# 第一章 绪论

## 1.1 选题的背景和意义

### 1.1.1 选题的背景

近年来，随着人脸识别技术的迅速发展，越来越多的基于人脸识别的应用进入了日常生活领域。从人脸识别考勤系统、人脸识别门禁系统等较为传统的人脸识别应用，到新近流行起来的“刷脸”支付的手机app、“刷脸”取款的ATM，人脸特征凭借着其方便、友好的交互方式，迅速取代了其他的冗余、繁琐的验证方式。但目前，绝大多数的人脸识别系统都把研发重心放置于身份识别的准确性以及算法效率上；却忽视了人脸识别的安全性问题，即——输入的人脸可以是非真实的脸（包括高分辨率的彩色打印照片、3D打印的人脸模型、人脸面具等），使用者以此可以仿冒他人身份通过人脸识别系统，对后续的应用安全性构成了极大地安全隐患。

任何一种验证系统，都会面临仿冒攻击的威胁。应对的方式无外乎以下两种：（1）通过保护验证信息，使仿冒者不易获取；（2）通过验证系统本身来鉴别验证信息是否有效。前者的主要应用场景就是密钥验证系统（包括对称密钥以及非对称密钥），使用此方式来保护验证系统对密钥持有者的要求较高，且密钥本身不易被他人获取。由于人脸的特殊性，它作为日常生活中人类的最重要的身份标示，被广泛地传播，极容易被他人获取；所以对于人脸验证系统而言，通过验证系统本身来鉴别验证信息的可靠性是唯一的解决方案。

因此，用于增强人脸识别系统的安全性的人脸活体验证技术受到了越来越多人的关注。人脸活体验证技术旨在通过人脸信息来验证目标的合法性，更加具体的说，即在传统的人脸识别系统之前，增加一道防火墙，所有试图进行人脸验证的输入都将首先进行人脸活体验证，通过验证的输入才能继续进行人脸识别，否则就认为是非法输入，从而拒绝访问。

### 1.1.2 选题的意义

信息安全的重要性和敏感性以及无需赘述。在人脸识别领域，使用者不仅仅需要方便快捷地进行身份验证，更希望验证系统能够保证其身份的安全性，即不被仿冒者盗用。本次课题就是针对这一应用环境，通过图像信息来对输入信息的合法性进行验证。

上一节已经提到，人脸活体验证系统是一个添加在人脸识别系统之前的认证模块，采用带有人脸活体验证模块的人脸识别系统有助于在保证信息真实有效并且可以使用的情况下大大提高人脸认证的效率。传统的人脸识别中，其可欺骗性会导致系统运行的结果不可靠等非预期性结果。所以在高敏感性的应用环境中，对于机器识别的结果，还需要人工进行筛查纠错，这就直接降低了人脸识别系统的实用性。因为动用人力来进行筛查纠错不仅成本高，而且效率极其低下，也容易发成错误。

人脸活体验证技术能够为人脸识别系统提供可靠的输入数据，使人脸识别系统的应用范围得到进一步的拓展。

## 1.2 国内外研究现状

### 1.2.1 人脸活体验证算法研究现状

随着人脸识别技术的成熟，在2000年左右，活体验证技术逐渐被各大领域重视。2006年，H. Bay和T. Tuytelaars等人提出了一种基于SFM（Structure from Motion）的活体检测模型。该模型通过定位眼睛、嘴巴等显著位置来预测特征点的三维深度，从而实现判断活体的方法。2009年，K. Kollreider , H. Fronthaler提出了利用光流来分析人脸各部位的移动量，最终进行活体检测的方法。2010年，孙霖提出了利用人脸识别进行多模活体验证的概念。其思想大多是将多个检测源的数据进行混合，然后联合多个带权检测结果，来进行评判。上述的算法虽然取得了较好的效果，均属于通用算法的范畴。由于算法的设计比较复杂，会导致较难进行实时性的结果计算。

### 1.2.2 人脸特征点定位算法研究现状

人脸特征点定位技术是指

## 1.3 本人主要工作

* 研究人脸活体验证系统的定位，了解相关的基本功能需求；并且通过阅读大量文献，掌握国内外有关人脸活体验证的发展现状。
* 根据系统的应用场景，明确系统角色，确定最后的项目功能性需求以及非功能性需求。
* 通过项目的需求分析，确定项目的整体架构以及相关系统功能模块的划分。
* 细化项目的各个模块设计，抽象出各个模块之间的通信接口，按照软件工程的项目规范把所有模块实现。
* 搭建系统的测试环境，并根据系统的功能性需求以及非功能性需求进行完善的测试。

## 1.4 论文组织结构

本论文的内容结构安排如下：第一章为绪论。该章论述了人脸活体检测系统的项目背景和项目意义，并且介绍了现阶段工业界和学术界相关的最新成果。第二章为相关技术介绍。

# 第二章 相关技术介绍

## 2.1 半朴素贝叶斯模型原理

半朴素贝叶斯模型（Semi-Naive Bayes Model）对朴素贝叶斯模型（Naive Bayes Model）的改进。该算法由Geoffrey I. Webb等人于2005年提出。该算法填补了朴素贝叶斯模型的两大缺陷：需要大量训练样本以及独立性假设与大部分特征不相符。假设输入的N维特征向量的形式为：f = (f1, f2, … , fN)，模型的输出类别为C∈{c1, c2, …, cK}。贝叶斯模型是一个通过计算后验概率并求出最大值所属的类别，并作为输出。贝叶斯模型的形式化表示为：

H(f) = argmaxk P(C=ck|f1,f2,…,fN)

使用贝叶斯公式可以将后验概率的形式变化为先验概率的形式：

H(f) = argmaxk P(f1,f2,…,fN | Ck) \* P(Ck)

在朴素贝叶斯模型中，假设所有的特征都是互相独立的，因此上述联合概率分布等于各个特征的先验概率的乘积，即为：

但是这个假设过于严格，在实际的应用中，该假设往往是不成立的，这就直接导致了朴素贝叶斯模型得出的结果与真实情况有较大的偏差。

半朴素贝叶斯模型取消了这个全局独立性假设，而是对特征进行分组，假设各个组内特征不独立，各个组间特征独立。假设N个特征被划分为L个组，每个组内含有S个特征，则上述的联合概率分布可以表示为：

通过计算每个组内的联合概率分布，并通过独立事件的组合来求得该先验概率，再通过最大化先验概率来得到整个模型的最终输出。

半朴素贝叶斯模型通过组内容量的大小来协调算法复杂度与准确度之间的关系。S的量级与算法准确度成正比，而L的量级与算法复杂度成正比。

## 2.2 随机蕨回归器原理

随机蕨回归器是由Dollár等人在2010年首次提出。该算法将半朴素贝叶斯模型进行了应用到了回归器的设计上。随机蕨回归器采用了级联式回归的设计方式，由多个回归器共同来约束输出结果。更具体地说，随机蕨级联回归框架的每一级回归器都是对上一级回归器所做出的结果的一个修正。该算法不同于之前流行的大部分算法，采用参数化的约束来控制输出结果；在随机蕨回归算法中，最终的输出是所有的训练集的一种线性组合，由此保证了在不依赖参数化约束模型的情况下，依然能够控制输出结果。

以图像的特征点定位的应用为例，随机蕨回归器在特征选择上，采用的是像素差异特征。选取的是随机生成的N对像素对的强度差，组成原始的特征向量。其形式如下：

X = {x1, x2, … , xn}

通过随机采样，将特征的数量缩减至S个，并且在每次迭代时生成S个划分阈值，将S个经过随机采样的特征映射成S个二进制位。映射规则如下：

在训练时，通过将S个二进制位的映射，可以把S维特征映射为0-2S-1之间的一个整数d，将该训练集存于序号为d的桶中，完成第一个回归器的训练。随后用第一个回归器预测出所有训练集图片的特征点，并将该误差作为第二级回归器的回归目标，以此类推，完成所有回归器的训练。

在预测时，对于每个回归器，对输入图片进行取“像素差异特征”后，再进行随机采样，并将映射的到的序号为d的桶中的所有回归目标中，选取一个残差最小的回归目标，作为本次回归器的输出。形式化表示如下：

其中，函数d(Si,S)表示第一个回归目标与序号为d的桶中所有回归目标的残差。

在预测时，对于整个级联回归器，将输入图片依次通过所有的回归器，将每个回归器输出的结果进行累加，即为整个级联回归器最终的输出。形式化表示如下：

RFinal = R0 + R1 + ... + RS

随机蕨回归器的训练，预测速度都非常快，并且由于采用了级联的方式约束结果，随着级联回归器数量的增加，回归器输出的误差粒度逐渐细化，可以得到比较准确的输出结果。由于级联框架中的每一级回归器的回归目标为上一级回归器与真实形状之间的残差，因此采用随机蕨这种较弱的回归器亦能取得较好的回归效果，并且能够极大地提高回归速度。

同样得益于级联回归的设计，经过Dollár等人证明，随机蕨回归器可以在每一个阶段以指数级的速度逐步收敛，因此该算法具有非常强的的鲁棒性，可以在不同的输入的情况下得到稳定的输出结果。

## 2.3 显式形状回归原理

## 2.4 主流人脸训练集介绍

## 2.5 Node.js介绍

Node.js是一款流行的高性能开源的网络服务与应用平台，使用JavaScript作为脚本语言。Node.js的构建基础是Google公司的V8引擎（Chrome JavaScript Runtime V8），而Google的V8引擎亦是目前全球效率最高的JavaScript解释器之一，这使得Node.js拥有非常好的并发性能以及非常快的执行速度。

Node.js能够提供一个使用事件驱动来实现异步开发的优秀解决方案。由于Node.js中绝大多数的API都是采用基于事件触发的异步调用模式设计的，非常适合进行高并发、高IO、低运算的网络开发。

Node.js旨在提供一种可伸缩、可扩展的高性能运行环境，以解决WEB服务器的性能瓶颈问题。由于传统的WEB服务器需要为每一个访问链接创建一个处理线程，会导致服务器只能响应十分有限的请求。而Node.js由于采用了单进程、单线程的事件驱动的设计方式，把每个新的请求加入事件循环（event loop）的轮询队列而非创建新线程，这个方式极大地减少了系统开销，提高了系统的处理并发连接的能力。

Node.js还是用模块化的设计来划分不同的功能包，这一特点也在很大程度上提高了开发效率。例如HTTP模块内封装了大量HTTP以及TCP/IP协议栈的操作函数，采用Node.js可以有效降低服务器的资源消耗，提高脚本的性能。

虽然Node.js与2013年才正式发布，但是已经得到了业界的一致认可，其中国际著名的社交网站LinkedIn、国际著名的开源项目托管网站GitHub和国内的阿里巴巴都把Node.js作为项目的重要组成部分。

## 2.6 MongoDB介绍

# 第三章 系统需求分析

## 3.1 系统的设计目的

## 3.2 系统功能性需求

### 3.2.1 离线训练数据

### 3.2.2 采集摄像头数据

### 3.2.3 目标眨眼检测

### 3.2.4 目标张嘴检测

### 3.2.5 目标摇头检测

## 3.3 系统非功能性需求

### 3.3.1 实时性

### 3.3.2 健壮性

### 3.3.3 可维护性

# 第四章 系统概要设计

## 4.1 系统整体架构

## 4.2 系统功能模块设计

# 第五章 系统详细设计

## 5.1 客户端详细设计

### 5.1.1 控制器模块设计

### 5.1.2 通信模块设计

### 5.1.3 图像采集模块设计

### 5.1.4 人脸特征点定位模块设计

### 5.1.5 眨眼检测模块设计

### 5.1.6 张嘴检测模块设计

### 5.1.7 摇头检测模块设计

## 5.2 服务器端详细设计

### 5.2.1 模型模块设计

### 5.2.2 控制器模块设计

### 5.2.3 视图模块设计

# 第六章 系统部署与测试

## 6.1 系统功能测试

## 6.2 系统性能测试

# 第七章 总结

## 7.1 论文总结

## 7.2 个人收获

## 7.3 改善空间

# 参考文献

1. Geoffrey I. Webb, Janice R. Boughton, Zhihai Wang. Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning, 58, 5–24, 2005 提出半朴素贝叶斯定理
2. Piotr Doll´ar Peter Welinder Pietro Perona. Cascaded Pose Regression. IEEE Conference on Computer Vision and Pattern Recognition, 2010 提出随机蕨回归器
3. Mustafa Özuysal, Michael Calonder, Vincent Lepetit, Pascal Fua. Fast keypoint recognition using random ferns. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2010 提出随机蕨分类器
4. 蒋秀鹏. 基于NodeJS的数字标牌系统的设计与实现[硕士学位论文]. 天津: 南开大学, 2014 参考了一些Node.js的概念
5. 王越. 基于nodejs的微博系统的设计与实现[硕士学位论文]. 成都: 电子科技大学, 2014 参考了一下nodejs以及mongodb的概念

# 致谢