



北京大学

本科生毕业论文

题目： 基于人脸识别的轻量级社交
平台

姓 名： 余俊峰

学 号： 1100012769

院 系： 信息科学技术学院

专 业： 计算机

研究方向： 移动互联网

导 师： 边凯归

二〇一五年五月

版权声明

任何收存和保管本论文各种版本的单位和个人，未经本论文作者同意，不得将本论文转借他人，亦不得随意复制、抄录、拍照或以任何方式传播。否则一旦引起有碍作者著作权之问题，将可能承担法律责任。

摘要

人际关系总是人们生活中非常重要一环，人们总是扩展自己的交际圈的需求无无所不在，在如今的互联网的时代下，移动应用成为了人们进行社交活动的重要方式。为了尽可能减少用户去扩展社交圈的障碍，令用用户可以更加有效地获取新的人际关系，很多的移动互联网网尝试用不同的方式去（例如 wechat 的“摇一摇”，QQ 的“漂流瓶”等）增强这用户的获取的难度。然而，这些现有的应用对于用户的信息利用不完全，往往不能提供令用户满意的效果。本文文提出了一种通过人脸识别的技术为主，用户的的其他信息为辅助来帮助用户去减少他们去扩展他们的社交圈的难度，在本系统中，将用户的人脸座位了一个非常重要的特征，来配合其他特征来进行匹配算法，为用户推荐好友。针对以上的想法，本文实现了一种适合多移动平台，覆盖广泛的应用，并设计了不同的匹配算法，通过实验衡量其性能和用户体验。最终确定了一个最优设计成为了最后一个的移动互联网网应用用，经实验证明该具有良好的体验，可以面向商用用进一一步推广广。而更进一步的，由于图片流的火热和现在 computer vision 技术的发展，可以根据人们在日常生活中所表现出的倾向进行学习和进一步的匹配，前景十分远大。

关键词：人脸识别，图片社交应用，移动设备，匹配算法

Test Document

Test (Some Major)

Directed by Prof. Somebody

Abstract

Test of the English abstract.

Keywords: First, Second

目录

第一章 引言	1
1.1 背景分析	1
1.2 主要研究任务	3
1.3 研究的目的是和意义	3
1.4 本文组织	3
第二章 相关工作	5
2.1 主要相关技术调研	5
2.1.1 人脸识别技术	5
2.1.2 移动应用	6
2.1.3 数据库调研	7
2.1.4 crawler	8
第三章 系统设计	11
3.1 匹配算法设计	11
3.1.1 基于人脸相似度的算法设计	11
3.1.2 基于用户信息的算法设计	11
3.1.3 基于机器学习的算法设计	12
3.2 应用端设计	12
3.3 face++ SDK for node.js	12
3.4 服务器设计	12
3.5 数据库设计	12

第四章 系统实现	13
4.1 crawler	13
4.2 front-end	13
4.2.1 提交界面	13
4.2.2 展示界面	13
4.2.3 详细资料界面	13
4.3 数据库	14
4.4 匹配算法设计	14
4.4.1 face++ 已有的人脸匹配算法	14
4.4.2 人脸特征	14
4.4.3 用户信息	14
4.4.4 机器学习算法	14
第五章 总结与展望	15
5.1 系统测试结果及分析	15
5.1.1 用户分析	15
5.2 匹配算法实验	15
5.2.1 根据人脸相似度的匹配	15
5.2.2 根据具体人脸特征的匹配	15
5.2.3 根据具体信息的匹配算法	15
5.2.4 机器学习算法	15
5.3 展望	15
5.3.1 基于用户特点喜好的更进一步的匹配算法	15
致谢	17

第一章 引言

1.1 背景分析

随着网络和移动设备普及，人们越来越多的用移动设备来进行自己的社交活动，现在层出不穷的社交类应用一种在涌现，更为显著的是 **wechat** 其生态圈已经成为了中国用户一种特定的生活模式，所以移动应用在人们社交关系中起到的作用不言而喻。

并且在这个信息爆炸时代下，由于大量的信息涌到人们眼前，而不断兴起的社交媒体更是进一步将信息碎片化。不像文字一般，照片、图像作为不易切割的单位在传递信息过程中备受用户青睐。而图片类社交应用因为其独特性而成为了新一代移动应用规定的一个热点，比如国外的 **instagram** 就是图片分享类社交的先行者。而 **flickr**，**snapchat**，乃至于现在出现在中国众多的社交应用都在充斥着人们的生活。甚至有人断言到“分享相片是社交网络的将来”。



图 1.1: flickr



图 1.2: instagram



图 1.3: snapchat

但与以往社交应用类似的是大多数的图片类社交应用要么是熟人社交圈或者弱关系社交圈。而在这种关系社交类应用中，存在着用户难以找到令他们感兴趣的用戶。

如今的社交应用都在尝试不同的方式帮助用户去寻找他们感兴趣的用戶。比如微信的“摇一摇”或者“漂流瓶”是一种具有随机性的方式，但除了一部分特定的人群，大部人的接受度并不高。而基于地点联系的方式，很多社交应用都为用户设置了查看附近的人，但使用这项功能的人并不是很多。而基于用户的兴趣，在

类似于 twitter,weibo 这种弱关系社交中，根据关注人的和用户的某些特性，推荐好友的功能，这给予了用户一个很好的体验。而在如今的 classification 中，deepwalk [引用]，line [引用，张铭的论文] 深度学习等算法都是未来根据用户关系来帮助用户去寻找他们感兴趣的人。

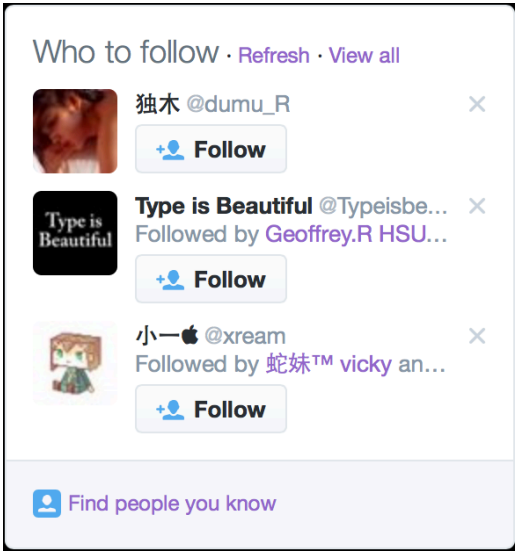


图 1.4: flickr



图 1.5: instagram

虽然现在的方式各种各样，但无可否认的是至今不存在一种很好的方式提供给用户去拓展他们的社交圈子。

这就产生了本研究的一个最初的动机：

能否通过一种图片交流的方式提供图片类应用的用户去认识新朋友的渠道？

而最近由于人脸识别技术的火热，市面上不断在涌现着与人脸识别关系密切的应用，比如 face++ 与阿里巴巴在支付宝上联合推出的“刷脸”支付功能。

这些给予了笔者以灵感，笔者拟应用人脸的特征去提供一种独特的方式给用户去拓展他们的社交圈子，结合了传统的“夫妻脸”的概念，其中最基础的方式就是以人脸相似度去实现一个应用。之后通过实验的不断调整，加入其他特征去调整最后的匹配算法，得出最后的适合用户的匹配好友算法。

1.2 主要研究任务

本研究主要的任务是对于图片社交应用的一种基于人脸识别方式的好友扩展方式的研究，为用户一个良好的方式去扩展他们的社交圈。因此本研究的主要工作总结如下：

- 1.
- 2.
3. 在移动平台上应用

1.3 研究的目的和意义

1.4 本文组织

第一章包括背景和对研究的分析, 及本文组织。第二章介绍论文相关技术和工作。第三章详细给出系统的算法和逻辑设计。第四章详细给出详细的系统的算法的。第五章进行总结和未来工作。

第二章 相关工作

2.1 主要相关技术调研

2.1.1 人脸识别技术

人脸识别是一项热门的计算机技术研究领域，它属于生物特征识别技术，是对生物体（一般特指人）本身的生物特征来区分生物体个体。生物特征识别技术所研究的生物特征包括脸、指纹、手掌纹、虹膜、视网膜、声音（语音）、体形、个人习惯（例如敲击键盘的力度和频率、签字）等，相应的识别技术就有人脸识别、指纹识别、掌纹识别、虹膜识别、视网膜识别、语音识别（用语音识别可以进行身份识别，也可以进行语音内容的识别，只有前者属于生物特征识别技术）、体形识别、键盘敲击识别、签字识别等。

Face++

Face++ 是新一代云端视觉服务平台，提供一整套世界领先的人脸检测，人脸识别，面部分析的视觉技术服务。Face++ 旨在提供简单易用，功能强大，平台通用的视觉服务，让广大的 Web 及移动开发者可以轻松使用最前沿的计算机视觉技术，从而搭建个性化的视觉应用。Face++ 同时提供云端 REST API 以及本地 API（涵盖 Android, iOS, Linux, Windows, Mac OS），并且提供定制化及企业级视觉服务。通过 Face++，您可以轻松搭建您自己的云端身份认证，用户兴趣挖掘，移动体感交互，社交娱乐分享等多类型应用。face++ 提供了相应的 http 请求的接口，并提供了 python, java, c++ 的 API 接口。在本项目中，作者对于 face++ 对于 nodejs 的 api 接口进行了补全，能够很好的支持 nodejs 对于 face++ 的接口应用，方便之后的 nodejs 开发者能够有效的利用该项技术。

caffe

Caffe (<http://caffe.berkeleyvision.org/>) 是一个清晰而高效的深度学习框架，其作者是博士毕业于 UC Berkeley 的贾扬清 (<http://daggerfs.com/>)，他目前在 Google 工作。Caffe 是纯粹的 C++/CUDA 架构，支持命令行、Python 和 MATLAB 接口；可以在 CPU 和 GPU 直接无缝切换。caffe 在图像处理方面有着自己独特的效果。

而比较了这两项技术之后，由于 face++ 是专门做人脸识别的技术，他的各项评测也遥遥领先，所以在本文准备使用 face++ 的 API 来实现应用。

Face++ 的 API 接口

2.1.2 移动应用

移动应用 (Mobile application) 安装和运行在移动设备上。它随着智能手机的推出，由于其移动性和娱乐性，在近年来达到了全新的巅峰。目前有三大主流的移动应用平台，分别为 IOS, Android 和 Windows Phone。据三大平台统计，从 2010 年到 2013 年，移动应用的数目增加了 100 多万个。因此，移动应用是目前软件开发的一种最流行的模式，移动设备庞大的用户量和较小的开发经费也吸引了越来越多的自由开发者加入到了移动应用的开发者行列之中。

2.1.2.1 angularJS

angularJS 是一款开源 JavaScript 函式庫，由 Google 维护，用來協助單一頁面應用程式運行的。它的目标是透過 MVC 模式 (MVC) 功能增强基于浏览器的应用，使开发和测试变得更加容易。函式庫讀取包含附加自定義 (標籤屬性) 的 HTML，遵從這些自定義屬性中的指令，並將頁面中的輸入或輸出與由 JavaScript 變量表示的模型綁定起來。這些 JavaScript 變量的值可以手工設置，或者從靜態或動態 JSON 資源中獲取。AngularJS 是建立在這樣的信念上的：即声明式编程應該用於構建用戶界面以及編寫軟件構建，而指令式編程非常適合來表示業務邏輯。[1] 框架採用並擴展了傳統 HTML，通過雙向的數據綁定來適應動態內容，雙向的數據綁定允許模型和视图之間的自動同步。因此，AngularJS 使得對 DOM 的操作不再重要並提升了可測試性。設計目標：將應用邏輯與對 DOM 的操作解耦。這會提高代碼的可測試性。將應用程序的測試看的跟應用程序的編寫一樣重要。代碼的

構成方式對測試的難度有巨大的影響。將應用程序的客戶端與服務器端解耦。這允許客戶端和服務器端的開發可以齊頭並進，並且讓雙方的復用成為可能。指導開發者完成構建應用程序的整個歷程：從用戶界面的設計，到編寫業務邏輯，再到測試。

2.1.2.2 ionic

Ionic 提供了一個免費且開源的移動優化 HTML，CSS 和 JS 組件庫，來構建高交互性應用。基於 Sass 構建和 AngularJS 優化，並形成了一個強大的 HTML5 應用程序開發框架，可以幫助您使用 Web 技術，比如 HTML、CSS 和 Javascript 構建接近原生體驗的移動應用程序。Ionic 主要關注外觀和體驗，以及和你的應用程序的 UI 交互，特別適合用於基於 Hybrid 模式的 HTML5 移動應用程序開發。且它是一個輕量的手機 UI 庫，具有速度快，界面現代化、美觀等特點。為了解決其他一些 UI 庫在手機上運行緩慢的問題，它直接放棄了 IOS6 和 Android4.1 以下的版本支持，來獲取更好的使用體驗。

Ionic 是現在 GitHub 上的最火的開源項目之一，具有超過 16,000 星及以上創建 600000 Ionic app。Ionic 遵循視圖控制模式，通俗的理解和 Cocoa 觸摸框架相似。在視圖控制模式中，我們將界面的不同部分分為子視圖或包含其他視圖的子視圖控制器。然後視圖控制器“驅動”內部視圖來提供交互和 UI 功能。一個很好的例子就是標籤欄（Tab Bar）視圖控制器處理點擊標籤欄在一系列可視化面板間切換。由於 ionic 在三大平台都能够使用，存在了很強大的適應性，並且針對於現有先進的移動設備。

2.1.3 数据库调研

為了收集可用的用戶信息，建立一個完整的圖片數據庫，我們需要調研一個基礎的能夠供我們對於用戶的實驗數據庫。

2.1.3.1 mongodb

Mongo DB 是目前在 IT 行業非常流行的一種非關係型數據庫 (NoSql)，其靈活的数据存儲方式備受當前 IT 從業人員的青睞。Mongo DB 很好的實現了面向對象

的思想 (OO 思想), 在 **Mongo DB** 中每一条记录都是一个 **Document** 对象。**Mongo DB** 最大的优势在于所有的数据持久操作都无需开发人员手动编写 **SQL** 语句, 直接调用方法就可以轻松的实现 **CRUD** 操作。**MongoDB** 的文档模型自由灵活, 可以让你在开发过程中畅顺无比。对于大数据量、高并发、弱事务的互联网应用, **MongoDB** 可以应对自如。**MongoDB** 内置的水平扩展机制提供了从百万到十亿级别的数据量处理能力, 完全可以满足 **Web2.0** 和移动互联网的数据存储需求, 其开箱即用的特性也大大降低了中小型网站的运维成本。

2.1.3.2 数据来源

根据本文所要实现的应用的特性, 是基于人脸的社交应用, 而无疑最符合这个性质的社交应用就是婚恋市场的社交应用。而也只有婚恋的用户才会在大概率上将他们的真实人物照片提交到应用上, 为了模拟这个应用的特性, 本文实验的数据库的来源也需要从类似的网站上抓取, 排除掉国外的婚恋网站, 作为国内的比较流行的婚恋网站有, 百合网, 珍爱网, 世纪佳缘。

通过比较这几个网站的数据, 发现百合网的会员资料的开放程度不及世纪佳缘, 珍爱网。而世纪佳缘的用户活跃度和注册数远远优于珍爱网。所以本文拟用世纪佳缘的数据作为本文应用的数据库来源。

2.1.4 crawler

Beatifulsoup

Beautiful Soup 提供一些简单的、**python** 式的函数用来处理导航、搜索、修改分析树等功能。它是一个工具箱, 通过解析文档为用户提供需要抓取的数据, 因为简单, 所以不需要多少代码就可以写出一个完整的应用程序。**Beautiful Soup** 自动将输入文档转换为 **Unicode** 编码, 输出文档转换为 **utf-8** 编码。你不需要考虑编码方式, 除非文档没有指定一个编码方式, 这时, **Beautiful Soup** 就不能自动识别编码方式了。然后, 你仅仅需要说明一下原始编码方式就可以了。**Beautiful Soup** 已成为和 **lxml**、**html6lib** 一样出色的 **python** 解释器, 为用户灵活地提供不同的解析策略或强劲的速度

API Overview

API Names		API Funtions
detect	/detection/detect	Detect the information of the given photo(e.g. face location, age, race, gender etc.)
	/detection/landmark	Return the facial parts and contour locations in the face.
train	/train/verify	Should be called once before calling /recognition/verify
	/train/search	Should be called once before calling /recognition/search
	/train/identify	Should be called once before calling /recognition/identify
recognition	/recognition/compare	Compare two faces and returns the similarity
	/recognition/verify	Given a person and a face, judge whether the face belongs to the person
	/recognition/search	Given a face and a faceset, find the most similar face (compared with the given face) in the faceset
	/recognition/identify	Given a face and a group, find the person in the group with the most likelihood (compared by the given face)
grouping	/grouping/grouping	Given a faceset, group the faces into several categories so that each category represents a person
person	/person/create	Create a person
	/person/delete	Delete a person
	/person/add_face	Add a face to a person
	/person/remove_face	Delete a face from a person
	/person/set_info	Add some information to a person
	/person/get_info	Get the information of a person
faceset	/faceset/create	Create a faceset
	/faceset/delete	Delete a faceset
	/faceset/add_face	Add a face to a faceset
	/faceset/remove_face	Delete a face from a faceset
	/faceset/set_info	Add some information to a faceset
	/faceset/get_info	Get the information of a faceset

图 2.1: *Face++API*

API Overview

API Names		API Funtions
detect	/detection/detect	Detect the information of the given photo(e.g. face location, age, race, gender etc.)
	/detection/landmark	Return the facial parts and contour locations in the face.
train	/train/verify	Should be called once before calling /recognition/verify
	/train/search	Should be called once before calling /recognition/search
	/train/identify	Should be called once before calling /recognition/identify
recognition	/recognition/compare	Compare two faces and returns the similarity
	/recognition/verify	Given a person and a face, judge whether the face belongs to the person
	/recognition/search	Given a face and a faceset, find the most similar face (compared with the given face) in the faceset
	/recognition/identify	Given a face and a group, find the person in the group with the most likelihood (compared by the given face)
grouping	/grouping/grouping	Given a faceset, group the faces into several categories so that each category represents a person
person	/person/create	Create a person
	/person/delete	Delete a person
	/person/add_face	Add a face to a person
	/person/remove_face	Delete a face from a person
	/person/set_info	Add some information to a person
	/person/get_info	Get the information of a person
faceset	/faceset/create	Create a faceset
	/faceset/delete	Delete a faceset
	/faceset/add_face	Add a face to a faceset
	/faceset/remove_face	Delete a face from a faceset
	/faceset/set_info	Add some information to a faceset
	/faceset/get_info	Get the information of a faceset

图 2.2: *Face++API*

第三章 系统设计

3.1 匹配算法设计

在应用上，

对于匹配算法的设计，主要的思想是探究出一种基于用户特征信息来推荐与用户志趣相投的人，所以其中必须要考虑到是现实用户对匹配算法实际效果的满意程度。之后对于不同的意见，对于匹配算法再进行不断对应的调整，从而得到一个相对最优的算法。故此，本文准备采取不同的匹配算法来实验，选取出实验效果最好的来当作最终实施在应用上的算法。同时，该算法可以根据用户的反馈来调整用户的信息的适应性，从而实现程序的自动调节。

3.1.1 基于人脸相似度的算法设计

设变量。。以 f 当作人脸相似度

$$C = \operatorname{argmax}_c \sum_t w_c^t \quad (3.1)$$

3.1.2 基于用户信息的算法设计

根据用户提供的信息，将此当作一个变量添加到匹配算法中，将用户信息提供的信息 i 设为 x_i ，而对应不同的信息 i 应该具有不同的匹配算法 f_i ，举个例子，比如年龄信息的可用正态分布的函数对于推荐用户年龄的进行一个匹配，即 $f_i(x_i)$ = 正态分布函数。

由此我们可以推导出我们新的匹配算法的公式

但由于用户的兴趣问题，我们针对于用户的猜测可能导致负相关，因此这里的变量越少越好。

3.1.3 基于机器学习的算法设计

3.2 应用端设计

根据社交应用的性质，关于应用端的逻辑应该能够提供给用户能够查看，交流，以及使用最主要的匹配功能，所以本文 **implement** 的应用端的逻辑如下图所示 [用苹果的 view 画一个图] 其中第一个 **View** 是主界面来展示用户的 **post** 信息，点入具体的一个 **card**，对应着具体的用户的 **post** 信息。

3.3 face++ SDK for node.js

由于 Face++ 并没有提供对应的 SDK 给 node.js，所以本研究需要为 face++ 的 http 请求的 api 实现一个 SDK 给 node.js。Face++ 的 API 如下图

3.4 服务器设计

[用 omnigraffle 画一个思维导图]

3.5 数据库设计

[用 er 图] [用 mysql 画的一个图]

第四章 系统实现

4.1 crawler

基于之前的数据来源的分析，并采用 beautifulSoup 技术来进行 crawler [伪代码 1] 提交到用户数据库的代码 [伪代码 2]

4.2 front-end

根据主界面的用户需求，系统需要连到数据库进行数据的获取，并需要对数据库进行刷新， [伪代码 2] [贴实际效果图]

4.2.1 提交界面

基于用户照片来源的不同 [伪代码] [实际效果图]

4.2.2 展示界面

[伪代码] [实际效果图]

4.2.3 详细资料界面

[伪代码] [实际效果图]

4.3 数据库

4.4 匹配算法设计

4.4.1 face++ 已有的人脸匹配算法

4.4.2 人脸特征

根据 face++ 能够得到用户的五官的位置

4.4.3 用户信息

根据网上的爬虫数据，能够得到男女，地区和星座年龄等信息，根据这一些进行匹配

4.4.4 机器学习算法

根据用户的感兴趣的图片，结合人脸的特征

第五章 总结与展望

5.1 系统测试结果及分析

5.1.1 用户分析

根据数据库的特性和年龄地区分布 [用户年龄分布图] [用户地区分布图]

5.2 匹配算法实验

5.2.1 根据人脸相似度的匹配

5.2.2 根据具体人脸特征的匹配

5.2.3 根据具体信息的匹配算法

5.2.4 机器学习算法

5.3 展望

5.3.1 基于用户特点喜好的更进一步的匹配算法

致谢

感谢我的母校北京大学,燕园浓厚的学术氛围,培养了一代代社会精英,祖国栋梁。与良师益友交游,谈笑有鸿儒,往来无白丁,这里的寸寸土地,都充满了智慧与知识。感谢所有教授过我各类课程的老师。他们丰富的学术积淀和睿智博学的个人风采都让我崇敬仰慕。感谢北京大学网络与信息系统研究所移动网络组为我提供的良好的实验环境。感谢边凯归老师和严伟老师,他们在我的工具实现,实验完成和论文写作方面给予了我充分的指导和鼓励。在进入移动网络组一年的时间里,我切身感受到边老师与严老师亲切的师长风采与迷人的学术魅力,每一次和老师的交流,都能得到许多启迪和灵感,这都是我人生中最宝贵的财富。感谢实验室和与我同级的同学还有朋友们:毛景树,薛易清,等等等等,他们在我的论文写作中,为我提供了充分的帮助,使我克服了许多困难,少走了许多弯路。还有很多参与了我毕设实验的亲人朋友还有同学们,正是有了他们对我的帮助,使我得以顺利完成论文。感谢我的父母,无论寒暑春秋都给我最无私的支持和鼓励。感谢所有关心我的亲人和朋友以及所有给与我帮助的人,我的每一份收获都离不开你们。

北京大学学位论文原创性声明和使用授权说明

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律结果由本人承担。

论文作者签名： 日期： 年 月 日

学位论文使用授权说明

（必须装订在提交学校图书馆的印刷本）

本人完全了解北京大学关于收集、保存、使用学位论文的规定，即：

- 按照学校要求提交学位论文的印刷本和电子版本；
- 学校有权保存学位论文的印刷本和电子版，并提供目录检索与阅览服务，在校园网上提供服务；
- 学校可以采用影印、缩印、数字化或其它复制手段保存论文；
- 因某种特殊原因需要延迟发布学位论文电子版，授权学校在 ☐ 一年 / ☐ 两年 / ☐ 三年以后在校园网上全文发布。

（保密论文在解密后遵守此规定）

论文作者签名： 导师签名： 日期： 年 月 日