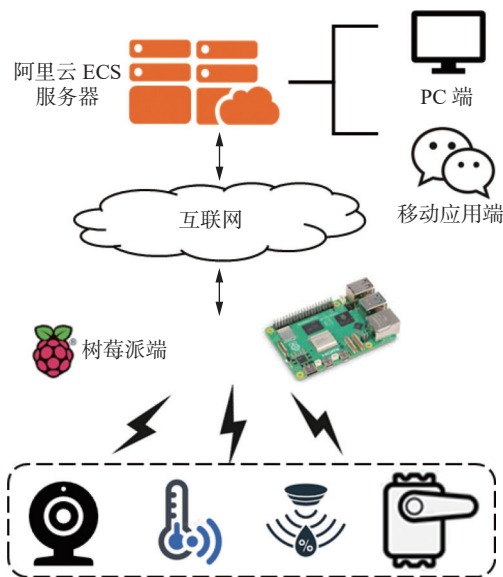


图 1 “云-边-端”示意图

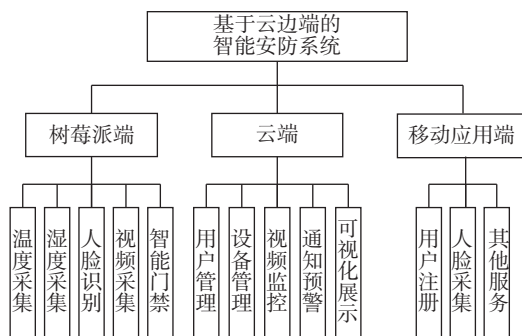


图 2 功能结构图

3 系统架构设计

在系统架构设计中, 融入了分层、模块化的设计思想, 这既符合新工科强调的工程化、系统化思维, 也有利于培养学生的团队协作和项目管理能力。

3.1 云端平台架构

云端平台主要基于 JAVA 开发, 如图 3 所示, 采用 3 层架构模式: 表现层、业务层、数据访问层。

1) 表现层使用 SpringMVC 框架, 具有调度器 (Servlet)、控制器 (Controller)、模型视图 (Modelandview)、前端组件等部分, 负责接收浏览器的请求操作, 并与业务层接口交互; 表现层采用 Thymeleaf^[5] 模板引擎, 负责云端平台的页面渲染。

2) 业务层使用 Spring 框架, Service 对象负责业务逻辑处理, 与表现层通过 Service 接口进行交互通信。

3) 数据访问层采用 MyBatis 框架，具有数据访问对象(data access objects, DAO)，与业务层通

过接口进行交互，并与数据库直接相连，完成数据的存取。

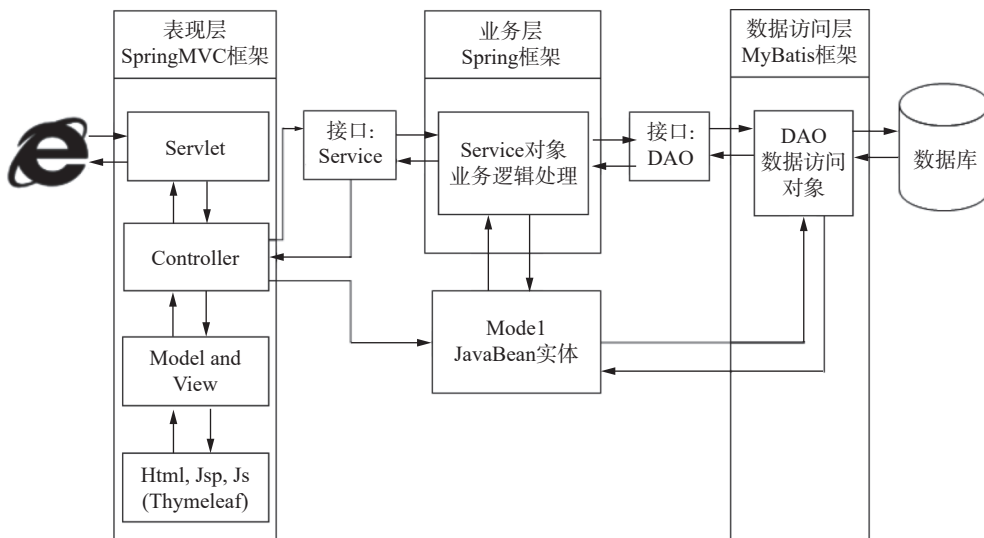


图 3 云端平台架构图

3.2 树莓派端架构

如图 4 所示，树莓派端主要使用星型拓扑结构，以树莓派为中心连接摄像头、舵机和多种传感器。

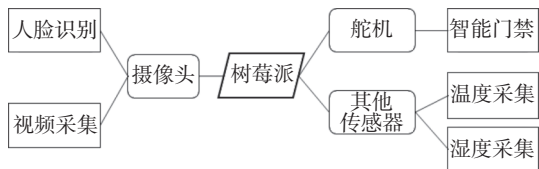


图 4 树莓派端架构图

树莓派作为核心控制器，负责处理来自各个设备的数据传输，以及执行相应的计算任务，并且作为边缘节点实现各设备模块之间的通信，如在摄像头实时捕捉到环境图像后进行数据的分析，控制舵机、显示屏等传感器操作，并将数据实时反馈给云端平台。

3.3 移动端架构

如图 5 所示，移动端使用微信小程序开发，整体框架构建包括逻辑层和视图层^[6]。逻辑层采用 JavaScript 语言开发，用于处理用户交互、数据请求、逻辑控制等任务。视图层采用微信标记语言(weixin markup language, WXML)和微信样式表(weixin style sheet, WXSS)进行构建。其中 WXML 用于构建页面的结构设计，类似于 HTML，而 WXSS 用于定义页面的样式，类似于层叠样式表(cascading style sheets, CSS)。

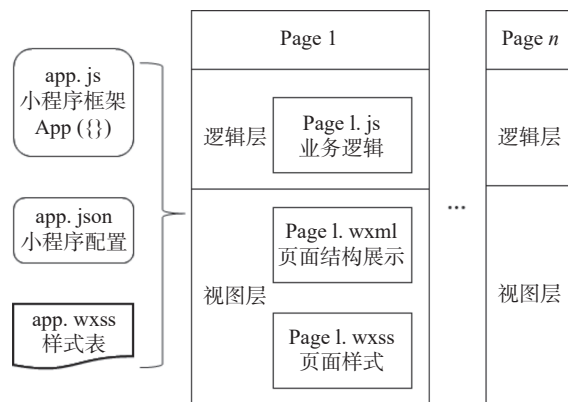


图 5 微信小程序端架构图

3.4 数据交互流程

如图 6 所示，基于云边端的智能安防综合项目涉及多端系统之间通信，其中有一个非常重要的需求是人脸采集与识别。下面以此为例讲述多端系统之间的数据交互流程。

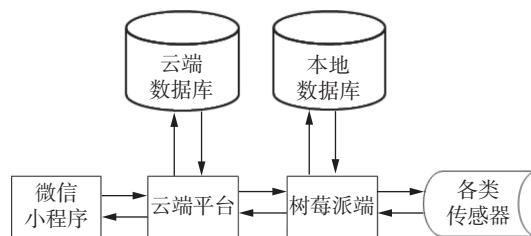


图 6 数据交互流程

1) 用户数据采集

微信小程序端通过调用手机摄像头采集用户人脸数据。

2) 用户数据存储

微信小程序端调用云端服务器接口, 将用户人脸数据传至云端服务器。

3) 用户数据拉取

树莓派请求云端服务器, 定时拉取人脸数据, 并存入本地人脸信息库。

4) 人脸信息识别

树莓派端通过摄像头获取实时人脸数据, 并进行分析比对, 识别是否为授权人员。

5) 舵机实现开关门

树莓派端根据识别, 通过触发舵机开关门模拟门禁系统。

6) 通行数据上报

树莓派端将通行人员的数据上传至云端服务器。

7) 通信记录查看

管理人员可以通过云端平台或通过微信小程序端查看到人员的进出数据。

3.5 数据交互接口设计

云端平台对外提供的接口均采用基于 RESTful^[7]风格的接口设计, 通信数据使用 JSON^[8]进行封装。请求数据时, 通过设置 HTTP 头的 Accept 属性为 application/json, 明确告诉服务器可以接收 JSON 格式。下面是请求温湿度信息时的接口:

GET/humiture/obtain HTTP/1.1

Accept: application/json

如果服务器正常响应, 会以 JSON 格式返回当时温湿度信息。

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "status": "success",
  "data": {
    "time": "2023-20-01",
    "humidity": "45",
    "temperature": "23"
  }
}
```

如果服务器验证用户未登录或会话过期, 也会以 JSON 格式返回响应的错误信息。

HTTP/1.1 401 Bad Request

Content-Type: application/json

```
{
  "status": "failed",
```

```
"data": {
  "message": "Session expired."
}
}
```

4 核心模块实现

在项目实施过程中, 老师会着重引导学生凝练出关键问题, 让学生明白在完成整个项目的过程中, 哪些环节是至关重要、须优先解决的。这样的引导不仅有助于提高学生的问题解决能力, 还能培养他们的逻辑思维和判断力。

4.1 用户数据采集与存储

用户数据的采集与存储主要由微信小程序完成, 如用户在使用人脸采集功能时, 微信小程序通过调用 wx.uploadFile 官方接口获取访问用户的摄像头与本地相册的权限, 并在用户选择人脸图像后将其发送至后端接口, 具体代码如下所示:

```
wx.uploadFile({
  url:'http://'+app.globalData.urlinfo+'8080
/uploadgraphbyfile', //后端接口地址
  filePath: tempFilePaths[0],
  name: "graph",
  header: {
    "Content-Type": "multipart/form-data"},
  formData: {
    id:that.data.id,
    ifStu:"false"},
  success: function (res) {
    var data = res.data
    console.log(that.data.id)
    wx.showToast({
      title: '成功上传! ',
      icon:'success',
      duration: 2000})
  }
})
```

4.2 用户数据拉取

用户的人脸数据通过微信小程序采集后都统一存储在云端服务器上。但是人脸信息识别是在树莓派端完成的, 因此需要将数据从云端服务器传到树莓派端。这里采用了树莓派端主动拉取的方式, 下面是其数据拉取并进行保存的关键代码:

```
url='http://192.168.207.83:8080/getgraph'#后 端
```

接口

```
response = requests.get(url)
data= response.json()
print(data)
for i in range(len(data['name'])):
    folder_name = data['name'][i]
    encoded_string_image = data['graph'][i]
    # 解码
    decoded_bytes = base64.
    b64decode(encoded_string_image)
    # 构建文件夹路径
    save_path="/home/admin/OPencv-face-recog-
nition/face_collect/%s"%(folder_name)
    os.makedirs(save_path, exist_ok=True)
    file_path = os.path.
    join(save_path, "new_self.jpg")
    with open(file_path, "wb") as image_file:
        image_file.write(decoded_bytes)
```

由于人脸数据库是动态变化的，因此需要定时对数据进行拉取。在树莓派端可以通过设置如下 crontab 任务，完成用户数据的定时拉取。

```
**/2 * * * /user/local/update.py
```

4.3 人脸信息识别

树莓派端使用 OpenCV^[9] 实时捕捉摄像头图像，然后调用 face_recongnition^[10] 进行人脸检测，检测到人脸图像后，对该图像进行特征提取并与人脸特征库中的人脸信息进行比对，最后根据设定的阈值返回比对的结果。face_recognition 是一个强大、简单、易于上手的人脸识别开源项目。它基于业内领先的 C++ 开源库 dlib^[11] 中的深度学习模型，用 Labeled Faces in the Wild^[12] 人脸数据集进行测试，有高达 99.38% 的准确率。人脸识别与比对的代码如下所示：

```
for face_encoding in face_encodings:
    # 人脸比对
    matches = face_recognition.compare_faces
        (known_face_encodings, face_encoding)
    name = "Unknown"
    face_distances = face_recognition.
        face_distance( known_face_encodings, face_
encoding)
    best_match_index = np.argmin(face_distances)
    if matches[best_match_index]:
```

```
name=known_face_names[best_match_index]
face_names.append(name)
```

如图 7 所示，人脸识别成功后会在视频中实时标识出人名。



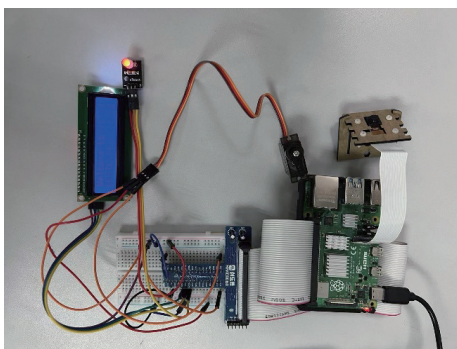
图 7 人脸识别

4.4 舵机实现开关门

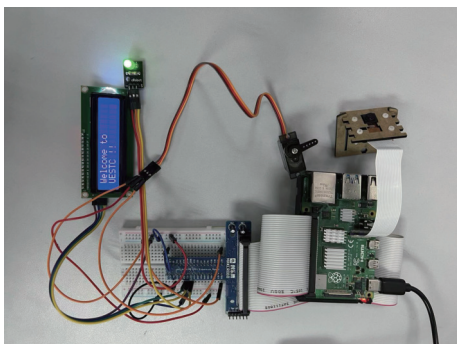
树莓派端通过 GPIO 引脚控制并驱动 MG90S 舵机。MG90S 舵机的控制信号是周期为 20 ms 的脉宽调制(pulse width modulation, PWM)信号，频率为 50 Hz，占空比为 2.5%~12.5%，线性对应舵机转动角度的 0~180°。也就是说，给舵机提供一定的脉宽，输出轴就会保持在一个相对应的角度上，直到给它提供一个另外宽度的脉冲信号，舵机才会改变输出角度到新的对应的位置上。在实现中，通过创建 PWM 对象，并设置占空比来控制舵机转动角度。控制舵机转动的关键代码如下所示：

```
# GPIO 引脚定义
servo_pin = 11
GPIO.setmode( GPIO.BOARD) # 采用实际的
物理管脚给 GPIO 口
GPIO.setwarnings(False)
GPIO.setup(servo_pin, GPIO.OUT)
pwm = GPIO.PWM(servo_pin, 50)
pwm.start(0)
def rotate_servo(angle):
    pwm.start(2.5 + angle / 18.0)
    pwm.ChangeDutyCycle(90)
    time.sleep(0.3)
    pwm.stop()
```

人脸识别成功前后舵机和显示屏的对比如图 8 所示。未识别到人脸时，舵机关闭，亮红灯，没有欢迎字样，如图 8(a)所示；人脸识别成功后，驱动舵机转动 90°，模拟开门，亮绿灯，并提示欢迎字样，如图 8(b)所示。



(a) 未识别到人脸



(b) 人脸识别成功

图 8 树莓派端舵机和显示屏对比图

5 安全性与隐私保护

除了注重功能性需求的设计与开发, 老师积极引导从多维度审视系统安全性, 培养他们的新工科思维。鼓励学生运用信息加密技术、搭建超文本传输安全协议(hypertext transfer protocol secure, HTTPS)安全通道以及实施精细化的权限管理等多种技术手段, 全方位保障系统的安全稳定运行。

5.1 用户信息加密

为了保护用户的个人信息和隐私, 项目对数据字典进行了梳理, 不存储身份证号、卡号等高度敏感的数据。对于用户密码采用 MD5 加盐加密存储, 即为每个用户生成不同的随机串, 与用户密码拼接后进行 MD5 加密存储, 不存储明文, 确保数据在存储和传输过程中不容易被未经授权的访问者窃取或窥视。

5.2 搭建 HTTPS

为了保护数据在网络传输中的安全性, 项目引入了 HTTPS^[13](超文本传输安全协议)。通过 HTTPS, 所有数据在客户端和服务端之间的传输都是加密的, 防止中间人攻击和数据窃取, 保证了用户与系统之间的通信安全, 确保用户信息

不容易被拦截或篡改。要支持 HTTPS 协议, 需要先申请 SSL^[14-15]证书, 该项目使用的证书是在阿里云上申请的。申请后将证书放置在“/home/tomcat/”目录下, 并按照如下方式修改 tomcat 的 server.xml 配置文件, 使其支持 HTTPS 协议:

```
<Connector port="443"
protocol="org.apache.coyote.http11.
Http11NioProtocol"
maxThreads="150" SSLEnabled="true"
scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="/home/tomcat/cn.pfx"
keystorePass="****"/>
```

5.3 权限管理

为了严格控制对系统的访问和数据的使用, 项目实施了基于角色的访问控制^[16-17]。不同用户对应不同的角色, 不同角色在系统中具有不同的权限, 只能访问其所需的信息和功能。这有助于降低数据泄露和滥用的风险。以下是权限认证相关的数据库表, 如图 9 所示。

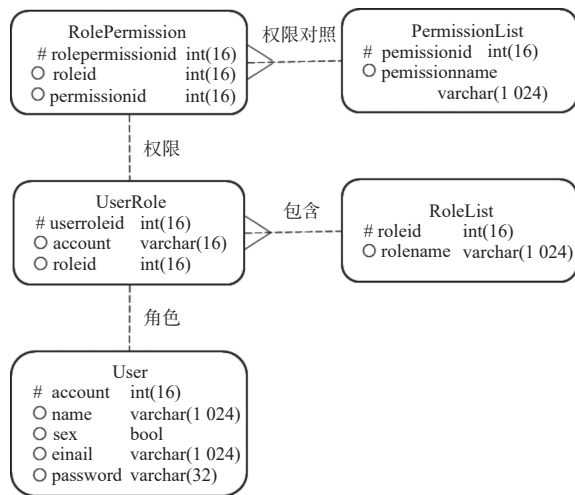


图 9 权限认证相关数据库表

6 实现效果

如图 10 所示, 云端平台完成了用户管理、校车管理、设备管理、通知预警等功能。可视化大屏中, 展示了实时监控画面, 并统计了用户入校信息、不同时段的校车预约流量、温湿度信息以及异常信息等。如图 11 所示, 移动端通过微信小程序提供了便捷的校园服务功能, 用户可以使用小程序进行进校预约、校车预约和用户管理等操作。

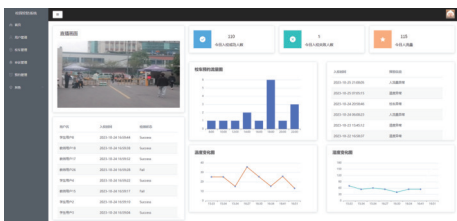


图 10 云端平台



图 11 微信小程序端

7 结束语

该综合项目的实施旨在培养大二、大三本科生的新工科素养，引导他们在日益复杂的云计算和物联网领域展开探索。项目将云边端相结合，为学生提供了一种全面的学习机会。通过云端平台、边缘设备和微信小程序的协同作用，学生能够深入理解云边端技术的核心原理，并熟练运用这些技术完成实际项目。

通过项目式学习，学生的跨学科协作能力、创新思维以及解决实际问题的能力均得到了显著提升。学生在项目中不仅仅是知识的接受者，更是问题的解决者和创新者。老师鼓励学生持续对项目进行改进和扩展，以积累更多的实践经验，为未来的工作和研究奠定坚实的基础。

参考文献

[1] 曾勇, 黄艳, 向桂君, 等. 从新生项目课开始: 新工科建设“成电方案”的设计与实践[J]. 高等工程教育研

- 究, 2020(1): 14-19.
- [2] 金东寒. 深化拓展新工科建设培养新时代卓越工程师[J]. 中国高等教育, 2022(12): 12-14.
- [3] 廖勇, 周世杰, 汤羽, 等. 面向新工科的软件工程专业核心课程体系建设[J]. 高等工程教育研究, 2022(4): 10-18.
- [4] 郭斌, 刘思聪, 刘琰, 等. 智能物联网: 概念、体系架构与关键技术[J]. 计算机学报, 2023, 46(11): 2259-2278.
- [5] 陈刚. 基于 SpringBoot+Thymeleaf+MySQL 的动态表单功能模块设计与实现[J]. 长江信息通信, 2023, 36(9): 100-102.
- [6] 李哲, 周灵. 微信小程序的架构与开发浅析[J]. 福建电脑, 2019, 35(12): 66-69.
- [7] CHEN Y, YANG Y, LEI Z, et al. Bootstrapping automated testing for RESTful web services[J]. IEEE Transactions on Software Engineering, 2023, 49(4): 1561-1579.
- [8] IEEE. IEEE Standard for learning technology: Javascript object notation (JSON) data model format and representational state transfer (RESTful) web service for learner experience data tracking and access: IEEE STD.2023.10273185[S]. New York: IEEE, 2023.
- [9] MILTON D K D, VELRAJ A R. Crowd size estimation and detecting social distancing using raspberry Pi and OpenCV[J]. International Journal of Electronics and Telecommunications, 2023, 69(1): 19-24.
- [10] ZHU Z, YAO C. Application of attitude tracking algorithm for face recognition based on OpenCV in the intelligent door lock[J]. Computer Communications, 2020, 154(3): 390-397.
- [11] 王梅霞, 徐怡彤, 杨照宇, 等. 基于 Dlib 与 OpenCV 的疲劳驾驶检测系统[J]. 中阿科技论坛, 2023, 6(3): 88-92.
- [12] RIM D, HASAN M K, PUECH F. Learning from weakly labeled faces and video in the wild[J]. Pattern Recognition, 2015, 48(3): 759-771.
- [13] HU Q W, ASGHAR M R. A large-scale analysis of HTTPS deployments: Challenges, solutions, and recommendations[J]. Journal of Computer Security, 2021, 29(1): 25-50.
- [14] PARIS I L B M, HABAEBI M H, ZYOUD A M. Implementation of SSL/TLS security with MQTT protocol in IoT environment[J]. Wireless Personal Communications, 2023, 132(1): 163-182.
- [15] MĂNESCU V A, ALEXANDRA N R, BARBU A. Analysis of SSL certificates trends and extended validation SSL usage for E-commerce websites and Internet of Things[J]. University Politehnica of Bucharest, Scientific Bulletin, Series C: Electrical Engineering and Computer Science, 2012, 83(4): 201-214.
- [16] 鞠博, 边臻. RBAC 模型在访问控制中的应用[J]. 福建电脑, 2022, 38(9): 37-40.
- [17] 王静宇, 张伟. 结合属性和 RBAC 的访问控制模型及算法研究[J]. 小型微型计算机系统, 2022, 43(7): 1523-1528.