

基于树莓派的人脸识别考勤系统的开发与实现

李玉鹏, 宋维, 程超伟

(武汉东湖学院 电子信息工程学院, 武汉 430212)

摘要: 提出一种基于树莓派的人脸识别考勤系统, 该系统通过使用人脸识别云服务可实时地对人员进行考勤。文中详细描述了该系统的搭建和实现过程, 为相关产品提供了一套很好的解决方案。对所设计的系统进行模拟测试, 其测试结果达到了设计目标。

关键词: 树莓派; 人脸识别; 考勤系统; 云服务; Python

中图分类号: TP11

文献标识码: A

Attendance System of Face Recognition Based on Raspberry Pi

Li Yupeng, Song Wei, Cheng Chaowei

(College of Electronic Information Engineering, Wuhan Donghu University, WuHan 430212, China)

Abstract: An attendance system of face recognition based on Raspberry Pi is proposed, which can realize the attendance using the face recognition cloud service. The construction and implementation process of the system are introduced, it provides a good solution for the related products. The test of the system is simulated, and the results achieves the design goal.

Key words: Raspberry Pi; face recognition; attendance system; cloud service; Python

引言

人脸识别^[1]一度被认为是生物特征识别领域甚至人工智能领域最困难的研究课题之一, 但随着科学技术的发展, 一系列关于人脸识别的难题相继被攻克, 人脸识别技术已被广泛应用于公安刑侦破案、门禁系统、摄像头监视系统、网络应用、身份辨别、支付系统等与国计民生密切相关领域。

云计算^[2]的兴起, 为人脸识别技术的广泛应用铺平了道路, 各大互联网企业相继开发了人脸识别技术云服务, 提供了人脸识别的一整套技术方案, 开发人员可以方便高效地使用其提供的服务接口对系统进行开发和调试。本文利用人脸识别云服务技术的特点, 基于树莓派^[3]开发了一个基于人脸识别云服务的考勤系统。

1 系统总体方案

基于树莓派的人脸识别考勤系统是将树莓派采集到的实时待考勤人员的脸部图像发送至云端服务, 待其处理完毕后再从云端传回考勤结果, 并将考勤人员的出勤结果录入数据库进行保存以备查询, 管理员通过登陆树莓派即可实现人机交互, 在数据库中查询人员的出勤情况, 因此, 本系统包括本地的树莓派终端和云服务部分。

为适应在树莓派上进行测试及实际应用, 本文采用编程语言 Python2.7^[4]进行系统的开发和设计, Python语言的高可读性以及面向对象的特性很适合本系统的开发, 同时人脸识别云服务也提供了基于 Python 的 SDK。本系统总体结构图如图 1 所示, 图中的本地终端为树莓派, 云服务采用了腾讯优图人脸识别云服务^[5]。

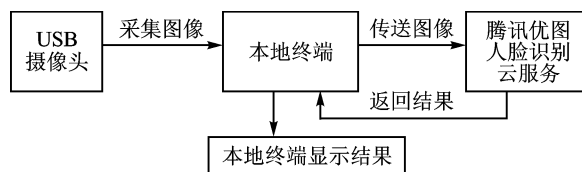


图 1 系统总体结构图

2 系统主要功能实现

本系统的设计具有全自动考勤的特点, 待考勤人员只需将其脸部适当的靠近考勤机摄像处, 系统将自动检测并锁定人脸, 即刻对其进行摄像并将图像发送至云端服务进行处理, 待处理完成后, 管理人员可通过树莓派终端或者通过远程 PC 登陆 MySQL 数据库, 查看人员的出勤情况, 获悉考勤结果。为便于读者的各种终端对本系统进行测试和实际应用, 开发的基于人脸识别云服务的软件完全可移植, 即直接可在各种终端平台上运行不需做其他改动。

系统的实时性是用户体验的关键要素,系统正常工作时考勤人员感觉不到卡顿状况,实时性良好。

2.1 人脸检测功能的实现

在人脸检测环节,用户无需进行其他操作,系统通过摄像头自动捕捉屏幕中出现的人脸,在成功捕捉到人脸后,通过摄像头进行拍照保存。以上过程通过在 Python 中使用 SimpleCV^[6] 库的函数 findHaarFeatures 来实现,这是一个在图像中搜索匹配某一特定模式的算法,在 SimpleCV 库中自带了一些模式,如脸、鼻子、眼睛、身体等。findHaarFeature 可以分析摄像头捕捉到的图像,并从图像中匹配出对应的模式,然后返回匹配的部分在图像中的坐标位置,使用此函数可以匹配图像中的人。首先,利用 SimpleCV 库中的 Camera 函数设置图像尺寸;其次,通过循环语句不断执行 faces = frame.findHaarFeatures('face') 来捕捉用户在考勤时的图像,当摄像头捕捉到人脸后,执行 frame.save() 就可将检测到的人脸图像保存在树莓派本地存储空间中备用。如果摄像头没有捕捉到人脸,系统将处于待工作状态,直至检测到人脸,以上实现了本系统的自动触发的功能。在树莓派上连接好 USB 摄像头后,测试结果如图 2 所示。需要说明的是,读者需了解人脸检测和人脸识别概念的区别。

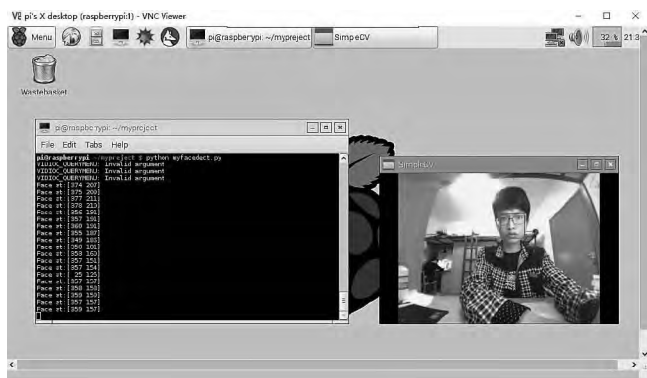


图 2 人脸检测功能的实现

2.2 腾讯云优图人脸识别云服务介绍

人脸识别云服务的识别率及实时性是本系统所必须考虑的问题。本文采用了腾讯云优图团队提供的人脸识别服务。腾讯云优图人脸识别是由腾讯云优图团队推出的生物识别技术,腾讯云优图团队立足腾讯社交数据大平台,拥有海量数据分析与人脸训练集,累计处理人脸已达数亿,积累数千万人脸身份。

腾讯云优图人脸识别 1:N 人脸检索技术,即输入一张人脸照片,可以从训练时产生的特定的特征集数据库中找出与之相似的候选人脸,可广泛应用在考勤、通关、门禁、签到、黑名单等场景,可针对不同人脸检索规模,为客

户提供稳定、方便、快捷的人脸检索解决方案。同时,该技术注重解决实际应用中的问题,包括人脸姿态、人脸光照、人脸遮挡、人脸年龄变化,通过海量真实场景的训练数据,得到世界领先的人脸模型。在目前公开的 LWF 测试中,人脸验证可达到 99.5% 的准确率。

2.3 本地终端与人脸识别云服务的交互实现

树莓派与人脸识别云服务的交互是本文设计的核心,腾讯优图团队为方便广大开发者使用其技术服务,开发了基于各种平台的 SDK,下面对其提供的基于 Python 的 SDK 的 API 进行调用以实现交互功能,主要包括服务鉴权签名、训练及识别过程。

首先是鉴权签名,SDK 的初始化过程中,需要使用项目 id 和密钥获取鉴权签名,否则将无法使用上述服务,人脸识别云服务通过签名来验证请求的合法性。登录腾讯云项目密钥管理页面新建密钥,选择需要生成密钥的项目即可产生签名所需的 SecretId, SecretKey。编程时使用函数 TencentYoutuYun.YouTu(appid, secret_id, secret_key, userid, end_point) 来完成签名工作,其中的参数 appid 为业务中的应用标识 AppId, secret_id 为密钥 SecretId, secret_key 为密钥 SecretKey, userid 为用户 id, end_point 服务后台路径。上述参数可在腾讯云服务开发平台的页面中获得。

接下来是训练过程,即分类器的设计,此过程主要生成可用于识别的参数。腾讯云优图提供的 SDK 的训练过程包括新建个体和增加人脸。首先新建个体,使用接口 NewPerson(self, person_id, image_path, group_ids, person_name="", tag="", data_type=0) 来完成,其中参数 person_id 为新建的个体 id,由用户指定,需要保证 app_id 下的唯一性; person_name 为个体对应的姓名; group_ids 为数组类型,由用户指定(组默认创建)的个体存放的组 id,可以指定多个组 id; image_path 为包含个体人脸的图片路径; tag 为备注信息,用户自解释字段; data_type 用于表示 image_path 是图片还是 url, 0 代表图片, 1 代表 url。

考虑到降低系统的复杂度,避免使用数据库,本系统将 url 值设置为 0,直接将图片保存在本地后通过此函数进行上传。在后期的管理当中,有事需要对系统进行管理,比如删除个体或人脸等,此时接口函数 DelPerson(self, person_id) 和 DelFace(self, person_id, face_ids) 就可根据需要选用,其中参数 person_id 为待删除的个体 id, face_ids 为数组类型,为待删除的人脸 id。

因为在进行识别之前,必须先对人脸库进行训练,所以实际上训练和识别在服务器端是异步执行的,相比其他人脸识别云服务,腾讯云优图团队开发的服务可自动进行训练操作,因此不需要开发者再进行训练操作。

最后就是识别过程,即通过比对获得的未知人脸参数和训练所得的参数完成人脸的分类和判别,返回识别结果。需调用人脸识别接口 FaceIdentify(self, group_id, image_path, data_type = 0)将本地保存的待考勤人员的图像上传至云服务端,进行人脸的识别。其中参数 group_id 为识别的组 id, image_path 为待识别的图片路径, data_type 用于表示 image_path 是图片还是 url, 0 代表图片, 1 代表 url。同理, 本文将此函数的 url 值设置为 0。

接口调用统一返回 Json 格式的返回结果, 可以通过返回信息了解各个参数的具体意义。

2.4 考勤系统数据库的建立

本次开发中使用的数据库为 MySQL^[7], 用于管理考勤人员的出勤信息, 存入数据库的信息只有管理员用管理帐号登录后才能查询。下面详细介绍如何在树莓派中建立考勤系统的数据库。

首先需要在树莓派上安装 MySQL, 先使用管理员权限运行 sudo apt-get update 获得最新的软件包的列表, 再继续使用管理员权限运行 apt-get 获取最新的 MySQL 及 Python 的编程接口(之后用于数据库编程): sudo apt-get install mysql-server python-mysqldb, 安装过程中需要输入 root 管理员的密码, 该密码之后用于访问数据库系统。

在本地 MySQL 创建库“test”, 表“myfacetest”和 3 个列“id”、“name”“attend_time”, 分别表示考勤人员的员工号、姓名和出勤时间。详细的过程如下: 使用“MYSQL-uroot -p;”命令登陆本地 MySQL, “CREATE DATABASE test;”创建数据库“test”, “USE test;”选择数据库“test”, “CREATE TABLE myfacetest;”创建表“myfacetest”, “CREATE TABLE myfacetest (id int(10), name varchar(20), attend_time char(25));”在表中创建 3 个列“id”、“name”“attend_time”, 完成后查看建好的表“myfacetest”, 如图 3 所示。

```
mysql> USE test
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| myfacetest      |
+-----+
1 row in set (0.00 sec)

mysql> DESCRIBE myfacetest;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(10) | NO | PRI | NULL | |
| name  | varchar(20) | NO | | NULL | |
| attend_time | char(25) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

图 3 建立的表“myfacetest”

上述过程详细地描述了本系统考勤人员数据库的建立过程, 在本系统 Python 主程序中使用 MySQL 数据库

的 Python 编程接口“conn = MySQLdb. connect(host = 'localhost', port = 3306, user = 'root', passwd = 'XXXXXX', db = 'test')”、“cur = conn. cursor()”、“cur. execute()”、“conn. commit()”进行编程, 当人脸识别云服务通过 json 格式返回相关的信息后, 根据相应的考勤人员的相关信息进行匹配登记, 即可将相关考勤人员的出勤信息进行入库保存并且实时更新, 管理人员可通过树莓派终端或者通过远程 PC 登陆 MySQL 数据库, 查看人员的出勤情况。

3 功能测试

经过上述的设计和开发, 本文最终实现了设计的预期, 采用基于 UVC 协议的 USB 摄像头, 通过调用 SimpleCV 的相关算法, 实现了系统无需进行任何人工干预的自动促发。由于本文提出的基于人脸识别云服务的考勤系统, 其人脸识别服务并不在树莓派上进行, 所以极大地节省了本地终端的资源耗费, 本文采用的“树莓派 2B+”, 其拥有 32 位 ARM-Cortex-A7 架构的双核 CPU, 并且配备了各种常用外设, 如以太网端口及 USB 端口等, 配合摄像头, 实验表明本系统可流畅地运行在树莓派上, 达到了高实时性的需求。

首先, 需要说明的是, 读者需区分识别成功率与识别率概念上的区别, 识别成功率表征的是总测试次数中成功地识别出该人员身份所占的次数, 而识别率表征的是各人员脸部特征与训练时对相应人员提取到的脸部特征的相近程度。显然, 识别率决定了识别成功率, 一旦识别率达到系统所设定的阈值, 该人员的考勤即被判为成功。在一般考勤系统的设计中, 根据考勤场合的严格程度, 一般此阈值在 60%~99.9% 之间。为充分而可靠地体现人脸识别云服务的识别率, 本系统测试时通过分别对多个不同的人进行识别测试, 识别结果如图 4 所示, 可以看出, 对 10 人进行测试, 识别成功率为 100%。

```
mysql> select * from myfacetest;
+-----+-----+-----+
| id | name | attend_time |
+-----+-----+-----+
| 1 | name_a | 20160509.17.50 |
| 2 | name_b | 20160509.17.50 |
| 3 | name_c | 20160509.17.51 |
| 4 | name_d | 20160509.17.51 |
| 5 | name_e | 20160509.17.51 |
| 6 | name_f | 20160509.17.52 |
| 7 | name_g | 20160509.17.52 |
| 8 | name_h | 20160509.17.52 |
| 9 | name_i | 20160509.17.52 |
| 10 | name_j | 20160509.17.52 |
+-----+-----+-----+
10 rows in set (0.00 sec)
```

图 4 系统识别成功率测试结果

图 5 为本系统总体测试后服务器端返回的 10 位测试人员中其中一人的识别率及其他参数, 其余 9 人的识别率分别为: 97.749 105、98.173 032、98.686 947、90.844 358、93.997 826、92.598 704、98.000 685、94.314 138、

进行处理。中间件定义一个状态值用于表明中间件当前的状态。中间件状态值初始为初态,根据输入消息的类型、代码和数值进行状态转换和相应的处理操作,当中间件的状态为可接受状态时,向应用程序上报触摸点信息并转入初态进行下一轮的处理。当中间件在某一状态接收到不能识别的消息时,转入初态,不能识别的消息是指该消息在该状态没有对应的转入状态。

本文设计的中间件在使用时可以作为系统服务运行于操作系统中,也可以作为函数库供用户程序单独使用。

3.2 实验测试

编写了一个中间件测试程序,该程序的功能是通过对触摸屏的操作实现图片的移动、缩放和旋转^[5],驱动程序由不同的厂商提供。经过实验,中间件可以识别和处理驱动程序上报的消息串,测试程序可以使用中间件提供的触摸点数据实现对图片的操作^[6]。

结 语

本文主要设计了可以识别 Linux multi-touch 协议消息串的有穷自动机模型,并依照此模型编写了多点触摸屏中间件,最后给出了中间件的关键代码并对中间件进行测试。使用本文设计的中间件,可以使开发者免去了

对触摸屏驱动程序原始数据的处理而提高开发效率,同时也为 Linux multi-touch 协议消息串的处理提供了一种思路。ME

参考文献

- [1] Multi touch (MT) Protocol[EB/OL]. [2016-06]. <https://www.kernel.org/doc/Documentation/input/multi-touch-protocol.txt>.
- [2] 宋宝华. Linux 设备驱动开发详解:基于最新的 Linux4.0 内核[M]. 北京:机械工业出版社,2015.
- [3] 林平凯. 基于 Linux 系统的多点触摸驱动设计[J]. 信息技术与标准化,2014(6).
- [4] 崔力升. 中间件技术的综述[J]. 科技视界,2014(3):198-198.
- [5] 刘斌. 基于有限状态机的 Linux 多点触摸屏驱动设计[J]. 微计算机信息,2012,28(3):54-56.
- [6] 泰国栋. 有限状态机的嵌入式 Linux 按键驱动设计[J]. 单片机与嵌入式系统应用,2010,4(1):79-81.

赵建峰(硕士研究生),主要研究方向为嵌入式系统与应用;陈朔鹰(副教授),主要研究方向为计算机操作系统、嵌入式系统与应用;罗一涵(硕士研究生),主要研究方向为嵌入式系统与应用。

(责任编辑:薛士然 收稿日期:2016-06-06)

30 99.106 476。从测试结果可以看出,人员平均识别率在 95.6%以上,最低为 90.8%,本系统完全可以胜任单位的人员考勤工作。

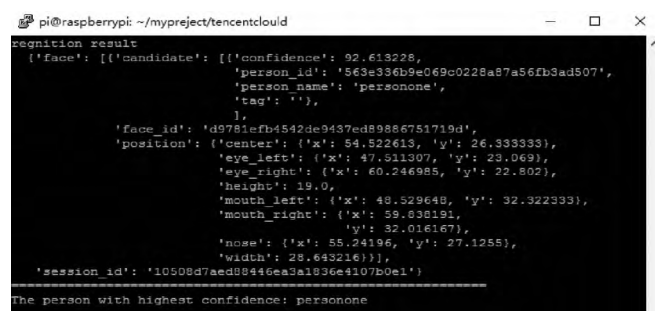


图5 系统识别率测试结果

结 语

本文依托人脸识别云服务,基于树莓派设计了一种考勤系统,其识别率及实时性均可满足设计者需求。通过详细地描述基于树莓派的人脸识别考勤系统的搭建和实现过程,为相关产品提供了一套解决方案,这方面的研究将有助于加速云服务及智能硬件发展的进程,具有较高的研究价值。ME

参考文献

- [1] 沈理,刘翼光,熊志勇. 人脸识别原理及算法——动态人脸识别系统研究[M]. 北京:人民邮电出版社,2014.
- [2] Michael J kavis,陈志伟. 云计算服务模式(SaaS、PaaS 和 IaaS)设计决策[M]. 北京:电子工业出版社,2015.
- [3] MattRichardson,ShawnWallace. Getting Started with Raspberry Pi[M]. Sebastopol:O'Reilly Media, 2012.
- [4] Magnus Lie Hetland. PYTHON 基础教程(修订版)[M]. 2 版. 司维,曾军葳,谭频华,译. 北京:人民邮电出版社,2014.
- [5] 腾讯云[EB/OL]. [2016-01]. <https://www.qqcloud.com/doc/product/277/优图人脸识别简介>.
- [6] NathanOostendorp,AnthonyOliver,KatherineScott. Practical Computer Vision withSimpleCV[M]. Sebastopol: O'Reilly Media, 2012.
- [7] 唐汉明,翟振兴,关宝军. 深入浅出 MySQL(数据库开发优化与管理维护第2版)[M]. 北京:人民邮电出版社,2014.

李玉鹏(本科),主要研究方向为单片机及嵌入式系统的开发。

(责任编辑:杨迪娜 收修改稿日期:2016-05-10)