

基于 FAST 和 Brief 算法使用 tracking.js 的 Web 人脸识别考勤系统

徐辉, 陈晨

(蚌埠学院, 蚌埠 233030)

摘要:

实现一个 Web 人脸识别考勤系统, 系统使用 HTML5 技术通过浏览器来获取人的面部图像, 再引入 tracking.js 库, 利用其中的 FAST 和 Brief 算法实现来对人面部图像的生物特征进行快速特征点提取、描述和匹配, 最终完成考勤过程。这些 HTML、JavaScript 代码和图像特征数据可以部署在 HTTP 服务器上, 通过 Web 访问, 形成一套 B/S 架构的人脸识别考勤系统。在任何地方, 只要有一台有摄像头、可以连接 Web 服务器的设备, 就能够很快进行考勤。相比已有的方案, 将具有价格低廉、访问快速等优点。

关键词:

FAST; Brief; tracking.js; 人脸识别

基金项目:

安徽省高等学校大学生创新创业项目(No.201511305008)

0 引言

基于生物特征的身份验证一直是人们研究的重点方向。其中, 人脸识别在近年来随着研究的深入和识别率的提升, 被愈来愈广泛的应用于生活的各个方面, 向民用领域不断地扩展, 如支付宝等软件的登录、日常学习工作中的人脸识别考勤等。传统的考勤方法包括人工记录、打卡、手机软件定位等, 这些方法均存在一定缺陷, 例如人工记录费时费力容易错误、记录结果可能被伪造, 打卡、手机软件定位则需要携带硬件设备, 也同样存在记录被伪造的可能。

基于以上原因, 基于人脸识别的面部考勤设备近年被逐渐地采用于各种场合。但是这样的考勤设备一般是硬件解决方案, 同样也存在安装调试过程繁琐、不具有适应多种场合的缺点。

同时, 我们注意到随着 HTML5 标准的提出、网络带宽的增加、浏览器技术的发展, Web 和浏览器具有了越来越强大的能力, 例如捕获摄像头视频的能力, 处理图像的能力等。这些能力为本文提出基于 Web 的人

脸识别考勤系统奠定了硬件和技术基础。在本文中, 我们将开发一个使用 HTML5 技术的、纯 JavaScript 的人脸识别考勤系统。这个系统使用 HTML5 技术通过浏览器来获取人的面部图像, 再使用纯 JavaScript 代码来对人面部图像的生物特征进行快速特征点提取和描述, 从而完成特征匹配, 最终确定摄像头前的人是谁。这些 HTML、JavaScript 代码和图像特征数据可以部署在 HTTP 服务器上, 通过 Web 访问, 形成一套 B/S 架构的人脸识别考勤系统。这样, 在任何地方, 只要有一台有摄像头可以连接 Web 服务器的电脑, 我们就能够很快进行考勤。相比已有的方案, 将具有价格低廉、访问快速等优点。

1 相关理论

经典的人脸识别算法有: 特征脸方法、线性判别分析和局部二值模式。

特征脸方法(Eigenface)^[1]是通过一定数量的训练样本来建立一组正交基, 并将人脸图像映射到一个低维的子空间, 因而达到降低图像维度的目的, 提取有利

于表征人脸图像的特征信息。

线性判别分析(LDA)又叫 Fisher 判别分析^[2]。LDA 也是主要通过训练样本数据来建立一个特征空间,并且这个特征空间特点是图像在这个空间经过投影后能够保证类内离散度最小,类间离散度最大,也就是最大可能保证类则之间的可分性,人脸识别应用中用于区别不同的人脸图像具有较好的效果。

局部二值模式又叫 LBP 算子^[3],是一种从纹理局部近邻定义中衍生出来,灰度范围内的纹理度量算子。最原始的 LBP 算子是由 Ojala 于 1996 年提出,用来分析图像纹理特征,并且描述了它在纹理分类中的强区分能力。由于其具有很强的分类能力、较高的计算效率、对单调的灰度变化具有不变性等特点,现已被延伸到人脸检测、人脸识别及表情分析和背景建模之中。LBP 对人脸描述的基本思想是利用纹理描述算子构建若干局部的人脸描述,然后把局部描述组合形成全局描述。

除上面这三种经典算法外,SVM、稀疏表示、HMM、一般神经网络、深度学习、3D 模型方法、Gabor、弹性束图匹配^[4-5]等也被用于人脸识别。

对于考勤系统的人脸识别算法,有一个要求是速度,虽然不需要实时,但是也不可耗费过长时间。因此本文使用一种快速的面部特征点描述算法,也就是将快速特征点提取算法 FAST 和快速特征点描述算法 Brief 应用于人脸识别,一方面采用局部匹配的策略,提高识别速度,另一方面在判断特征点是否匹配问题上,采取多项约束,确保匹配精确度。我们采用这 2 种算法有两个原因,一是它们的运算速度很快,满足考勤系统的要求,二是后文系统实现所采用的 tracking.js 有这 2 种算法的实现。

FAST 全称是 Features From Accelerated Segment Test。Rosten 等人将使用 FAST 寻找特征点(或称角点)的过程定义为:若某像素点与其周围领域内足够多的像素点处于不同的区域,则该像素点可能为特征点,也就是某些属性与众不同,考虑灰度图像,即若该点的灰度值比其周围领域内足够多的像素点的灰度值大或者小,则该点可能为特征点。具体流程为:

- (1)在图像中任选一点 p , 其像素值为 l_p ;
- (2)以 3 为半径画圆,覆盖 p 点周围的 16 个像素;
- (3)设定阈值 t ,如果这周围的 16 个像素中有连续

的 n 个像素的像素值都小于 $l_p - t$ 或者有连续的 n 个像素都大于 $l_p + t$, 那么这个点就被判断为特征点。本文中使用的 tracking.js 中 t 取值为 40, n 取值为 16。

FAST 不产生多尺度特征而且 FAST 特征点没有方向信息,这样就会失去旋转不变性。但是这个缺点在人脸识别中问题不大,我们可以人脸的方向和摄像头的方向都是固定的。

找到 FAST 特征点后,用 Brief 进行描述,为这些特征点建立描述子。Brief 的全称是 Binary Robust Independent Elementary Features, 它的主要任务是是对已检测到的特征点进行描述,它是一种二进制编码的描述子,主要应用于视觉目标跟踪、图像配准、目标定位和图像融合等领域,由于其能够快速建立特征点的描述信息,并且得到的是二值的特征向量,通过计算汉明距离来判断特征点之间的匹配程度,因此更加适合计算机的快速计算。主要思想是在特征点附近选择一个区域,并且在此区域里随机选择若干对像素点,通过比较像素点对的灰度值大小来建立一个二进制串,并将这个二进制字符串作为特征点的描述子。由于摒弃了利用区域灰度直方图描述特征点的传统方法,大大加快了特征描述符建立的速度,同时也极大地降低了特征匹配的时间。具体流程为:

(1)为减少噪声干扰,先对图像进行高斯平滑滤波。方差 σ 参数的取值一般为 0 到 3,滤波窗口有文献指出为 7×7 或 9×9 时效果较好。

(2)假设 p 为图像中的一个特征点,以 p 为中心,取 $S \times S$ 的邻域窗口。在窗口内随机选取两个点 x 和 y , 在这个邻域内,定义一个 τ 测试,比较二者像素的大小:

$$\tau(p; x, y) = \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{if } p(x) \geq p(y) \end{cases}$$

其中, $p(x)$, $p(y)$ 分别是随机点 $x=(u_1, v_1)$, $y=(u_2, v_2)$ 在图像经过平滑后的灰度值。

(3)在窗口中随机选取 N 对随机点,重复步骤 2 的二进制赋值,形成一个 N 维的二进制比特串,这个比特串就是对特征点的描述,即特征描述子。 N 一般取 128、256 或 512。

经过上面的 BRIEF 特征描述算法,对于一幅图中的每一个特征点,都得到了一个 N 比特的二进制编码,接下来对有相似或重叠部分的两幅图像进行匹配。Brief 利用汉明距离进行判决,对两个比特串进行异或运算,

并统计结果为 1 的个数,这个数就是汉明距离。一幅图上特征点与另一幅图上特征编码对应比特位上相同元素的个数最多的特征点配成一对。两个特征编码对应比特位上相同元素的个数小于 $N/2$ 的,一定不是配对的。

2 系统流程

我们希望建立一个 Web 的人脸识别考勤系统,因此本文使用 tracking.js 库中的 FAST 和 Brief 算法实现。tracking.js 是一个独立的轻量级 JavaScript 库。它可以利用 HTML5 技术通过浏览器来获取图像、视频或摄像头数据,并运行机器视觉算法,如目标跟踪、特征匹配等,也可以对其进行计算机图形学处理,如在视频图像上绘制图形等。

通过第 2 部分的分析,系统的工作过程分为两阶段:第一阶段,特征提取阶段。对于没有在数据库中存储特征的新人,使用 FAST 和 Brief 算法为其建立特征描述子——一个 N 维的二进制比特串,同其他的相应信息,如姓名等,一同存储在数据库中;第二阶段,特征匹配阶段,当通过浏览器获取到人脸图像时,使用 FAST 和 Brief 算法为其建立特征描述子,同数据库中已经存储的比特串进行异或运算匹配,从而判断某人是否出勤,完成考勤。由于考勤只是在一定范围内进行,如一个班级、一个部门、一个公司等,不用担心遍历数据库造成的效率降低。实际部署中,特征数据可以由 Web 服务器在浏览器端登陆的时候通过 WebSocket 发送到前端,特征描述子的汉明距离计算匹配过程在前端通过 JavaScript 完成;也可以存储在位于 Web 后端的数据库中,浏览器在前端生成特征描述子后,通过 WebSocket 发送至 Web 服务器,Web 服务器计算汉明距离并将匹配结果返回。前者的优势在于减少 Web 服务器端压力,后者的好处一是避免大量数据的传送,二是便于数据库的更新,无论在哪里登录并使用 Web 人脸识别考勤系统,特征数据都是最新的。本文中,由于数据量较小,使用方案一,汉明距离的计算放在前端通过 tracking.js 来进行。整个算法流程见图 1:

tracking.js 中所用到的一些重要函数如下:

(1)使用 FAST 算法的函数是:

```
var corners = tracking.Fast.findCorners (pixels, width, height);
```

pixels 指的是一个灰度图线性数组,可以用 tracking.js 相关函数处理得到,返回的是特征点或称角点的坐标数组 $[x_0, y_0, x_1, y_1, \dots]$, (x_0, y_0) 即表示寻找到的一个特征点。

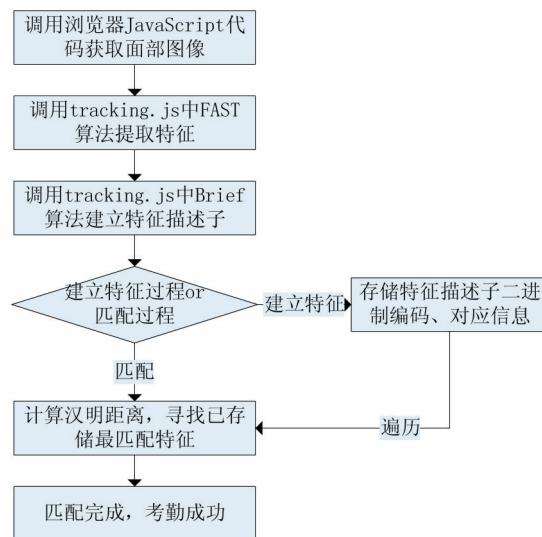


图1 基于FAST和Brief算法的Web人脸识别考勤算法流程

(2)使用 Brief 算法得到描述子的函数是:

```
var descriptors1 = tracking.Brief.getDescriptors (pixels, width, corners1);
```

```
var descriptors2 = tracking.Brief.getDescriptors (pixels, width, corners2);
```

pixels 指的是一个灰度图线性数组, corners1 和 corners2 指两幅图像 1 和 2 之前提取出的特征点数组,返回的是一串 N 比特的二进制编码。本文使用的 tracking.js 中 N 取值为 512。

(3)匹配的函数是:

```
var matches = tracking.Brief.match (corners1, descriptors1, corners2, descriptors2);
```

corners1、descriptors1 是图像 1 的特征点数组和描述子数组, corners2、descriptors2 是图像 2 的特征点数组和描述子数组,实现方法即用的汉明距离,tracking.js 的实现考虑到人脸识别的特殊性,进行了局部匹配的优化,返回一个数组,其中每项是两幅图像中匹配的特征点及其坐标。假设图像 1 中有 m 个特征点,图像 2 中有 n 个特征点,返回的数组为 o 项,即 o 个特征点匹配,当 $o/\min\{m,n\} > 0.75$ 时,我们认为图像 1 和图像 2 匹



图2 Web人脸识别考勤系统工作过程

配,0.75 是一个经验值。

3 实验

我们请来 5 位同学帮助我们进行实验,为避免噪声干扰 FAST 产生不必要的特征点,我们将摄像头面向墙壁,墙壁白色,没有其他物体。tracking.js 中为 FAST 算法 t 取值为 40, n 取值为 16。Brief 算法 N 取值为 512。部署好系统后,浏览器通过摄像头获取人头部图像并使用 FAST 寻找特征点,再使用 Brief 算法建立特征描述子并匹配的过程见图 2。

基于隐私考虑,截图未使用同学的实际面部图像。我们使用了网络上的图片来说明工作过程,图 2 中的左图是原图,中图是提取出的特征点,右图是匹配过程,横线表示匹配上的特征点。

第一阶段,分别为 5 位同学建立特征描述子并存储下来。第二阶段,分别对每位同学进行 20 次人脸识别,同时使用 JavaScript 记录主要函数的总体运行时间,结果见如表 1。

从表 1 的结果可以看出,识别率已经到了基本可用的程度,运行时间也在可接受范围内。所以我们认为 Web 人脸识别考勤系统已经具有了一定的实用性。

表 1 正确识别次数及运行时间

实验者	识别正确次数	实验次数	识别正确率 (%)	运行时间 (s)
同学 1	18	20	90	0.724
同学 2	20	20	100	0.679
同学 3	17	20	85	0.821
同学 4	19	20	95	0.763
同学 5	19	20	95	0.758

4 结语

本文使用 tracking.js 提供的 FAST 和 Brief 算法实现了一个 Web 人脸考勤系统,通过为人面部图像建立特征描述子再匹配来实现考勤。同传统方法相比,本文的方法优势在于对软硬件设备要求低,只要有一台带有摄像头并可以访问网络的设备即可进行。实验表明,系统在具有较高识别准确率的情况下也保证了运行速度,已经具有了一定的实用性,可以应用到实际生产生活中。

后续的研究方向是继续提高识别准确率,并且在复杂背景下如何排除干扰,提取正确有效特征。

参考文献:

- [1]Kirby M, Sirovich L. Application of the Karhunen–Loeve Procedure for the Characterization of Human Faces[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(1): 103–108.
- [2]Belhumeur P N, Hespanha J P, Kriegman D J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, 19(7): 711–720.
- [3]Ojala T, Pietikainen M, Maenpaa T. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 971–987.
- [4]Sun Y, Wang X, Tang X. Deep Learning Face Representation from Predicting 10,000 Classes[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1891–1898.

[5]Liu C, Wechsler H. Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition[J]. IEEE Transactions on Image Processing, 2002, 11(4): 467-476.

作者简介:

徐辉(1994-),男,蚌埠学院计算机科学与技术系 2013 级软件工程

陈晨(1988-),女,蚌埠学院助教,研究方向为数字图像处理、模式识别

收稿日期:2017-04-21 修稿日期:2017-07-12

Web Face Recognition Attendance System Using tracking.js Based on Fast and Brief Algorithm

XU Hui, CHEN Chen

(Bengbu University, Bengbu 233030)

Abstract:

Realizes a Web face recognition attendance system, this system uses HTML5 technology to obtain the human face image through the browser, then introduces the tracking.js library, uses the FAST and Brief algorithm to realize the quick feature point extraction, description and matching of the biological features of the human face image, and finally completes the attendance process. These HTML, JavaScript code, and image feature data can be deployed on an HTTP server and accessed through the Web to build a B/S based face recognition attendance system. In any place, as long as there is a device with camera which can connect the Web server equipment, attendance can be quickly processed. Compared to existing programs, this system has the advantages of low price, quick access and so on.

Keywords:

FAST ; Brief; tracking.js; Face Recognition