

基于百度云 AI 的自动售检票系统人脸识别应用可行性分析

王 琮 韩 骁

(天津轨道交通运营集团有限公司, 300380, 天津//第一作者, 高级工程师)

摘 要 人脸识别作为生物识别技术的重要分支, 因其非接触性特点, 在身份认证方面已经获得广泛应用。百度云 AI 作为智能云平台已经开放轻量级人脸识别和认证的应用服务。基于该服务不需要重复研发人脸识别算法, 短期即可完成“刷脸”乘车环境搭建。分析了基于百度云 AI(人工智能)技术快速建立轨道交通 AFC(自动售检票)系统人脸识别应用场景的可行性。

关键词 轨道交通; 自动售检票系统; 人脸识别; 百度云 AI

中图分类号 F530.7; U293.22

DOI: 10.16037/j.1007-869x.2020.03.034

Feasibility Analysis of Face Recognition Applied in Automatic Fare Collect System Based on Baidu Cloud AI

WANG Cong, HAN Xiao

Abstract As an important branch of biological recognition technology, face recognition is widely used in identity authentication because of the non-contact characteristics. Baidu Cloud AI is an intelligent cloud platform, it has opened the application service of lightweight face recognition and authentication. Since the service does not need repeatedly develop the face recognition algorithm, the riding environment with “face swiping” can be built in a short time. Based on Baidu Cloud AI technology, the feasibility of fast establishment of face recognition application scenario for rail transit automatic fare collection (AFC) system is analyzed.

Key words rail transit; AFC system; face recognition; Baidu Cloud AI

Author's address Tianjin Railway Transit Operation Group Co., Ltd., 300380, Tianjin, China

人脸识别作为生物识别技术的重要分支, 因其非接触性特点, 在身份认证方面已经获得广泛应用。百度云 AI(人工智能)(以下简称“云 AI”)作为智能云平台已经开放轻量级人脸识别和认证应

用服务。利用该技术可以快速建立轨道交通 AFC(自动售检票)系统在线人脸身份识别的应用场景, 不需要重复研发人脸识别算法, 无需高性能设备投入, 短期即可完成“刷脸”乘车环境搭建, 同时为优化图像采集设备集成方案提供技术基础。

1 云 AI 人脸识别的技术优势

云 AI 人脸识别服务基于行业普遍采用的深度学习人脸识别方案, 可以提供复杂场景下的人脸捕捉和识别信息数据, 支持多种编程语言开发, 灵活适应不同开发环境的 AFC 系统。

因云 AI 已经实现人脸识别、认证及管理的全部功能, 所以在现有 AFC 系统上快速实现人脸识别应用的优势是自建人脸识别算法系统所无法企及的, 可在最少投资和最短时间内建立完整的在线“人脸”购票、过闸、登记管理等业务应用。同时, 便于开发人员集中精力完善票务应用体系和与乘客交互流程的设计; 便于验证“人脸识别”前端系统的准确性; 便于提高 AFC 系统交易容错性设计, 在 AFC 系统人脸识别体系的先导设计验证等方面有巨大的技术优势。

现有 AFC 系统设备增加人脸识别购票和过闸功能, 只需要增加摄像头及增加云 AI 的交互代码和交易处理程序即可, 具有开发成本低、易于实现的特点。摄像头需加装在现有售票机和闸机外壳的特定位置上。

云 AI 人脸识别还具有如下技术优势:

1) 云 AI 人脸识别采用可见光图像分析方式, 只要基本保证环境亮度要求, 普通摄像头即可采集认证图像, 且支持低照度和较低分辨率识别。

2) 面部图像信息的处理及人脸库存储和管理全部在云端完成, 本地无需任何附加设备。数据信息的云端处理可以最大限度降低数据同步风险, 实

现即时注册即时乘车。

3) 云 AI 平台支持 API(应用程序编程接口)和 SDK(软件开发工具包)两种开发方式,可以灵活地在现有 AFC 系统平台上实现人脸识别功能的开发和测试。快速降低延迟的处理能力,可以满足试验环境下的人脸识别业务应用。

4) 准确识别多种人脸属性,提供人脸信息关键点定位、人脸认证对比、在线活体检测等关键信息;实现复杂图像中对人脸的检测,并对人脸进行分析,获取眼、口、鼻轮廓等关键点定位(见图1)。



图1 五官及面部轮廓信息定位

5) 适应大角度侧脸,以及遮挡、模糊、表情变化等各种实际环境。在实际应用中可有效减少乘客面部对正时间。

2 云 AI 人脸识别的扩展方案

AFC 系统设备增加人脸识别功能,可在现有上位机软件逻辑层上扩展人脸识别模块或者通过读卡器操作系统扩展。图2是以闸机为例描述的增加人脸识别功能模块的构成方案。其中,方案1可通过闸机上位机系统软件实现扩展人脸识别功能。该方案在闸机上位机软件增加人脸识别模块,控制摄像头采集人像,与云 AI 交互识别人像信息。方案1扩展方式相对简单,可以充分利用上位机计算资源,图像处理延迟低,人像捕捉开发难度较低,可

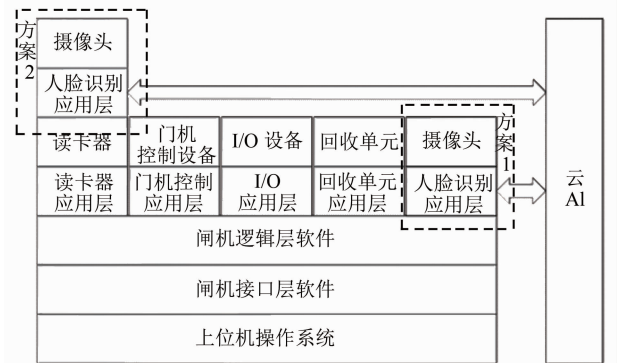


图2 闸机扩展人脸识别功能模块图

扩展性好;但交易过程没有读卡器的直接参与,交易生成过程需要进一步优化。方案2为通过大读卡器实现扩展人脸识别功能,其实现方式与闸机上位机软件相似。方案2需要在读卡器内开发软件,受限于读卡器计算资源有限,人像捕捉开发难度相对较高,图像处理延迟高,可扩展性受限。优势是可与现有交易方式有机结合,可充分利用现有交易加密体系,安全性高。

3 云 AI 人脸识别的开发测试

为了验证云 AI 的应用可行性,搭建试验环境测试云 AI 的交互过程和人脸信息的本地捕捉方法。由于开发环境受限,只采用方案1在现有编译环境下实现,开发平台采用 Python 2.7+OpenCV 3.2.0+云 AI 开放平台人脸识别 API 服务。

人脸识别交互过程通过两个独立程序分别开发实现:第一部分,本地图像采集和人脸图像预处理;第二部分,云 AI 交互及人脸信息分析和身份认证。两部分功能由两段独立代码实现,通过图像文件交换实现交互。采用此种方式设计,目的是降低程序的耦合性,便于发现开发过程中的错误,尽量适应不同 AFC 厂商的开发习惯。实际应用中应根据实际情况将两部分优化整合,以提高运行效率。

3.1 本地图像采集和人脸图像预处理流程

1) 通过 OpenCV(开源计算机视觉库)调用摄像头视频流,获取视频流中的帧图像。

2) 调用 OpenCV 人脸检测接口函数,分析帧图像中的人脸信息,获得人脸坐标、高度和宽度信息。人脸检测模板使用的是 OpenCV 标准正面人脸检测参数模板(harcascade_frontalface_default.xml),没有进行调参操作。

3) 调用 OpenCV 模糊评价接口函数,生成人脸图像模糊评价价值(灰度图像模式),合理设定图像模糊评价阈值,分析采集图像是否能够达到云 AI 认证的最低标准。

4) 裁剪人脸图像,生成图像交换文件。

本地图像采集和人脸图像预处理概要流程如图3所示。流程中只包含图像采集和人脸图像预处理的必须流程,实际应用中还需加入图像采集频度控制、脸部对正提示、同框人脸数限制、频繁合法图像采集限制等控制逻辑,用以减少图像数据采集数量,加快图像采集速度,避免身份认证混淆和重复采

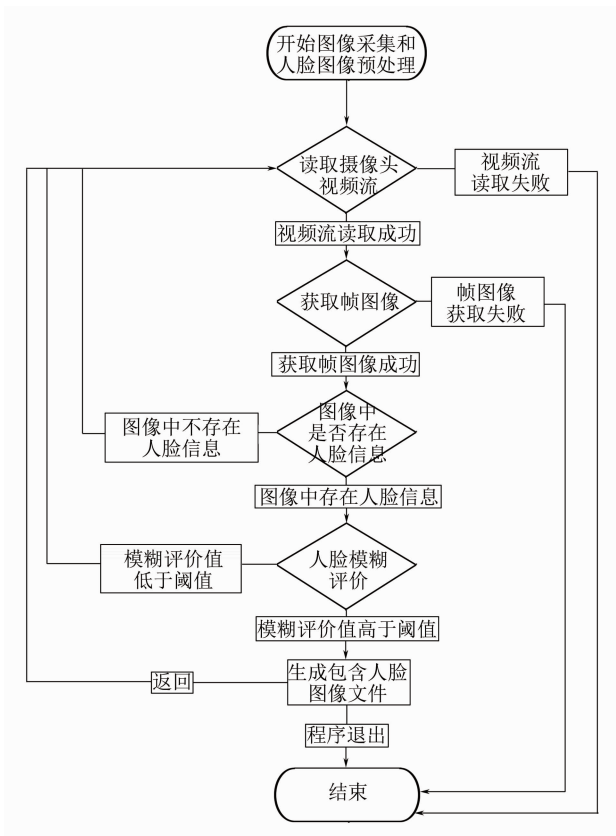


图 3 本地图像采集和人脸图像预处理概要流程图

集等问题。

3.2 云 AI 交互及人脸信息分析和身份认证流程

- 1) 初始化云 AI 人脸识别环境,提交终端身份信息和安全密钥,进行应用安全认证;
- 2) 获取图像交换文件,申请人脸识别服务,提交人脸图像信息和用户分组信息;
- 3) 接收人脸识别服务返回的分析结果,包括用户 ID(身份标志)、身份相似度和活体指标等数据;
- 4) 处理分析结果,判定图像身份;
- 5) 根据返回的身份认证信息生成交易数据。

云 AI 交互及人脸信息分析和身份认证流程如图 4 所示。

以上流程只是已经完成身份认证的合法用户在进行人脸识别的处理流程。完善的处理流程还应包含用户登记、采集面部图像、用户 ID 管理、支付账户绑定等必要流程。

4 云 AI 人脸识别的测试结果

经过一段时间的调试,测试程序可实现调用云 AI 人脸识别服务,并可在较低分辨率图像的基础上实现面部信息身份识别。如图 5 所示,测试程序扫

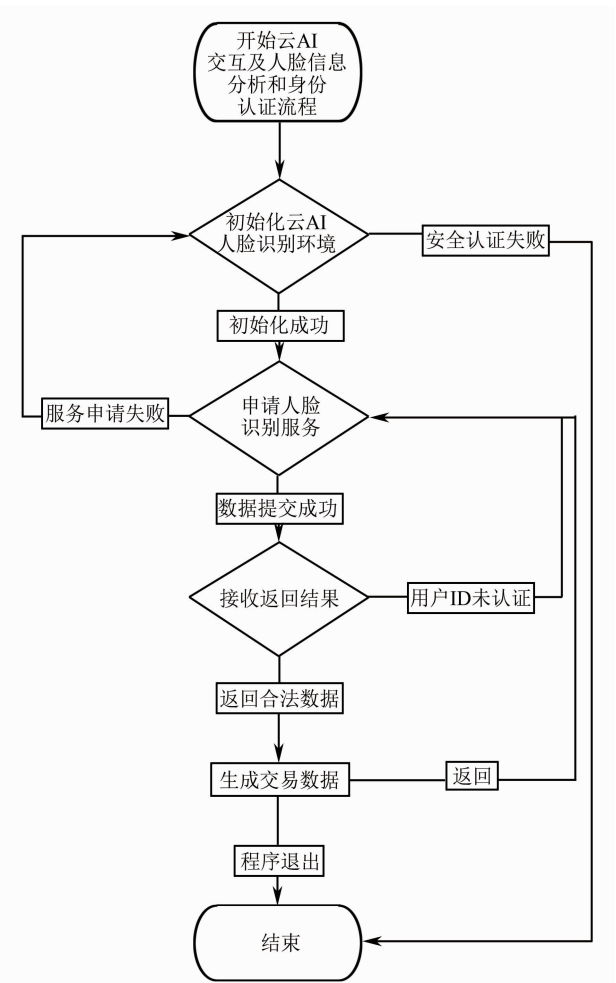


图 4 云 AI 交互及人脸信息分析和身份认证流程

描读取面部信息交换文件后,提交至云 AI 后台进行处理,然后通过程序解析人脸认证信息处理结果后实现人脸的显示。在测试环境下基本实现人脸识别应用交互的关键流程,人像采集和云 AI 图像身份识别认证功能基本通过测试验证。对于 AFC 系统最为敏感的通行速率和票款安全性要求等,进行了初步验证。



图 5 测试程序运行截图

4.1 通行速率

云端处理延迟会直接影响乘客使用体验,特别是对于闸机而言,过高的延迟会影响通行速率。为了测试云 AI 的处理性能,采用批量模拟识别请求的方法,检查本地与云端交互的总体处理延迟。由于百度云 AI 为免费测试账户,只能提供 1.5 QPS(吞吐量)峰值访问性能,故只针对处理延迟做出分析。

在测试环境下,本地多次发起与云端建立连接请求,并连续发送识别交互(即从图像提交云 AI 到接收分析数据返回)请求。测试结果为:云端处理延迟平均时间为 395 ms,首次连接请求处理延迟平均时间为 1 098 ms。以测试数据分析,云 AI 平均处理延迟时间基本满足闸机对通行速率的要求,延迟时间介于 M1 卡和 CPU(中央处理器)卡的处理时限之间,特别是连续图像处理的平均延迟时间只有 226 ms。但是,首次连接请求处理延迟过高,需通过技术手段减少对实际应用的影响。云 AI 识别处理延迟统计详见表 1。

表 1 云 AI 识别处理延迟统计表

交互 次序	建立连接/ms				平均延迟时间/ms	
	第 1 轮	第 2 轮	第 3 轮	第 4 轮	首次 连接	全部 连接
1	1 675	1 107	296	1 312	1 098	395
2	214	227	180	220		395
3	184	183	199	197		395
4	223	QPS 限制	216	179		395
5	204	168	QPS 限制	628		395
6	QPS 限制	187	248	246		395

4.2 视频流分析和图像采集中存在的问题

4.2.1 OpenCV 人脸参数模板的调参问题

在通过 OpenCV 分析视频流中是否包含人脸的过程中,一些特定背景会被识别成人脸坐标,导致图像采集错误。因此,目前采用的 OpenCV 标准正面人脸检测参数模板还需要进一步在参数细节上调校,以进一步适应地铁光线环境和装修背景。

4.2.2 摄像头焦距和拍摄角度的设计问题

普通定焦摄像头在拍摄过程中同时采集多个人像的概率很高,即使在适合焦距外采集的清晰度较低的人像也能达到云 AI 的识别要求,这样就会产生同一张画面内有多个合格面部信息,导致本次采集失败。如果背景中的人像不能移开,将导致无法

采集唯一合格人像,直接影响闸机的通过速率。

摄像头的拍摄角度也会对人像采集产生影响,特别是闸机摄像头的拍摄角度。高仰角设计可以减少多个人像同时采集的概率,但无法覆盖较低身高(特别是儿童)的人像采集;低仰角设计的问题正好相反。因此需合理安排拍摄角度。

5 结语

通过开发测试基本可以验证基于云 AI 的人脸识别方案是可行的,开发难度基本可控,现有计算机系统基本可以满足视频信息采集和人脸识别的资源消耗要求,云 AI 响应速度和并发性能在试验环境下测试满足性能要求。在识别算法授权方面,OpenCV 是基于开源许可协议(BSD)发行的跨平台计算机视觉库,没有授权费用,可在现有计算机系统上分发使用;百度 AI 开放平台人脸识别 API 服务在 AFC 生产环境中使用时需要付费,以保证快速任务处理和高并发性能。

人脸识别不同于现有的刷卡形式,其交互方式已经发生根本变化,不再是以读卡器为核心的 IC(集成电路)卡或模拟 IC 卡数据交换方式,而转为以人像数据为核心的非数据交换方式。交易过程不再有乘客数据信息载体,即 IC 卡,乘客信息全部由 AFC 系统捕捉和处理,对 AFC 系统交易信息的完整性和容错能力提出了更高的要求。虽然借助云 AI 系统可以大幅减少 AFC 系统后台处理压力,降低人脸识别信息复杂度,但是如何使人机交互界面更加自然,提高系统可靠性需要进一步深入研究。随着轨道交通安全性要求的不断提高,人脸识别的“实名制”优势将逐步显现,预计很快将成为主流支付方式之一。

参考文献

[1] HETLAND M L. Python 基础教程[M]. 3 版. 北京:人民邮电出版社,2014:219.

[2] PYTHON O. Python 2.7.15 documentation[EB/OL]. (2010-07-03) [2018-04-27]. <https://docs.python.org/2/>. <https://docs.python.org/2/tutorial/index.html>.

[3] 百度. 人脸识别 API 文档-V2[EB/OL]. (2018-04-027) [2018-04-27]. <http://ai.baidu.com/docs#/Face-Detect-V3/top>.

(收稿日期:2019-03-21)