

应用 Camshift 跟踪算法提高视频中人脸检测速度^①



孙凯旋

(西南交通大学 电气工程学院, 成都 610031)
通讯作者: 孙凯旋, E-mail: kxsun617@163.com

摘要: 在 OpenCV 中提供的 CascadeClassifier 级联分类器利用 Haar 特征进行人脸检测时, 检测速度很慢无法满足视频对实时性的要求, 而且光照的影响也很大. 基于这两点提出一种新的人脸检测算法, 采用 Camshift 目标跟踪与人脸检测相结合提高检测速度并利用直方图均衡化减弱光照的影响. 该算法首先把 CascadeClassifier 级联分类器方法检测到的人脸区域设为 ROI 区域, 对 ROI 区域操作并用 Camshift 算法进行目标跟踪, 其次要定时进行一次人脸检测用来更新 ROI 区域保证跟踪的准确性. 经过实验结果分析表明: 利用改进后的算法, 人脸检测的速度有明显提高 (约为 40%), 并且减小了光照的影响.

关键词: 人脸检测; OpenCV; Haar 特征; Camshift; 图像处理

引用格式: 孙凯旋. 应用 Camshift 跟踪算法提高视频中人脸检测速度. 计算机系统应用, 2020, 29(9): 231-236. <http://www.c-s-a.org.cn/1003-3254/7645.html>

Improving Face Detection Speed in Video Using Camshift Tracking Algorithm

SUN Kai-Xuan

(School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: When the CascadeClassifier cascade classifier provided in OpenCV uses Haar features for face detection, the detection speed is too slow to meet the real-time requirements of the video, and the impact of lighting is also great. Based on these two points, a new face detection algorithm is proposed, which uses Camshift target tracking and face detection to improve the detection speed and uses histogram equalization to reduce the impact of light. The algorithm first sets the face area detected by the CascadeClassifier cascade classifier method as the ROI area, operates on the ROI area and uses the Camshift algorithm for target tracking, and secondly performs face detection regularly to update the ROI area to ensure the tracking accuracy. The analysis of the experimental results shows that: with the improved algorithm, the speed of face detection has been significantly increased (about 40%), and the impact of light is reduced.

Key words: face detection; OpenCV; Haar features; Camshift; image processing

人工智能正在影响着人类的生产和生活, 对视频的处理随处可见. 其中, 人脸检测是数字图像处理领域的重要研究领域^[1], 是很多涉及人脸的应用程序的基础^[2], 研究和处理人脸在视频监控中的情况是一项非常有前景的方向^[3]. 但在实践中人脸识别技术还存在一些不

足, 如人脸检测速度较慢^[4], 因此研究提高人脸检测的快速性和准确性是非常有意义的^[5].

OpenCV 是常见的人脸检测工具, OpenCV 被广泛应用于人机互动、物体识别、图像分割、运动跟踪、机器人、机器视觉、汽车安全驾驶等领域^[6]. AdaBoost

① 收稿时间: 2020-02-26; 修改时间: 2020-03-17, 2020-04-24; 采用时间: 2020-04-28; csa 在线出版时间: 2020-09-04

(Adaptive Boosting) 方法是基于学习, 借助 Haar 特征, 把一组弱分类器提升为强分类器, 级联检测效率较高^[7], 相对于其他的检测方法效果较好, 但是还存在着检测率和检测效率不足的问题^[8]. OpenCV 自带工具中, 其用于筛选大量 Haar 特征和生成级联分类器的训练阶段非常繁琐且需要大量计算^[1]. 传统人脸检测方法速度较慢, 很难满足对实时性要求较高的场合. 基于检测速度的提高, 本文提出在人脸检测中应用跟踪算法, 实用的目标跟踪算法不仅要求能够准确实时地跟踪目标, 而且要求在光照变化、目标相似等复杂的背景下也能有较强的抗干扰能力^[9]. 目标跟踪是计算机视觉领域发展较快的一个方向. 近些年, Meanshift 算法在目标跟踪方面应用广泛, 但也有不足之处, 它无法对目标模型进行更新. 为解决该算法无自适应问题, Bradski 在 Meanshift 算法的基础上提出基于颜色概率信息的 Camshift 算法^[10].

本文将人脸检测算法检测出来的人脸区域作为 Camshift 算法的初始窗口^[11], 实现由每一帧的人脸检测转变为人脸区域跟踪以提高检测效率和跟踪实时性^[11]. 此外, 针对光照影响也做出了相应处理, 并在 VS 2017+OpenCV 3.4.1 实验环境下实验验证, 结果达到预期效果.

1 整体设计

在 OpenCV 视觉库中, 图片是以 Mat 数据类型存储的, Mat 类可以被看作是 OpenCV 中 C++ 版本的矩阵类, 对图片的操作就是对 Mat 对象的操作. 视频是由一帧一帧的图片组成, 所以处理视频也就是处理视频中的每一帧图片. 传统的人脸检测方法是对每一帧图像用级联检测器进行人脸检测, 然而对于采用 Haar 特征的检测方法, 由于大都是浮点数运算, 虽然相对于 LBP 特征检测的精度较高, 但是检测的速度很慢. 由于视频对实时性的要求很高, 传统的人脸检测方法并不能满足实际需求. 而目标跟踪算法可以实现准确的跟踪目标并且速度很快, 可以满足视频对实时性的需求. 因此将目标跟踪结合到人脸检测中以改进传统算法.

本文提出的算法利用了直方图均衡化消除光照影响并对检测到的人脸区域设为 ROI 区域然后进行跟踪, 同时要设定一个阈值, 每隔固定帧数进行一次人脸

检测, 用来更新数据.

整体流程如图 1 所示.

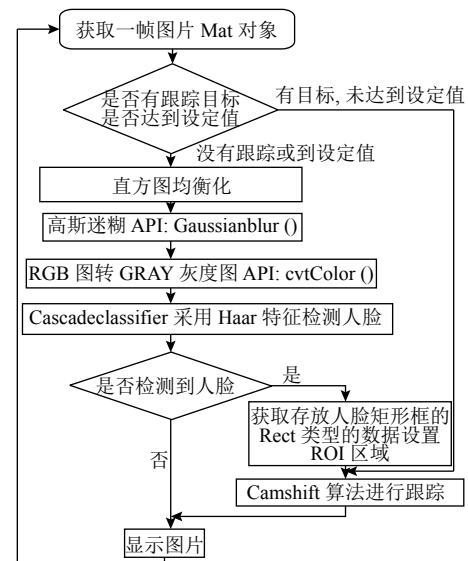


图 1 系统整体流程图

2 人脸检测

在 OpenCV 中 CascadeClassifier 是包含级联分类器的一个类, 用于人脸检测, 该分类器可以使用 Haar 和 LBP 两种特征. Haar 特征和 LBP 特征有以下区别:

(1) Haar 中数据是浮点数, 但 LBP 中数据是整数.

(2) LBP 训练需要的样本数量要比 Haar 大.

(3) 同样的样本空间, Haar 训练出来的数据检测结果要比 LBP 准确.

(4) LBP 的速度一般可以比 Haar 快几倍.

2.1 AdaBoost 算法

Haar 特征是一种矩形特征由 Papageorgiou 等提出, 并被 Viola 等应用到人脸识别系统中. 它的定义是图像中相邻区域内灰度值差的总和, 也就是相邻区域内白色像素减去黑色像素之差的和, 反应了相邻区域内白色区域到黑色区域的梯度变化趋势. Haar 矩形特征经过不断的扩展其特征数量不断的增加, 经过 Lienhart 等的不断努力, 现在 Haar 矩形特征集已经扩展到 16 种^[12] 如图 2 所示.

一种 Haar 特征就是一个弱分类器, 定义如下:

$$\eta = \begin{cases} 1, & \lambda f(x) < \lambda\omega \\ 0, & \lambda f(x) \geq \lambda\omega \end{cases}$$

其中, $f(x)$ 为 Haar 矩形特征值, λ 是调节符号, ω 定义为:

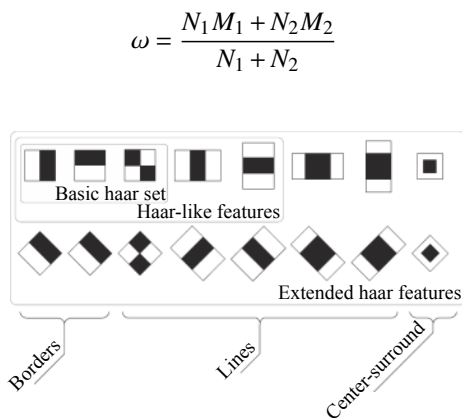


图2 Haar 据性特征集合

N_1 为正样本数, N_2 为负样本数, M_1 为正样本平均特征值, M_2 为负样本平均特征值, 如 $M_1 > \omega$, $\lambda=1$, 否则 $\lambda=-1$. 当 η 为 1 时, 判定为目标, 即检测到人脸目标, 否则, 未检测到目标^[7].

弱分类器的检测能力很弱, 将若干个弱分类器以线性方式组合可以形成强分类器. Adaboost 算法就实现了该过程, 其算法核心是对同一训练集来训练多个弱分类器, 最终对这些弱分类器进行结合组成分类能力更强的强分类器.

AdaBoost 算法的流程: (1) 增加其中被上轮弱分类器错误分类的样本的权重, 对于正确分类的样本要降低它们的权重. (2) 利用加权多数表决议. 具体做法是: 提升误差率较小的弱分类器的权重, 让它增加在表决议时的影响, 降低误差较大的弱分类器的权重, 减小在表决议时的影响.

2.2 CascadeClassifier 级联分类器

级联分类器是多个强分类器的组合, 对目标图像采用多个强分类器检测得到检测结果. CascadeClassifier 是 OpenCV 下 objdetect 模块中一个类, 可以实现对目标的检测. 实现方式是对级联分类器与滑动窗口机制进行结合, 支持利用 OpenCV 中的多种特征进行目标检测.

CascadeClassifier 实现了几个重要的方法, 其中包括 load() 和 detectMultiScale(). load() 用来加载已经训练好的模型, detectMultiScale() 则实现了多尺度的检测过程, 利用 CascadeClassifier 实现人脸检测的流程如图 3 所示.

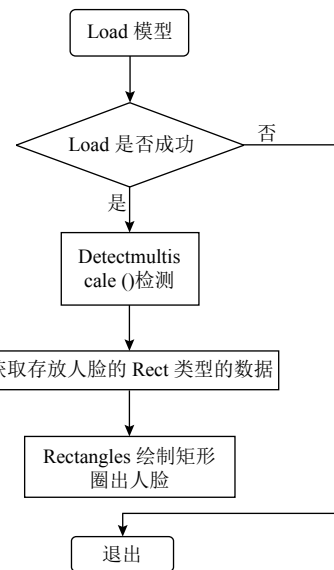


图3 人脸检测流程图

3 Camshift 目标跟踪

Camshift 是 Meanshift 跟踪算法的改进算法, 可以做到连续自适应, 能够实现自动调整窗口大小, 可以检测尺寸发生变化的目标. 该算法核心是对每帧图像均作 Meanshift 运算, 然后把上一帧的结果 (窗口中心位置与大小) 设为下一帧 Meanshift 算法的窗口的初始值, 按照此方法一直迭代下去. Camshift 算法最重要的一点是当目标发生形变的时候, 能够做到自适应的改变区域达到继续跟踪的目的.

3.1 反向投影 (BackProject)

反向投影是一种方式: 就是记录指定图像中每个像素是怎样适应直方图模板中像素布局的.

RGB 颜色空间对光照亮度变化敏感, 在目标跟踪中为了减少光照强度变化对目标跟踪的影响, 将目标图像由 RGB 颜色空间映射到 HSV 颜色空间进行处理. HSV 颜色空间中的 H 表示物体颜色色调, 它直观地反映了物体的颜色信息, 根据图像中 H 通道的颜色色调目标直方图, 计算得到目标图像的颜色概率查找表^[13].

利用 Hue 直方图实现反向投影步骤如下:

(1) 读取被测对象中的每个像素点的 Hue 数值, 同时在直方图找到数值的 bin 位置.

(2) 查询 Hue 直方图中对应 bin 的数值.

(3) 然后把这个值放在新的图像中存储, 同时可以把 Hue 直方图中的数据进行归一化处理 (0-255), 从而实现 BackProject 单通道图像直接显示.

(4) 通过对测试图像每个像素采取以上步骤, 可以得到最终的 BackProject 图像。

3.2 Meanshift 算法

Meanshift 算法的核心是 mean(均值)+shift(偏移). 假如空间上一点 x , 在其周围分布着很多个点 x_i 通过算出点 x 偏移到各个点的偏移量的总和, 并求出平均值, 该值就是平均偏移量, 该偏移量中包含大小和周围分布较密集的方向. 之后点 x 就会向着平均偏移量的方向进行移动, 然后以该点当做新的起点一直迭代到满足一定的条件为止。

如图 4 所示为均值+偏移的过程, 并最终找到中心点。

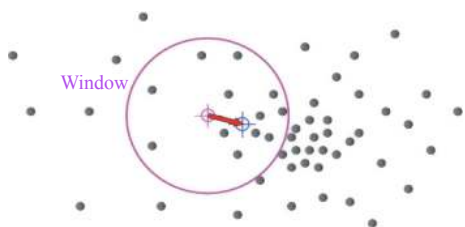


图 4 均值+偏移示意图

3.3 Camshift 算法

Camshift 算法就是对视频里的每一帧图像全部做 Meanshift 运算, 并且把计算出来的目标的大小及位置当做下一帧 Meanshift 算法中搜索框的大小起始值及它的中心位置. 利用迭代运算的形式实现对视频里的目标进行跟踪. Camshift 算法流程如图 5 所示。

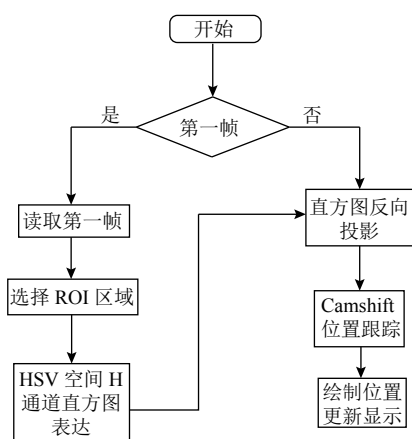


图 5 Camshift 算法流程图

4 图片预处理

在人脸检测之前非常重要的步骤就是对视频图片

做预处理, 去噪^[14] 是图片预处理很重要的一步. 去噪主要是去掉图像中会对检测结果有影响的噪点, 在光照不足的情况下会更容易产生这种噪点, 同时光照太强也会影响检测结果. 本文将采用高斯平滑进行去噪, 利用直方图均衡化来保持光照不变性, 同时为了保证检测的准确性还要在人脸检测之前对图像进行灰度处理。

4.1 高斯平滑

高斯平滑也叫高斯模糊, 是图像预处理的重要手段, 以数学角度看, 高斯平滑的过程就是图像与正态分布做卷积的过程, 对于图像来说高斯模糊就起到一个低通滤波器的作用。

高斯模糊是利用正态分布来计算图像里每个像素点的变换. 空间为 N 维时的正态分布方程如下:

$$G(r) = \frac{1}{\sqrt{2\pi\sigma^2 N}} e^{-r^2/(2\sigma^2)}$$

在二维空间定义为:

$$G(u, v) = \frac{1}{2\pi\sigma^2} e^{-(u^2+v^2)/(2\sigma^2)}$$

上述表达式中 r 是模糊的半径, u 、 v 是图像的像素坐标, σ 为正态分布的标准偏差. 通常 σ 的数值越大说明平滑度越好^[15]。

4.2 直方图均衡化

直方图均衡化就是把图像的直方图平坦化, 其作用是减少由于光照变化引起的影响. 均衡化就是利用变换函数将输入图像的灰度图像经过非线性转换后, 使其直方图的分布变得更加均匀, 从而使图像中某些局部偏亮或偏暗的部分变得均匀化^[16]. 直方图均衡化能够提高图像对比度, 在光照较强和光照较暗的环境下可以有效的进行补偿, 该方法主要减弱图像微小的细节。

- (1) 首先给出原始图像灰度级 $S_k: k = 0, 1, \dots, L-1$.
- (2) 统计出图像中的各灰度级像素个数 n_k .
- (3) 根据下面公式算出各个灰度级的频率:

$$P(S_k) = \frac{n_k}{n}$$

其中, n 是像素的总个数, n_k 是灰度级 S_k 像素个数。

- (4) 计算图像的累计直方图函数:

$$t_E = \sum_{i=0}^k P_S(S_i)$$

- (5) 对结果进行取整运算:

$$U_k = \text{int} \left[(N-1)t_k + \frac{k}{N} \right]$$

(6) 得出原直方图灰度级到新直方图灰度级的映射:

$$S_k \rightarrow U_k$$

(7) 算出 U_k 中总的像素个数 n_k .

(8) 运算得出原始直方图对应的新直方图:

$$P(t_k) = \frac{n_k}{n}$$

4.3 灰度处理

RGB 是彩色图片, 将 RGB 转为灰度图片可以大大降低检测的难度, 灰度变换是通过加权平均法的方式实现 RGB 到 GRAY 的格式转换, 公式为: $Gray = 0.587 \times green + 0.114 \times blue + 0.299 \times red$.

5 测试和分析

经过测试, 传统算法进行人脸检测时, 不仅检测时间长而且视频会出现卡顿现象. 而利用改进之后的算法有效的缩减了人脸检测的时间, 满足视频对实时性的需求.

如图 6 所示, 每达到预设值就会进行一次人脸检测用来更新数据即 ROI 区域. 通过对大量的人脸检测运行时间 (face detect run_time) 和利用 Camshift 算法改进之后的人脸检测运行时间 (Camshift run_time) 进行对比分析, 可以得到, 用 Camshift 目标跟踪算法改进的人脸检测速度提升 40% 左右, 不会出现视频播放不流畅的情况, 且跟踪效果比较好.

```

C:\Users\Administrator\Desktop\监控视频项目\C++\监控视频服务端\Debug\监控视频
face detect run_time: 0.849797s
Camshift run_time: 0.564936s
ROI.x= 364, ROI.y= 221, width= 222, height= 222
Face detect run_time: 0.817222s
Camshift run_time: 0.534831s
ROI.x= 228, ROI.y= 16, width= 386, height= 386
Face detect run_time: 0.851238s
Camshift run_time: 0.583799s
ROI.x= 241, ROI.y= 241, width= 212, height= 212
Face detect run_time: 0.852722s
Camshift run_time: 0.561823s
ROI.x= 257, ROI.y= 216, width= 252, height= 252
Face detect run_time: 0.868386s
Camshift run_time: 0.579875s
ROI.x= 318, ROI.y= 198, width= 253, height= 253
Face detect run_time: 0.898444s
Camshift run_time: 0.609586s
Face detect run_time: 0.822839s
Face detect run_time: 0.794885s
Face detect run_time: 0.883772s
Face detect run_time: 0.838546s
ROI.x= 298, ROI.y= 162, width= 264, height= 264
Face detect run_time: 0.888386s
Camshift run_time: 0.581633s

```

图 6 实验对比数据

此中的设定值不是固定不变的, 应根据实际工作场合进行调整, 设定值的大小会影响检测的准确性.

如图 7 所示, 为人脸区域 H 通道的直方图, 用于 Camshift 算法中直方图反射投影, 从而对人脸跟踪.

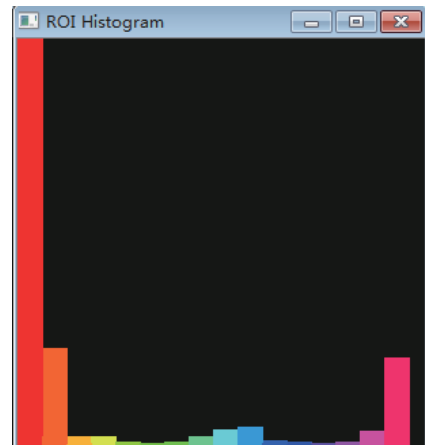


图 7 ROI 区域数据直方图

如图 8、图 9 所示, 利用改进的算法之后, 图中的人脸矩形区域就是 Camshift 的 ROI 区域, 通过图 8 和图 9 对比可知, 基本消除了光照对检测的影响.



图 8 检测到的人脸区域

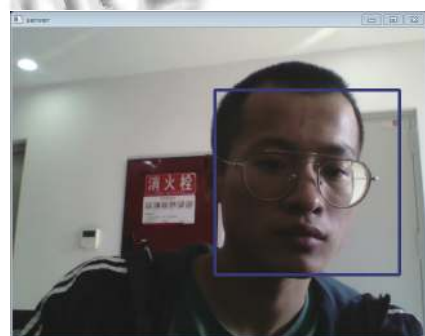


图 9 光照对比检测

6 结论与展望

本文提出了一种新的人脸检测算法, 在传统人脸检测算法中融合 Camshift 目标跟踪, 该算法可以很好的弥补传统人脸检测算法中无法满足实时性需求的缺点, 并通过图像处理手段消除噪声和光照的影响, 最后

通过测试证实该方法的可行性。该算法有效的改善了检测速度和准确性,适用一定的工作场合,今后的工作会在此基础上完善算法使其更适合当今的图像处理问题。

参考文献

- 1 Mutneja V, Singh S. Modified Viola-Jones algorithm with GPU accelerated training and parallelized skin color filtering-based face detection. *Journal of Real-time Image Processing*, 2019, 16(5): 1573–1593. [doi: [10.1007/s11554-017-0667-6](https://doi.org/10.1007/s11554-017-0667-6)]
- 2 Nanni L, Brahnam S, Lumini A. Face detection ensemble with methods using depth information to filter false positives. *Sensors*, 2019, 19(23): 5254. [doi: [10.3390/s19235242](https://doi.org/10.3390/s19235242)]
- 3 程实, 沐燕舟, 赵赟. 一种基于改进 JDA 算法的人脸检测方法. *计算机与数字工程*, 2019, 47(11): 2871–2875. [doi: [10.3969/j.issn.1672-9722.2019.11.045](https://doi.org/10.3969/j.issn.1672-9722.2019.11.045)]
- 4 吴纪芸, 陈时钦. 一种改进的 MTCNN 人脸检测算法. *软件导刊*, 2019, 18(12): 78–81.
- 5 刘卫华, 吴丹. 基于 OpenCV 的人脸检测. *科技创新与应用*, 2019, (31): 18–19.
- 6 石泽琼, 毛俊强, 李双伟, 等. 基于 OpenCV 的运动目标识别跟踪. *中国集成电路*, 2019, 28(4): 20–25. [doi: [10.3969/j.issn.1681-5289.2019.04.004](https://doi.org/10.3969/j.issn.1681-5289.2019.04.004)]
- 7 丁业兵. 基于 Adaboost 算法的人脸检测及 OpenCV 实现. *电脑知识与技术*, 2018, 14(27): 167–169. [doi: [10.14004/j.cnki.ckt.2018.2965](https://doi.org/10.14004/j.cnki.ckt.2018.2965)]
- 8 刘燕, 贺松, 成雨风. 一种改进型 Adaboost 算法的人脸检测. *智能计算机与应用*, 2019, 9(5): 98–101, 110.
- 9 陈艺. 改进 Camshift 算法的多行人目标跟踪方法. *导航定位学报*, 2019, 7(4): 30–36. [doi: [10.3969/j.issn.2095-4999.2019.04.006](https://doi.org/10.3969/j.issn.2095-4999.2019.04.006)]
- 10 马洪涛, 吴赛敏. 基于 Kalman 滤波的 CAMShift 目标跟踪算法. *科技风*, 2019, (23): 32, 36. [doi: [10.19392/j.cnki.1671-7341.201923029](https://doi.org/10.19392/j.cnki.1671-7341.201923029)]
- 11 张晓青, 马牧燕, 燕必希, 等. 改进 Haar-like 特征联合 CamShift 算法用于车辆跟踪. *实验室研究与探索*, 2019, 38(9): 86–91. [doi: [10.3969/j.issn.1006-7167.2019.09.019](https://doi.org/10.3969/j.issn.1006-7167.2019.09.019)]
- 12 刘伍丰, 何前磊, 郑维. 基于 HAAR 特征与 BP 神经网络的车牌识别技术研究. *电子测量技术*, 2019, 42(8): 61–67. [doi: [10.19651/j.cnki.emt.1802282](https://doi.org/10.19651/j.cnki.emt.1802282)]
- 13 王伟程. Camshift 目标跟踪算法改进研究. *石油化工自动化*, 2019, 55(1): 35–38, 65. [doi: [10.3969/j.issn.1007-7324.2019.01.009](https://doi.org/10.3969/j.issn.1007-7324.2019.01.009)]
- 14 余建波. 基于 OpenCV 的视频人脸识别系统设计. *信息通信*, 2018, (12): 40–42. [doi: [10.3969/j.issn.1673-1131.2018.12.016](https://doi.org/10.3969/j.issn.1673-1131.2018.12.016)]
- 15 肖阳. 基于 OpenCV 的人脸识别的算法研究与实现 [硕士学位论文]. 武汉: 湖北工业大学, 2018.
- 16 霍焰焰. 基于 PCA 和 LBP 改进算法的人脸识别研究 [硕士学位论文]. 哈尔滨: 哈尔滨理工大学, 2015.