计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(12):37−42 [doi: 10.15888/j.cnki.csa.006100]

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

基于人脸的活体检测系统①

张高铭, 冯　瑞

(复旦大学 计算机科学技术学院, 上海 201203)

摘　要: 人脸识别技术由于其成本低、用户友好、效率高等特点被广泛应用, 同时也出现了针对人脸识别的身份伪造攻击, 主要包括照片人脸攻击、视频人脸攻击、三维人脸模型攻击等方式, 对于这些攻击方式的防范方法都是围绕着基于人脸的活体检测这个中点进行展开. 本文着重研究的活体检测方法为眨眼检测与背景分析算法, 通过区域增长算法进行人眼定位、形态学操作进行人眼张合判断、感知 Hash 算法进行背景差异对比, 构造出一个复合的活体检测系统. 基于复合的眨眼检测与背景分析算法, 本文设计了一个包含眨眼检测模块与背景分析模块的活体检测系统, 使用 OpenCV2.4.9 与 vs2012 的 MFC 架构实现了一个可以抵御照片攻击与视频攻击的活体检测系统, 并对系统进行实验与评估, 在与其它同类型的系统进行比较的结果来看, 本文实现的系统性能表现优异.

关键词: 人脸识别; 活体检测; 眨眼检测; 人眼定位; 环境背景分析

引用格式: 张高铭,冯瑞.基于人脸的活体检测系统.计算机系统应用,2017,26(12):37–42. http://www.c-s-a.org.cn/1003-3254/6100.html

Liveness Detection System Based on Human Face

ZHANG Gao-Ming, FENG Rui

(School of Computer Science and Technology, Fudan University, Shanghai 201203, China)

Abstract: The face recognition technology is widely used for its low cost, user-friendly and high efficiency. At the same time, identity forgery attack has also been the corresponding occurrence. The face recognition system attacks include photo face attacks, video face attacks and three-dimensional face model attacks, etc. For these attacks, prevention methods are carried out around the midpoint of in liveness detection based on human face. This paper focuses on the blink detection and background analysis algorithm, and carries out eye location with regional growth algorithm. The morphological operation is used to judge the human eye state, and the Hash algorithm is used to compose the background difference. These methods construct a Liveness Detection Systems. Based on the blink detection and background analysis algorithm, this paper designs a liveness detection system including blink detection module and background analysis module; uses the MFC architecture and OpenCV2.4.9 to build a liveness detection system which can resist photo attack and video attack; makes the experiment and evaluation of the system. In comparison with other similar types of systems, the system performance of this paper is excellent.

Key words: face recognition; liveness detection; blink detection; eye localization; environmental background analysis

随着计算机视觉技术的飞速发展, 基于人脸识别的身份验证系统正在被广泛地应用在各个场所[1], 包括门禁系统, 软件登录系统, 人群监测系统等. 在人脸识别技术广泛应用的背景下, 针对人脸识别系统的身份伪造攻击也相应地逐渐出现, 最主要的攻击手段包括照片人脸攻击与视频人脸攻击. 对于这些身份伪造攻击手段, 主要采用的防治方法为基于人脸的活体检测技术. 比如分析人脸旋转时的面部光流分析法[2], 通过

① 基金项目: 临港地区智能制造产业专项 (ZN2016020103)
收稿时间: 2017-03-07; 修改时间: 2017-03-27; 采用时间: 2017-04-07

分析人脸旋转时产生的光流特征判别活体与照片视频; 傅利叶频谱与纹理分析法[3], 通过分析活体人脸与照片、视频人脸的傅利叶频谱以及纹理特征的差异来区分活体用户与伪造攻击; 动态纹理分析法[4], 通过动态分析人脸的纹理特征来区别活体人脸与二维人脸. 以上介绍的基于特征的检测方式对特征提取方式的依赖较大, 当环境变化时, 特别是光照条件差异悬殊时, 特征提取效果就较为不理想, 从而影响活体检测的结果, 所以本文另辟蹊径, 从行为方面着手进行活体检测. 对人对于照片人脸攻击, 可以使用基于眨眼检测的方式进行判别; 对于视频人脸攻击, 则可以使用背景分析的方式判别. 使用眨眼检测与背景分析法, 构造一个复合的活体检测系统, 只有当待检测用户同时通过眨眼检测模块与背景分析模块, 才认为待检测用户通过活体检测系统, 否则不通过. 在实际的应用中, 这样的活体检测系统可以有效地检测照片人脸攻击与视频人脸攻击.

1 系统概述

复合活体检测系统主要分成两个大的模块, 一个是环境背景分析模块, 一个是眨眼检测模块, 各模块的功能与系统框架如图 1 所示.

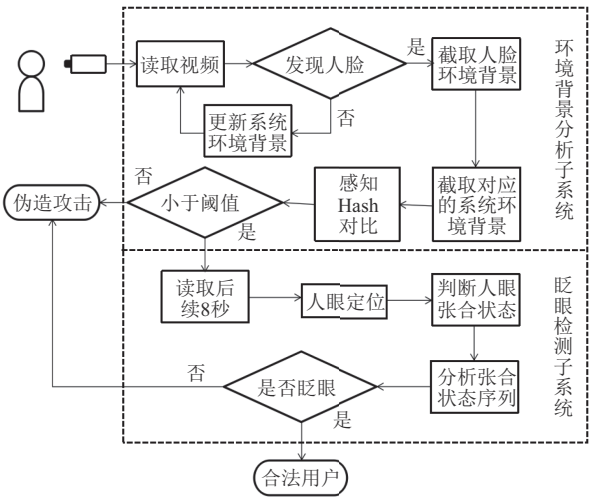


图 1 系统流程图

环境背景分析模块: 分析人脸环境与系统环境背景的区域图像是否一致, 一致, 则环境背景分析检测通过, 否则拒绝.

眨眼检测模块: 分析待检测对象在规定时间内是否完成眨眼动作, 若在一定时间内未眨眼, 则认为待检测对象为照片攻击.

2 系统各模块的设计与实现

本系统的实现主要基于表 1 和表 2 的硬件与软件系统

表 1 硬件配置

设备名称	型号
中央处理器	Intel core i5-3470
摄像头	Logitech PRO C920

表 2 软件配置

软件名称	配置
操作系统	Windows 8
集成开发环境	Visual Studio 2012
依赖库	OpenCV 2.4.9

2.1 环境背景分析模块

环境背景分析[5,6]的原理是, 系统初始化时先保存摄像头前的图像作为系统背景, 当检测到人脸时, 提取人脸周围区域的图像, 并提取系统环境对应区域的图像, 进行比较, 判断人脸背景与系统背景的一致性.

环境背景分析模块主要包括几个子模块:

① 读取视频模块.

② 系统背景更新模块.

③ 人脸环境截取与系统环境截取模块.

④ 感知 Hash 值对比模块.

2.1.1 读取视频

OpenCV2.4.9 提供的类库屏蔽了繁杂的硬件层的工作, 使用其封装的类可以轻松完成读取视频的工作, 其中主要用到的类:

cv::VideoCapture;

这个类封装了读取视频流的方法, 包括读取文件视频流以及摄像头视频流, 在类构造函数中传入 0, 则可以获取默认摄像头的视频流. 之后每隔一段时间就调用该类的方法:

VideoCapture::bool read(cv::Mat& image);

就可以将视频流的一帧读入到 cv::Mat 类型的 image 变量中, 将图像绘制到系统界面上, 则可以显示视频流图像.

2.1.2 系统背景更新模块

对于系统环境背景, 设置一个变量 cv::Mat background 表示, 系统刚启动时, 将视频的第一帧赋值给 background 变量, 此后每一帧判断是否检测到人脸, 若未检测人脸, 判断距离上次更新环境背景的时间是

否超过 60 秒, 是, 则将当前帧图像拷贝给 background:

background=img.clone();

其中 img 表示当前帧.

2.1.3 人脸环境截取与系统环境截取

当摄像头传来的当前帧检测到人脸时, 提取人脸周围区域的图像, 并提取系统背景相对应区域的图像. 如图 2, 由于在人脸照片或人脸视频中, 人脸的背景与系统的背景不相同, 所以通过检测背景的一致性, 就可以判断是否遭受到照片人脸攻击或视频人脸攻击了.

(a) 活体人脸周围背景 (b) 对应区域的系统背景

(c) 视频人脸周围背景 (d) 对应区域的系统背景

图 2 人脸环境背景与系统环境背景

2.1.4 感知 Hash 值比对

提取出系统与人脸的背景图像, 接下来就要进行图像的相似度比较, 这里使用感知 Hash 算法[7,8], 算法的主要过程如下:

① 将图像缩小到 8×8 尺寸, 目的是为了去除图像的细节, 只保留大体的结构以及明暗等信息.

② 将缩小尺寸后的图像转换为 64 级灰度级.

③ 计算 64 个像素的平局值.

④ 将每一个像素与平均值进行比较, 大于平均值则记为 1, 否则记为 0.

⑤ 按照像素的位置顺序排列 64 个 0 或 1 数字, 得到一个 64 位的二进制数, 被称为图像的“指纹”.

比较两张图像 Hash 值的汉明距离, 如图 3, 距离越小, 图像相似度越高; 距离越大, 图像相似度越低.

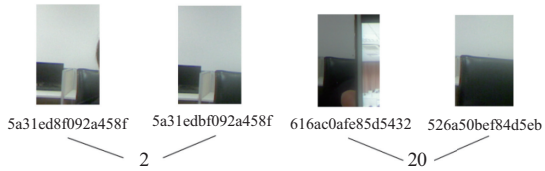


图 3 相似图像与不相似图像的 Hash 值差异

2.2 眨眼检测模块

眨眼检测模块的主要工作原理是, 活体人眼会进行不自觉的眨眼动作, 而照片等静态人眼图像是无法进行眨眼动作的, 所以进行眨眼检测, 可以有效地区分照片攻击与合法的用户登录.

眨眼检测主要有以下几个子模块:

① 人眼定位模块.

② 人眼张合状态判断模块.

③ 张合状态序列分析模块.

2.2.1 人眼定位

使用区域增长算法[6,7]进行人眼定位, 该算法的主要流程:

① 通过人脸检测模块首先估计出鼻尖的坐标位置 (x_0,y_0), 估计鼻尖-瞳孔的矩形框初始大小为w_0、h_0, 并定义阈值 D.

② 设矩形框初始的左下角为 (x_0,y_0), 矩形框的初始宽度为w_0, 初始高度为 h_0, 计算初始平局灰度值 $I_{mean(0)}$.

$$I_{mean(0)}=\frac{1}{w_0\cdot h_0}\sum\nolimits_0$$

$$\sum\nolimits_0=\sum_{\substack{x_0\le x<x_0+w_0\\ y_0\le y<y_0+h_0}} I(x,y)$$

③ 进行迭代, 在 i+1 步, 矩形框保持宽高比例进行固定, 左下角坐标固定, 进行向右、向上增大, 计算每次迭代新的平均灰度值.

$$\sum\nolimits_{i+1}=\sum\nolimits_i+\sum_{(x,y)\in L_{new}} I(x,y)$$

$$I_{mean(i+1)}=\frac{1}{w_{i+1}\cdot h_{i+1}}\sum\nolimits_{i+1}$$

其中L_{new}表示在第 i+1 次迭代时新增加的像素集合.

④ 平滑处理矩形框, 找到矩形框I_{i+1}中灰度值最小的像素$I_{i+1}(x',y')$.

⑤ 计算若$I_{mean(i+1)}-I_{i+1}(x',y')<D$, 重复③、④步.

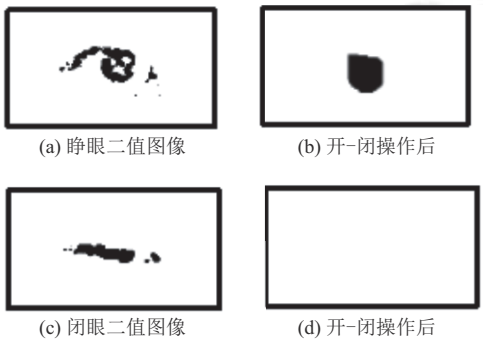
⑥ 在点(x',y')附近设置搜索框, 利用一个领域最小模板找到瞳孔的位置, 然后截取一个固定尺寸的矩

形, 即为人眼区域.

2.2.2 人眼张合状态判断

提取人眼区域中的人眼图像, 做灰度值转化、自适应阈值二值化后, 就得到一个人眼区域的二值图像, 分析人眼张开与闭合时的形态: 当人眼张开时, 眼珠露出, 二值图像中黑色区域较大; 人眼闭合时, 人眼二值图像只有一条狭窄的眼缝黑色区域.

先对人眼图像做一次形态学开操作[11], 消除眼珠中的高亮区域; 再做一次闭操作, 消除狭窄的眼缝区域; 如图 4, 形态学操作后图像中黑色像素的个数, 根据人眼张合状态的不同呈现出巨大的差异, 所以可以统计操作后黑色像素的个数来判断人眼的张合状态.



(a) 睁眼二值图像 (b) 开-闭操作后

(c) 闭眼二值图像 (d) 开-闭操作后

图 4 不同状态人眼形态学操作后

2.2.3 张合状态序列分析

通过对一系列帧人眼图像的张合状态判断, 得到一个人眼张合状态的序列.

定义一个人眼张合状态的集合 Q:

$$Q=\{O,C\}$$

其中, O 代表睁眼状态, C 代表闭眼状态. 则两次眨眼动作包含以下序列:

$$O\to\cdots\to C\to\cdots\to O\to\cdots\to C\to\cdots\to O$$

在规定时间内若检测到张合状态序列中包含以上子序列, 则认为眨眼检测通过, 否则不通过.

3 系统展示及性能评估

3.1 系统展示

通过将系统各个模块组织起来, 利用 VS2012 与 OpenCV 库, 实现了一个基于 Windows 系统的活体检测

系统应用程序, 其主界面如图 5 所示, 界面上有包括显示实时视频图像区域、背景分析区域以及眨眼检测区域, 其中背景分析区域与眨眼检测区域包含参数设置以及中间结果显示的功能. 两个模块任一个检测结果为拒绝则活体检测结果为拒绝, 两个模块检测结果均为通过活体检测为通过.

3.2 活体检测系统行为实验

3.2.1 照片攻击

架好摄像头并开启活体检测系统, 调整参数后, 将包含人脸的静态照片放在摄像头前, 观察系统的反馈行为 (图 6).

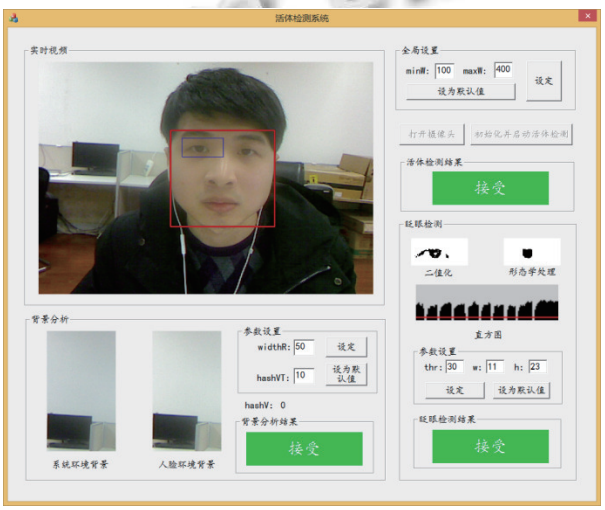


图 5 系统界面

如图 6, 当数码照片放置在摄像头前时, 首先观察背景分析的结果, 通过系统界面可以看到, 人脸环境背景与系统环境背景的对比值为 13, 超过阈值 5, 所以背景分析模块给出了拒绝的检测结果.

图 6 数码照片攻击

对于眨眼检测模块, 从直方图上可以看出, 由于静态

照片的人眼不发生眨眼行为, 所以直方图上没有表现出明显的波谷, 所以眨眼检测模块也给出了拒绝的检测结果.

综合眨眼检测模块与背景分析模块, 活体检测系统的检测结果为拒绝.

另一种照片攻击的方式是使用实体照片, 将照片打印出, 并沿着人脸轮廓剪下, 用裁剪下的人脸进行攻击, 如图 7 所示.

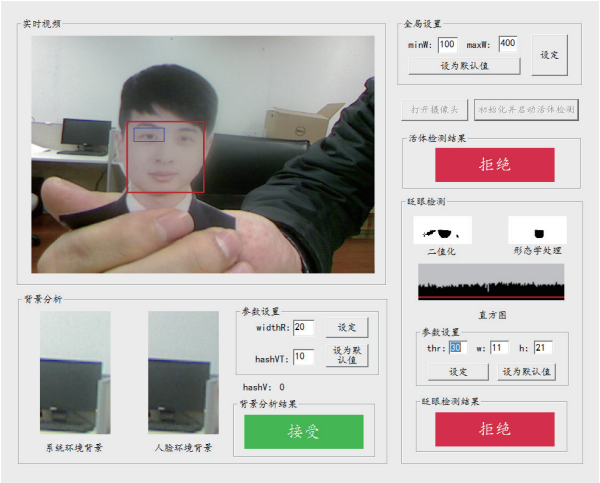


图 7 裁剪人脸攻击

由于使用裁剪的人脸, 不同于数码照片, 人脸周围的环境图像与系统环境图像相同, 环境分析给出了接受的检测结果, 同时眨眼检测的结果为拒绝, 所以活体检额的综合结果为拒绝.

3.2.2 视频攻击

将活体检测系统启动后, 将带有眨眼动作的人物视频在摄像头前播放, 观察系统的反馈行为 (图 8).

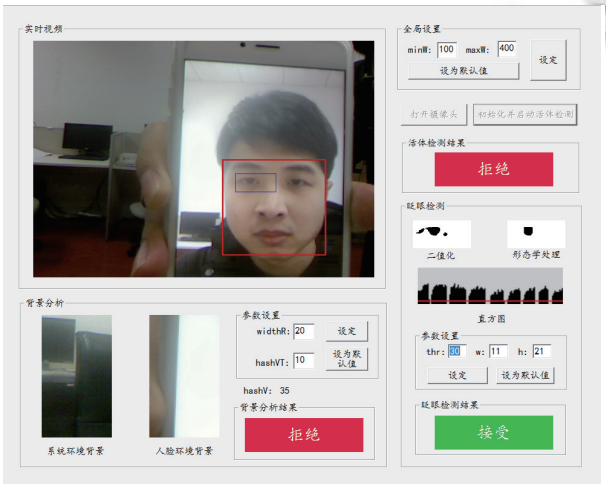


图 8 视频攻击

如图 8, 当包含眨眼动作的视频放置在摄像头前时, 首先观察背景分析的结果, 通过系统界面可以看到, 人脸环境背景与系统环境背景的对比值为 35, 超过阈值 10, 所以背景分析的结果为拒绝.

再观察眨眼检测的结果, 由于视频人脸拥有眨眼行为, 通过观察直方图可以看出, 直方图上有许多低谷, 代表人眼闭合的瞬间. 通过分析人眼状态, 眨眼检测模块给出了接受的结果.

由于背景分析与眨眼检测有一个模块给出了拒绝的结果, 所以活体检测系统给出的整个检测结果为拒绝.

3.2.3 合法登录

活体检测系统启动后, 合法的活体用户在摄像头前保持几秒, 观察系统的反馈行为 (图 9).

如图 9, 当合法活体用户站在摄像头前时, 首先观察背景分析的结果, 通过系统界面可以看到, 人脸环境背景与系统环境背景的对比值为 0, 低于阈值, 所以背景分析的结果为接受.

图 9 合法登录

再观察眨眼检测的结果, 由于活体用户拥有眨眼行为, 通过观察直方图可以看出, 直方图上有许多低谷, 代表人眼闭合的瞬间, 通过分析人眼状态, 眨眼检测模块给出了接受的结果.

背景分析与眨眼检测均给出了接受的结果, 所以活体检测系统给出的整个检测结果为接受.

3.3 性能评测

为了测试结合了眨眼检测与环境背景分析的复合活体检测系统, 使用 100 张照片、20 段视频以及 20 次合法登录, 记录并分析结果, 如表 3 所示.

通过实验, 对活体检测系统进行 120 次照片、视频攻击, 116 次正确拒绝, 4 次错误接收, 错误接受率(FAR): 3%. 对活体检测系统进行 20 次合法登录, 1 次错误拒绝, 19 次正确接受, 错误拒绝率 5%. 故系统的等错误率 (EER): 4%. 总体来说, 本算法在能以较低的错误接受率来识别照片攻击与视频攻击的同时, 能以较低的错误拒绝率来辨别合法的人脸识别登录, 性能良好.

表 3 照片攻击、视频攻击及合法登录对比

登录方式	照片攻击	视频攻击	合法登录
登录次数	100	20	20
拒绝次数	98	18	1
接受次数	2	2	19

为了测试系统的鲁棒性, 在不同光照条件下对系统进行实验, 结果如表 4 所示.

表 4 不同光照条件下的等错误率

光照	白天(亮)	傍晚(中)	夜晚(暗)
等错误率(%)	4	6	85

可以看出, 系统在光线较为充足的情况下表现良好, 而对于光线很暗的环境里, 系统无法工作.

为了对本系统的性能有个具体的了解, 与同类型的其它文献所提出的方法进行比较, 情况如表 5.

表 5 各系统等错误率对比

系统	文献[2]	文献[3]	文献[4]	本系统
等错误率(%)	7	6.5	4	4

表 5 中, 文献[4]使用的是动态纹理法, 文献[3]采用傅利叶频谱与纹理分析结合的方法, 文献[2]采用的是面部光流法, 文从对比中可以看出, 本系统的等错误率比文献[2]、文献[3]都要高, 而与文献[4]相同, 在同类系统中处于相对优秀的水平.

4 结语

本文探讨了对人脸识别的攻击手段, 并针对照片攻击与视频攻击设计、实现了一个活体检测系统, 包括背景分析与眨眼检测模块.

对于背景分析模块, 关键技术有背景更新、背景提取、背景比对, 对于眨眼检测模块, 关键技术包括人眼定位、人眼张合判断、张合状态序列分析等. 利用 VS2012 以及 OpenCV 开源库, 实现了一个基于 Windows 操作系统的桌面应用程序, 在对系统的实验测评结果来看, 本系统在同类系统中表现较好.

参考文献

1 Chellappa R, Wilson CL, Sirohey S. Human and machine recognition of faces: A survey. Proc. of the IEEE, 1995, 83 (5): 705–741. [doi: 10.1109/5.381842]

2 Smiatacz M. Liveness measurements using optical flow for biometric person authentication. Metrology and Measurement Systems, 2012, 19(2): 257–268.

3 Kim G, Eum S, Suhr JK, *et al*. Face liveness detection based on texture and frequency analyses. 2012 5th IAPR International Conference on Biometrics. New Delhi, India. 2012. 67–72.

4 de Freitas Pereira T, Komulainen J, Anjos A, *et al*. Face liveness detection using dynamic texture. EURASIP Journal on Image and Video Processing, 2014, 2014: 2.

5 Yan JJ, Zhang ZW, Lei Z, *et al*. Face liveness detection by exploring multiple scenic clues. 2012 12th International Conference on Control Automation Robotics & Vision. Guangzhou, China. 2012. 188–193.

6 Komulainen J, Hadid A, Pietikäinen M, *et al*. Complementary countermeasures for detecting scenic face spoofing attacks. 2013 International Conference on Biometrics. Madrid, Spain. 2013. 1–7.

7 Buldas A, Kroonmaa A, Laanoja R. Keyless signatures' infrastructure: How to build global distributed hash-trees. Proc. of the 18th Nordic Conference on Secure IT Systems. Ilulissat, Greenland. 2013. 313–320.

8 Phash POHO. pHash.org: Home of pHash, the open source perceptual hash library. Phash.

9 Yuille AL, Hallinan PW, Cohen DS. Feature extraction from faces using deformable templates. International Journal of Computer Vision, 1992, 8(2): 99–111. [doi: 10.1007/BF00127169]

10 Deng JY, Lai FP. Region-based template deformation and masking for eye-feature extraction and description. Pattern Recognition, 1997, 30(3): 403–419. [doi: 10.1016/S0031-3203(96)00086-6]

11 Mohammed AA, Anwer SS. Efficient eye blink detection method for disabled-helping domain. International Journal of Advanced Computer Science & Applications, 2014, 5(5): 202–206. [doi: 10.14569/IJACSA.2014.050530]