计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2019,28(3):93−98 [doi: 10.15888/j.cnki.csa.006802]

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

基于 Storm 的大容量实时人脸检索系统①

王晨曦, 范春晓, 吴岳辛

(北京邮电大学 电子工程学院, 北京 100876)
通讯作者: 王晨曦, E-mail: wangchenxi@bupt.edu.cn

摘　要: 针对公共安全领域能够获取的人脸图像数据急速增长, 传统的人工方式辨别人物身份工作量大、实时性差、准确度低, 本文设计了一种大容量实时人脸检索系统. 该系统通过 Storm 分布式平台实现人脸抓拍图像的实时存储与检索, 通过 HBase 分布式存储系统实现大容量非结构化人脸数据的存储与维护. 多组实验结果表明, 该系统具有良好的加速比, 在大容量人脸图像数据检索场景下具有良好的可扩展性和实时性.

关键词: 人脸检索; 大容量; Storm; 实时

引用格式: 王晨曦,范春晓,吴岳辛.基于 Storm 的大容量实时人脸检索系统.计算机系统应用,2019,28(3):93–98. http://www.c-s-a.org.cn/1003-3254/6802.html

Large-Scale Real-Time Face Retrieval System Based on Storm

WANG Chen-Xi, FAN Chun-Xiao, WU Yue-Xin

(School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: The face image data that can be obtained in the field of public security has grown rapidly. The traditional manual method to identify people has large workload, poor real-time performance, and low accuracy. This study designs a large-scale real-time face retrieval system. The system implements the real-time storage and retrieval of captured face images through the distributed platform Storm, and implements the storage and maintenance of large-scale unstructured face data through the distributed storage system HBase. The results of multiple experiments show that the system has a good speedup, good scalability, and real-time performance in the application scenarios of large-scale face image data retrieval.

Key words: face retrieval; large-scale; Storm; real-time

图像检索是指根据目标图像的内容, 通过一种自动分类算法来判断目标图像的相应类别. 在图像检索的发展中, 主要经历了三个阶段: 基于元数据的图像检索、基于文本标注的图像检索和基于内容的图像检索[1], 人脸检索则可以归类于基于内容的图像检索. 与人脸识别不同的是, 人脸检索需要实现目标照片的人脸检测、特征提取, 并在大规模人像库中进行匹配. 在大数据环境下, 这种相似人脸检索技术, 在安防、军事以及娱乐领域有着广泛的应用价值, 已成为人脸图像研究中的一个热点[2].

现有的视频监控系统虽然已经遍布社会生活的各个角落, 但是只实现了单纯的视频存储功能, 监控视频信息的分析仍然依靠人力因而不具备实时性. 因此, 在视频监控的同时抓拍和记录高清晰度的人脸图像并提供识别、比对、报警、查询等功能, 是新一代视频监控追求的目标, 也是实际应用中迫切需要的功能[3]. 而随着国家天网工程的部署, 公共安全领域能够获取的人脸图像急速增长, 使得现有人脸图像检索技术在处

① 收稿时间: 2018-08-24; 修改时间: 2018-09-20; 采用时间: 2018-10-09; csa 在线出版时间: 2019-02-22

理海量图像数据的过程中, 在实时性、扩展性、并发性和准确性等方面面临严峻的考验[4].

人脸检索系统在海关对入关人员的身份排查、大人流量下的安检工作等应用场景下对检索的实时性有极高的要求, 而传统的基于单节点架构的人脸图像检索系统已经不能满足人们对于检索性能的要求, 因此本文采用分布式技术并设计了一种大容量实时人脸检索系统, 将人脸识别技术与视频监控融合, 通过 Storm 分布式平台实现基于实时人脸抓拍照片的识别、存储、检索、报警、查询等功能, 解决了现有人脸图像检索技术在处理海量图像数据的过程中, 在实时性、扩展性上的需求.

1 相关研究

随着图像数据的快速增长, 许多研究人员都对大规模图像数据的分布式处理进行了研究, 这类研究主要分为三类: 对分布式图像检索系统的研究、对分布式图像特征提取算法及检索算法的研究以及对各类分布式平台图像处理性能的研究.

文献[5]通过 Hadoop 平台进行大规模图像数据的特征提取与检索, 将提取到的特征值存入 HBase 中, 但通过 MapReduce 进行图像检索时, 每开启一次任务都需要重新从 HBase 读取一次图像特征值, 增加了不必要的 I/O 开销. 文献[6]则是将提取的图像特征以文件形式存储于 Hadoop 的分布式文件系统 HDFS 中, 通过 MapReduce 进行图像检索, 考虑到大量的小文件会造成在计算时会产生过多的 Map 和 Reduce 任务, 极大延迟计算完成时间, 提出了将多幅图像的特征小文件合并为后存储于 HDFS 中, 以实现大规模图像的快速存储和读取, 但每次的检索仍然存在大量的 I/O 开销. 文献[7–10]也都基于 Hadoop 平台针对图像特征提取效率、图像检索效率进行了研究, 但都没有对图像检索的实时性进行考虑. 文献[11]则对比了 Hadoop、Spark 以及 Storm 三者在大规模图像检索中的性能, 其实验结果表明, Storm 在图像检索性能上, 比 Hadoop 与 Spark 有更低的延迟.

基于上述研究, 本文设计了一种大容量实时人脸检索系统. 在 HBase 中建立注册库用于存储大容量人物信息, 包括人脸照片、人脸特征值以及人物基本信息. 通过 Storm 分布式平台实现基于实时抓拍照片的实时存储与检索.

2 系统设计

本文设计的人脸检索系统主要适用于海关对入关人员的身份排查、大人流量下的安检工作等应用场景下, 系统需要实现对摄像机人脸抓拍照片的实时存储与检索, 并提供报警、查询等功能, 同时当摄像机数量增加时, 单位时间内各个摄像机所产生的人脸抓拍照片的总量也会线性增长, 系统需要有良好的扩展性以应对负载的增加. 传统的基于单节点架构的人脸图像检索系统在性能上已经无法满足设计需求, 因此本文采用分布式技术设计了基于 Storm 的大容量实时人脸检索系统. Storm 的流式处理方式使得人脸抓拍照片能够源源不断的进行实时存储与检索, 而通过集群中计算节点的增加以及 Storm 中实时抓拍照片存储与检索任务的并行运行则可以满足摄像机数量增加时对系统处理能力的需求.

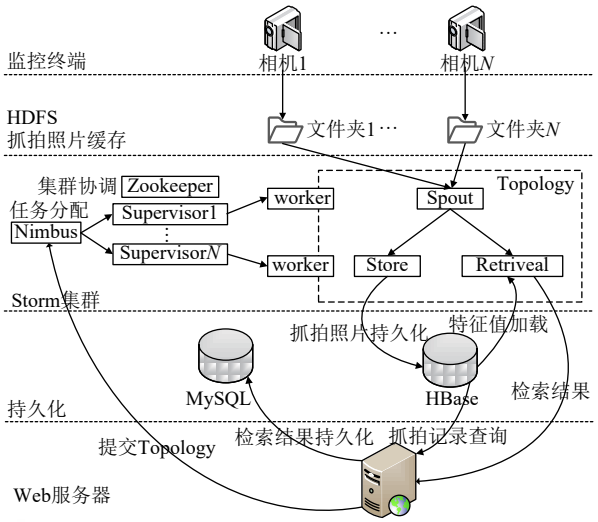

图 1 基于 Storm 的大容量实时人脸检索系统总体框架

本文设计的基于 Storm 的大容量实时人脸检索系统整体设计框图如图 1 所示, 主要包含: 监控终端、HDFS 抓拍照片缓存、Storm 集群、持久化、Web 服务器.

(1) 监控终端: 系统采用的监控终端为海康深眸系列的智能人脸筒型网络摄像机 DS-2CD7627FWD/F-LZ, 该摄像机支持人脸抓拍功能, 支持对运动人脸进行检测、跟踪、抓拍、评分, 并自动筛选输出最优人脸图. 通过配置摄像机的参数, 将摄像机的实时人脸抓拍照片存储到 HDFS 中对应的文件夹下做缓存.

(2) HDFS 抓拍照片缓存: HDFS 中的文件夹与摄

像机一一对应, 摄像机的人脸抓拍照片会自动保存到其对应的文件夹下, 同时这些文件夹也是 Storm 集群中运行的实时抓拍照片存储与检索任务的数据源.

(3) Storm 集群: Storm 集群中每台摄像机均有唯一的实时抓拍照片存储与检索任务与之对应. 这些任务在集群中并行运行, 一方面从 HDFS 中读取摄像机的抓拍照片, 进行特征提取后以抓拍时间为行健存储到 HBase 中对应摄像机的抓拍库中; 另一方面将抓拍照片的特征值与用户导入的注册库中的大容量人脸特征值进行相似度计算, 筛选出相似度超过阈值的检索结果, 并将检索结果反馈给 Web 服务器. 用户在向集群提交任务时需指定该任务所对应摄像机在 HDFS 中的缓存文件夹的路径作为数据源, 并指定需要进行检索的注册库.

(4) 持久化: 大容量、非结构化的人脸数据采用 HBase 进行存储, 而实时人脸检索结果以及其它结构化数据则采用 MySQL 进行存储. 实时抓拍照片存储与检索任务从数据源获取的抓拍照片经处理后会存储到 HBase 中摄像机所对应的抓拍库中, 检索结果则会反馈给 Web 服务器并经过格式处理后存入 MySQL 以供用户查询.

(5) Web 服务器: 为用户提供实时抓拍照片存储与检索任务的开启、实时人脸检索结果的持久化、报警、报警记录查询、摄像机抓拍记录查询等功能, 并为各项功能提供可视化界面.

3 大容量实时人脸检索系统的实现

3.1 人脸数据存储

系统中的人脸数据包括两个部分, 第一部分数据需要在系统部署时导入, 这部分数据包含了人物的人脸照片、人脸特征值以及基本信息, 其中人物照片数量不做限制, 即用户可以针对同一人物导入其不同年龄、不同表情、不同角度的人脸照片, 该部分数据为非结构化数据. 第二部分是系统开始运行后各个摄像机的实时人脸抓拍数据, 这部分数据为结构化数据, 在存储时只包含了抓拍时间、抓拍照片以及人脸特征值三个字段, 但单台相机每年抓拍的数据量能达到 1.5 TB. 为区分两部分数据, 第一部分的数据称为注册库, 第二部分的数据称为抓拍库.

为解决大容量、非结构化人脸数据的存储, 系统采用开源的、面向列的分布式存储系统 HBase[12]作为存储人脸数据的数据库. 图 2 中给出了存储非结构化人脸数据的注册库在 HBase 中的存储结构.

Row							
RowKey	列簇1			列簇2			列簇3
证件号	照片1	照片2	.....	特征值1	特征值2	.....	基本信息
证件号	照片1	.....		特征值1	.....		基本信息

图 2 注册库在 HBase 中的存储结构

3.2 人脸识别算法

SeetaFace 是一个开源的 C++人脸识别引擎, 可以不依赖第三方库函数并在 CPU 上进行运行, 它包含搭建一套全自动人脸识别系统所需的三个核心模块: SeetaFace Detection(人脸检测模块), SeetaFace Alignment(面部特征点定位模块) 和 SeetaFace Identification(人脸特征提取与比对模块)[13]. SeetaFace 所采用的人脸特征值为 VIPLFaceNet FC2 层的 2048 个结点的输出, 即长度为 2048 的 float 型数组, 特征值对比采用了 cosine 计算相似度. 假设 $V(I)$ 为人脸 I 的特征向量, $V(J)$ 为人脸 J 的特征向量, 则人脸 I 与人脸 J 的相似度计算如下:

$$\text{sim}(V(I),V(J)) = \frac{V(I)\cdot V(J)}{\|V(I)\| * \|V(J)\|} \quad (1)$$

JNI (Java Native Interface) 是 Java 平台的一部分, 它提供了若干的 API 以实现 Java 和其他语言的通信. 本文中 Storm 集群上运行的任务由 Java 语言编写并在 Java 虚拟机上运行, 而 SeetaFace 由 C++编写, 为实现在 Storm 任务中调用 SeetaFace 中的算法, 本文在 SeetaFace 的基础上进行了二次开发. test_face_verification.cpp 是 SeetaFace 当中的一个测试文件, 它演示了一次完整的人脸相似度计算流程: 照片加载、人脸检测、人脸特征值提取以及人脸相似度计算. 本文基于该测试文件, 将人脸检测、人脸特征值提取、人脸相似度计算三个模块的代码分离出来, 各自编写为可以单独调用的新函数, 并新增了一对多人脸相似度计算函数, 在 Linux 平台上编译生成动态链接库, 最终 Storm 任务通过 JNI 实现对以上二次开发算法的调用. 此外, 经二次开发后, 照片加载不再是由 OpenCV 中的 Mat 类从指定的路径下加载, 而是修改为 Mat 类直接加载 JNI 调用时由 Java 语言传参过来的照片数据.

通过 JNI 调用 SeetaFace 中的各类算法, 降低了 Storm 任务与 SeetaFace 的耦合性. 本文设计的大容量实时人脸检索系统, 其实时性由分布式计算来实现, 但人脸检索的准确性由系统中所采用的人脸识别算法决

定. 降低 Storm 任务与人脸识别算法的耦合性, 便于后续在系统更新时, 选择更优的人脸识别算法进行替换, 即对于识别性能更好的人脸识别算法, 其只需实现相应的 JNI 接口后编译生成动态连接口, Storm 任务即可通过 JNI 来调用新的算法来进行识别、检索.

3.3 实时抓拍照片存储与检索任务

Storm 是一个开源、分布式、高容错的实时大数据处理系统. Storm 实现了流式计算, 弥补了 Hadoop 批处理所不能满足的实时性要求. Storm 经常用于在实时分析、在线机器学习、持续计算、分布式远程调用和 ETL 等领域[14].

系统中的实时抓拍照片存储与检索任务在 Storm 分布式平台上运行, 图 3 给出了实时抓拍照片存储与检索任务的拓扑结构. 任务主要包括 6 个部分: 1) 照片抓取, 2) 人脸检测, 3) 特征提取, 4) 人脸检索, 5) 结果统计, 6) 存抓拍库. 任务在提交到 Storm 上运行时, 会预先读取注册库中的特征值到内存中, 在任务运行过程中, 抓拍照片的特征值会与预先读取到内存中的特征值进行相似度计算, 从而减少频繁的从 HBase 中读取所产生的 I/O 开销.

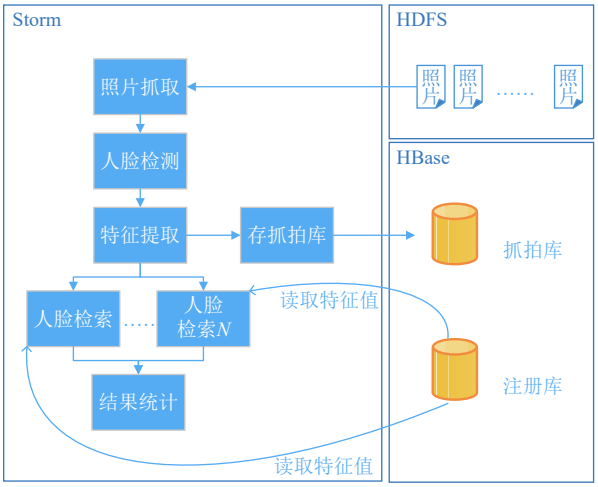

图 3 实时抓拍照片存储与检索任务

4 实验与结果分析

4.1 实验环境

本文提出的实时人脸检索系统的计算集群部署在三台服务器上, 单台服务器的硬件配置为: 两个 Intel(R) Xeon(R) X5650 @ 2.67 GHz CPU、16 GB 内存, 操作系统为 CentOS 7, 集群的软件版本在表 1 中给出.

表 1 集群软件版本

Configuration	Value
Java Version	openjdk 1.8.0_161
Zookeeper Version	3.4.11
Hadoop Version	2.7.5
Hbase Version	1.3.2
Storm Version	0.10.2

本文使用的人脸数据集为 LFW (Labeled Face in Wild) 数据集[15]. LFW 数据集包含了 5749 名人物, 共计 13 233 张照片. 为避免无效人脸影响实验结果, 本文通过 SeetaFace 的人脸检测算法与特征提取算法对 LFW 数据集进行筛选, 最终筛选得到 13 220 张有效人脸.

4.2 SeetaFace 性能

本文在 SeetaFace 的基础上进行了二次开发, 主要针对 SeetaFace Identification 部分样例代码进行修改, 将人脸检测算法、特征提取算法从样例代码中提取并编写为新的接口, 同时增加了一对多人脸相似度计算算法. 经实验环境下测试, 本文修改后的算法通过 JNI 调用时单张照片人脸检测速度为 19.3 ms、特征提取速度为 286.6 ms. 图 4 给出了一对多人脸相似度计算算法性能测试结果.

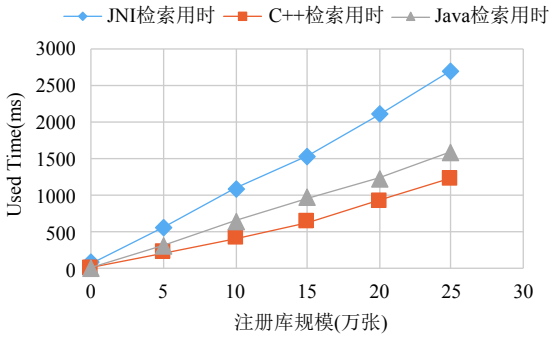

图 4 一对多人脸相似度计算算法性能测试结果

从图中可以看出, 在实验环境下, 通过 Java 计算相似度比通过 JNI 调用 C++算法有更高的效率.

4.3 加速比

加速比定义为同一个任务在单机系统和分布式系统中运行消耗的时间的比率, 用来衡量分布式系统的性能和效果[16]. Storm 任务的运行分为本地模式和远程模式, 其中本地模式是指 Storm 任务运行在本地机器的单一 JVM 上, 而远程模式是指 Storm 任务被提交到集群当中, 经任务调度后运行在不同节点的多个 JVM 上, 此时 Storm 任务中的各个线程是并行运行的, 且

Storm 任务中各模块并行度的改变会影响任务的计算效率. 本文以实时抓拍照片存储与检索任务在 Storm 本地模式下且任务中各模块并行度为 1 时运行的耗时为单机系统下的耗时, 而实时抓拍照片存储与检索任务在 Storm 远程模式下运行的耗时即为分布式系统下的耗时.

在实验中, 本文保持 HBase 中注册库的大小为 50 000 张人脸照片不变. 在本地模式下且任务中各模块并行度为 1 时运行实时抓拍照片存储与检索任务测得平均耗时为 9866.9 ms. 此后在远程模式下逐渐增大实时抓拍照片存储与检索任务中人脸检索模块的并行度并测得不同并行度下实时抓拍照片存储与检索任务的耗时, 最终获得了图 5 所示的不同并行度下实时抓拍照片存储与检索任务用时与加速比.

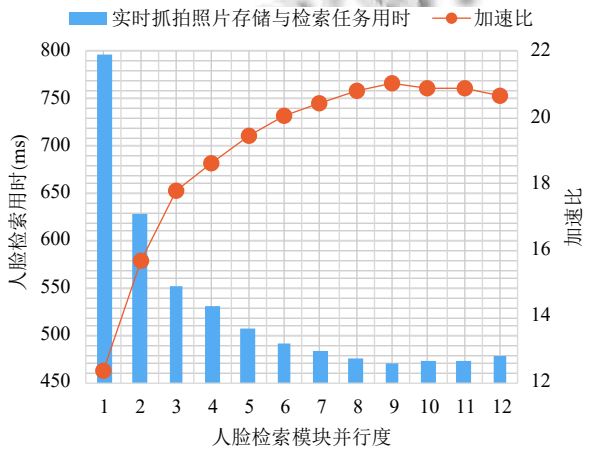

图 5 实时抓拍照片存储与检索任务用时与加速比

从图 5 中可以看出, Storm 任务运行在远程模式会有更好的实时性, 同时随着人脸检索模块的并行度的提升加速比会有显著的增长, 但由于在整个实时抓拍照片存储与检索任务中, 人脸检测与特征值提取也会占据大量的耗时, 因此加速比与人脸检索模块的并行度并不成正比. 当人脸检索模块耗时随着该模块并行度的提高而接近甚至低于人脸检测与特征值提取的耗时时, 加速比的增长逐渐趋于平缓. 此外, 随着人脸检索模块并行度的逐渐增大, Storm 任务中线程间通信的耗时也会增加, 这也是加速比的增长逐渐趋于平缓的原因之一, 而当线程间通信耗时在整个任务耗时中的占比足够大时最终加速比会出现回落. 实验中当并行度为 9 时, 加速比达到最大值 21.05.

5 结论与展望

本文所设计并实现的大容量实时人脸系统将人脸识别技术与视频监控融合, 实现了视频监控数据的实时处理与反馈, 减轻了后期人力调取历史图像数据的工作量. 通过多组实验表明, 该系统可以有效利用 Storm 平台的并行计算能力, 具有良好的实时性和扩展性.

本文的人脸检索仍采用了穷举搜索方式, 即对给定的一张人脸图像, 需要将其其特征描述向量和注册库中每张人脸图像的特征依次进行相似度匹配计算. 穷举搜索的时间复杂度为 O(n), 会随着注册库数据规模的增大而线性增加, 所以下一步的研究方向包括: 1) 注册库基于特征值的索引建立; 2) 基于索引的人脸检索.

参考文献

1 王倩, 谭永杰, 秦杰, 等. 基于 Hadoop 分布式平台的海量图像检索. 南京理工大学学报, 2017, 41(4): 442–447.
2 陈雯柏, 黄至铖, 刘琼. 一种基于 *P* 稳定局部敏感哈希算法的相似人脸检索系统设计. 智能系统学报, 2017, 12(3): 392–396.
3 顾志松, 沈春锋, 姚文韬, 等. 高清人像抓拍检索系统的设计与实现. 控制工程, 2015, 22(S1): 68–71.
4 陈新荃, 陈晓东, 蒋林华. 基于 Spark 平台的人脸图像检索系统. 计算机工程, 2018, 44(2): 251–256.
5 Cheng WC, Qian J, Zhao ZC, *et al*. Large scale cross-media data retrieval based on hadoop. Proceedings of the 2015 11th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness. Taipei, China. 2015. 133–138.
6 朱珊, 艾丽华. 基于 Hadoop 的大规模图像存储与检索. 计算机与现代化, 2017, (6): 61–66, 83.
7 Premchaiswadi W, Tungkatsathan A, Intarasema S, *et al*. Improving performance of content-based image retrieval schemes using Hadoop MapReduce. Proceedings of 2013 International Conference on High Performance Computing & Simulation. Helsinki, Finland. 2013. 615–620.
8 Uttarwar D, Agarwal A, Kadiwar R, *et al*. Distributed content based image search engine using hadoop framework. Proceedings of 2017 International Conference on Communication and Signal Processing. Chennai, India. 2017. 1706–1710.
9 Sabarad AK, Kankudti MH, Meena SM, *et al*. Color and texture feature extraction using apache hadoop framework. Proceedings of 2015 International Conference on Computing

Communication Control and Automation. Pune, India. 2015. 585–588.

10 Tungkasthan A, Premchaiswadi W. A parallel processing framework using MapReduce for content-based image retrieval. Proceedings of 2013 Eleventh International Conference on ICT and Knowledge Engineering. Bangkok. 2013. 1–6.

11 Hedjazi MA, Kourbane I, Genc Y, *et al*. A comparison of Hadoop, Spark and Storm for the task of large scale image classification. 2018 26th Signal Processing and Communications Applications Conference. Izmir, Turkey. 2018. 1–4. [doi: 10.1109/SIU.2018.8404688]

12 Apache Software Foundation. Apache HBase. https://hbase.apache.org/. [2018-08-20]

13 Visual Information Processing and Learning (VIPL) group. SeetaFace. https://github.com/seetaface/SeetaFaceEngine. [2018-08-20]

14 Apache Software Foundation. Apache storm. http://storm.apache.org/. [2018-08-20]

15 Huang GB, Ramesh M, Berg T, *et al*. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Workshop on Faces in RealLife Images: Detection, Alignment, and Recognition. Marseille, France, 2008. Inria-00321923.

16 朱为盛, 王鹏. 基于 Hadoop 云计算平台的大规模图像检索方案. 计算机应用, 2014, 34(3): 695–699.