

数学学院案例分析报告

案例名称：人脸检测及识别分析

报告参与人：魏丹怡（202032164）

完成时间：2021.06.17

目 录

一、绪论.....	1
1、研究背景.....	1
2、研究意义.....	1
二、人脸检测.....	2
三、人脸识别.....	2
1、人脸录入.....	2
2、特征提取.....	3
3、人脸对比.....	4
四、实践总结.....	4

一、 绪论

1、研究背景

生物特征识别技术是利用人体生物特征进行身份认证的一种技术。生物特征是人的内在属性，具有很强的自身稳定性和个体差异性，因此是身份验证的理想依据。生物特征识别系统对生物特征进行特征提取并组成特征模板，当人们应用该识别系统进行身份认证时，识别系统提取其特征并与数据库中的特征模板进行对比，以确定是否匹配，从而决定接受或拒绝该人。一般来说，人类的身份识别方式分为三类：

- 1) 特征物品:包括各种证件，如身份证、学生证和护照等；
- 2) 特殊知识:包括各种密钥如(密码、口令等)和暗号等；
- 3) 人类生物特征:包括各种人类的生理和行为特征，如人脸、指纹、掌纹、虹膜、声音等。

前两类特征识别方法属于传统意义上的身份识别技术，有着方便、快捷的特点。但这些特征识别方法致命的缺点是安全性差、易伪造窃取。相比较而言，人体生物特征由于其稳定性和独特性，成为最理想的身份识别特征。相比于其他生物特征，基于人脸面部特征的识别具有主动性、非侵犯性和用户友好性等许多优点，它是一种更方便直接、更友好、更容易被人们接受的识别方法。

与此同时，人脸自身存在着诸如表情、姿态、光照强度变化以及饰物影响等，都会使人脸识别方法的效果及稳定性受到很大的影响。在过去的几十年中，研究者们主要致力于从人脸识别算法的角度来提高生物特征识别的精度。时至今日，很多生物识别技术(如:指纹识别、人脸识别、声音识别、虹膜识别等)都具有了很高的识别精度且有行对较好的用户友好性。

2、研究意义

人脸识别是当前人工智能和模式识别的研究热点，它是近 20 年来才发展起来的，20 世纪 90 年代后更成为科研热点。最初的应用源于公安部门关于罪犯照片的存档管理和刑侦破案，随着科学技术的发展和社会的进步，进行快速、有效、自动的人脸识别的技术要求日益迫切。不仅如此，目前该技术在国家公共安全、社会安全及商业等领域都有很多应用，主要有证件验证主要有证件验证、刑侦破案、入口控制、信息安全、视频监控等等。因此，这项课题有着非常广阔的前景，值得深入研究。

二、人脸检测

人脸检测 (Face Detection), 就是给一幅图像, 找出图像中的所有人脸位置, 通常用一个矩形框框起来。我使用 WIDER FACE 数据集训练了自己的人脸检测模型, 并对目标图像进行了人脸检测, 结果如下:



图 1 人脸检测

三、人脸识别

本次人脸识别的实现利用了 Python 开发, 借助 Dlib 库捕获摄像头中的人脸并提取人脸特征, 通过计算特征值之间的欧氏距离, 来和预存的人脸特征进行对比, 判断是否匹配以达到人脸识别的目的, 其核心为提取 128D 人脸特征。

首先, 可以从摄像头中抠取人脸图片存储到本地, 然后提取构建预设人脸特征; 其次, 根据已有的同一个人多张人脸图片提取 128D 特征值, 然后计算该人的 128D 特征均值; 最后, 和摄像头中实时获取到的人脸提取出的特征值, 计算欧氏距离, 判定是否为同一张人脸。

1、人脸录入

人脸识别需要将提取到的图像数据和已有图像数据进行比对分析, 因此, 首要进行人脸录入。我截取了三位同学每个人各 3 张图片作为预存的人脸, 以此来提取人脸特征。

运行 `get_faces_from_camera.py` 文件, 然后根据键盘输入进行人脸捕获。“N”的意思是新录入人脸, 新建文件夹 `person_X` 来存储第 X 个人的人脸图像; “S”的意思是开始捕获人脸, 将捕获到的人脸放到 `person_X` 文件夹中; “Q”是指退出窗口。摄像头的调用利用了 `opencv` 库的 `cv2.VideoCapture(0)` (其中参数为 0 代表调用的是笔记本的默认摄像头)。代码运行输出结果如下:

```
新建的人脸文件夹 / Create folders: data/data_faces_from_camera/person_1  
写入本地 / Save into: data/data_faces_from_camera/person_1/img_face_1.jpg  
写入本地 / Save into: data/data_faces_from_camera/person_1/img_face_2.jpg  
写入本地 / Save into: data/data_faces_from_camera/person_1/img_face_3.jpg
```

人脸截取窗口展式如下：

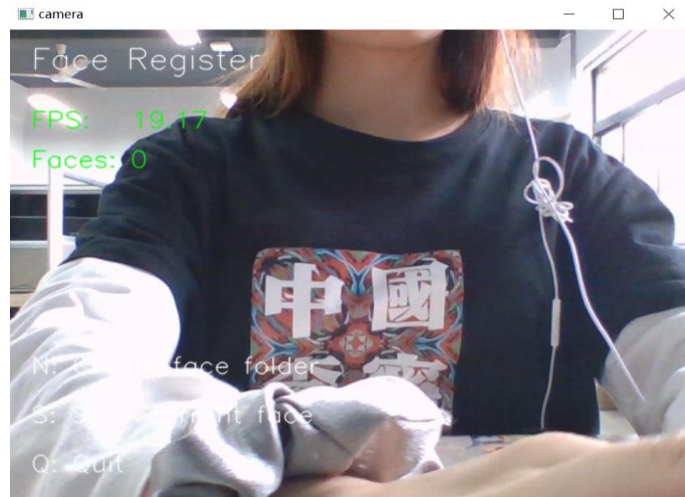


图2 人脸截取

截取到的一组人脸示例如下：

识别code > data > data_faces_from_camera > person_3



图3 人脸示例

2、特征提取

在得到三组人脸图像后，先将人脸图像对应的名字存入 `name.txt` 文件中，之后开始提取人脸的 128D 特征，然后计算出每个人人脸数据的特征均值存入 CSV 中，方便之后识别时候进行比对。

运行 `features_extraction_to_csv.py` 文件，则可以得到每张人脸图像的 128D 特征。之后就需要对人脸图像进行批量化操作，提取出 128D 的特征，然后计算特征均值，存入 `features_all.csv` 文件中，`features_all.csv` 是一个 n 行 128 列的 CSV 文件，“n”是指录入的人脸数，“128”是指每个人的 128D 特征。

得到的三个人的 128D 特征展式如下：

表 1 人脸特征

Person_1	-0.0240	0.0427	0.0593	-0.0956	...	0.092875	-0.0140
Person_2	-0.0551	0.0921	0.0752	-0.0844	...	0.042278	-0.0277
Person_3	-0.0968	0.0689	0.0664	-0.0919	...	0.116294	0.0560

3、人脸对比

在录入人脸数据之后，调用摄像头捕获摄像头中的人脸，将摄像头中的人脸提取出 128D 特征，即可和已录入人脸的这些特征值进行比对，如果欧式距离比较近的话，就认为是同一张人脸，否则不是一个人。

运行 face_reco_from_camera.py 文件，我们发现在人脸之下，可以显示出对应的名字，若出现未预存的人脸，则会显示 unknow 。

人脸识别窗口展式如下：

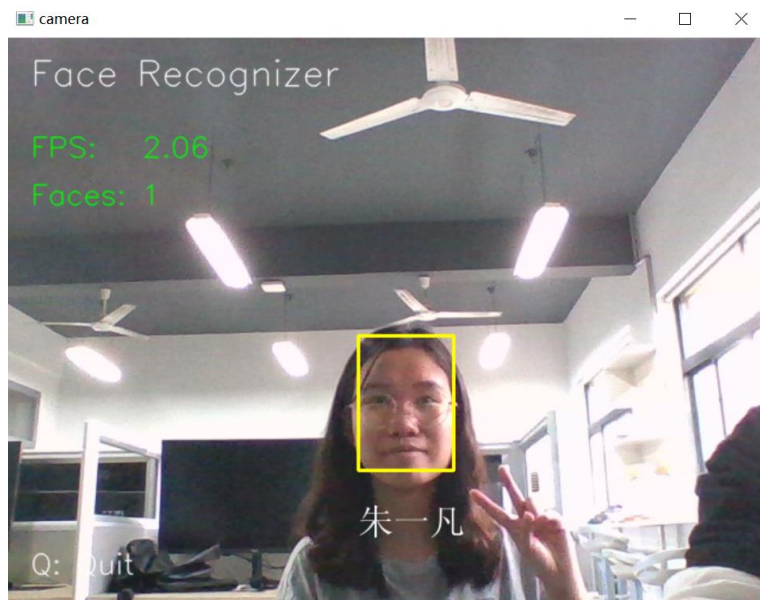


图 4 人脸识别

四、 实践总结

本次实验进行了人脸检测与人脸识别。首先使用 WIDER FACE 数据集训练了自己的人脸

检测模型，并对目标图像进行了人脸检测；其次，通过调用 Dlib 库实现了人脸录入、特征提取及人脸识别，当已录入人脸出现在摄像头画面中时，可以显示对应的名字。我们的实验过程中，在安装环境以及下载某些函数库时，出现很多问题，在解决这些问题的过程中，极大地提高了我对编程软件的熟悉程度。