

# 基于OpenMV的人脸识别智能门锁设计

唐孟雪, 邢晓燕

(西华师范大学, 四川 南充 637002)

摘要: 得益于现代科技的飞速发展, 人们的生活水平得到了极大的提升, 与此同时人们对安全的要求也越来越高。门锁一直是一个家庭安全保障的关键性物件, 门锁的发展更是经历了从普通物理门锁到密码锁, 指纹锁到如今利用生物特征识别技术的门锁。其中人脸识别技术更是在近年来得到了大众的追捧和普遍认可, 并已逐渐被应用于一些高科技产品设计上。

该次设计初探人脸识别门锁制作, 采用OpenMV智能采集人脸信息, 利用其内置的STM32F65VI AMR Cortex M7处理器计算处理信息, 通过对比图像LBP算子的方式, 识别判断人脸信息。同时将OpenMV与单片机相连接, 外接LCD1602显示屏, 矩阵键盘, 继电器, 蜂鸣器完整模拟一款可以进行用户人脸信息注册、人脸识别、电子显示、开锁、报警等多功能的智能识别人脸门锁。

关键词: OpenMV; LBP算子; 人脸识别; 门锁

中图分类号: TP18 文献标识码: A

文章编号: 1009-3044(2020)25-0005-04

DOI: 10.14004/j.cnki.ckt.2020.2953

开放科学(资源服务)标识码(OSID):



## 1 背景

人脸识别技术是近代以来飞速发展的一种生物识别技术, 由于每个人长相不同, 脸部轮廓幅度不同, 用人脸信息作为识别身份的依据是可靠的。因而现如今也出现了人脸识别的广泛应用, 比如支付宝刷脸支付、手机刷脸解锁, 公安监控系统的人脸识别等。但就对于普通家庭而言, 门锁才是安全的第一保障, 而传统的钥匙门锁易丢失, 易复制; 密码锁密码需定时更换, 且容易被录像窃取; 指纹锁在智能人脸门锁的特征优势下也显得相形见绌。虽然人脸识别门锁成本较高, 但具有可记录图像, 处理和分析图像的功能, 不仅防止了外来非法人员的进入, 更可以帮助使用者了解试图闯入人员的面部信息, 起到很好的安全防范作用。

人脸识别是通过对图像中人脸各个维度信息进行分析, 再将分析后的数据与数据库内的数据进行对比判断, 从而达到识别的效果。一般步骤为, 定义局部区域, 局部区域特征的提取, 经过样本训练后得到的变换矩阵将人脸图像向量映射为人脸特征向量; 然后进行分类。分类器多采用组合分类器的形式, 每个局部特征对应一个分类器, 后可用投票或线性加权等方式得到最终识别结果。

人脸识别算法的分类按维数分类可大致分为二维算法和三维算法。

二维算法是利用人脸上分布的由低到高的80个节点来检测测量眼睛、颧骨、下巴等之间的距离从而实现身份认证。其主要算法有: 基于模板匹配的方法; 基于奇异值特征方法; 子空间分析法; 局部保持投影(Locality Preserving Projections, LPP); 主成分分析(PCA); 其他方法还有弹性匹配方法、特征脸法(基于

KL变换)、人工神经网络法、支持向量机法、基于积分图像特征法(adaboost学习)、基于概率模型法。

由于三维算法可以有效地解决二维图像中姿态, 光照, 表情, 面部遮挡物等方面对识别精度的影响, 自90年代初期, 科学家们就开展了对三维算法的研究, 并取得了极大的进展。目前三维人脸识别方法有基于图像特征的方法和基于模型可变参数的方法。

## 2 总体设计

### 2.1 系统总体框图

本系统主要是由五个硬件部分组成, 其中核心的处理器为OpenMV搭载的STM32F65VI AMR Cortex M7芯片, 输入模块有按键和摄像头; 输出模块有LCD显示和继电器模拟门锁开关。

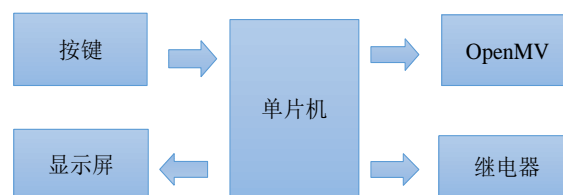


图1 系统总体框图

### 2.2 系统总体方案

本设计按照用户具体操作情况可以分为两大部分, 一是管理员操作模块, 二是用户使用模块。其中管理员操作模块分为三个部分: 验证管理员信息; 采集人脸信息; 退出管理员系统。用户使用模块又可细化为下面四个步骤: 摄像头采集人脸, 处

收稿日期: 2020-06-18

作者简介: 唐孟雪(1998—), 女, 四川达州人, 本科在读, 研究方向为通信工程; 邢晓燕(1981—), 女, 山西平遥人, 讲师, 硕士, 研究方向为嵌入式系统、移动平台。

本栏目责任编辑: 王 力

本期推荐 5

理人脸信息,根据人脸信息选择是否开锁,系统复位。具体描述如下:

- 1)通过键盘选择进入管理员操作系统或用户系统。
- 2)若检测到指定按键被按下,摄像头立刻采集人脸图片三次。
- 3)处理器处理采集到的人脸图片,与系统内储存的合法人脸相对比,若三次采集的图像数据对比吻合度都达到阈值要求,则打开门锁。
- 4)门锁打开后,按任意键关闭门锁。
- 5)若用户想进入管理员系统需按下按钮并输入管理员密码。
- 6)检测管理员密码是否正确,若正确进入管理员操作系统;若错误发出提示音,若连续三次输入错误,则锁住按键,不能更改。
- 7)录入人脸信息20次。
- 8)按任意键退出管理员系统。

### 3 硬件设计

本设计主要是由OpenMV、STC89C51单片机、矩阵键盘、液晶显示器、继电器控制模块等部分组成。

#### 3.1 OpenMV

OpenMV摄像头是一款低功耗,容易上手,扩展性强的电路板,其使用python编程语言可以帮助使用者较为轻松地完成机器视觉应用的实现。它由STM32F65VI AMR Cortex M7处理器搭配OV7725感光元件构成,其中处理器有一个全速USB(12Mbps)接口,可供OpenMV插入电脑端;一个 $\mu$ SD卡槽;一个SPI总线;一个I2C总线,CAN总线,和一个异步串口总线(TX/RX);一个12-bit ADC和一个12-bit DAC;三个I/O引脚用于舵机控制;一个RGB LED(三色),两个高亮的850nm IR LED(红外)。

OpenMV是一个开源的,搭载了一个强大的32位处理器的摄像头,达到了本项目的技术要求水平,并且可以使用SD卡可以更方便将图像流数据提取出来,是一个较为完整地开发系统。相较于在STM32单片机上搭载摄像头,OpenMV更加的便捷高效。相较于用树莓派搭载摄像头采集人脸信息,然后调用opencv与face++的API对照片进行识别而言,OpenMV会更加方便快捷。

#### 3.2 STC89C51 单片机

本设计中,管理员系统采用了STC89C51作为主处理芯片,STC89C51是电子类大学生普遍接触的一款单片机类型,具有低功耗,高性能的特点。STC89C51含有32个双向I/O口,2个16位可编程定时器、计数器中断,2个串行中断,2个外部中断,2个读写中端口,具有低功耗空闲和掉电模式,可软件设置睡眠和唤醒功能。

由时钟电路和复位电路构成单片机最小系统,时钟电路为单片机提供运行振荡时钟,是单片机工作的前提;复位电路提供快速复位操作,一般单片机都提供了上电自动复位和手动复位。

#### 3.3 LCD1602 显示屏

LCD1602是一款点阵式LCD,由16\*2个显示单元组成,每个显示单元可以显示一个字符,一个显示单元由5\*7的点阵字符显示位构成。显示单位的点阵字符的明暗反映出显示的内

容,而屏幕上的显示单元又与显示RAM区字节一一对应,实现了控制显示。

LCD1602共具有16个引脚,其中2个电源引脚VSS和VCC,一个显示屏对比度调整端VL,一个寄存器选择引脚RS,一个读写引脚R/W,一个使能引脚E,一正一负两个背光源引脚,七个双8位双向口数据线D0~D7。

LCD1602的读写,光标和显示模式的控制操作则都是由指令编码完成。一共有十一条指令:清除、返回、输入模式设置、显示开关、控制、以为、功能设置、CGRAM地址设置、DDRAM地址设置、读忙信号和光标地址、写数据、读数据。

#### 3.4 AT24C02 存储模块

AT24C02是一款美国Atmel公司生产的低功耗CMOS型E2PROM,具有掉电存储功能,利用IIC总线数据传送协议,它共有6个引脚,其中有串行时钟线SCL,串行数据/地址线SDA, A0-A2器件地址输入端,WP写保护。

#### 3.5 继电器控制模块

电磁继电器是利用电磁感应原理,可以通过小电压控制较大电流,其主要组成部分有铁芯、线圈、衔铁、触点簧片。当给线圈两边施加一定电压,线圈周围将会产生磁场,从而吸引衔铁使之向铁芯方向移动,同时也导致了衔铁的动触点与静触点吸合,使输出回路连通。当断电后,衔铁在弹簧的拉力作用下返回原来位置,动触点与静触点断开,输出回路断开。在此设计中,利用继电器控制门锁的开关是合适的。

#### 3.6 矩阵键盘模块

本次设计需要用户输入密码,这里采用的是最普遍的0-9数字密码,因而选用了4\*4矩阵键盘。将键盘设计成矩阵的形式可以减少I/O口的占用,4\*4一共16个仅占用一个端口即可设计完成。

### 4 软件设计

此次开发设计选用OpenMV作为图像处理处理器,而OpenMV的二次开发一般选用Python程序语言。Python具有上手简单;不用考虑内存的申请和释放;已有开发团队封装好的各种算法和模块可以直接调用;还有MicroPython库可以直接用等的优点,并且Python在图像处理,机器学习,网络编程中都得到了广泛应用。

#### 4.1 主函数程序框图

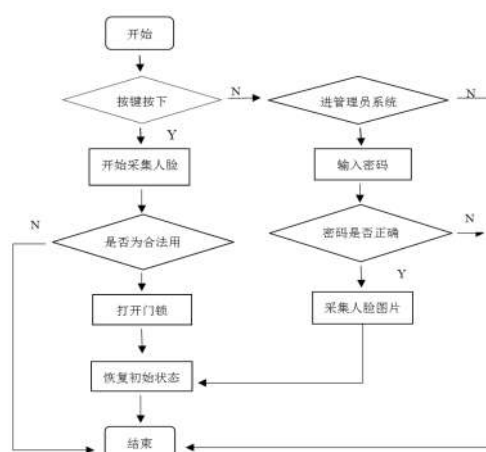


图2 主函数程序流程图

## 4.2 各子程序设计

### 4.2.1 OpenMV 采集人脸程序设计

OpenMV 采集人脸只需要将当前摄像头所拍摄图像信息储存到用户库中即可。为了提高样本集,在此处设置每个用户拍摄20张照片,并且拍摄过程中需要用户尽可能调整自己的面部表情。

采集人脸程序代码:

```
import sensor, image, pyb
red_led = LED(1)
green_led = LED(2)
blue_led = LED(3)
lcd.init() # 初始化lcd.
sensor.reset() # 初始化摄像头.
sensor.set_pixformat(sensor.GRAYSCALE)
sensor.set_framesize(sensor.B128X128)
sensor.set_windowing((92,112))#设置屏幕窗口大小
sensor.skip_frames(10)
sensor.skip_frames(time = 2000)
num = 3#被拍摄人脸信息编号
n = 10 #每个用户拍摄10张照片
while(n):

    red_led.on()#亮红灯
    sensor.skip_frames(time = 2000) # 暂停2s
    blue_led.on()
    sensor.snapshot().save("photo/s%s/%s.pgm" % (num, n) ) #
    将用户人脸信息保存到SD卡里面
    n -= 1
    blue_led.off()
    print("注册人脸信息已成功")
```

### 4.2.2 OpenMV 识别人脸程序设计

为了提高识别人脸时数据采集的精准性,使用了判断标识关键字的方式,利用for循环语句,进行三次人脸采集,分别判断当前人脸与用户库人脸数据的对比数据是否超过阈值,当差异度小于阈值,将标识字赋值为1,否则为0。当关键字为0时,退出识别人脸程序,并闪烁红灯,标识人脸识别失败。当三次识别均成功时,闪烁绿灯,打开继电器。

识别人脸代码:

```
import sensor, time, image, pyb, lcd
from pyb import Pin
from pyb import LED
red_led = LED(1)
green_led = LED(2)
lcd.init() #
sensor.reset() # 初始化摄像头.
sensor.set_pixformat(sensor.GRAYSCALE)
sensor.set_framesize(sensor.B128X128)
sensor.set_windowing((92,112))#设置屏幕窗口大小
sensor.skip_frames(10)
sensor.skip_frames(time = 2000)
NUM_SUBJECTS = 3 #图库中的用户人数
```

```
NUM_SUBJECTS_IMGS = 10 #每个人的合法样本数
flag = 1
p_in = Pin('P6', Pin.IN, Pin.PULL_UP)# P6 为输入引脚,并
上拉电阻
value = p_in.value() # get value, 0 or 1#读P6引脚的值
print(value)
def min_value(a, b, c):
    global num
    if a<pmin:
        pmin=b
        num=c
    return a
if value==0 and flag ==1:
    for n in range(0,3):
        # 拍摄当前人脸。
        Img_now = sensor.snapshot()
        #####display###
        for i in range (1,100):
            lcd.display(Img_now) #拍摄当前照片并显示出来
            #####
            L0 = Img_now.find_lbp((0, 0, img.width(), img.height()))
            #将当前人脸的lbp特征保存在L0中
            Img_now = None
            pmin = 999999#最大化
            num=0
            for s in range(1, NUM_SUBJECTS+1):
                dist = 0
                for i in range(2, NUM_SUBJECTS_IMGS+1):
                    img = Img_now.Image("photo/s%d/%d.pgm"%(s, i))
                    L1 = Img_now.find_lbp((0, 0, Img_now.width(),
                    Img_now.height()))
                    #将文件夹中的lbp特征保存在L1中
                    dist += Img_now.match_descriptor(L0, L1)#计算人
                    脸的特征差异度。
                    print("Average dist for subject %d: %d"%(s, dist/
                    NUM_SUBJECTS_IMGS))
                    pmin = min_value(pmin, dist/NUM_SUB-
                    JECTS_IMGS, s)#打印当前照片与用户库中照片的特征差异度
                    if (pmin<6500):
                        flag = 1
                    else:
                        flag = 0
                print(pmin)
                print(num_value) # num为当前最匹配的人的编号。
                if(pmin<6500):
                    print("the door is open.")
                    Img_now.draw_string(200,200,"the door is open.", col-
                    or = (255, 0, 0))
                    green_led.on()
                    time.sleep(500)
                else:
                    print("the face is not detected,please try again.")
                    Img_now.draw_string(200,200,"the face is not detected,
```



```

please try again.")
    red_led.on()
    time.sleep(500)
else:
    print("the face is not detected,please try again.")
    Img_now.draw_string(200, 200, "the face is not detected,
please try again.")
    red_led.on()
    time.sleep(500)

```

### 4.2.3 密码验证程序设计

在验证管理员信息时需要对输入的密码与 AT24C02 中储存的密码进行比较验证,并且设置了三次密码错误启动键盘锁,此时将不能再通过按键输入密码。

#### 4.2.4 按键程序设计

本次设计采用的是矩阵键盘,判断是否有按键按下,需要CPU对所有的键盘进行实时监控,判断按下的是哪个键盘需要按行按列依次扫描键盘确认位置,并且扫描时要注意去除抖动。

## 5 调试

### 5.1 人脸识别特征差异阈值

由于LBP特征识别算法是将图像的局部纹理特征转化为一个LBP值,因此在对比图像时,LBP的差异度即特征差异度越小,说明检测到的人脸与样本人脸更为相似。

在本设计中,用于区分是否为本人的这个特征差异度称之为阈值。这个阈值的选择需要实际实验测得数据来选择,同时还考虑到了样本照片,光线强弱,人物背景,人物表情等因素对特征差异度的影响。下面是控制变量法下的实验数据。

表1 样本特征差异度

		样本一无表情 20 张	样本一各种表情 20 张	样本一各种表情 20 张 2	样本二
1	无表情	5074	5055		
2	笑	6214	5955		
3	怪表情	3586	4536		
4	无表情	4599	5012	5208	
5	笑	6940	6516	6459	
6	无表情	5680	5607	5518	9985
7	散发	4971	5464	5789	10648
8	去掉眼镜	5314	5822	5939	10539
9	亮度低	6292	6211	5434	10182
10	1-9 平均值	5407.777778	5575.333333	5724.5	10338.5
12	样本一照片 1 近照	7397	8820	8870	11052
13	样本一照片 2 近照	6940	7384	7231	11673
14	样本一照片 2 近照	7028	6469	7434	11413
15	样本一照片 3	6936	6734	6734	10909
16	样本一側照	8029	8284	8284	11251
	样本一照片 4	7992	7689	7684	11352
12-16 均值		7387	7563.333333	7706.166667	11275

由表中数据可以看到,通过图片检测的特征差异度普遍在7000-8000之间,而真实人脸检测的特征差异度可以在6500左右。因此在实际选择阈值时选择设置阈值为6500,并且进行多次检测的方式减少误差比较合适。

## 5.2 OpenMV 调试

对 OpenMV 调试主要是将其连接至电脑端,单独运行人脸识别和人脸存储的代码。由于代码是用 python 语法,因此有一个常见的错误提醒:SyntaxError: invalid syntax,这种错误常由于在 def,class,if,elif,for,while 等语句末尾没有加上“:”或是因为缩进不正确。由于 python 是一个没有中括号的编程语言,语言的逻辑包含关系全都由缩进体现,因此在整个编程中要特别注意缩进。另外在调试代码时出现了没有语法错误,但是出来的结果并非期望值的情况,这时就只能一步一步地看代码,在关键代码处可以添加一行输出来查看这时一些关键值的变化,这样可以快速定位出错的逻辑位置。

还有一个小问题值得一提,由于OpenMV保存人脸信息是保存至本地的SD内存卡中,在提取人脸信息算法中要输入相应的路径,该路径下存有已经保存的人脸信息分别放至s1~s6的文件夹中。开始本以为路径和文件夹都可以自动生成,但是测试时发现都不行,而且在提取完人脸信息后,文件夹内是不会立即更新的,需要将OpenMV重启后才能看到。

### 5.3 LCD 显示屏调试

本次设计中的管理员系统采用了LCD1602显示屏,用于用户输入密码显示。整体来说LCD1602的调试比较简单,可以通过在51学习板上进行范例代码的测试,由此判断显示屏是否工作正常,只要显示成功余下的就是编码的问题了。

## 5.4 矩阵键盘调试

矩阵键盘测试的重点是硬件方面,在焊接时一定要仔仔细细再仔细,每次焊接完一行或者一列都需要用万用表测试是否连通,是否存在虚焊的情况。用万用表确认基本无误后,将在学习板测试成功的简单按键操作代码修改引脚后用于测试。

## 5.5 系统安装

1)设计需求分析,按照本次设计的需求现将设计成为相应模块,并构思好系统的框架结构。

2)按照模块需求,列出每个模块需要的元器件,并汇总为总元器件清单。

3)按照元器件清单购买元器件,并根据设计的原理图将元器件按照每个模块焊接完成,并用万能表测试焊接情况,初步排除虚焊,漏焊等情况。

4)在相应的编程软件中,按照模块顺序编写相应程序,如果方便可以先在学习板上测试程序功能。

5)在软硬件准备工作完成后,将所有的模块相连,挨个测试模块功能是否完成。模块测试完成后再编写主程序,耐心调试。

6)在主程序测试成功后,对整体代码逻辑顺序进行梳理,多次测试完善,直到最终达到理想效果。

设计安装完成及调试过程如图3。

(下转第18页)

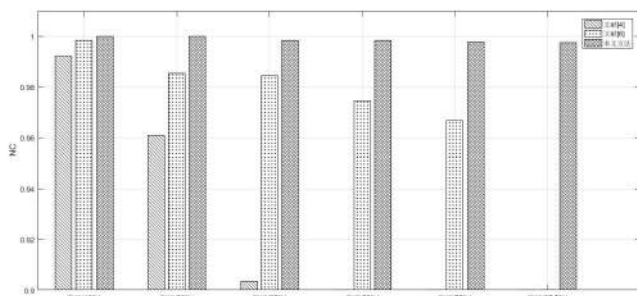


图6 与文献[4,6]针对不同程度剪切攻击(NC)的对比图

#### 4 结论

为提高零水印方案抵抗强剪切攻击的能力,结合NMF和改进的Sobel算子边缘提取等算法提出一种抗强剪切攻击的零水印方案。利用NMF的局部感知全局的特性,对受强剪切后的图像进行重构,提取出尽可能精确地水印。实验表明,当剪切面积高达96.875%时,提取水印精确度为99.67%,实现了抗强剪切的目的。同时对其他包括缩放、滤波、噪声、压缩等攻击具有很强的鲁棒性。

#### 参考文献:

- [1] Rani A, Bhullar A K, Dangwal D, et al. A zero-watermarking scheme using discrete wavelet transform[J]. Procedia Computer Science, 2015, 70: 603-609.
- [2] Fang Z, Zhao Y. Image watermarking resisting to geometrical attacks based on histogram[C]//2006 International Conference on Intelligent Information Hiding and Multimedia. 18-20 Dec. 2006, Pasadena, CA, USA. IEEE, 2006: 79-82.
- [3] 凌贺飞, 卢正鼎, 邹复好. 抗几何攻击的数字水印技术综述[J]. 计算机工程与科学, 2006, 28(11): 42-47, 50.
- [4] Verma V S, Jha R K, Ojha A. Digital watermark extraction using

support vector machine with principal component analysis based feature reduction[J]. Journal of Visual Communication and Image Representation, 2015, 31: 75-85.

- [5] JiaXL, Zhou Y, QiYL, et al. The anti-geometric attack digital watermarking algorithm based on image normalization of local information[C]//2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference. 20-22 May 2016, Chongqing, China. IEEE, 2016: 1120-1124.
- [6] Hurrah N N, Parah S A, Loan N A, et al. Dual watermarking framework for privacy protection and content authentication of multimedia[J]. Future Generation Computer Systems, 2019, 94: 654-673.
- [7] 温泉, 孙铁峰, 王树勋. 零水印的概念与应用[J]. 电子学报, 2003, 31(2): 214-216.
- [8] 肖振久, 宁秋莹, 张晗, 等. NMF和增强奇异值分解的自适应零水印算法[EB/OL]. [2019-10-08]. <http://www.aocmag.com/article/02-2020-04-030.html>.
- [9] Yang K Y, Wang W W, Yuan Z H, et al. Strong robust zero watermarking algorithm based on NSCT transform and image normalization[C]//2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). 12-14 Oct. 2018, Chongqing, China. IEEE, 2018: 236-240.
- [10] 刘万军, 孙思宇, 曲海成, 等. 一种抗几何旋转攻击零水印算法[J]. 计算机应用研究, 2019, 36(9): 2803-2808.
- [11] Lee D D, Seung H S. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788.
- [12] 沈德海, 侯建, 鄂旭, 等. 基于Sobel的多方向算子模板边缘检测算法[J]. 现代电子技术, 2015, 38(4): 91-93, 97.

【通联编辑:唐一东】

(上接第8页)



图3 调试过程

#### 参考文献:

- [1] 何玉. 基于Python语言图像边缘检测及其算法分析[J]. 计算

机产品与流通, 2018(6): 147.

- [2] 郭向, 星田斐. 智能人脸识别门锁控制系统设计[J]. 电子技术与软件工程, 2019(17): 115-116.
- [3] 李勇. 基于RFID与人脸识别技术的智能门禁系统研究与设计[D]. 重庆: 重庆大学, 2009.

【通联编辑:谢媛媛】