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

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 version 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　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户。

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户。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 classfication 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。

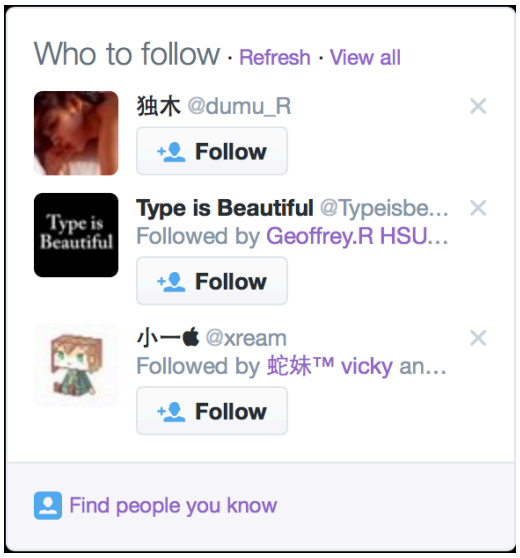

图 *1.4: flickr*

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好友推荐算法，得出最后的适合用户的匹配好友算法。

1.2 方案概述

本文提出基于人脸识别技术的图片社交平台，利用人脸识别的技术，不但能够提高人脸识别技术的通用性，并能提高用户对于社交应用的体验。根据不同的用户脸部特征，个人信息，本文对于基于用户特征值的匹配算法的可行性进行了系统的研究。基于已有的特征值，推荐给用户各项指标都符合其标准的好友，使得可以为用户提供一个良好的方式去扩展他们的社交圈。

因此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，基本的工作可以总结为如下：

1. 了解现有的社交应用给予用户的扩展好友的方式，调研相关的人脸识别的技术和算法，提出一种基于人脸特征的好友推荐匹配算法
2. 基于 face++ 提供的 api，完善在 node.js 平台上 sdk 的支持，并调研和准备应用的数据，存储到数据库中。
3. 基于 angularJS+Ionic 框架，开发出一款能够应用在三个移动手机平台上的应用。该应用不但能有效的解决用户好友来源的问题，并且可以在好友推荐中，为用户推荐符合用户的好友。同时，根据用户的反馈能够及时的调整对应的匹配算法。
4. 对于基于不同算法的好友推荐算法，进行实际的用户效果的测试，通过实验数据的比较取得一种较高满意度的推荐算法。实验结果表明本应用的好友推荐算法具有较高的实用价值和扩展性。

1.3 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。

2.1.1.1 技术选用

通过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。

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 函式库，由 Google 维护，用来协助单一页面应用程式运行的。它的目标是透过 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。函式库读取包含附加自定义（标签属性）的 HTML，遵从这些自定义属性中的指令，并将页面中的输入或输出与由 JavaScript 变量表示的模型绑定起来。这些 JavaScript 变量的值可以手工设置，或者从静态或动态 JSON 资源中获取。AngularJS 是建立在这样的信念上的：即声明式编程应该用于构建用户界面以及编写软件构建，而指令式编程非常适合来表示业务逻辑。[1] 框架采用并扩展了传统 HTML，通过双向的数据绑定来适应动态内容，双向的数据绑定允许模型和视图之间的自动同步。因此，AngularJS 使得对 DOM 的操作不再重要并提升了可测试性。设计目标：

1. 将应用逻辑与对 DOM 的操作解耦。这会提高代码的可测试性。
2. 将应用程序的测试看的跟应用程序的编写一样重要。代码的构成方式对测试的难度有巨大的影响。
3. 将应用程序的客户端与服务器端解耦。这允许客户端和服务器端的开发可以齐头并进，并且让双方的复用成为可能。

指导开发者完成构建应用程序的整个历程：从用户界面的设计，到编写业务逻辑，再到测试。

2.1.2.2 ionic

Ionic 提供了一个免费且开源的移动优化 HTML，CSS 和 JS 组件库，来构建高交互性应用。基于 Sass 构建和 AngularJS 优化，并形成了一个强大的 HTML5 应用程序开发框架，可以帮助您使用 Web 技术，比如 HTML、CSS 和 Javascript 构建接近原生体验的移动应用程序。Ionic 主要关注外观和体验，以及和你的应用程序的 UI 交互，特别适合用于基于 Hybird 模式的 HTML5 移动应用程序开发。且它是一个轻量的手机 UI 库，具有速度快，界面现代化、美观等特点。为了解决其他一些 UI 库在手机上运行缓慢的问题，它直接放弃了 IOS6 和 Android4.1 以下的版本支持，来获取更好的使用体验.

Ionic 是现在 GitHub 上的最火的开源项目之一，具有超过 16,000 星及以上创建 600000Ionic app。Ionic 遵循视图控制模式，通俗的理解和 Cocoa 触摸框架相似。在视图控制模式中，我们将界面的不同部分分为子视图或包含其他视图的子视图控制器。然后视图控制器“驱动”内部视图来提供交互和 UI 功能。一个很好的例子就是标签栏（Tab Bar）视图控制器处理点击标签栏在一系列可视化面板间切换。

2.1.2.3 技术选用

由于 angularJS+ionic 的框架在三大平台都能够使用，存在了很强大的适应性，并且针对于现有先进的移动设备.

2.1.3 数据库调研

为了收集可用的用户信息，建立一个叫完整的图片数据库，我们需要调研一个基础的能够供我们对于用户的实验数据库。

2.1.3.1 mongodb

Mongo DB 是目前在 IT 行业非常流行的一种非关系型数据库 (NoSql), 其灵活的数据存储方式备受当前 IT 从业人员的青睐。Mongo DB 很好的实现了面向对象的思想 (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

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算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

$$C = \underset{c}{\operatorname{argmax}} \sum_{t} w_c^t \tag{3.1}$$

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 xi，而对应不同的信息 i 应该具有不同的匹配算法 fi，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 fi(xi)= 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 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

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 likelyhood (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 *3.1: Face++API*

3.3 服务器设计

[用 omnigraffle 画一个思维导图]

3.4 数据库设计

为了完善我们的数据库的逻辑，本文对于数据库进行了 ER 图的描述，并用关系行数据库对于 NOSQL 进行了比对 [用 er 图] [用 mysql 画的一个图]

第四章　系统实现

4.1　crawler

基于在本文第二部分中的数据来源和爬虫的分析，采用 beautifulSoup 技术来进行扒取数据。

4.1.1　爬取基本数据

[伪代码 1]

4.1.2　导入数据库

提交到用户数据库的代码 [伪代码 2]

4.2　应用端

该应用的实现由三步组成: 状态查询, 信息源添加, 和匹配反馈。具体的应用端的逻辑，由一个类似的 iOS 框架图来类比。

4.2.1　状态查询及展示

用户在使用应用程序查看他人已经存在的用户信息，需进行对于用户信息的一个预览，以及对于具体应用信息的查询。

根据用户需求可以查看具体的信息，并可以支持用户随时对于需要的信息的刷新，以及链接用户更进一步的个人主页的途径。针对［伪代码 2］［贴实际效果

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 likelyhood (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 *4.1: Face++API*

图]

4.2.2 信息源添加

信息发布者在使用用应用用程序添加信息源时, 需进行行上下文文采集和添加详细信息两方方面面操作。由于前文文提到的传感器数据在精确度和稳定性上的弊端, 为了改善整个系统的工工作效果, 上下文文采集过程需要在应用用程序自自动采集传感器数据的基础上, 增加一一定的人人工工干干预以进行行优化。对于位置信息, 应用用程序将采集到的坐标显示示为地图上的一一个大大头针, 如果用用户认为此位置不准确, 可以拖动地图以调整到准确的位置。对于磁场计信息和 iBeacon 信号, 由于它们总是随着时间不停地变动, 应用用程序将它们以友好的形式显示示出来 (磁场计信息显示示为设备方方向角角度,iBeacon 信号显示示为每

个 iBeacon 的 minor identifier 和距离值), 用用户可以随时观察这些数值的变化, 并冻结在比比较满意的数值上 (图 1-a)。

信息发布者通常希望其发布的信息具有多种多样的形式, 考虑到移动设备较难编辑和显示示富文文本信息, 应用用程序被实现为能够编辑、存储和显示示具有多种类型的字段, 包括标题、详细介绍、URL 或应用用程序跳转链接 * 和图片片 (图 1-b)。

基于用户照片来源的不同 [伪代码] [实际效果图]

4.2.3 匹配信息展示

由于服务端将客户端提交的图片以及文本信息直接调用对应匹配算法的接口, 因此, 用用户在选择了信息识别编码后, 服务端便具有了足足够的信息以匹配出用用户想要了解的信息源。在服务端返回给客户端匹配出的结果后, 客户端将详细信息显示示在用用户界面面上 (图 3)。如果具有多条匹配结果, 客户端将按照匹配程度顺序将结果显示示为列表的形式。 [伪代码] [实际效果图]

4.2.4 详细资料界面

[伪代码] [实际效果图]

4.3

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

由于匹配系统不仅需要识别每个密码的键盘输入值, 还需要识别每个手指点击的二进制标签值 (非大即小)。

本文将密码识别问题看做一个分类问题 (或一个监督学习问题), 即根据用户本身图片的特征值和过去用户感兴趣的图片信息的历史特征值来对一个手指点击事件分类。TapLock 密码识别问题的目标是确定新发生的手指点击事件的标签 (即测试用例)。在密码设定过程中, 我们收集了一组过去的手指点击事件 (即训练用例), 用 D 表示这个集合。每个用例 d D 能被两个特征值和一个点击标签描述。令这两个属性 2 为 ,1 = h(), ,2 = (), 若用例 d 被分类为小 (或大) 则点击标签为 = 0(或 = 1)。

第五章　总结和展望

5.1　系统测试

为了分析和评估本论文提出的基于人脸识别的匹配算法的有效性和可用性，我们使用了该算法编写的基于婚恋数据的匹配好友的应用进行了实验测试。本实验主要真了基于不同算法的匹配系统进行了分析评估。本实验对于两个方面进行了分析和评估：

1. 数据库用户分析
2. 测试用户对于不同匹配算法的满意度

5.1.1　数据库用户分析

通过数据库的采取的 3000 条数据进行分析，对于数据库的用户数据进行了统计。测试用户和数据库的用户的年龄分布如下图

通过上表我们可以看出在数据库中的用户的年龄都偏大，超过半数的用户都超过了 30 岁，而测试用户却相反的半数以上都为 30 岁以下的青年人。带来这种差异的原因主要是数据库数据来源的特性，由于是婚恋网站，所以大多数的用户年龄都偏大，以及本文所邀请的用户的局限性主要限定在大学周围。由于年龄段的不同，用户有着不同的兴趣和审美观，而年龄差异过大的也并不适合称为一个社交圈子里的人。这种差异给匹配算法带来了一些不确定性。

但随着图片技术的发展（例如 Microsoft 的 How old，可以大致检测出人的年龄），使得我们可以在之后的工作中，为用户添加上这一特征值，从而进一步提供匹配算法的的准确性，降低这种差异给我们算法带来的不确定性。

5.1.2 实验结果

总共 20 人参加了该实验的测试，该实验的满意度的分数为 1-10 分，测试用户对于一个测试算法给出的 5 个推荐好友进行评分，最后去一个平均值作为最终的检测结果。而对于不同的匹配算法的满意度的结果如下图。

从表中可以看出由于加入了更多的一些特征到了匹配算法中，用户的满意度有了一定的提升。

本文总结

本文提出了一种新型的图片社交平台，通过人脸识别技术为主导的好友推荐匹配来有效的扩展用户的社交圈以及提升用户活跃的程度。在本应用中用户可以通过他们的信息特征来寻找与他们兴趣相符的好友。

本文实现了基于 angularJS + ionic 的客户端应用程序，以及以世纪佳缘为数据源导入的数据库，以 face++API 实现的 nodejs 的 SDK，已经对应的服务器程序。目前的实验表明本文的匹配算法具有良好的用户体验，能够有效的为用户提供适合用户特征的好友，具有良好的可扩展性和实用性。

实验数据毕竟具有一定的随机性，且得到的数据较为主管，因此在之后为了更好的去探索人脸识别技术图片社交平台上的应用，需要在之后我们考虑更广泛的实验，并且将此应用发布并推广出去去获得更多的用户的反馈进一步来精确原算法的阈值，获得更多用户的特征数据来提升算法。

而在图片识别等技术的飞速发展上，应用能够更多的通过图片来收集用户的信息，将用户的日常信息不断分类，从而更为精确的为用户匹配好友。我们可以期待图片社交应用更远大的未来。

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, 使我得以顺利完成本论文。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