计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(12):73−83 [doi: 10.15888/j.cnki.csa.008222]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

军用车辆驾驶远程授权及监控系统①

司维超[1], 宋　超[1], 满曙光[2], 齐玉东[1], 马迅骁[3]

[1](海军航空大学, 烟台 264001)
[2](烟台市公安局, 烟台 264001)
[3](92975 部队, 宁波 315000)
通讯作者: 司维超, E-mail: luckydevilsi@163.com

摘　要: 目前, 传统的军用车辆管理方法已经无法适应部队信息化的发展, 迫切需要使用一种高效的、融合新技术的管理模式. 为此, 本文针对管理中涉及的安全认证和授权方面, 设计了一款基于人工智能的军用车辆远程授权及监控系统. 该系统采用了最新的人脸识别技术和网络技术, 由车辆端、服务器端和客户端 3 部分组成. 其中, 车辆端部署于车辆驾驶室内的 NVIDIA Jeston Nano AI 开发板上, 使用人脸识别模型对驾驶员进行认证, 并与服务端进行数据交互. 服务端部署于云服务器, 用于管理车辆端和客户端信息、汇总车辆端上传数据、在车辆端和客户端建立数据交互. 客户端部署于移动智能终端, 用于进行远程授权、远程监控等. 通过测试表明, 本系统能够实现车辆的自动化认证、远程授权和远程监控, 防止了车辆的非法使用行为, 有力保障了车辆的使用安全.

关键词: 车辆管理; 安全认证; 人脸识别; 远程授权; 远程监控

引用格式: 司维超,宋超,满曙光,齐玉东,马迅骁.军用车辆驾驶远程授权及监控系统.计算机系统应用,2021,30(12):73–83. http://www.c-s-a.org.cn/1003-3254/8222.html

Driving Remote Authorization and Monitoring System for Military Vehicle

SI Wei-Chao[1], SONG Chao[1], MAN Shu-Guang[2], QI Yu-Dong[1], MA Xun-Xiao[3]

[1](Naval Aeronautical University, Yantai 264001, China)
[2](Yantai Municipal Public Security Bureau, Yantai 264001, China)
[3](Troops 92957, Ningbo 315000, China)

Abstract: The traditional management method for military vehicles has failed to adapt to army informatization, and it is urgent to use an efficient management mode integrating new technology. Therefore, aiming at the security authentication and authorization involved in management, this study designs a remote authorization and monitoring system for military vehicles based on artificial intelligence. The system adopts the latest face recognition technology and network technology, comprising vehicle, server, and client terminals. The vehicle terminal is deployed on the NVIDIA Jeston Nano AI development board in the driver's cab, and it depends on a face recognition model to authenticate the driver and carries out data interaction with the server terminal. The server terminal is placed in the cloud server to manage the information of the vehicle and the client terminal, summarize the data uploaded by the vehicle, and establish data interaction between the vehicle and the client. In addition, the client terminal is deployed in the mobile intelligent terminal for remote authorization, remote monitoring, etc. The test shows that the system can realize the automatic certification, remote authorization, and remote monitoring of vehicles and prevent the illegal use of them to guarantee their safety.

Key words: vehicle management; security authentication; face recognition; remote authorization; remote monitoring

① 收稿时间: 2021-02-22; 修改时间: 2021-03-18, 2021-04-06; 采用时间: 2021-04-13

依据军车使用正规化管理要求以及军用车辆本身的特殊性质, 不仅要在使用准入环节进行批准授权, 而且还要能够实时掌握使用过程环节中的相关信息, 如位置信息、驾驶员信息等. 目前, 大多数部队单位对于车辆的使用管理仍旧采用人工管理的传统方法, 批准授权效率较低, 且无法实时掌握车辆使用过程中的情况. 为此, 需要设计一种新的车辆使用授权及监控系统.

目前, 对于车辆的使用管理和监控问题已经进行了很多研究. 例如, 文献 [1] 从驾驶员、车辆、调度、服务、安全等五大基本元素入手, 阐述了规范公务车辆日常使用与管理的措施和途径. 文献 [2] 从探索公车改革, 提升车辆的管理效益, 降低单位经济费用支出等方面分析了车辆的管理模式. 文献 [3] 设计了一套基于 ARM+4G+传感器架构的车载远程监控系统, 实现装备信息非接触采集与远程实时传输. 文献 [4] 提出了一种具有实时安全预警功能的车辆远程监控系统, 包括车载数据采集端、手机客户端及监控中心, 主要是对车辆行驶安全进行监控, 未对使用授权进行管理. 文献 [5] 基于物联网技术和 OBD2 远程监控车辆的排放和运行参数, 主要是对汽车状态进行管理, 未对驾驶员情况进行监控.

另外, 随着科技的创新驱动, 人工智能技术的普及推广为现代生活带来了极大的便利, 人工智能改变了人们的生活习惯并逐渐发展成为一种极其重要的工程技术. 若将人工智能运用到车辆使用安全管理中, 则会极大的提高管理效率.

综上所述, 为了实现对军用车辆远程授权和远程监控, 本文设计了一款基于人工智能的军用车辆驾驶远程授权及监控系统. 该系统采用人脸识别技术、网络技术以及远程连接授权等计算机技术[6], 能够实现车辆的自动化认证、远程授权和远程监控, 进而防止车辆的不正当使用行为, 有力保障车辆管理安全.

1 系统总体设计

1.1 系统功能结构

本文设计的军用车辆驾驶远程授权及监控系统由 3 个终端组成, 即车辆终端、服务器端以及移动客户端, 如图 1、图 2 所示.

(1) 车辆端

车辆端包括 5 个功能模块: 驾驶员人脸识别、车辆报警、远程传输、车辆定位以及实时抓拍. 其硬件主要采用英伟达 NVIDIA Jeston Nano AI 人工智能开发板, 并搭载摄像头、报警器、GPS 定位模块、无线模块以及显示器. 软件则是通过 Python 语言编程实现人脸识别和远程传输. 通过这 3 个模块的软硬件建设, 可以实现车辆安全认证系统在车辆终端的管理以及使用.

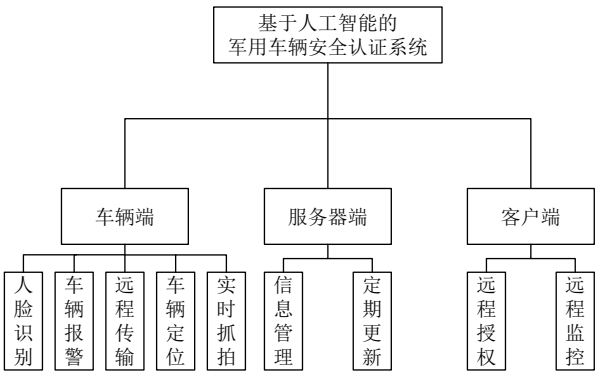

图 1 系统功能结构图

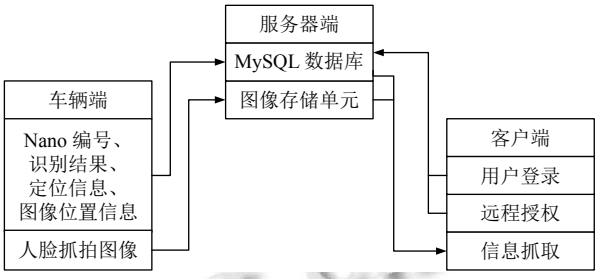

图 2 系统各端结构设计图

人脸采集: 对于某车辆来说, 其具有专属驾驶员集合, 为了能够保证该集合中驾驶员具有驾驶车辆的权限, 必须首先要采集驾驶员的人脸信息集, 形成人脸特征数据包, 作为后续人脸识别的比照对象集[7,8]. 预置的人脸特征数据包即为合法驾驶员在光线条件较好环境下的多张多角度无遮挡拍摄的正式照片.

人脸识别: 即利用人脸识别技术[9], 对驾驶员进行识别. 当驾驶员进入驾驶室后, 人脸自动识别系统首先进行预处理, 即对人脸进行检测与定位, 从输入图像中找到人脸存在的位置, 并将人脸从整体图像中分割出来, 然后提取人脸图像的特征, 最后与预置的人脸特征数据包比对, 确认是否有相匹配的人脸图像, 最后输出判断结果[10].

实时抓拍: 该终端能够在车辆驾驶途中对驾驶员进行检测. 其主要通过车辆端的摄像头对驾驶员脸部

进行抓拍, 并将抓拍到的图像远程传输至服务端, 进一步转发至客户端, 以此来实现对车辆驾驶员的动态监测.

车辆定位: 该终端能够实现车辆行驶位置信息的数据采集. 在执行任务时, 通过车辆位置信息实时掌握车辆动态, 能够有效监测任务执行情况, 防止驾驶员在执行任务期间偏离预定路线, 便于对车辆的严格管理.

远程传输: 当人脸识别或抓拍成功后, 将车辆端输出的信息通过 HTTP 协议远程传输至服务器端, 等待服务器端转发客户端的授权与否的指示.

车辆报警: 当收到拒绝授权信号后, 连接在车辆终端的蜂鸣器发出报警, 同时控制车辆启动电源禁止启动.

(2) 服务器端

服务器端的主要功能是管理车辆端和客户端基本信息、汇总车辆端采集上传的车辆信息数据, 并在车辆端与车辆管理者客户端之间建立数据连接.

(3) 客户端

移动客户端包括两个主要功能模块: 远程授权、远程监控. 其硬件要求为移动智能手机或平板终端. 软件则是基于 H5+Bootstrap+Spring Boot 开发的网页版 APP[11], 可运行于 Android 和 iOS 系统.

远程授权: 在服务端已经建立了客户端与车辆端之间的对应关系. 客户端登录后会看到所管理的所有车辆端信息, 可以接受车辆端所发起的授权申请. 在接到申请授权请求后, 管理人员根据现场情况, 判断是否符合驾驶任务情况, 从而做出同意授权或者拒绝授权的处理.

远程监控: 客户端可以随时向服务器端请求车辆当前信息, 包括车辆定位信息、驾驶员人脸抓拍信息等, 实现实时监控车辆和驾驶员的功能.

1.2 系统数据交互

系统整体数据的传输流程如图 3 所示.

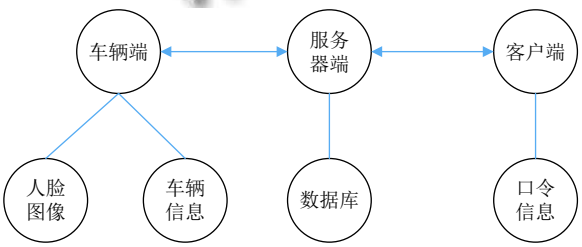

图 3 系统数据传输流程图

数据传输过程描述:

(1) 车辆端软硬件安装于车辆上, 进行驾驶员人脸识别或抓拍, 将结果通过服务端提供的 API 远程上传至服务端.

(2) 服务器端除了管理维护车辆端和客户端基本信息外, 还作为车辆端和客户端之间的数据汇总和中转站, 负责接收车辆端识别或抓拍信息, 并转发至客户端.

(3) 客户端从服务器获取识别或抓拍信息后, 进行人为判断, 进而给出授权或抓拍指令, 并上传至服务器端.

(4) 服务器端响应客户端授权或抓拍操作指令, 进一步将该指令传输至车辆端.

(5) 车辆端收到服务器中转过来的操作指令后, 进行确认授权、拒绝使用或抓拍操作.

具体实施步骤如图 4 所示.

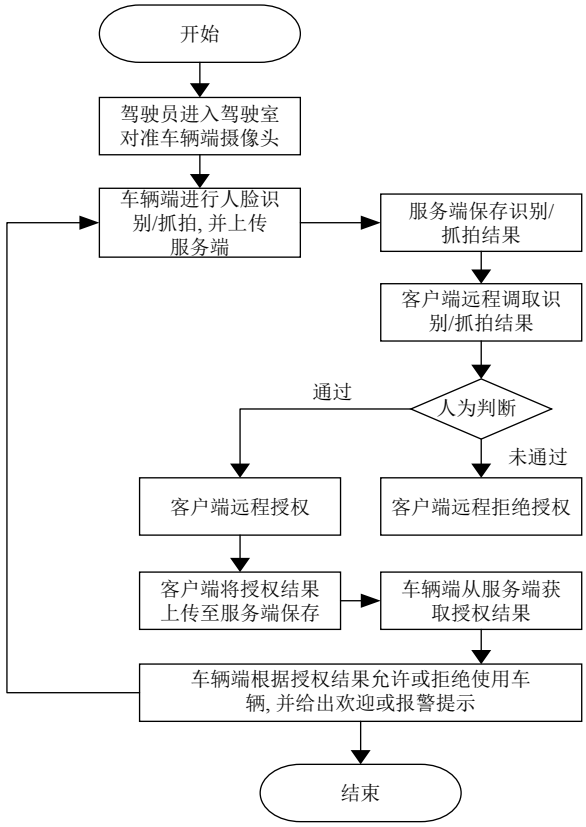

图 4 安全认证系统流程图

图 4 中, 在安全认证系统启动前, 还有一个驾驶人员注册的流程. 其具体操作为: 摄像头抓拍驾驶者的面部图像, 经面部特征提取算法得到人脸特征的信息, 与驾驶者的姓名信息一起写入车辆终端的 SD 卡内. 这时, 注册后的人员就具有了安全认证的标识. 当驾驶员

进入驾驶室启动车辆的时候, 车辆端的摄像头会抓拍当前人员的面部图像并进行人脸特征的提取. 根据车辆端获取的人脸特征结合 SD 卡内保存的特征模块进行人脸识别.

2 服务端模块设计

服务端部署于云服务器上, 操作系统使用 Windows Server 2012, 主要完成管理车辆端和客户端基本信息、汇总车辆端采集上传的车辆信息数据, 并在车辆端与车辆管理者客户端之间建立数据连接.

2.1 数据库设计

(1) 实体属性关系 (E-R) 图

① 客户端信息

维护了客户端的基本信息, 用于对客户端用户登录进行身份认证, 数据项内容为: id、客户编号、客户名称、性别、电话号码、地址、密码、备注. 其中, 客户编号默认采用其手机号码; 密码存储的内容经过 MD5 进行加密. 实体-属性图如图 5 所示.

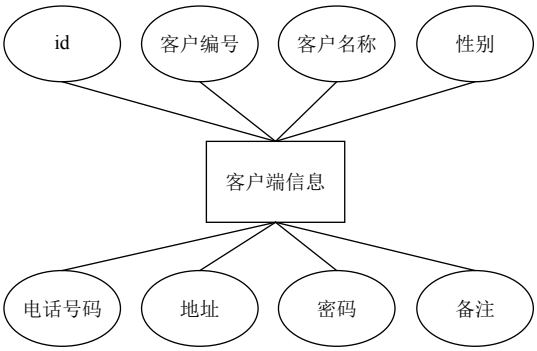

图 5 客户端信息实体-属性图

② 双端匹配信息

维护了车辆端 Nano 基本信息、客户端与车辆端 Nano 之间的对应关系, 用于将客户端与车辆端进行匹配, 数据项内容为 id、Nano 编号、客户编号、是否启用、备注 (存储当前 Nano 所部署车辆的车牌号). 其中, 客户编号来自于客户端信息表; 是否启用标记当前 Nano 是否可用 (若不可用则无法提供车辆端功能). 实体-属性如图 6 所示.

③ 识别操作结果

用于汇总车辆端识别/抓拍结果信息, 记录当前客户端对车辆端的操作指令等, 数据项内容包括 id、Nano 编号、日期、识别结果、图像存储位置、定位信息、授权指令、抓拍指令等内容. 其中, 识别结果为车辆端进行人脸识别后的结果; 图像存储位置是车辆端抓取的当前驾驶员图像上传服务端后保存路径; 定位信息存储车辆端上传的当前车辆所处的经度和纬度; 授权指令记录客户端是否对车辆端进行授权使用与否 (1 表示允许; 0 表示不允许; −1 表示未下达指令); 抓拍指令记录客户端是否要让车辆端提供抓拍服务 (1 表示抓拍; 0 表示不抓拍). 实体-属性图如图 7 所示.

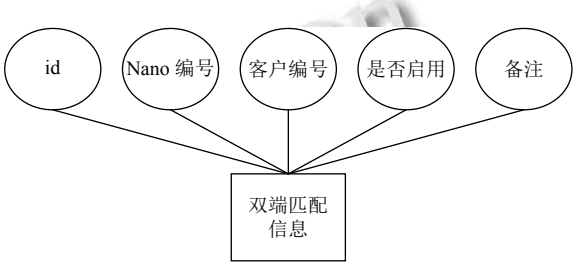

图 6 双端匹配信息实体-属性图

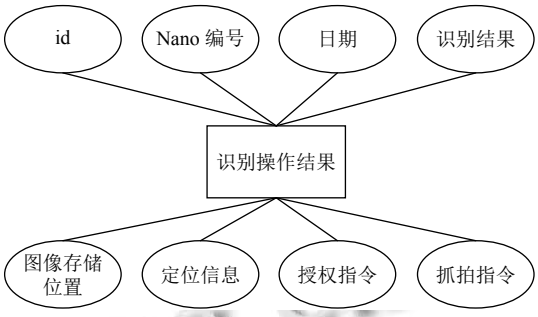

图 7 识别操作结果实体-属性图

通过对以上 3 个实体进行整合, 形成总 E-R 图如图 8 所示 (省略了实体属性):

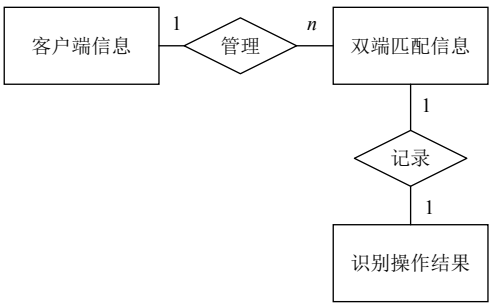

图 8 服务端系统 E-R 图

其中, 一个客户端可以管理多个车辆端的 Nano; 每个车辆端的 Nano 在识别操作结果表中只有唯一一行记录.

(2) 数据表结构设计

根据上述的 E-R 图, 数据库中的表设计如图 9–图 11 所示.

名	类型	长度	小数点	不是 null	
id	int	11	0	☑	🔑1
nanoid	varchar	255	0	☐	
rq	datetime	0	0	☐	
drivername	varchar	255	0	☐	
picpath	varchar	255	0	☐	
latitude	varchar	255	0	☐	
longitude	varchar	255	0	☐	
canuse	varchar	255	0	☐	
catchpic	varchar	255	0	☐	

图 9 识别操作结果表

名	类型	长度	小数点	不是 null	
id	int	11	0	☑	🔑1
khbh	varchar	255	0	☐	
khmc	varchar	255	0	☐	
xb	varchar	255	0	☐	
dhhm	varchar	255	0	☐	
dz	varchar	255	0	☐	
mm	varchar	255	0	☐	
bz	varchar	255	0	☐	

图 10 客户端信息表

名	类型	长度	小数点	不是 null	
id	int	11	0	☑	🔑1
nanoid	varchar	255	0	☐	
khbh	varchar	255	0	☐	
khmc	varchar	255	0	☐	
inuse	varchar	255	0	☐	
bz	varchar	255	0	☐	

图 11 双端匹配信息表

2.2 接口设计

服务端其中一个主要功能是汇总车辆端识别/抓拍信息, 为此必须提供相应的接口.

(1) 车辆端上传识别/抓拍信息至服务端

功能说明: 车辆端通过调用该接口, 将识别或抓拍的图像、识别结果、经纬度等信息上传至服务端进行汇总保存. 其中人脸图像保存至服务端磁盘固定位置, 而识别结果、经纬度和图像保存路径则直接保存至数据库“识别操作结果表”中.

请求地址: http://IP:8000/CarDriverCertification/CheckResult/SaveCZDResults.

请求参数: file-存放上传的图片; nanoID-存放 Nano 编号; name-存放识别结果; longitude-存放车辆端经度信息; latitude-存放车辆端纬度信息.

返回结果: 成功-success; 失败-failure.

(2) 车辆端获取服务端指令信息

功能说明: 车辆端通过调用该接口, 可以获取服务端当前需要车辆端执行的指令.

请求地址: http://IP:8000/nanoAdmin/GetWhatToDo.

请求参数: nanoID-要获取哪个 Nano 编号的指令信息.

返回结果: canuse-获取当前车辆端是否被授权使用 (0: 拒绝使用; 1: 允许使用); catchpic-是否执行抓拍 (0: 不抓拍; 1: 抓拍).

(3) 车辆端获取服务端指令信息

功能说明: 当车辆端执行完毕抓拍后, 可以通过调用该接口, 将服务端的抓拍指令恢复至不抓拍状态, 避免车辆端重复抓拍.

请求地址: http://IP:8000/nanoAdmin/cancleZP.

请求参数: nanoID-要取消哪个 Nano 编号的抓拍指令信息.

返回结果: 成功-success; 失败-failure.

2.3 系统设计

服务端系统是基于 H5+Spring Boot+MySQL 框架开发的 Maven 工程, 开发工具使用 STS, JDK 版本为 jdk1.8.0_161, 浏览器建议使用 Firefox 或 Google Chrome, 应用服务器采用 Apache-Tomcat-8.5.29. 该工程框架结构如图 12 所示.

(1) web-admin

该模块包含了系统所有的网页页面、JS 文件以及负责业务逻辑控制的 Controler, 其中页面均基于 thymeleaf 模板.

(2) web-bean

该模块主要包括系统运行所需的实体类, 如 CheckResult.java 对应的是数据库中的识别操作结果表; kehu.java 对应的是客户端信息表; Nano.java 对应的是双端匹配信息表. 另外还有系统运行所必需的其它非业务实体类.

(3) web-core

该模块是整个系统的核心, 包含了系统运行所必需的核心内容, 如系统配置、工具类、查询类等, 其它模块均基于该核心进行运作.

(4) web-dao

该模块用于采用建立一些操作数据库的接口, 该接口均继承了 JpaRepository, 提供了一些基于 JPA 访问数据库的方式.

(5) web-service

该模块提供了对外访问数据库的服务, 包括接口和实现. 可以直接基于在 web-dao 中建立的数据库接口具体执行数据库操作. 另外, 该模块在操作数据库时可以将每个操作函数与缓存 Cache 建立联系. 等下次再次执行该语句获取相同 id 的记录时, 系统无需再去数据库中查询, 而是直接从缓存中取出即可, 提高了运行效率.

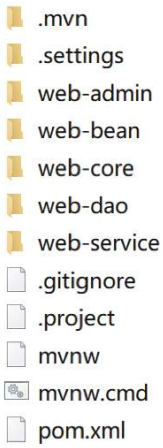

图 12 服务端系统框架

3 车辆端模块设计

车辆端系统部署于车辆上, 软件利用 Python 语言进行开发, 完成人脸采集、人脸识别、实时抓拍、车辆定位、远程传输、车辆报警等功能.

3.1 硬件设计

车辆端的硬件设计上需要考虑到以下几点: 第一, 低功耗. 由于驾驶途中需要一直处于工作状态, 所以我们需要为车辆终端提供稳定电源支持其长时间的工作; 第二, 体积小. 由于驾驶室封闭且环境狭窄的原因, 要求车辆终端足够小巧便于安装拆卸且不影响驾驶员进行正常驾驶. 第三, 要能够实现对人员位置实时掌握, 保持数据传输通畅. 最后要在军队中大规模的推广, 因此要求各元器件的成本要低. 基于以上几个方面, 最终车辆终端的硬件结构按模块组成如下: 采用 NVIDIA Jetson Nano 作为核心开发板, 无线模块采用迅捷 FW150、定位模块采用北斗定位模块、摄像头选用 Nano 专用摄像头、音频模块选用 USB 音响, 总体硬件结构如图 13 所示.

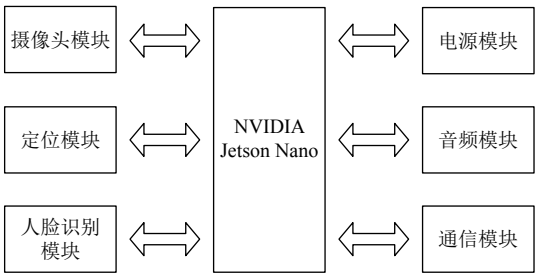

图 13 系统硬件模块设计图

系统硬件实物图如图 14 所示.

图 14 系统硬件结构图

3.2 系统设计

(1) 人脸采集

人脸采集模块目的是形成针对某车辆的合法用户的人脸特征数据包. 在采集时, 基于 OpenCV 对人脸进行检测[12], 并将已检出的人脸突出出来, 转化为灰度图像以此减少计算机所需要处理的计算量. 针对每一个合法驾驶员可采集多张不同角度的图像, 以增加后续人脸识别的准确性. 如图 15 所示.

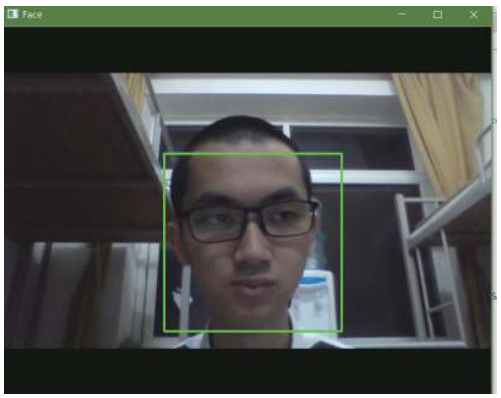

图 15 人脸采集结果图

每个驾驶员的多张人脸图像集中放置于以其“编号-姓名”为命名的文件夹中, 如图 16、图 17 所示.

图 16 合法驾驶员人脸特征数据包

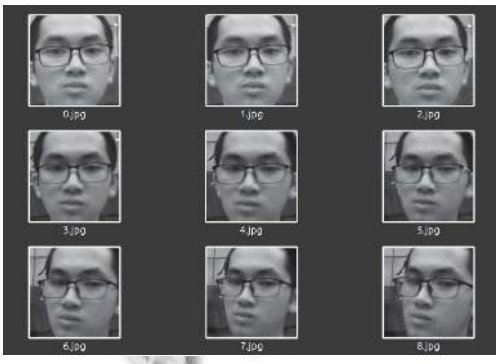

图 17 某一合法驾驶员人脸特征数据

(2) 人脸识别

在正常使用系统时, 需要对驾驶员进行人脸识别, 以判断是否为合法驾驶员.

本系统采用了卷积神经网络 (CNN) 用于建立人脸识别模型, 在模型中我们使用了 Keras 所提供的处理多分类问题的 categorical_crossentropy 损失函数. 根据需要设计各神经网络层并按顺序添加至模型中, 模型建立如图 18 所示.

模型建立完毕后, 根据人脸样本的设计方案, 采用了 LFW 人脸数据集进行模型训练. 模型训练完毕后, 会形成 aggregate.face.model.h5 模型文件, 供后续识别人脸调用. 执行识别时, 将训练好的模型加载出来, 通过指定的摄像头设备实时捕获视频流, 并根据人脸识别分类器采用循环检测的方式, 读取视频流的每一帧图像, 将彩色图像转化为灰度图像以降低数据处理的复杂度. 并将灰度图像中所识别的人脸图像作为输入传递到模型中去识别. 最后将识别好的人脸信息以及人脸图片存储至指定文件夹中, 并调用服务端提供的 SaveCZDResults 接口上传相应的信息至服务端.

(3) 远程授权/拒绝及报警

车辆端上传识别结果信息后, 等待客户端进一步指令, 以判断是否允许使用车辆. 车辆端周期性调用服务端的 GetWhatToDo 接口, 判断其中的 canuse 状态, 若为 0 则拒绝使用, 并调用报警器进行蜂鸣报警; 若为 1 则允许使用车辆, 给出欢迎使用提示.

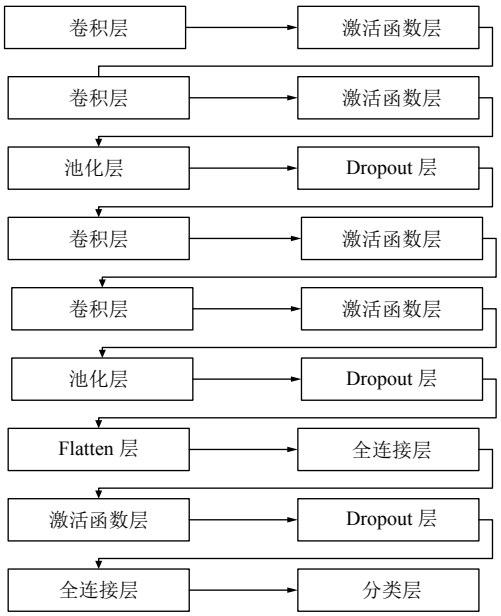

图 18 模型层次设计图

(4) 实时抓拍

车辆端必须能够响应客户端实时抓拍的功能. 车辆端周期性的调用服务端的 GetWhatToDo 接口, 判断其中的 catchpic 状态, 若为 0 则不执行抓拍; 若为 1 则执行抓拍, 会对当前驾驶员图像进行抓拍, 并将结果上传至服务端, 进一步供客户端获取查看使用.

(5) 车辆定位

车辆定位信息会同识别结果或抓拍信息一起上传服务端, 存储在识别操作结果数据表中.

(6) 数据远程传输

在 Python 中调用服务端 URL 接口需要使用到 requests 库. Requests 库是一个用于 HTTP 请求的模块, 性质和 urllib 是一样的, 作用就是向指定的网站后台服务器发起请求, 并接收服务器端的响应和返回的内容. 在 requests 库中又包含了 get、post 等多种方法, 不同的方法表示使用 HTTP 协议中的请求方式不同, 但其最终目的都是对目标网站发起请求, 将本地的数据上传至服务器. 此处需要注意的是: 服务端的 SaveCZDResults 接口是同时上传图像文件和普通参数, 需要使用 MultipartFile 格式.

4 客户端模块设计

客户端部署于智能移动设备上, 如平板和手机等, 主要完成对远程驾驶员信息查看、判断, 下达授权和抓拍等指令.

4.1 数据库设计

客户端数据库与服务端数据库位于同一个数据库, 详见 2.1 节.

4.2 接口设计

(1) 验证客户端账号信息

功能说明: 当客户端登录时, 对其进行账号验证.

请求地址: http://IP:8000/CLSQ/check.

请求参数: khbh-客户编号 (此处以客户的手机号作为其编号); mm-客户密码 (经过 MD5 加密).

返回结果: 验证成功-success; 验证失败-failure; 客户不存在-nonexistent.

(2) 获取客户端对应的 Nano 列表

功能说明: 一个客户端可以管理多个车辆端的 Nano, 通过此接口可以获取 Nano 列表.

请求地址: http://IP:8000/nanoAdmin/selectFill_Nano.

请求参数: khbh-客户编号 (此处以客户的手机号作为其编号).

返回结果: 成功-对应的 Nano 列表; 失败-failure.

(3) 某个车辆端 Nano 详细的识别操作结果信息

功能说明: 客户端选择某个车辆端的 Nano 后, 通过此接口可以获取该 Nano 的识别操作结果信息.

请求地址: http://IP:8000/nanoAdmin/getSelectData_Nano.

请求参数: khbh-客户编号 (此处以客户的手机号作为其编号); nanoInfo-所选择 Nano 的信息 (格式为: nanoid+"@#&"+车辆端车牌号).

返回结果: 成功-对应的 Nano 识别操作结果信息; 失败-failure.

(4) 对某车辆端 Nano 进行授权

功能说明: 客户端选择某个车辆端的 Nano 后, 通过此接口可以对该 Nano 进行远程授权, 即将识别操作结果表中的 canuse 设置为 1.

请求地址: http://IP:8000/nanoAdmin/sq.

请求参数: id-该 Nano 对应识别操作结果表中的 id 行号.

返回结果: 成功-将对应的 Nano 的 canuse 设置为 1; 失败-failure.

(5) 对某车辆端 Nano 进行拒绝授权

功能说明: 客户端选择某个车辆端的 Nano 后, 通过此接口可以对该 Nano 进行远程拒绝授权, 即将识别操作结果表中的 canuse 设置为 0.

请求地址: http://IP:8000/nanoAdmin/jj.

请求参数: id-该 Nano 对应识别操作结果表中的 id 行号.

返回结果: 成功-将对应的 Nano 的 canuse 设置为 0; 失败-failure.

(6) 令某车辆端 Nano 进行抓拍图像

功能说明: 客户端选择某个车辆端的 Nano 后, 通过此接口可以令该 Nano 进行远程抓拍驾驶员图像, 即将识别操作结果表中的 catchpic 设置为 1.

请求地址: http://IP:8000/nanoAdmin/zp.

请求参数: id-该 Nano 对应识别操作结果表中的 id 行号.

返回结果: 成功-将对应的 Nano 的 catchpic 设置为 1; 失败-failure.

(7) 对某车辆端 Nano 的信息进行刷新

功能说明: 车辆端执行完客户端命令后, 执行状态和图像会发生改变, 可以通过此接口刷新当前页面的 Nano 信息.

请求地址: http://IP:8000/nanoAdmin/sx.

请求参数: id-该 Nano 对应识别操作结果表中的 id 行号.

返回结果: 成功-刷新对应的 Nano 的信息; 失败-failure.

4.3 系统设计

客户端系统是基于 H5+BootStrap+Spring Boot+MySQL 框架开发的网页版 APP, 开发工具使用 STS, JDK 版本为 jdk1.8.0_161, 浏览器建议使用 Firefox 或 Google Chrome, 应用服务器采用 Apache-Tomcat-8.5.29.

5 系统实现

(1) 车辆端测试

首先, 在启动安全认证系统前录入合法驾驶员的人脸信息. 选择光线条件较好的位置根据提示进行面部信息录入. 如图 19 所示.

将图片转换为均等且合适大小. 然后将它们读取

到数组中. 图片读取完毕后, 人脸识别模型通过卷积神经网络进行训练. 将训练结束后的模型保存至指定目录下, 以便识别时调用. 如图 20 所示.

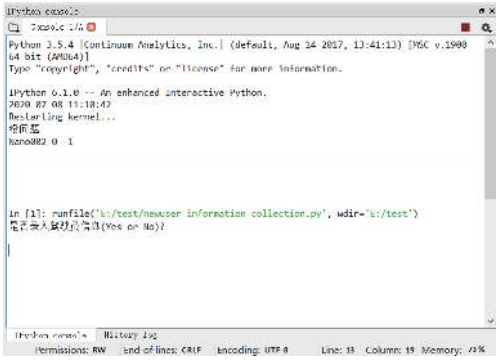

图 19 信息录入界面

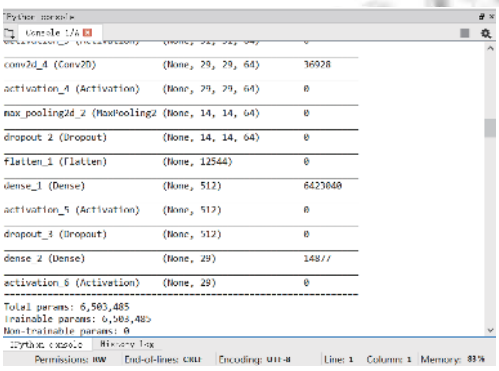

图 20 模型训练过程

驾驶员使用车辆前, 车辆端会通过摄像头抓取驾驶员图像, 并自动调用人脸识别程序, 将识别结果及捕捉图像保存至本地并上传至服务端. 在驾驶员使用车辆过程中, 车辆端会通过摄像头自动抓拍驾驶员图像, 并上传至服务端. 如图 21 所示.

(2) 服务器端测试

在 Tomcat 中启动服务端 Maven 工程, 进入网址 (http://localhost:8000/login), 然后使用管理员账户登录服务器. 如图 22 所示.

登录后进入车辆识别结果界面可以根据 NanoID 查询车辆启动时间、车辆端识别结果、驾驶员人脸图像以及客户端的授权与抓拍信息, 如图 23 所示.

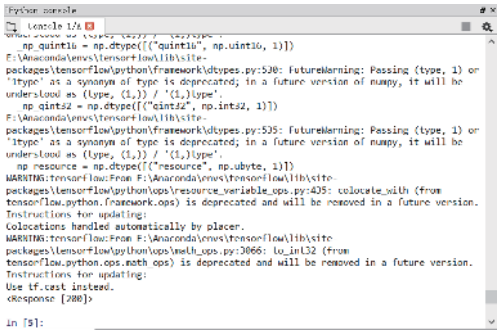

图 21 识别结果及抓捕图像上传

图 22 服务器登录界面

	ID	日期	Nano编号	识别姓名	图像	经度	纬度	授权	抓拍
	3	20-07-01 16:47	Nano002	mxx		116.397128	39.916527	-1	0
	2	20-07-01 16:51	Nano001	zh		121.26757	37.49794	-1	0

图 23 车辆识别结果界面

在左侧系统管理中可以进行客户端客户账号管理、车辆端 Nano 管理以及服务器端用户管理 (可配置权限和角色), 如图 24–图 26 所示, 图中信息均为虚拟数据, 仅用于系统测试.

(3) 客户端测试

移动客户端只需在智能终端自带的浏览器中输入指定网址 (http://localhost:8000/CLSQClient/CLSQLogin), 进入客户端登陆界面. 根据在服务端注册的账户登录客户端, 如图 27 所示.

进入客户端后选择对应的 Nano 编号或车牌号可以查看使用情况, 使用情况包括 Nano 编号、识别日期、识别结果、使用授权以及是否抓拍这几项. 客户可根据实际情况在客户端发送进行授权以及抓拍等操作命令, 如图 28 所示.

6 结论与展望

本文设计的车辆驾驶远程授权及监控系统采用时下十分热门的人脸识别技术, 结合人工智能深度学习的特点提高认证的准确性. 目前部队的车辆管理大都采用人工管理的传统方法, 在车辆管理难度大、安全事故突发性强的背景下, 人工操作自然会带来诸多弊端. 然而, 随着军队的加速发展, 车辆类型和数量不断增加, 车辆管理难度也不断攀升. 本文针对车辆管理中存在的众多隐患, 以及人工管理的不当行为与不足之处进行分析, 完成了车辆安全认证系统的方案设计. 该系统利用远程传输、人脸识别与深度学习技术, 不仅能够实现对驾驶员身份认证, 也可以从远程客户端实时掌握车辆驾驶情况.

图 24 客户端用户管理界面

图 25 Nano 端管理界面

图 26 服务器端管理界面

图 27 客户端登录界面

图 28 客户端控制界面

参考文献

1 万正国. 浅谈公务车辆使用与管理. 汽车实用技术, 2014, (5): 128–130. [doi: 10.3969/j.issn.1671-7988.2014.05.033]

2 周蔚洪. 创新车辆管理模式 提升车辆管理效率. 中小企业管理与科技, 2015, (10): 45. [doi: 10.3969/j.issn.1673-1069.2015.10.031]

3 江鹏程, 李志浩, 齐晓辉. 4G 网络装备车辆远程监控系统. 兵工自动化, 2020, 39(2): 24–27.

4 李一龙, 卢军. 车辆远程监控系统的设计与实现. 网络安全技术与应用, 2014, 5: 39–40. [doi: 10.3969/j.issn.1009-6833.2014.02.026]

5 廖燕辉. 基于 OBD2 的车辆远程监控系统. 时代汽车, 2018, (4): 28–29. [doi: 10.3969/j.issn.1672-9668.2018.04.011]

6 战岳祥. 全方位安全认证系统的研发 [硕士学位论文]. 长春: 吉林大学, 2008.

7 王宇航. 基于面模特征的嵌入式安全认证终端的设计与实现 [硕士学位论文]. 长春: 吉林大学, 2016.

8 Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798–1828. [doi: 10.1109/TPAMI.2013.50]

9 Dong Y, Li D. Deep-learning and its applications to signal and information processing. Signal Processing Magazine, 2011, 28(1): 145–154. [doi: 10.1109/MSP.2010.939038]

10 王天庆. Python 人脸识别: 从入门到工程实践. 北京: 机械工业出版社, 2019.

11 邓笑. 基于 Spring Boot 的校园轻博客系统的设计与实现 [硕士学位论文]. 武汉: 华中科技大学, 2018.

12 钟官长. 基于 OpenCV 的人脸识别算法研究与实现 [硕士学位论文]. 南昌: 江西师范大学, 2015.