

中山大学硕士学位论文

分布式人脸检索系统
设计与实现

Design and Implementation of Distributed
Face Retrieval System

学位申请人： 汤善敏

导师姓名及职称： 朝红阳 教授

专业名称： 软件工程

院、系（所）： 软件学院

答辩委员会主席： _____

委员： _____

二零一零年五月

论文题目： 分布式人脸检索系统设计与实现

专 业： 软件工程

硕 士 生： 汤善敏

指导教师： 朝红阳 教授

摘 要

随着“平安城”在全国各个城市的推广，分布在各地的视频监控系统正逐渐得到整合、规范和互联，为各级政府职能部门实现图像信息共享以及全市视频监控系统建设奠定了基础。然而目前一些智能视频分析技术的应用只是停留在实现单个区域目标的分析、跟踪和警告功能，并没有对整个网络的视频数据资源进行整合，形成一个系统的持久的信息源，从而难以实现跨区域的目标跟踪和调查，也没有充分发挥城市互联优势。随着城市互联，视频信息的整合和共享必然是视频监控智能化发展的要求。

本文针对上述问题，尝试提出分布式人脸检索系统的解决方案。该方案旨在通过从视频流中获取人脸信息，并建立面像索引库的方式来实现各地面像信息的共享。该方法能够在一定程度上满足整合视频信息的要求，同时分布式的面像信息还可以在在一定程度上为跨区域的调查、侦察活动提供帮助。

本文在该方案的基础上实现了一个简单的考勤系统。该系统能够自动完成图像预处理、人脸检测、跟踪、识别和面像库的建立等任务。面像库的记录还能反映人员的出入情况，以供用户查询之用。另外本文还分析和研究了目前主流的人脸检测、跟踪和识别技术，并通过选择适当的技术来保证系统设计的合理性和科学性。本论文按照软件工程的思想对系统进行详细的设计，并针对系统的功能需求和性能需求设计测试用例，对系统处理的速度、检测和识别的准确率等性能指标进行了测试，验证了所设计的系统在实际应用中的可行性。

关键词： 分布式系统，人脸检索，人脸检测，人脸识别，人脸跟踪

Title: Design and Implementation of Distributed Face Retrieval System
Major: Software Engineering
Name: Shanmin Tang
Superviso: Professor Hongyang Chao

Abstract

With the avocation and promotion of the "Safe City" plan in the range of many cities around the country, the video monitoring system scattered in various regions are gradually integrated, normalized and inter-connected. This has built the cornerstone of the image information sharing pool for different level government functional departments and the city video monitoring system.

However, the current video analysis techniques only support the functions of analyzing, tracking and warning of single object, but fail to integrate the video data to form a systematic and continuous information source, thus are not able to track and investigate the object in a cross-region manner. With the inter-connection of different cities, the sharing and integrating of the video information becomes a necessary requirement of intelligent video monitoring.

This thesis tries to propose a distributed face retrieval scheme to solve the aforementioned problems. The distributed face retrieval technique aims to build the image sharing pool for different regions by extracting face information from the video streams and establishing the index library. This method, in some extent, meets the requirement of video information intergradations. Meanwhile, the distributed face retrieval information would also be of help for the cross-region investigation and detection activities.

This paper implemented a fully-functional attendance checking system based on the distributed face retrieval scheme. This system is capable of images pre-processing, face detection, tracking and recognition, as well as image pool establishing. The image pool could also provide useful information for the users about the arriving and departing records of people. In addition, this paper further analysis the current popular schemes for face

detection, tracking and recognition, hence proper schemes can be selected to guarantee the scientificity and rationality of the system. This paper also brings forward a detailed design according to the idea of software engineering. Based on the functional demand and performance requirement, testing examples are designed to test the processing speed, accuracy of detections and recognition, thus verifying the feasibility of the system when applied in practical systems.

Keywords: distributed system, faces retrieval, face detection, face recognition, face tracking

目 录

摘 要.....	I
ABSTRACT	II
目 录.....	IV
第一章 前言	1
1.1 项目背景和意义	1
1.2 国内外发展现状.....	2
1.3 难点与论文的主要工作	7
1.4 论文结构简介	8
1.5 本章小结	9
第二章 核心算法和关键技术	10
2.1 核心算法	10
2.2 关键技术.....	28
2.3 本章小结	30
第三章 分布式人脸检索系统需求分析	31
3.1 系统概述.....	31
3.2 系统需求分析	31
3.3 系统主要用例分析	34
3.4 系统领域模型分析	41
3.5 本章小结	42
第四章 分布式人脸检索系统总体设计	43
4.1 系统架构及原理.....	43
4.2 分布式系统设计	44
4.3 系统功能设计	46
4.4 本章总结	49
第五章 应用实例-考勤系统.....	50
5.1 考勤系统功能分析	50
5.2 考勤系统软件结构	51
5.3 考勤系统硬件结构	52
5.4 考勤系统用例分析	52
5.5 考勤系统领域模型分析	56
5.6 考勤系统数据流分析	57
5.7 本章小结	60
第六章 考勤系统设计	61

6.1 线程池的设计 61

6.2 用例实现 62

6.3 系统数据库的设计 74

6.4 本章小结 75

第七章 系统详细设计 76

7.1 监控端模块详细设计 76

7.2 服务端详细设计 77

7.3 客户端模块详细设计 81

7.4 图像采集模块详细设计 82

7.5 人脸识别算法分析模块详细设计 83

7.6 本章小结 85

第八章 实现及测试结果分析 86

8.1 开发工具及运行环境介绍 86

8.2 软件测试 87

8.3 本章小结 95

第九章 总结与展望 96

总结 96

展望 97

参考文献 98

致谢 102

第一章 前言

1.1 项目背景和意义

随着经济的发展，城镇建设速度的加快，城市人口越来越密集，流动人口也越来越多，引发了城市建设的交通、社会治安等城市管理问题，这些正逐步对人们的公共安全构成威胁。因此旨在构建和谐社会的“平安城市”创建活动正在全国范围内展开。平安城市是一个覆盖整个城市的大型视频监控系统，业务范围涵盖治安、交管、消防、刑侦、内保等多个公安警种，应用范围也非常广泛。对于如此庞大的视频监控系统而言，传统的视频监看、录像回放等功能已经远远不能满足实际的需求，如何对海量的视频信息进行有效的开发利用，使得视频监控智能化是监控系统技术发展的重要趋势。视频监控智能化是指能够借助计算机强大的数据处理功能，利用智能视频技术对视频画面中的海量数据进行高速分析，提取有用的关键信息而过滤用户不关心的内容，从而实现充分整合视频资源，提高视频监控系统能力，以较低的整体成本实现安全性的提高。目前视频智能分析技术主要的应用是采用目标识别、跟踪和行为判断相结合的方法，通过分离场景中的背景和目标来分析并追踪摄像机监控范围内出现的目标。用户可以利用视频内容分析的功能，根据不同摄像机的场景预设不同的报警规则，一旦目标在场景中出现了违反预定义规则的行为，系统将自动发出报警，使用户能及时采取相关的措施。该分析技术虽然发挥了自动“电子眼”的作用，帮助执法人员提高审视各点监控图像的效率，然而却没有实现有效整合和分析所有视频资源的功能。随着视频监控系统的普及，庞大的视频数据能为侦察、调研等活动提供丰富的信息来源。为了充分利用这些信息，本文尝试提出分布式人脸检索系统方法，通过整合分布各地的视频数据和分析视频数据的方法，提高视频数据的有效利用率，从而进一步提高整个视频监控系统的智能化水平。

分布式人脸检索系统动态从视频流获取人脸信息，进行识别和跟踪，并建立面像索引库，同时实现各地的面像信息库的共享。这在一定程度上满足了视频信息整合的要求。同时分布式的人脸信息库还可以为跨区域的调查、侦察活动提供一定程度上的帮助。对人脸信息库的挖掘还可以寻找潜在的犯罪模式，对犯罪活动进行预测，防止罪

恶的发生,提高执法效率。另外在系统设计和应用实现过程中,对核心的技术进行分析和比较,它在一定程序上反映了目前的主流技术在实际应用过程中存在的各种问题,为日后技术改进提供一定的研究方向。

1.2 国内外发展现状

1.2.1 视频监控系統国内外发展现状

视频监控是安全防范系统的重要组成部分。最初,它以模拟闭路电视设备为主,系统通过电缆或光缆传输模拟视频信号,然后通过视频矩阵,将视频信号输出到监视器。由于在传输过程中信号容易受到外界干扰,视频信号容易失真,图像质量得不到保证。另外对于已经建好的系统,要增加新的监控点很困难,系统的扩展能力差,因此这种系统只适合小范围的区域监控。随着社会的发展,人们对安全防范越来越重视,传统的模拟设备监控系统已经远远不能满足人们的需要,因此出现了模拟与数字混合的监控系统。该系统采用模拟设备采集数据,并将模拟信号转换为数字信号,通过局域网传输数据,最后利用计算机的高速数据处理能力对视频进行处理,大大提高了视频质量。随着多媒体技术、视频压缩技术的发展以及计算机存储容量的提高,视频监控系统全面进入数字化时代。数字监控系统除了拥有传统闭路电视监控系统的所有功能外,还具有远程视频传输与回放等功能,其功能和性能对比前两种系统得到了极大的提高^[1]。由于在监控过程中人作为主要监控者,使各类视频监控系统或多或少存在报警精确度差、误报和漏报现象多、报警响应时间长、录像数据分析困难等缺陷,从而导致整个系统在安全性和实用性的降低。因此监控系统在数字化基础上,又引入了智能化分析技术。

目前智能化分析技术典型的应用如物体跟踪,人脸识别,车辆识别、烟火检测、人流量统计、人群控制、人体行为分析、交通流量控制等。国外的智能视频监控技术发展比较成熟,其中有美国国防高级研究计划局(DARPA)开发的视频监控和监测系统(VSAM)^[2],该系统集成了移动物体实时检测和跟踪、识别通用对象、人步态分析等一系列先进技术,能够对监控地区进行全方位的实时监控。美国国际商机公司(IBM)

与美国马里兰大学联合开发的实时视觉监控系统W4^[3]，它通过结合形态分析和跟踪技术定位人和其身体各部分，建立外观模型实现多人跟踪，同时它还可以检测人是否携带物体等简单行为，从而实现在室外环境下实时监测人在视觉监控范围内的活动。国内也有些高校和公司进行这方面的研究，清华大学计算机谢树煌教授等人利用实时视频图像识别技术，对人进行检测与统计。上海复旦大学施容杰教授等人利用行为分析等技术，对人群聚集与突发行为进行分析和检测^[4]。中科院李子清等人利用基于人脸识别中子空间的统计学习方法实现“靠人脸开启门”，其识别速度达到 0.01 秒。另外一些it公司如中兴、华为等，也近年来陆续推出如全球眼、新连同“神眼”，中移动“移动全球眼”等视频监控产品。

由以上可知，目前视频监控系统大致发挥实时检测、跟踪、识别、行为分析、及时警告等功能，却缺乏一个持久的信息化系统对检测或者识别结果进行保存及利用。本文的提出的分布式人脸检索系统正是希望通过对检测结果进行保存，建立一个持久化的信息系统。在该系统基础上，可以挖掘一些潜在犯罪模式或者跨区域跟踪和调查等。分布式人脸检索应用的技术主要有图像检索、人脸检测、跟踪和识别。以下将会大概介绍这几项技术的发展状况。

1.2.2 图像检索国内外发展现状

图像检索的发展大致分为两个阶段，第一个阶段主要以文本描述为主。它将图像作为数据库存储的对象，用关键字或者文本对图像进行描述，然后根据查询关键字或者给定文本内容，查找符合查询条件的图像。然而这种技术存在严重的问题。一方面图像的文字信息很小，人工智能很难自动完成图片的标注。因此必须依靠人工对图像进行标注。另一方面，不同的人对图像的理解不同，例如一幅含有海滩和森林的图片，有人认为是森林的图片，也有人认为是海滩的图片，因此不同的人对同一幅图片标注不同，主观理解差异成为影响图像检索正确性的重要因素。另外图像的视觉特征也很难用文本进行描述。为了解决这些问题，第二阶段采用基于内容的图像检索技术。它融合图像处理、模式识别、图像理解等多种技术，直接从图像提取信息线索，避免人工描述的主观性，大大减少工作量。用户只要选择具有代表性的图像进行查询，它就能通过提取特征和计算特征相似度方式，查找符合用户条件的图像。图像检索的关键在于提取特征值，目前关于图像特征值的描述有颜色特征、纹理特征和形状特征等。

颜色空间的选取有有HSV, RGB等; 纹理特征可以采用MSAR特征, 也可以采用Gabor小波特征, 形状特征如hog、haar等。

1.2.3 人脸检测国内外发展现状

人脸检测最早来源于人脸识别。识别系统要求对一般环境的图像具有一定的适应能力, 也就是说能够完成人脸自动提取到识别过程, 人脸检测成为自动人脸识别系统的关键环节。目前国内外关于人脸检测的文献有很多, 其算法大概分为 3 类。

1. 基于肤色模型的方法

该方法主要通过肤色模型将图片划分肤色区域, 然后在肤色区域寻找眼睛、鼻子、轮廓等人脸固有特征, 从而达到检测人脸的目的。其代表有R. L. Hsu^[5]和Yu-Ting Pai^[6]。他们的方法大致都是在YCrCb空间通过光照补偿技术和非线性颜色变换解决不同光照下的肤色分割问题, 然后利用基于特征的技术提取眼睛和嘴特征, 从而实现人脸的检测。这些方法的主要缺点是YCrCb颜色空间并不能够完全摆脱光照的影响, 当图像亮度较高时, 即使进行光线补偿, 也无法检测肤色特征, 算法失效。

2. 基于模板匹配的方法

该方法主要考虑人脸图像经过归一化处理后, 人脸图像结构具有很强的相似性, 从而可以通过比较目标模板与候选图像区域之间的相似度来设定相应的阈值来判断候选图像区域是否为人脸区域。其代表有Yullie^[7]提出的基于弹性模板匹配的人脸检测方法。该算法主要的缺点是过分依赖模板, 然而对模板的选取又没有明确的定义, 算法计算量大, 精度不高。

3. 基于机器学习的方法

该方法主要通过提取大量人脸和非人脸的特征如gabor、haar等, 然后利用机器学习方法对这些特征进行训练, 构造一些分类器, 最后通过分类器检测人脸区域。其代表有Viola^[8]等人采用类Haar特征, 然后利用AdaBoost学习算法构建级联分类器的人脸检测系统。W. Kienzle^[9]等人改进的SVM分类方法构造快速级联分类器的人脸检测系统。由于通过提取人的脸部形状特征进行训练, 所以不受光照影响, 漏检情况较小, 但是这样算法也容易受类人脸区域影响, 出现误测情况。

1.2.4 人脸跟踪国内外发展现状

人脸跟踪是指根据初始帧的人脸目标模型，在后续帧确定该模型位置的过程。目前国内外关于人脸跟踪的文献有很多。其算法大致分为三种：

1. 基于运动信息对目标进行跟踪

通过差分法、光流法等提取运动区域，然后根据人脸信息对运动区域进行人脸检测。其代表有清华大学的艾海舟等人采用差分法对办公桌前的人脸进行了定位跟踪^[10]。Anandan等人将人脸划分为若干个平面，然后根据光流法估计这些平面的位置实现人脸跟踪^[11]。然而差分法容易受光照变化和遮挡的影响，运行物体完整性。差光流法计算大，难以满足实时要求。

2. 基于目标特征进行跟踪

通过检测人脸区域，建立目标特征模型，用卡尔曼滤波、粒子滤波和均值移动等算法估计目标模型的位置。目标特征模型包括肤色模型和轮廓面部特征等。其代表有Jebara^[12]等人在初始帧用肤色模型提取人脸，然后在以后视频序列用扩展Kalman滤波获取目标位置信息。该算法的缺点是卡尔曼滤波跟踪的精度不高。X. Xu和 B. Li^[13]利用一组带有权值的状态空间中的采样粒子给出离散化形式的后验概率分布。由于在实际应用中采样的粒子个数必须充分大，才能保证离散形式的概率分布与原始连续形式的概率分布等价，所以粒子滤波运算量比较大，难以达到实时性的要求。Khan et al^[14, 15]提出一种MCMC (Markov chain Monte Cralo) 的采样方法，减少粒子的采样个数，提高粒子滤波的跟踪效率，然而都难以达到实时要求。Comaniciu^[16]等人提出基于跟踪目标颜色概率模型的均值移动目标跟踪算法，该算法计算复杂度低，速度快，满足实时要求，现已被广泛应用在人脸跟踪。Bradski^[17]等人利用目标颜色特征提出均值移动的自适应算法-CamShift。该方法的优点是速度快，不受人脸姿势和表情变化的影响，但是容易受背景中类颜色区域及光线变化的影响。

3. 基于统计和学习方法进行跟踪

Shai Avidan^[18]提出EnSemble Tracking的方法。该方法利用上一帧的像素构造特征向量，并通过Adaboost训练出强分类器，强分类器对当前帧的像素进行分类，分为属于跟踪对象和不跟踪对象两类，然后根据分类置信徒来确定当前帧感兴趣的区域，进而训练出新的Adaboost强分类器。

1.2.5 人脸识别国内发展状况

人脸识别最早始于Bledsoe^[19]，他以人脸特征点的间距、比率等参数为特征建立一个半自动人脸识别系统。随着人脸识别算法的发展，识别算法从特征提取角度可以分为以下三类：

1. 基于几何特征的人脸识别方法

该算法主要提取人脸主要器官特征信息及其之间的几何关系，利用几何特征进行识别。其代表有Brunelli等人^[20]采用改进的积分投影法提取人脸特征，并用欧氏距离进行分类。

2. 基于弹性模板匹配的人脸识别方法

该算法主要是将人脸上的一些特征点作为基准点构成弹性图，每个基准点存储一串具有代表性的特征矢量，形成稀疏人脸描述结构，最后根据图像库中的人脸描述结构与待识别人脸描述结构的相似度来实现人脸识别。其代表有Yuille^[21]采用弹性模板来提取眼睛、嘴巴和脸部轮廓，Huang和Chen^[22]采用Snakes模型提取眉毛、下巴和鼻孔等不确定形状信息。

3. 基于线性与非线性子空间人脸识别

基于线性和非线性子空间人脸特征分析方法。它主要通过寻找一个线性或者非线性空间变换，将图像中对分类最有用的特征留在子空间，使得高维向量在子空间的分布更紧凑，从而达到压缩原始数据维度的目的，降低计算复杂度。目前比较成功线性子空间分析方法有主成分分析PCA、线性鉴别分析LDA、独立主元分析ICA和非负矩阵因子NMF等。非线性主要有核主成分分析KPCA和核FISHER鉴别分析。基于子空间的特征提取强调的是不同人脸的差异而不是光照条件、人脸表情及方向的变化，因此有助于提高识别效果。

如果从分类的角度又可以把算法分为以下四类：

1. 最近邻接法的人脸识别算法

该方法主要把提取后的人脸特征向量与图片库所有图片的特征向量进行比较，选择使得相似度最小的向量，则该向量所属的类为待测图片所属的类。相似度的计算方式有欧氏距离、马氏距离等。

2. 基于模型的人脸识别算法

该方法在通用人脸模型的基础上设定特定人脸参数来表征一类人的人脸属性，进行分类设计。目前，常用的人脸模型有隐马尔科夫、三维可变形模型等。

3. 基于机器学习的人脸识别算法

该方法主要通过提取人脸特征进行样本学习，为每一个模式类别分别设计分类器，最后通过分类器决定待测人脸所属的类。目前比较出名的机器学习方法有神经网络、SVM和Adaboost。神经网络分类器的优点是减少相邻类之间的误分类，但是随着样本数目的增加，神经网络的性能容易下降。

4. 基于压缩感知的人脸识别算法

有别于机器学习的方法，该方法不针对具体的某一类图片，而是利用所有类的图片构成一个特征向量集，并假设向量集足够稀疏，待测图片的向量由该向量集线性组成，通过求解L1-norm问题，求出与待测图片最接近的向量，从而决定待测图片属于哪一类图片。

1.3 难点与论文的主要工作

分布式人脸检索系统在设计 and 实现过程中，难点在于：

1. 系统对人脸检测算法、人脸跟踪算法以及人脸识别算法有很高要求。系统的可行性依赖于这些算法。人脸检测算法能够从视频流中准确检测到人脸。人脸跟踪算法能够准确快速地跟踪多个目标，同时能够解决目标之间相互遮挡等问题。人脸识别算法要求很高的识别率，即使在图像失真或者遮挡的情况下，也能够正确识别物体。由于目前存在各种个样的人脸检测、人脸跟踪以及人脸识别算法，系统这些算法的选取应该具备科学性和合理性。

2. 系统的性能和并发处理能力要求较高。由于系统每时每刻都要处理来自各个监控点大量的识别任务，识别任务应该如何传递、保存和并发处理是系统设计的关键。

3. 系统的稳定性要求较高。由于网络连接的不稳定性导致服务端/客户端，服务端/从机，服务端/监控端之间连接处理复杂化。如何处理各种网络连接以及在连接异常时应该采用什么样的处理机制也是系统设计的难点。

为了解决以上问题，论文的主要工作有以下几点：

1. 为了选择合适的算法应用到本系统，作者实现目前几种主流人脸检测、人脸跟踪以及人脸识别算法，并对其进行分析选取较为满足系统设计需求的算法。在人脸检测

算法方面，作者还结合肤色模型改进adaboost人脸检测算法。在人脸跟踪方面，作者结合hsv直方图实现基于EnSemble-track算法的多目标跟踪。

2. 为提高任务处理的速度，作者根据MapdReduce程序设计模型对分布式人脸检索系统并行处理任务的函数接口进行具体设计。使用线程池技术对考勤系统并行处理任务的接口进行设计和实现。

3. 提出整个分布式人脸检索系统的设计方案。其中包括对需求进行详细分析，建立系统的领域模型，并对系统架构以及功能进行详细设计。

4. 根据分布式人脸检索系统的设计方案实现一个简单应用-考勤系统，并对考勤系统的功能进行详细的分析和设计，最后为系统设计测试用例，测试系统是否满足实际需求。

1.4 论文结构简介

本文来源于对视频智能化技术的研究，为实现视频信息的整合和共享，本文提出分布式人脸检索系统的设计方案，并根据该设计实现一个具体的应用系统-考勤系统。

本文的组织结构如下：

第一部分：主要介绍项目背景和意义，通过国内外技术的发展状况，阐述本系统的技术路线和研究难点。

第二部分：主要介绍分布式人脸检索系统应用的核心算法和关键技术。核心算法包括人脸检测技术、跟踪技术、人脸识别，通过分析这些算法的特点，对这些算法进行适当的修改使其满足系统要求要求。

第三部分:主要介绍分布式人脸检索系统的总体设计方案。包括系统需求，系统总体架构，功能的设计等等。

第四部分：介绍分布式人脸检索系统的具体应用-考勤系统的设计和实现。包括描述具体系统的需求分析，详细设计与实现。

第五部分：对具体应用系统模块进行测试和总结应用系统设计与开发的难点，并就系统的改进和展望进行探讨。

本人主要负责该项目的需求调研和分析、方案设计，系统软硬件平台的选择以及具体系统的设计和实施全过程。

1.5 本章小结

本章主要介绍项目的背景和意义，以及视频监控系统国内外发展状况，通过其分析，得到系统构思的来源。另外本章还介绍本系统应用的几种技术的国内外发展状况，为日后系统选取适当的技术提供一个理论基础。另外本章还介绍了项目的目标和范围，本系统将围绕这个目标和范围开展设计。下文将会介绍具体的某个技术的理论知识。

第二章 核心算法和关键技术

本章主要介绍分布式人脸检索系统应用的核心算法和关键技术。核心算法主要有脸检测、人脸跟踪和人脸识别；关键技术主要有分布式计算、多线程技术。在人脸检测方面，作者对Viola^[8]的Adaboost人脸检测算法和Yu-Ting Pai^[9]肤色模型的人脸检测算法进行对比和分析，实验表明Viola^[8]的方法漏检率低，但容易出现误检，肤色模型的人脸检测算法正确率高，但漏检严重。因此作者结合这两种算法进行人脸检测，它在一定程度上改善了人脸误检情况。在人脸跟踪方面，作者介绍了meanshift和ensemble track两个人脸跟踪算法，并在其算法的基础上进行了稍微的改进，使其满足自适应目标大小和多目标跟踪的要求。实验结果表明ensemble track跟踪的效果比meanshift好，但是对于快速行走或相互遮挡的物体，这两种算法都容易失效。在人脸识别方面，作者采用eigenface, fisherface和Laplacianfaces提取特征，并采用最近邻接法、支持向量svm和基于稀疏表示的分类方法进行识别，实验表明基于稀疏表示的分类方法比其余两种方法好，但是算法的好坏仍然依赖图片库和测试集的质量，随着测试集的质量的下降，识别率也逐渐降低。从目前的分析状况可见，人脸检测、人脸跟踪和人脸识别算法在实际应用中仍然存在不少问题，需要进一步研究和改进。随着这些算法的改进，系统的准确性（检测准确性、跟踪准确性、识别准确性）和运行效率等都会得到极大的提高。在关键技术方面，本文介绍了分布式计算的MapReduce分布式程序设计模式，本文的分布式系统在该模式基础上进行设计。

2.1 核心算法

本节主要介绍人脸检测、人脸跟踪和人脸识别算法。

2.1.1 人脸检测算法

人脸检测是指根据输入的图像确定是否存在人脸以及人脸大小、位置过程。本文主要采用基于矩形特征的Adaboost人脸检测算法。由于Adaboost算法训练的强分类器只具有最小误检率，而不是最高检测率，虽然漏检情况很低，但存在误检的情况，所以需要其他算法进一步确认检测结果。肤色是人脸的主要特征之一，基于肤色特征的人

脸检测的优点是对旋转、尺度及表情变化等都不敏感，而且肤色特征很稳定，在区别其他物体颜色方面具有很强的鲁棒性和稳定性，而且基于肤色特征的人脸检测算法简单，检测速度非常快，因此本文采用基于肤色特征的检测算法对其进一步确认，虽然检测结果有进一步改善，但是容易受类肤色区域影响，因此人脸检测算法尚待改进。下面将会介绍基于矩形特征的AdaBoost的人脸检测和基于肤色特征模型人脸检测算法的原理。

2.1.1.1 基于AdaBoost的人脸检测算法

Haar特征最早由Papageorgiou等人应用到人脸检测。Papageorgiou在研究人脸检测问题时，为了提高检测所需的分辨率和增加了模型的表达能力^[4]，他们使用3类二维的haar小波函数用于提取对象的特征。但是Viola^[8]发现正方形的Haar滤波不足以描述人脸的所有特征，例如人的眼睛和鼻梁上侧的灰度差的绝对值很大，是一个矩形的区域特征，如果用正方形的Haar滤波描述就会丢失该信息。因此他将Haar滤波扩展为矩形滤波，并将特征原型扩展为3种类型4种形式的矩形特征原型。这些特征原型在图像中平移或伸缩得到的特征称为矩形特征，然后再将这些特征通过Adaboost的训练得到基于矩形特征的分层人脸检测算法。

该方法的具体实现过程如下：

1. 首先使用矩形特征表示人脸，使用“积分图”实现特征数值的快速计算。计算的方法如公式 2-1 到 2-2^[8]。

$$\text{haar-like 特征: } feature = \sum_{i \in (1, \dots, N)} w_i \text{RecSum}(r_i) \quad (2.1)$$

其中 $\{1 \dots N\}$ 表示该特征由N个矩形构成， w_i 为第i个矩形的权值， $\text{RecSum}(r_i)$ 为第i个矩形内所有像素之和。

$$\text{积分图定义为}^{[8]}: ii(x, y) = \sum_{x' \leq x} \sum_{y' \leq y} I(x', y') \quad (2.2)$$

其中 $I(x', y')$ 为图像在 (x', y') 处的像素值。

图像的积分图递推公式为^[8]：

$$\begin{cases} s(x, y) = s(x, y-1) + i(x, y) \\ ii(x, y) = ii(x-1, y) + s(x, y) \end{cases} \quad (2.3)$$

其中 $s(x, y) = \sum_{y_1 \leq y} i(x, y_1)$ ，表示原图像 (x, y) 点所在列纵坐标不超过该点的所有像素灰度的和。有了图像的积分图就可以快速地计算图像中任意矩形内所有像素灰度积。

2. 然后使用Adaboost算法挑选出一些最能代表人脸的矩形特征(弱分类器)，并将这些弱分类器组成一个强分类器。弱分类器的选取为公式 2-4

Adaboost的弱分类器 $h(x, f, p, Q)$ 由特征值 f ，阈值 Q 和指示不等号方向的 p 组成^[8]：

$$h(x, f, p, \theta) = \begin{cases} 1, & pf(x) < p\theta \\ 0, & \text{else} \end{cases} \quad (2.4)$$

Adaboost的选取过程^[2]：

- a. 计算每一种矩形特征原型对应的所有样本的特征值 $f(x)$ ，并进行排序
- b. 扫描这些特征原型产生的特征值 $f(x)$ ，计算其对应的最小分类误差 ε ^[8]。

$$\varepsilon = \min((S^+ + (T^- - S^-)), (S^- + (T^+ - S^+))) \quad (2.5)$$

T^+ 、 T^- 表示全部人脸样本权重和与非人脸样本的权重和， S^+ 、 S^- 表示排在该特征值前的人脸样本权重与非人脸样本权重和。 $S^+ + (T^- - S^-)$ 表示在此特征前的人脸权重和加上后面非人脸权重和，如果这个值较小，说明在此特征前的特征为非人脸，后面的为人脸，而且误检率较小，反之亦然。

c. 选择最小分类误差对应的矩形特征原型的长、宽和 (s, t) ，即为弱分类器的特征 f ，这个分类器的阈值为 Q ，判断最小分类误差是 $S^+ + (T^- - S^-)$ 还是 $S^- + (T^+ - S^+)$ ，如果是前者，则表示小于阈值的为非人脸，大于阈值的是人脸，那么 p 为-1， p 为+1。

这样就确定了一个弱分类器，并且这个弱分类器对应的分类误差为 ε 。

d. 经过 T 次循环，可以得到 T 个弱分类器 $h_1(x), h_i(x)$ 及其对应的分类误差 $\varepsilon_1, \dots, \varepsilon_T$ 。通过公式 2-6^[8]，构造强分类器。

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{else} \end{cases} \quad (2.6)$$

$$\alpha_t = \log \frac{1}{\beta_t} = \log \frac{1 - \varepsilon_t}{\varepsilon_t} = -\log \varepsilon_t$$

其中

3. 最后将训练得到的若干个强分类器组成一个级联的多层分类器。由于多层级联分类器要求每一层最小检测率为 99%，而最大误差率为 50%，而Adaboost算法训练的强分类器只具有最小误检率，而不是最高检测率，要达到多层级联分类器各层的要求，需要对Adaboost训练出来的强分类器的阈值进行调整。对于每个人脸样本 x_j ，其弱分类器加权求和为^[8]：

$$PD[j] = \sum_{t=1}^T \alpha_t h_t(x_j) \quad (2.7)$$

对 PD 进行排序，选择满足多层级联分类器要求的每层最高检查率的阈值 Th。
 $Th[i] = PD[p(1 - d_{\min})]$ 。其中 p 为人脸个数。即保证 $p \cdot d$ 个人脸被检测出来。

2.1.1.2 基于肤色特征的人脸检测算法

肤色是人脸的主要特征之一，基于肤色特征的人脸检测的优点是对旋转、尺度及表情变化等都不敏感，而且肤色特征很稳定，在区别其他物体颜色方面具有很强的鲁棒性和稳定性。因此本文采用文献[9]的方法进行人脸检测。

其方法大致如下：

1. 对图像进行光线补偿。
 - a. 计算图片 YCbCr 颜色空间的 Y 分量，其公式如下：

$$Y = 0.3R + 0.6G + 0.1B \quad (2.8)$$

- b. 归一化 Y 分量然后求出其平均值 Yavg
- c. 求出补偿的图片

$$R'_{ij} = (R_{ij})^\tau \quad (2.9)$$

$$G'_{ij} = (G_{ij})^\tau \quad (2.10)$$

$$C_{ij} = \{R'_{ij}, G'_{ij}, B_{ij}\} \quad (2.11)$$

$$\tau = \begin{cases} 1.4, Y_{avg} < 64 \\ 0.6, Y_{avg} > 192 \\ 1, otherwise \end{cases} \quad (2.12)$$

2. 重新计算图片的 YCbCr 颜色空间的 Cr 分量，肤色特征主要几种在该分量：

$$C_r = 0.5R' - 0.419G' - 0.081B \quad (2.13)$$

3. 对图像进行二值化

$$S_{ij} = \begin{cases} 0, 10 < Cr < 45 \\ 1, otherwise \end{cases} \quad (2.14)$$

4. 使用 5*5 的低通滤波过滤图像的噪音，即是过滤离散的肤色点

5. 划分肤色区域

6. 找出每个检测区域其眼睛和嘴区域

$$\theta = \cos^{-1} \left(\frac{0.5(2R' - G' - B)}{\sqrt{(R' - G')^2 + (R' - B)(G' - B)}} \right) \quad (2.15)$$

那么属于嘴的二值区域为：

$$M_{pq} = \begin{cases} 0, \theta < 90 \\ 1, otherwise \end{cases} \quad (2.16)$$

属于眼睛的区域为：

$$E_{hw} = \begin{cases} 0, 65 < Y < 80 \\ 1, otherwise \end{cases} \quad \text{其中 } \bar{h} \text{ 为 } h_m - w_{\max} \quad (2.17)$$

2.1.1.3 两种算法实验分析

1. 实验数据

本文从实际应用出发，在实验室采集的三个视频，并用这两种算法对这些视频的图像序列进行人脸检测。视频一：实验室门的打开，整个实验室比较光亮，有个人从实验室的门口走近摄像头。视频二：有人从实验室门口走近摄像头，但是实验室的整体光线均匀。视频三：人在实验室内走动。

3. 实验结果分析



图 2-1 视频一肤色模型人脸检测算法结果



图 2-2 视频一 adaboost 人脸检测算法结果

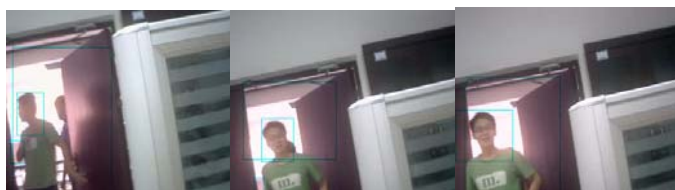


图 2-3 视频二 肤色检测人脸算法结果



图 2-4 视频二 adaboost 人脸检测算法结果



图 2-5 视频三肤色模型人脸检测算法结果



图 2-6 视频四 adaboost 人脸检测算法结果

上图为从视频一、二、三采集部分图像序列进行人脸检测的结果。从实验结果分析可知，adaboost 算法检测的人脸区域比肤色模型检测的人脸区域更准确。肤色模型检测算法容易受类肤色的影响，检测的人脸区域往往比实际的人脸区域要大；当光

线比较暗时，检测算法容易失效。Adaboost 人脸检测算法不受光线颜度的影响，人脸检测的区域比上述算法准确，但当人脸比较小和特征不明显时，adaboost 也难以检测到人脸，有时候 adaboost 还会出现误差。总得来说，adaboost 更符合实际应用需求，所以本文采用 adaboost 进行人脸提取，采用肤色模型进行人脸确认。

2.1.1.4 结合肤色特征与 AdaBoost 的人脸检测算法

1. 首先应用Adaboost进行人脸检测, 其大致过程为^[8]：
 - a. 假设待检测图像大小为 $W \times H$ ，计算其积分图
 - b. 初始化检测窗口大小为： $w \times h$ ，设置检测窗口最大放大倍数为 \maxSize
 - c. 对于多层级联分类器的每一层强分类器，使用放大倍率为 $size$ 的检测窗口对图片进行检测，如果是人脸，则标注为 1，如果不是，则标注为 0。当检测窗口小于某一个值时，就停止检测。
 - d. 最后对于所有检测到人脸的窗口，如果两个检测窗口的中心位置差值不超过两者的长度、宽的均值，则认为检测出的是同一个人脸，那么将其合并，取其均值。
2. 检测结果后，建立图片的肤色模型，计算肤色区域的面积与非肤色区域的面积之比 P 。如果图像是人脸，那么其肤色一定占绝大部分，所以假设本文中的设定阈值 $T=0.5$ 。如果 $P>T$ 则认为是人脸，否则认为是非人脸，直接抛弃。

2.1.1.5 实验结果

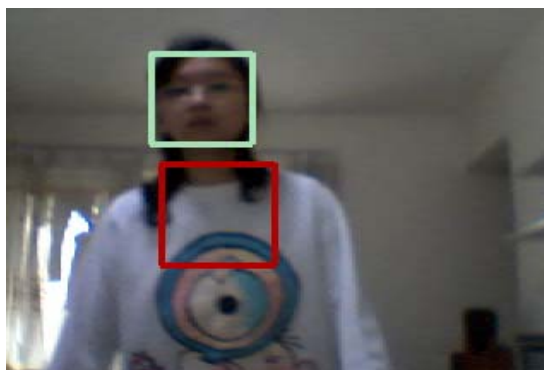


图 2-7 测试图片



图 2-8 检测图片 A

图 2-9 检测图片 B

图 2-8 和图 2-9 是对图 2-7 用基于 Hog 的 Adaboost 人脸检测的结果，从图中我们可以看出图 2-9 是误测的结果，而图 2-8 是正确的结果，如何去除图 2-9，而保留图 2-8 的结果呢？下面将采用了肤色检测进一步确认结果。

对图 2-8 和图 2-9 进行肤色特征检测，其二值化图像分别如下：



图 2-10 图片 A 二值化图片



图 2-11 图片二值化图片

由上图可知图 2-8 肤色区域占整个区域的 0.5278，图 2-9 肤色区域只占整个区域的 0.1071。由于截取的人脸区域容易受周围头发等影响，所以本文取图像的中心区域作为检测。再次检测后的图像结果：



图 2-12 图片 A 中心区域



图 2-13 图片 A 中心区域

此时图 2-12 的人脸区域占整个区域的 0.910，而图 2-13 的人脸区域为 0，本文设定阈值为 0.5。所以图 2-9 的图片可以去除，而图 2-8 的图片保留。

2.1.2 多目标人脸跟踪

人脸跟踪是指根据初始帧的人脸检测结果，在后续帧序列中确定人脸位置的过程。由于跟踪目标容易受光照变化和遮挡等影响，因此对跟踪算法的鲁棒性要求较高。另外，考虑到系统实时性，对跟踪算法也要求具有比较高的运动效率。因此，在复杂的环境下快速，正确地跟踪目标是跟踪算法的关键所在。人脸跟踪算法目前较常采用 meanshift 和 camshift 跟踪算法，这两个算法虽然满足实时性要求、鲁棒性较好，但是 Camshift 是一种自适应的跟踪算法。它在一定范围内，寻找和目标模型相似的颜色区域，然后再用 meanshift 寻找目标对象的位置，从而达到自适应改变跟踪窗口的目的。Camshift 最大的缺点是容易受类颜色区域的影响，使得其跟踪窗口在实际应用中经常出现比目标对象大很多的情况，不适合多目标跟踪。而传统的 Mean shift 跟踪算

法目标尺度一旦确定后无法改变,另外初始化的颜色直方图在跟踪过程中容易收到光线的影响。为了解决这些问题,本文尝试在跟踪过程中结合人脸检测,不断更新目标模型,从而达到自适应的多目标人脸跟踪目的。另外本文还对Ensemble track进行了稍微的改进,使其满足多目标跟踪要求。Ensemble Track是2005年Shai Avidan^[29]提出的新一类跟踪算法,它把跟踪问题看成是分类问题,其跟踪效果基本上达到实时要求,对于单个目标跟踪有较强的鲁棒性。从实验结果可以看出,改进后的Ensemble track比改进后的meanshift跟踪效果要好,其跟踪窗口更准确,然而对于快速运动的物体跟踪这两种算法都比较差,对于交叉或重叠的物体跟踪都容易出错,这些问题有待进一步研究。

2.1.2.1 基于 Mean Shift 的目标跟踪算法

Mean-shift算法是一种非参数概率密度估计算法,它可以通过迭代快速收敛于概率密度函数的局部最大值。文献[22]提出基于MeanShift的目标跟踪算法。该方法的算法过程大致如下:

1. 在起始跟踪帧通过加权直方图方法建立目标的颜色概率分布。

假设目标区域的中心为 x_0 ,则目标模型的概率密度分布定义如下^[36]:

$$q_u = c \sum_{i=1}^n k \left\| \frac{x_0 - x_i^*}{h} \right\|^2 \delta[b(x_i^*) - u] \quad (2.18)$$

核函数^[36]:

$$k(x) = \begin{cases} 0.5C_d^{-1}(d+2)(1-x), & x \leq 1 \\ 0, & \text{else} \end{cases} \quad (2.19)$$

其中 x_i 表示第 i 个像素的坐标。 h 是核函数的带宽,由公式上可以看出对于距离中心点较近的像素, $k(x)$ 分配较大的值,这是因为此像素的很可能是背景;而对于距离目标中心较近的像素, $k(x)$ 分配较大的权值,这是因为此像素很可能是被跟踪的目标。函数 b 是像素 x_i 在直方图的颜色索引。 $\delta[b(x_i^*) - u]$ 是判断目标区域的像素值是否属于第 u 个 bin (为了量化颜色空间,将颜色空间均匀划分为 m 个相等的区间,每个区间成为一个 bin,从而构成特征空间)。 C 是归一化常

量系数，使得所有特征值的概率和为 1.

2. 然后用同样的方法建立当前帧候选目标的颜色概率分布

候选目标在当前帧以 y 为中心，选用相同的核函数和核宽度，则候选模型的概率密度分布描述如下^[22]：

$$q_u(y) = c \sum_{i=1}^n k \left\| x_i - \frac{y}{h} \right\|^2 \delta[b(x_i) - u] \quad (2.20)$$

其中 x_i , $k(x)$, h , b , C 的含义和目标模型的相同

3. 接着用Bhattacharyya系统来描述两个分布的相似度^[22]

$$\rho(y) = \rho(p(y), q) = \sum_{u=1}^m \sqrt{p_u(y)q_u} \quad (2.21)$$

$\rho(y)$ 在 $[0,1]$ 之间，值越大表示两个模型越相似。其中使得 $\rho(y)$ 最大的候选模型就是目标在本帧的位置。

4. 以两个分布的相似性最大为原则，通过 mean-shift 向量的迭代运算，使搜索窗口沿密度增加最大的方向运动，收敛于最佳位置。

Mean Shift向量 $m_{hG}(y)$ ^[22]：

$$m_{hG}(y) = y_1 - y_0 = \frac{\sum_{i=1}^{nh} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{nh} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} - y_0 \quad (2.22)$$

其中 $g(x) = -K'(x)$, $w_i = \sum_{u=1}^m \sqrt{\frac{q_u}{p_u(y_0)}} \delta[b(x_i) - u]$, Mean Shift 算法反复迭代，最后得到当前帧目标最优位置。

该方法的主要缺点是目标尺度一旦确定后无法改变，另外初始化的颜色直方图在跟踪过程中容易收到光线的影响，为了解决这些问题。作者在跟踪过程中通过不断进行人脸检测，从而不断改变目标尺度的大小，另外重新计算颜色直方图，从而可以大

大提高 mean shift 的跟踪效果，改进的 Mean Shift 的人脸跟踪算法流程：

- a. 在开始帧，初始化目标跟踪模型
- b. 在下一帧根据目标模型，通过 Mean shift 估计目标模型的位置
- c. 对当前帧检测人脸位置
- d. 判断该帧人脸的位置是否与 Meanshift 估算后的目标模型的位置有 90% 的区域，如果有，那么将目标模型和该检测人脸视为同一个人，重新计算人脸所在区域的目标颜色概括分布，更新 Meanshift 跟踪目标模型。如果没有，那么这次的跟踪会被认为失败的跟踪，放弃跟踪，重新计算人脸所在区域目标颜色概率分布，重新建立 Meanshift 跟踪目标模型。
- e. 显示跟踪的结果。

2.1.2.2 Ensemble Track 的目标跟踪算法

Ensemble Track 将追踪问题看作分类问题来处理。它通过为图像的每一个像素创建特征向量，并用 Adaboost 训练一个强分类器来区分来自目标的像素和来自背景的像素，把对象从背景中区分出来。例如给定一个新的视频帧，它使用分类器来预测这些像素并形成分类置信图（confidence map），然后使用 mean shift 算法来找该图的峰值，即是物体移动的位置。

1. 分类器的设计

（1）特征值的定义：

Shai Avidan^[22] 使用局部方向直方图加颜色求解特征值。局部直方图的量化维度可以随意设定，一般取 8 bin 的颜色空间选择 RGB 颜色空间，维度为 3，因此总共的特征维度为 11。下面是具体求解过程。

使用 Sobel 算子分别求解图像 $I(x, y)$ 的水平和垂直梯度^[22]

$$G_x(x, y) = Sobel_x * I(x, y) \quad (2.23)$$

$$G_y(x, y) = Sobel_y * I(x, y) \quad (2.24)$$

那么图像边缘能量值^[22]：

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \quad (2.25)$$

去掉图像噪音后边缘能量值为^[22]:

$$G'(x,y)=\begin{cases} G(x,y), & \text{if } G(x,y)\geq T \\ 0, & \text{otherwise} \end{cases} \quad (2.26)$$

边缘梯度为^[22]:

$$\theta(x,y)=\arctan\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (2.27)$$

量化维度则^[22]

$$\psi(x,y)=\begin{cases} G'(x,y), & \text{if } \theta(x,y)\geq bin \\ 0, & \text{otherwise} \end{cases} \quad (2.28)$$

那么区域R对应的特征值^[22]:

$$F(R)=\left[\sum_{(x,y)\in R} \psi(x,y) \quad \sum_{(x,y)\in R} I_r(x,y) \quad \sum_{(x,y)\in R} I_g(x,y) \quad \sum_{(x,y)\in R} I_b(x,y) \right] \quad (2.29)$$

(2) 弱分类器的定义^[23]:

$$h_i(x) = \begin{cases} 1 & \text{if } F_i(x)\geq T_i \\ -1 & \text{otherwise} \end{cases} \quad (2.30)$$

其中 $F_i(x)$ 表示第 i 个特征值, T_i 表示阈值。每个特征值训练相应的弱分类器

(3) 强分类器定义^[23]:

$$H_t(x) = \text{sign}\left(\sum_{i=1}^n a_i h_i(x)\right) \quad (2.31)$$

(4) 类标签定义^[23]:

$$y_i = \begin{cases} +1 & \text{inside}(R_j, p_i) \\ -1 & \text{otherwise} \end{cases} \quad (2.32)$$

2. Ensemble Tracking 算法流程

假设有 n 个视频帧分别为 $I_1 \dots I_n$ 在第 i 帧, 初始化跟踪目标窗口为 r_1 。那么 Ensemble Tracking 的跟踪过程如下^[23]:

在第 i 帧, 初始化 r_1 的样本点 $\{x_i\}_{i=1}^N$ 的类标签 $\{y_i\}_{i=1}^N$

1. 初始化权值 $\{w_i\}_{i=1}^N$ 为 $1/N$

2. For $t=1 \dots T$

a) 训练弱分类器 $h_t(x_i)$

b) 设置误差 $err = \sum_{i=1}^n w_i |h_t(x_i) - y_i|$

c) 设置弱分类器系数 $a_t = \frac{1}{2} \log \frac{1-err}{err}$

d) 更新样本点的权重 $w_i = w_i e^{(a_t |h_t(x_i) - y_i|)}$

3. 得到强分类器为 $H(x) = \text{sign}(\sum_{t=1}^T a_t h_t(x))$

在第 $i+1$ 帧，提取样本点 $\{x_i\}_{i=1}^N$

4. 用强分类器测试样本点，生成置信图

5. 使用 meanshift 在分类置信图寻找峰值 r_j

6. 重新定义 r_j 对应的类标签 $\{y_i\}_{i=1}^N$

7. 去掉前 k 个弱分类器

8. For $l=k+1 \dots T$

a) 选择误差最小的弱分类器 $h_l(x)$

b) 更新弱分类器的系数 a_l 及其对应的样本点权重

c) 去掉 $h_l(x)$

9. For $t=1 \dots K$

a) 训练分类器 $h_t(x)$

b) 计算相应的 err 和 a_t

c) 更新样本点权重

10. 最后更新的强分类器 $H(x) = \text{sign}(\sum_{t=1}^T a_t h_t(x))$

3. 改进的 Ensemble Tracking 的人脸跟踪算法流程

- a. 在开始帧，提取目标对象及其附近的特征样本（假如目标对象为 r ，那么提取样本点的范围为 $2r$ ），设置类标签，根据样本训练强分类器 $H(x)$ 。提取目标对象的 hsv 颜色直方图。
- b. 在下一帧提取该范围的特征样本，根据强分类测试样本，生成置信图，保存目标对象的颜色直方图。
- c. 使用 $meanshift$ 在分类置信图寻找峰值，即为跟踪目标所在位置
- d. 对当前帧检测人脸位置
- e. 判断该帧人脸的位置是否与估算后的目标模型的位置有 70% 的重叠区域，如果有，那么进一步比较该目标的 hsv 直方图与跟踪模型的颜色直方图的相似度，如果相似度达到 0.8 以上，那么将目标模型和该检测人脸视为同一个人，标记根据区域。
- f. 对于没有标记为跟踪模型的人脸重新计算其颜色直方图，并新建一个跟踪模型。
- g. 重新提取新目标对象及其附近的样本，设计类标签，根据样本重新训练强分类器。重复上述过程，直至结束。

2.1.2.3 实验结果

图 2-14 和图 2-15 是应用改进后 $meanshift$ 和 $ensemble track$ 算法进行单个目标跟踪的结果。其中红色和紫色框代表检测的人脸，黑色代表该人脸跟踪的窗口。在连续 7 帧中，由这两个算法得出的目标跟踪窗口偏离目标对象有 3 帧，出现这个问题的主要原因是无论 $meanshift$ 还是 $ensemble track$ 在建立目标模型的过程中都主要使用目标跟踪对象的颜色信息（颜色概率分布和局部直方图），因此当目标对象周围出现相同的颜色区域时，目标跟踪窗口容易偏离目标对象。在图中可以看出人与脖子之间有相同的颜色区域，所以跟踪窗口容易向下偏移。应用这两个算法进行多个人跟踪的结果。当人与人之间出现相互遮挡时，跟踪对象会产生错误，这主要是在跟踪具体对象时只是应用位置信息进行检测，而缺乏人脸具体特征信息，所以无法准确判断具体的对象。这个问题将会作为下一步研究的重点。从实验结果可得， $ensemble track$ 比 $meanshift$ 跟踪的效果较准确而且比较稳定。



图 2-14 meanshift 单人跟踪结果



图 2-15 ensemble track 单人跟踪结果

2.1.3 人脸识别

人脸识别主要通过提取人脸特征，并根据这些特征进行身份识别的过程。目前比较成功的人脸识别算法主要用线性和非线性的子空间分析方法提取特征值，并选择适当的分类器进行相似度计算。下面将会介绍主流的子空间分析技术。

2.1.3.1 主成份分析 PCA

主成分分析 PCA 是指将多个变量通过线性变换以选出比较重要的变量多元统计分析方法。它最早由 Kirby 等人提出应用在人脸识别上，通过将样本集进行线性变换找到最优的单位正交基，用它们的线性组合来重构原样本，使得重构后的样本与原样本误差最小。后来 Turk 等人将它发展成为特征脸。特征脸就是将人脸的图像区域看作随意向量，然后通过主元分析得到一组特征脸图像，给定任意的人脸图像都可以近似为这组特征脸图像的线性组合，用组合的系数作为人脸的特征向量。

假设训练集 $\Gamma_1, \dots, \Gamma_M$ ，那么训练集的平均值 $\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$ ，计算每一张图片与训

练集的差值 $\Phi_i = \Gamma_i - \Psi$ 。计算训练集协方差的特征值和特征向量的公式为^[24]：

$$C = \frac{1}{M} \sum_{i=1}^m \Phi_i \Phi_i^T = AA^T \quad (2.33)$$

$A = [\Phi_1 \Phi_2 \dots \Phi_m]$ ，选择前 k 个最大的特征值 λ_k 和对应的特征向量 u_k 。 u_k 具有与人脸相似的形状，称为特征脸。将人脸图像映射到由特征脸张成的特征子空间

$\omega_k = u_k^T (\Gamma_i - \Psi)$ ，选择适当的分类器进行分类。

2.1.3.2 线性判别分析 LDA

Fisher线性判别分析方法最早由P. N. Belhumeur^[24]等人提出。他们指出特征值大的特征向量并不一定是分类性能最好的方向，主元分析忽视了类内与类间的关系^[32]。因此FLDA提出从高维特征空间里提取最具判别能力的低维特征，这些特征能够将同一类的所有样本聚集在一起，而不同类别的样本尽量分开，也就是选择能够使样本类间离散度和样本类内离散度比值最大的特征。

假设训练集 $\Gamma_1, \dots, \Gamma_M$ ，那么训练集的平均值 $\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$ ，每一类的平均值

$$\bar{\Gamma} = \frac{1}{N_i} \sum_{i=1}^{N_i} \Gamma_i, \quad C \text{ 表示训练集的种类数, } N_i \text{ 表示每一类的样本数目。}$$

定义类间的矩阵^[24]：

$$S_b = \sum_{i=1}^c N_i (\bar{\Gamma}_i - \Psi)(\bar{\Gamma}_i - \Psi)^T \quad (2.34)$$

定义类内的矩阵^[24]：

$$S_w = \sum_{i=1}^c \sum_{\Gamma_k \in \Gamma_i} (\Gamma_k - \bar{\Gamma}_i)(\Gamma_k - \bar{\Gamma}_i)^T \quad (2.35)$$

选择使得样本类间离散度与样本类内离散度比值最大的特征特征向量，其公式为^[32]：

$$W_{opt} = \arg \max_w \frac{|W^T S_b W|}{|W^T S_w W|} = [W_1 W_2 \dots W_m], \quad (2.36)$$

由于高维的特征空间计算复杂度比较大，FLDA 利用 PCA 将高维特征空间投影到低维的特征子空间。

$$W_{opt}^T = W_{fld}^T W_{pca}^T \quad (2.37)$$

$$\text{其中: } W_{fld} = \arg \max \frac{|W_{opt}^T S_b W_{pca} W|}{|W_{opt}^T S_w W_{pca} W|}$$

那么人脸图像在 FLDA 计算的特征向量张成的特征子空间的投影 $\omega_k = W_{opt}^T (\Gamma_i - \Psi)$ 。

2.1.3.3 局部保留映射 LPP

PCA和LDA都是保留图像空间的全局结构，然而在实际应用中，有时候局部结构更重要。何晓飞^[25]等人提出了LPP局部特征保留映射算法。LPP试图保留数据的几何特性和局部结构，其目标函数^[25]：

$$\min \sum_{i,j} (\Gamma_i - \Gamma_j)^2 S_{ij} \quad (2.38)$$

假设训练集 $\Gamma_1, \dots, \Gamma_M$ ，LPP的算法流程如下^[25]：

1. 构造邻接表。如果节点 i 和 j 相连，则 i, j 之间有边。

判断相连方法：

- a) 选择阈值 ε 。如果 $|\Gamma_i - \Gamma_j|^2 < \varepsilon$ ，则 i, j 之间有边
- b) 计算 i 与其他节点的值 $|\Gamma_i - \Gamma_1|^2, |\Gamma_i - \Gamma_j|^2, \dots, |\Gamma_i - \Gamma_m|^2$ ，选择前 k 个最小的值对应的节点，则 i 与这些点有边。

2. 计算边的权重

- a) 如果节点 i 和 j 相连，则其边的权重为： $W_{ij} = e^{-\frac{|x_i - x_j|^2}{t}}$ ，否则为 0
- b) 如果节点 i 和 j 相连，则其边的为 $W_{ij} = 1$ ，否则为 0

3. 求映射矩阵：

$$XLX^T w = \lambda XDX^T w \quad (2.39)$$

其中 D 为对角矩阵，其对角元素对应矩阵 S 相应行的元素和， $L=D-S$ 。映射矩阵 w 为式 xx 前 k 个最小的特征值对应特征相连构成的矩阵。

4. 求变换矩阵

在人脸识别过程中，由于训练人脸图像的数目往往比一幅人脸图像的像素少，因此 XDX^T 通常具有奇异性，为了解决这个问题，Laplacianfaces首先用PCA将图像映射到PCA子空间，然后再在该空间采用LPP方法，最后得到的变换矩阵为^[25]：

$$W = W_{pca} W_{lpp} \quad (2.40)$$

其中 $W_{lpp} = [w_1, \dots, w_m]$ ，该方法能够很好地以线性方式保留人内在的几何特征，

映射矩阵 W 为该训练集的特征子空间，其他人脸可以映射到该空间。

2.1.3.4 基于稀疏表示的分类方法 SRC

基于稀疏表示的分类方法假设每一类有足够多的训练样本，那么测试样本可以用同一类的训练样本的线性组合来表示，而这种表示很自然是稀疏的，只会涉及到整个训练数据很小部分，因此这种方法实际上是求解测试样本关于训练样本最稀疏的线性表示方法，通过求解 $L1$ 模的最小化问题来近似原图，最后使得最接近原图的训练集即测试图片所属的类。

算法流程：

假设训练集有 k 类 $A=[A_1, A_2, \dots, A_k] \in R^{m \times n}$, A_i 由其对应的样本集组成，给定测试图片 y ，误差 ε

1. 归一化矩阵 A 和向量 y

2. 求解 L^1 最小化问题：

$$\hat{x}_1 = \arg \min_x \|x\|_1 \text{ subject to } \|Ax - y\|_2 \leq \varepsilon \quad (2.41)$$

3. 计算每类样本集对应的剩余误差： $r_i(y) = \|y - A_i \delta_i(\hat{x}_1)\|_2 \quad i \in (1 \dots k)$

4. 则使得剩余误差最小的类就是 y 所属于的类： $identity(y) = \arg \min_i r_i(y)$

本文除了采用稀疏表示的分类方法外，还采用SVM和最近邻接法这些分类器，在此不一一论述，其理论可以参考文献^[26]。

2.1.3.5 实验结果

本文采用 PCA、LDA 和 LPP 分布提取 YaleB 人脸库的 eigenface、fisherface 和拉普拉斯的人脸特征，并应用最近邻接法 NN、支持向量基 SVM、基于稀疏表示的分类方法 SRC 进行识别。识别结果如表 2-1。从实验结果可以看出，基于稀疏表示的分类方法比其他方法识别高，无论测试集的数据是否完整，它都可以根据训练集的内容重构测试的数据。最近邻近法和 svm 的识别结果都与特征的类型和维数有关。其中 svm 需要根据数据类型选择适当的核函数，并且不断尝试才能够达到高的识别率，最近邻

接法只是根据数据特征向量之间的距离进行比较得到的识别结果，当图像的某部分缺失时，其识别结果就更差。因此本文选择基于稀疏表示的分类方法 SRC。

表 2-1 实验数据

识别算法	特征类型	图像缩小的倍数				
		3	5	6	8	10
NN	Eigenface	0.92	0.93	0.91	0.92	0.90
	Fisherface	0.92	0.92	0.89	0.91	0.88
	Laplacianfaces	0.92	0.92	0.89	0.90	0.90
SRC	Eigenface	1	1	1	1	1
	Fisherface	0.94	0.92	0.91	0.92	0.89
	Laplacianfaces	0.955	0.94	0.92	0.93	0.90
SVM	Eigenface	0.81	0.81	0.81	0.81	0.81
	Fisherface	0.82	0.83	0.83	0.81	0.78
	Laplacianfaces	0.82	0.83	0.82	0.81	0.81

2.2 关键技术

2.2.1 分布式计算（MapReduce）

随着计算量的增加，分布式计算被应用在分布式系统中用于解决计算的效率问题。它将计算任务分配到各台机器上共同完成，从而提高整体运算速度。从概念上我们很容易理解分布式计算的原理，但是在实际操作中它却涉及很多问题，例如如何分发数据、并行计算、处理错误等，这些问题综合起来，使得原本简单的计算，需要大量复杂的代码来处理而变得复杂。因此Google实验室^[27]提出了MapReduce分布式程序设计模型。它从底层上封装了数据分布、容错支持、并发处理、负载均衡等细节，使得用户将精力集中在任务的表述上。MapReduce 将应用在分布式计算过程的大部分操

作归结为Map和Reduce的过程。Map负责任务的分配,Reduce负责合并任务处理的结果。

MapReduce的工作原理如下^[27]:首先应用程序中的MapReduce函数将输入文件分成M块,并将处理每块过程作为一个map任务,合并每块处理的结果作为一个Reduce任务,然后交给主控程序master。master选择空闲的worker分配这些map任务或reduce任务。分配了map任务的worker根据输入数据分析出key/value对,然后传递给用户自定义的map函数。map函数将产生的中间结果key/value对定时刷写到本地的硬盘,这些数据通过分区函数分成R个区。这些中间结果在本地位置信息被发送到主控程序master, master把这些位置信息传送给执行reduce任务的worker。执行reduce任务的worker接收通知后,通过远程连接从map worker的本地硬盘上读取缓冲的中间数据。当reduce的worker读取所有中间数据后,使用中间key对数据排序。reduce worker根据中间key遍历所有排序后的中间数据,并且把key和其对应的中间数据集合传递给用户定义的reduce函数。reduce函数的输出结果。

当所有的 map 任务和 reduce 任务都已经完成了的时, master 激活用户程序。在这时候 MapReduce 返回用户程序的调用点。

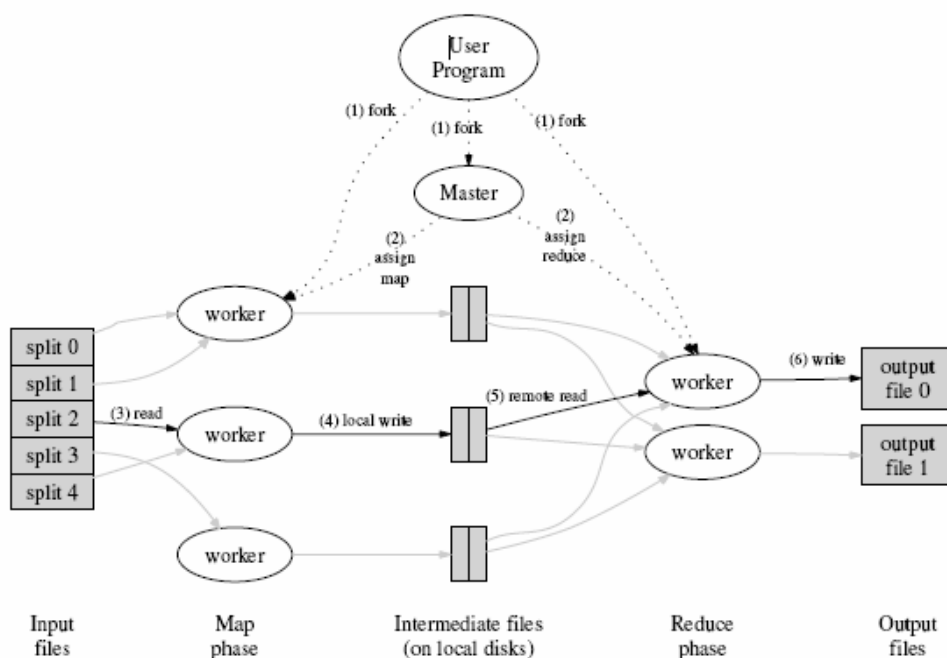


图 2-16 MapReduce执行流程^[4]

2.3 本章小结

本章主要对本系统应用的关键算法和应用进行简单的介绍和分析,为后面对系统的设计和实现提供指导和技术支撑。在设计和实现过程中,将根据实现技术的不同特点和优劣,结合系统应用的特点,选择合适的方案。

第三章 分布式人脸检索系统需求分析

本章作者首先根据系统的具体应用背景提出分布式人脸检索系统的主要设计思想,然后根据该设计思想分析系统在实际应用中应该具备哪些基本功能,接着从技术出发,分析系统在实现过程中需要应用的技术。另外考虑到系统的实用性,作者还分析了系统应该具备的性能需求。最后为了进一步细化功能的需求,作者还对具体功能的关键用例进行分析,并抽象系统的实体概念和关系建立领域模型图。

3.1 系统概述

随着“平安城”在全国各个城市的推广,分布在各地或各部门的视频监控系统正逐渐得到整合、规范、和互联,然而随之产生的庞大视频数据却没有得到更有效的利用,视频监控智能化也只是停留在发挥“电子眼”,视频录像处理等作用。为了充分利用这些视频信息,提高视频监控智能化水平,本文尝试提出分布式人脸检索系统解决方案。该方案旨在通过提取视频关键信息,并根据关键信息建立关键数据库,提供一个跨区域快速查找信息的平台。该平台为日后视频信息的整合、共享、分析、挖掘奠定了一定基础。

本系统的主要设计思想:将人脸识别、人脸检测和跟踪技术融合到视频监控系统,使监控系统具备智能化功能。在监控过程中,系统自动地从动态视频流中捕捉人的面像,从周围背景中抽取出来,保存人脸,存档备查,并对头像进行识别,为监控建立起面像索引库。另外利用图像检索技术自动检索分布在不同的监控点的存档,并返回检索的结果,其结果可以用于跟踪给定人像等,这样实现跨区域的信息检索,降低调查难度,大大提高调查的效率。

3.2 系统需求分析

3.2.1 系统功能需求分析

从系统的主要设计思想出发,对系统进行功能分析:监控过程和查询过程是

两个不同性质的过程。监控过程是指根据分布在监控地点的摄像头实时采集和处理图片的过程，其使用地点是固定的。而查询过程是用户执行查询和显示结果的过程，其使用地点是不固定的。因此按照使用地点和功能的不同，将系统划分为客户端和监控端。由于一般分布在监控端的 PC 机性能不高，如果由监控端完成整个人脸检测、跟踪、识别、检索以及消息回复等过程，那么监控端的运行效率将会非常低。另外考虑到识别算法本身的时间复杂度高，监控端的识别任务量大，系统应该将识别任务的处理独立出来，由一组机器专门完成这些任务。在这一组机器里应该有一台机器专门负责处理任务的分配，其他负责执行识别任务。最后系统应该划分为客户端、监控端和服务端，它们分别实现不同的功能。

1. 监控端本来应该自动从视频流中提取人脸，并将人脸图片传送给服务端，让服务端处理图片。然而考虑到图片比较大，如果监控端与服务端之间采用 udp 传送，图片容易丢失。而如果采用 tcp 进行点对点传送，由于监控端是实时采集图片的，图片传送的频率势必很高，如果每次传送都要建立连接，连接时间比较长。另外在传送过程中，其他监控端也难以发送图片给服务端。综合上述考虑，服务器应该建立识别任务数据库，监控端自动从视频流中检测人脸后，将人脸信息封装为识别任务直接传送给服务端的识别任务数据库。服务端从识别任务数据库中提取识别任务。识别任务的信息应该包括人脸出现的时间、地点以及人脸特征。

2. 如果监控端每次添加识别任务后，通过传递消息命令的方式通知服务端提取和处理识别任务，当服务端重启或消息命令丢失时，服务端就无法提取和处理识别任务。因此服务端应该选择定期从识别任务库中提取和处理识别任务，而无需监控端的通知。另外服务端完成识别任务处理后，应该从识别任务库中删除该任务记录。考虑到识别任务比较繁重，服务端由一组机器完成。其中有一台主机负责定期提取识别任务并分配给其他从机完成。识别任务处理过程包括识别该人脸图像是谁、保存识别结果以及删除识别记录。为了方便检索识别结果，系统应该建立面像索引库，其记录信息包括人的名称、特征、出现的时间以及地点。同时为了便于查询，该服务端的主机还提供信息查询服务，其他客户端可以向该主机发出查询请求。查询的服务应该可以按照时间、地点、姓名以及图片的特征的任意组合进行查询。最后为了方便管理员对面像索引库进行管理，服务端的主机还应该提供管理界面供管理员查看、查询、删除面像库的记录。

3. 客户端应该提供用户界面供用户发出查询请求和查看查询结果。用户应该可以按照时间、地点、姓名以及图片的特征的任意组合进行查询。

4. 由于人脸识别是根据人脸库的信息进行识别的。因此系统应该提供图像采集模块，方便用户采集图像和注册人员信息。图像采集的过程应该符合人性化设计，使用户容易查看采集结果，并进行相应的添加、删除操作。由于系统通过摄像头采集图像，其图片质量并不能满足用户需求，有些用户选择其他设备采集图像，因此图像采集模块还应该支持导入其他图像库的图片。另外采集的图像只需人脸部分，而且所有人脸图像的大小都要符合一定的规格，如果手工对这些图像进行处理，除了工作量非常大外，还容易受主观影响，人脸的处理出现不一致等情况，因此该模块应该提供人脸提取批处理功能。另外在拍摄过程中，图像容易受姿势、光照等影响，该模块还应该提供人脸扶正、光照补偿、平滑处理的功能，从而方便用户对图像进行处理，减少人工干预。最后该模块应该提供人脸信息库管理界面供管理员查看、删除、更新和添加人员注册信息。

3.2.2 系统技术需求分析

由系统的功能分析可知，系统大致需要人脸检测，人脸识别技术、图像检索技术。然而在人脸检测过程中，如果每帧都进行人脸检测和识别，数据量将会非常大。为了减少数据量，提高计算效率，系统在进行图像识别前，需要判断该图像是否与前面已检测的图像类似。如果是，那么就不对该图像进行识别处理。然而人在运动过程中，人脸容易受光照、姿势等影响，传统的 SIFE 特征、颜色直方图特征、纹理特征无法设定阈值确定前面检测的人脸与当前检测的人脸是否类似。因此系统需要增加多目标跟踪技术，其功能能够在检测人脸后建立目标跟踪模型，在下一帧估计该人脸的位置，通过位置信息判断检测的人脸与跟踪的人脸是否是同一个人，从而减小识别的计算量。

目前有很多流行的人脸识别算法，为了增加系统设计的科学性，系统应该具备人脸识别算法分析能力，它能够实现目前流行的几种人脸识别算法并进行对比，为用户选择合适的人脸识别算法。在特征值提取方面，目前主要有 `enginface`，`fisherface`，`Laplacianfaces` 特征脸，因此系统应该提供这几种特征脸的提取技术。

人脸识别算法的实现以人脸库为前提。人脸识别算法的识别率与人脸库的图片质

量有很大关系。如果拍摄的角度、光照、表情和姿势等没有得有合理的调整，那么该人脸库的识别率会很低。另外如果每张图片都要手动修改，不但费时，工作效率低，图片的质量也得不到保障，所以系统还需要提供人脸扶正，人脸提取，光线补偿，平滑处理等图像预处理技术。

另外由于服务端需要多台机器共同完成识别任务，所以系统还需要应用分布式计算技术，完成任务的分配和任务的处理。

最后系统还应该有一套完整的科学的图像采集技术，从而保障图像的质量。

3.2.3 系统性能需求分析

分布式人脸检索系统是有由监控端与服务端共同协作完成整个人脸检测到识别、记录处理的过程，它还支持客户端的信息检索。因此它们三者的性能是系统稳定的基础。以下是设计系统时需要满足的整体性能需求。

1. 稳定性

系统稳定、可靠是应用的基本条件，因此它是系统需要重点考虑的事情，也是系统设计需要达到的首要目标。

2. 并发处理的能力强

中心数据库作为支撑整个分布式信息系统的关键构件，必须具有强大的吞吐能力来支持庞大的信息数据交换，而不能成为整个系统的瓶颈，因此高性能的设计是平台的重要目标之一。

3. 易管理

易管理是衡量系统可用性的重要指标。易管理性可以加快系统建设和使用的效率，降低运营成本，及时发现问题解决故障。

4. 可扩展

软件系统的目标之一是使得系统的各个构件之间耦合性小，可以为其他构件提供公开的外部接口，因此，在系统的设计过程，应该着重考虑可扩展性。

3.3 系统主要用例分析

根据上述需求分析，对系统的模块的主要用例进行分析。

3.3.1 监控端

表 3-1 添加识别任务用例分析

主要参与者	用户、行人
前置条件	启动监控端服务程序
后置条件	监控端与服务器的中心数据库成功建立数据连接，识别任务库更新记录结果
主成功场景	
<ol style="list-style-type: none"> 1. 用户发出建立数据连接请求 2. 系统提供数据连接界面。界面显示系统配置的监控端属性，包括摄像头属于前摄像头还是后摄像头，监控地点以及连接服务器的 IP 地址。 3. 用户点击连接 4. 系统提示数据库连接成功 5. 用户开启摄像头或导入视频文件，系统展示从摄像头或视频源采集图像的界面 6. 系统自动完成对采集图像的人脸检测，并将检测结果封装成识别任务，添加到中心数据库。 7. 系统自动为检测结果建立跟踪对象，并显示跟踪结果。对不同的跟踪目标，系统用不同的颜色框表示。 8. 用户关闭摄像头或视频，系统停止进行图像采集，关闭数据库连接。 9. 用例结束。 	
扩展	
<ol style="list-style-type: none"> a. 用户提出建立数据库连接请求后，系统连接等待超时，系统提示失败 b. 用户在开启摄像头过程中，尝试导入视频文件，反之亦然 <ol style="list-style-type: none"> 1. 系统提示关闭摄像头或者关闭视频文件 2. 用户关闭视频源 3. 用户可以选择其他视频源 c. 在添加识别任务过程中，数据库突然断开连接 <ol style="list-style-type: none"> 1. 系统提示数据库连接断开，系统停止采集图像 	

3.3.2 服务端

表 3-2 识别任务处理用例分析

主要参与者	无
前置条件	启动主机和其他从机。系统已完成训练各地人脸库图像，并将训练结果保存在各个从机
后置条件	主机成功连接中心数据库；识别任务已经处理并从数据库中删除，面像库成功添加识别结果
主成功场景	
<div>1. 主机连接中心数据库，</div> <div>2. 主机定期从识别任务库中提取任务，并将任务分配给其他从机。</div> <div>3. 从机接受识别任务后，启动识别线程，处理识别任务。</div> <div>4. 识别线程处理识别任务后，向面像库添加识别结果，并从识别任务库中删除识别任务。</div> <div>5. 系统重复 2-5 过程，直到识别任务库为空。</div>	
扩展	
<div>a. 用户重启系统或系统出现故障，系统重启后恢复上次处理结果</div> <div>b. 数据库突然断开连接或数据库操作失败，系统提示错误信息，系统停止执行任务</div>	

表 3-3 查询请求处理用例分析

主要参与者	无
前置条件	启动主机，系统建立网络连接
后置条件	系统成功连接中心数据库，系统成功接收查询请求，系统与客户端成功建立 tcp 连接，客户端成功接收查询结果文件。
主成功场景	
<div>1. 系统接收查询请求</div> <div>2. 系统根据查询内容从数据库中检索所有符合查询内容的结果</div> <div>3. 系统将结果写入文件中</div>	

4. 系统与客户端建立 TCP 连接
5. 系统发送查询结果文件给客户端
6. 发送结束
7. 用例结束
扩展
a. 系统连接不成功，等待超时，系统放弃连接，等待下一个请求
b. 系统发送过程中，客户端断开连接，系统提示发送失败，断开连接，等待下一个请求

表 3-4 面像库管理用例分析

主要参与者	用户
前置条件	启动中心处理程序
后置条件	系统成功连接中心数据库，数据库更新用户处理结果
主成功场景	
1 用户提出管理面像数据库请求 2 系统提供面像数据库管理界面 3 用户查询和查看面像数据库 3.1.用户选择查看全部，系统显示所有查询记录结果 3.2 用户选择查询具体符合条件的记录，系统提供查询界面，用户填写查询内容，系统查找所有符合用户请求的记录并显示 4 系统显示查询结果后，系统还提供删除选项，用户可以任意删除一行或多行记录，系统保存用户处理结果。 5 用例结束	
扩展	
a. 数据库连接失败，系统提示连接失败信息	
b. 系统突然重启，系统恢复上一次处理结果	

3.3.3 客户端

表 3-5 查询用例分析

主要参与者	用户
前置条件	无
后置条件	客户端与中心处理模块建立网络连接，中心处理响应请求，客户端接收查询结果。
主成功场景	
<div>1. 用户发出查询请求</div> <div>2. 系统提供用户查询界面，该界面提供选择检索类型按钮，如姓名或学号、图像等，用户输入检索类型对应的内容。该界面还提供时间段和地点选项，用户可以输入相应的内容。</div> <div>3. 用户点击查询确认</div> <div>4. 系统向服务端的主机发出查询请求</div> <div>5. 系统接收服务端的主机发过来的查询结果</div> <div>6. 系统列出所有符合条件（时间地点和姓名）的查询结果。如果结果中含有陌生人的记录，系统显式显示出来。</div> <div>7. 系统提供查询结果保存选项。如果用户希望保存查询结果，用户可以根据系统的提示保存文件路径。</div> <div>8. 用例结束。</div>	
扩展	
<div>a. 用户没有填写查询内容，却点击查询确认，系统默认查询当天的所有地点所有人的出入记录。</div> <div>b. 用户填写查询内容后，点击取消查询，系统提示是否确定取消。如果确认取消，退出查询界面；如果不是，继续填写查询内容。</div> <div>c. 用户只填写查询选项的一条或者多项内容，系统自动组合查询结果。</div> <div>d. 系统发出查询结果，中心处理模块没有响应请求，系统等待超时后，向用户提示无法连接服务器信息。</div>	

3.3.4 图像采集

表 3-6 人员注册用例分析

主要参与者	用户
前置条件	无
后置条件	与中心数据库连接成功，人脸信息数据库更新处理结果
主成功场景	
<ol style="list-style-type: none"> 1. 用户打开摄像头，系统展示当前采集的场景 2. 用户选择新建注册表，系统提供注册表，用户在注册表填写注册信息并保存，系统自动保存该人的信息到人脸数据库 3. 系统为该人脸新建一个目录 4. 用户点击保存人脸图片，系统从摄像头采集图片并保存到新建的目录，同时展示当前用户拍摄的图像内容 5. 用户重复该过程直到结束 6. 用户关闭摄像头 7. 用例结束 	
扩展	
<ol style="list-style-type: none"> a. 用户不满意当前拍的照片，选择删除某张图片，系统处理结果并保存 b. 用户填息注册表后，取消添加到人脸数据库，系统提示是否确定取消，并进行相应处理 c. 系统添加注册信息失败，系统提示失败信息，并终止下一步操作 	

表 3-7 人脸信息库管理用例分析

主要参与者	用户
前置条件	无
后置条件	与中心数据库连接成功，人脸信息数据库更新处理结果
主成功场景	
基本流程： <ol style="list-style-type: none"> 1. 用户提出人脸信息库管理请求 	

<div>2. 系统提供人脸信息库管理界面</div> <div>分流：</div> <div><div>➤ 用户选择添加注册员信息</div><div><div>▪ 系统提供注册表格，用户填写信息并保存</div><div>▪ 系统保存结果</div></div><div>➤ 用户选择查询查看注册员信息</div><div><div>▪ 系统提供查询界面，用户选择查询条件和填写查询内容，并确定</div><div>▪ 系统搜索所有符合用户查询条件的内容</div><div>▪ 系统显示查询结果</div></div><div>➤ 用户选择更新某人信息</div><div><div>▪ 系统显示该人信息表格</div><div>▪ 用户修改具体某项内容并保存</div><div>▪ 系统保存结果</div></div><div>➤ 管理选择删除人脸库</div><div><div>▪ 用户选择删除一行或多行查询结果</div><div>▪ 系统删除结果并保存</div></div></div>
扩展
<div>a. 数据库连接失败，系统提示连接失败信息</div> <div>b. 系统突然重启，系统恢复上一次处理结果</div> <div>c. 用户添加信息后忘记保存结果，系统提示是否保存结果</div>

3.3.5 人脸识别算法分析

表 3-8 人脸识别算法分析模块用例分析

主要参与者	用户
前置条件	存在人脸库
后置条件	无
主成功场景	

1. 用户设置基本信息，包括测试图像片库和训练图片库来源，人脸的大小，特征的提取算法
2. 用户选择具体的某种算法
3. 系统用该算法对人脸图片进行训练
4. 用户选择测试
5. 系统根据训练图片进行测试
6. 系统显示测试结果
扩展
无

3.4 系统领域模型分析

图 3-1 为系统的领域模型，它以可视化的形式描述系统中的各个实体及其之间的关系。监控端抽象为实体 Monitor，它产生识别任务 TaskData，并通过识别任务管理器 TaskDataManger 保存识别任务。识别任务管理器管理识别任务库 TaskDataBase。客户端抽象为实体 Client，它通过 Socket 向主机 Master 发出查询请求，主机 Masker 通过面像管理器 FaceDataManager 查询所有符合条件的内容。Master 通过识别任务管理器 TaskDataManger 提取识别任务，并将任务分配给从机 Slaver。从机 Slaver 将识别任务封装为每个 job，并通过 Map 接口处理每个任务 job。

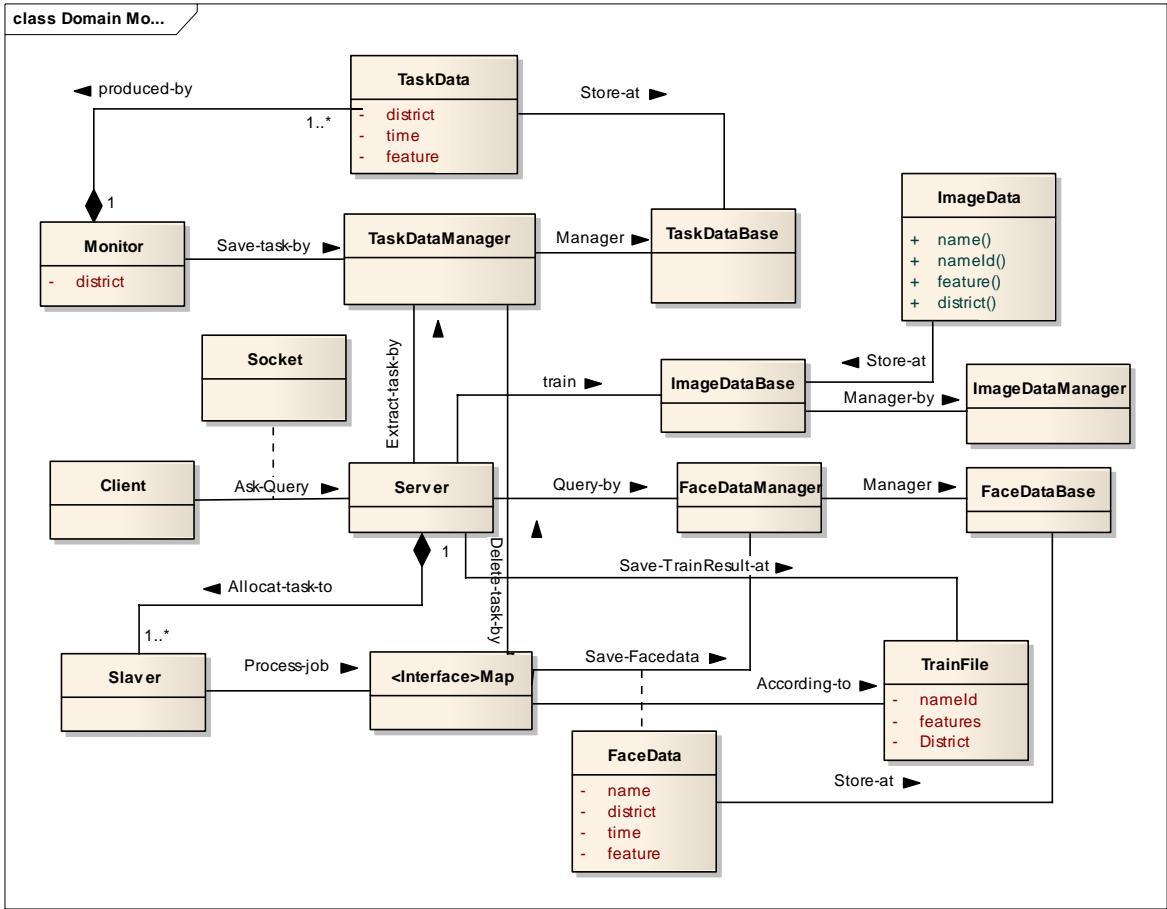


图 3-1 分布式人脸检索系统领域模型

3.5 本章小结

本章主要从系统概述、系统需求和用例分析、领域模型四方面出发对分布式人脸检索系统的需求进行设计。系统概述描述了项目背景以及系统设计的主要思想。整个系统的架构和功能设计围绕该思想进行。系统需求从功能、技术、性能三方面进行分析。功能需求分析从系统实现的目标以及用户使用的角度上总结系统应该具备基本功能。系统技术需求分析描述了系统在实现过程中需要应用的技术以及原因。性能需求分析描述系统应该具备的性能要求，为系统的设计和实现提供了现实依据。

第四章 分布式人脸检索系统总体设计

本章作者结合分布式人脸检索系统的总体架构以及特定的通信模式，描述了系统各个模块的详细划分情况。另外由于系统涉及的计算量大，作者采用 MapReduce 分布式程序设计模式实现任务的并行处理。在客户端与服务端的通信方面，作者采用点对点的通信模型，并结合 TCP 和 UDP 的通信协议实现客户端与服务端之间的通信。

4.1 系统架构及原理

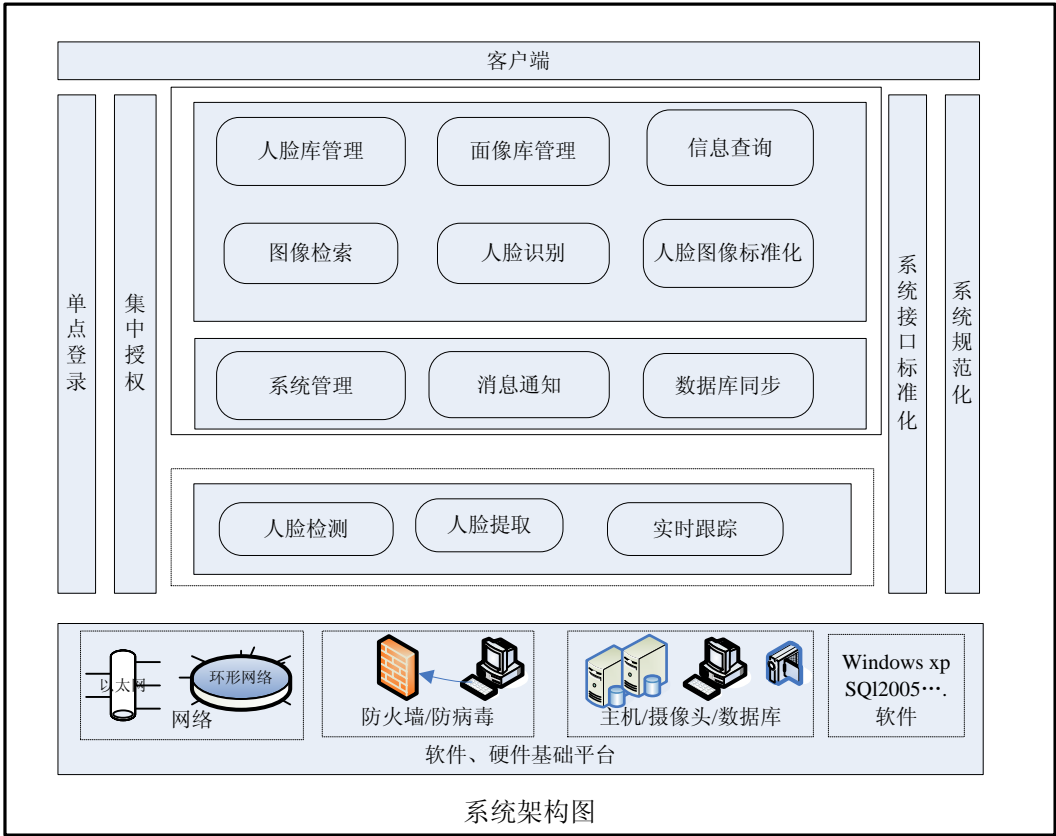


图 4-1 系统架构图

分布式人脸检索系统的总体结构可以划分为三部分：软硬件基础平台、通用组件以及运行于平台上面的各应用系统。

- 1. 软件/硬件平台：硬件平台提供了承载系统的服务器以及连接各客户端和监控端的网络环境。软件平台提供了操作系统、数据库、负载均衡等系统软件，提供了一个应用程序运行的基础软件环境。

2. 通用组件：将一些通用基础的功能封装成一系列功能相对独立的组件，包括数据库同步、网络通信（消息通知）等通用功能，并以组件的形式实现。
3. 系统应用：运行于软硬件平台之上，利用通用组件设施提供的通用功能，实现的应用系统包括服务端子系统，监控点子系统，客户端子系统。服务端子系统包括人脸检索，信息查询，人脸库的管理等；监控端子系统包括人脸检测，信息提取，实时跟踪等；客户端子系统包括发送请求，结果显示等。

4.2 分布式系统设计

分布式人脸检索系统其分布式主要体现在两方面：一方面是指使用多台机器共同完成所有的识别任务；另一方面是指能够查询跨区域信息。本文使用 MapReduce 分布式程序设计模型实现任务的并行处理，使用点对点通信模式实现信息的查询。

（1）任务的并行处理设计

根据 MapReduce 模型，要并行处理识别任务，首先要将识别任务进行分割，然后将分割信息通过输入传递给并行的 Map 操作，各个 Map 操作对各自的识别任务进行独立处理，由于识别结果不需要返回给主机，所以系统不需要 Reduce 操作，Map 在完成识别后，直接将识别处理结果保存在面像数据库中，其过程大致如下：

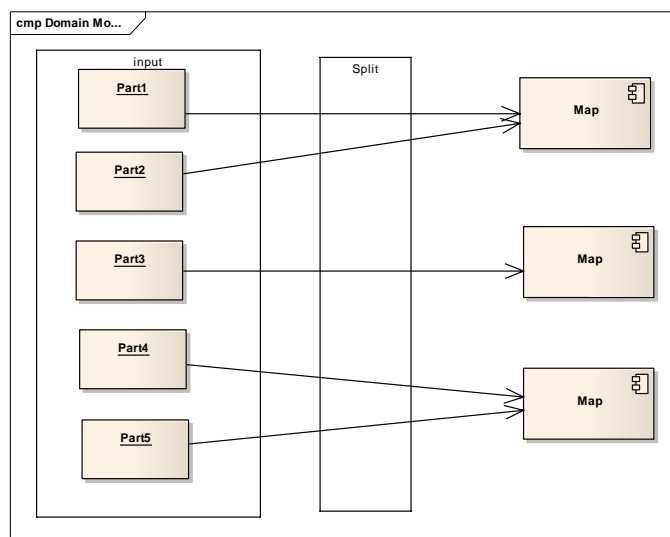


图 4-2 Map 工作流程

1. 定义识别任务文件

由上一章的用例分析可知，识别任务记录主要保存在识别任务数据库中，然而 MapReduce 的执行是以文件为单位的。因此主机在提取识别任务记录时需要将任务记

录保存在文件里。文件格式：

表 4-1 文件格式

时间	地点	人脸特征
----	----	------

2. 定义 Map 函数的 Key/Value 对

将识别文件的每一行作为 **Key**，将每一行的内容作为 **value**。

3. Map 函数设计

Map 操作对输入的 Value 参数提取时间、地点、人脸特征信息，并利用人脸识别算法对人脸特征进行识别，最后将识别结果保存在面像库中。

(2) 通信模式的设计

服务端采用点对点的通信模式来处理客户端的请求。客户端向服务端发出查询请求，服务端接受查询消息后，将查询结果发送给客户端。如图所示，客户端与服务端直接使用 Socket 连接。对于每一个连接，服务端分配一条线程使之接收并处理该连接中发送的消息。服务端与客户端之间通过该 Socket 维持点对点的通信。

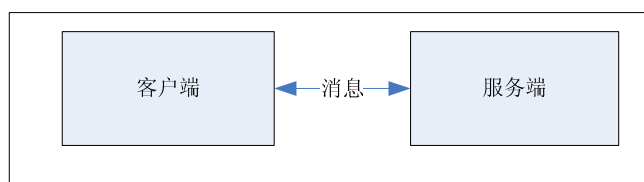


图 4-3 点对点模式

1. 通信协议的设计

客户端与服务端主要采用 TCP/IP 协议实现点对点的通信模式。TCP/IP 协议是一组包括 TCP 协议、IP 协议、UDP 协议、ICMP 协议和其他一些协议的协议组。

➤ TCP 协议

TCP 是一种可靠的面向连接的传送服务，使用面向连接的流 Socket 通信来实现。套接字是通信的基础，流套接字是基于 TCP 协议的，面向连接，需要三次握手建立虚拟的连接，并利用收到应答等方式实现其可靠性。

➤ UDP 协议

UDP 是基于数据报或分组的协议，它可以直接发送和接收数据报文而不需要建立过程，因此是不可靠的。它可以使用面向非连接的数据报式套接字 Socket 通信来实现。

由于服务端查询的结果比较大，而且需要稳定可靠地发布给客户端服务程序，因此客户端和服务端采用 TCP 传送查询结果。另外如果每次启动时都要重新建立 socket 连接，比较耗时间。系统在启动时应该建立 socket 连接，绑定端口。另外因为 TCP 每次连接都需要三次握手过程，比较麻烦，因此一些数据量比较少的消息如控制消息可以通过 UDP 来传递。下面是服务端与客户端之间传递消息和传送文件的过程。

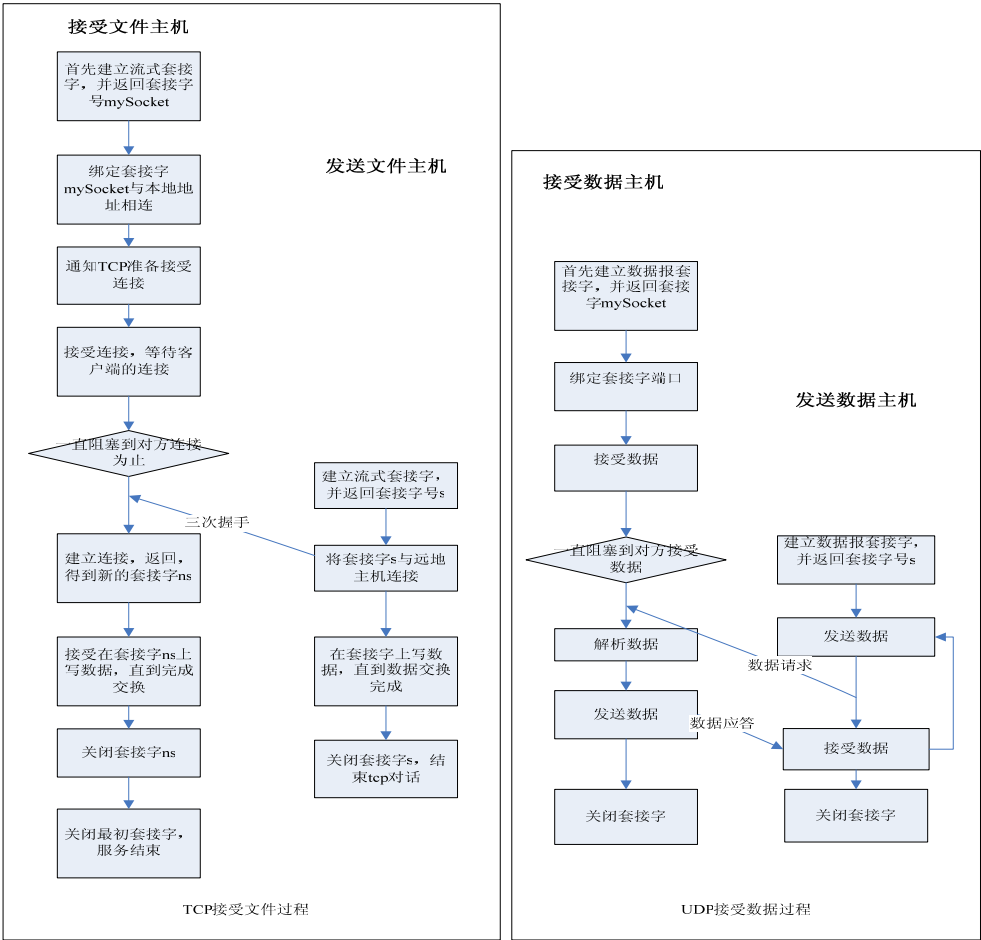


图 4-4 服务端/客户端发送/接受文件 图 4-5 客户端/服务端发送/接受数据

4.3 系统功能设计

4.3.1 监控端功能设计

监控端由界面层、业务层以及数据访问层组成。

界面层主要提供可视化的界面供用户进行相关的操作，其中包括连接数据库、开启/关闭摄像头、导入视频文件、播放/关闭视频、继续播放/停止视频、设置视频属性和系统配置。它们之间的逻辑关系如下：系统启动时，自动加载配置文件，用户点击连接数据库，系统连接成功后，用户开启摄像头或者导入视频文件，然后系统自动完成从视频流中检测人脸、进行跟踪和添加识别任务的过程。在这一过程中用户可以点击系统配置，系统显示配置文件的信息包括监控端的位置信息和连接的数据库 IP 地址。监控端的位置信息会作为识别任务的其中一个标志。用户可以修改配置文件的信息，但修改后需要重新建立与数据库的连接。

业务层提供系统的所有业务操作，其中包括人脸检测、人脸跟踪、数据库的管理。为方便日后算法的扩展，将不同的人脸检测算法检测人脸的操作抽象为一个统一接口，将不同的跟踪算法添加跟踪对象、估计跟踪对象的位置抽象为统一接口，其他算法实现该接口以提供系统调用。数据库的管理通过封装数据访问的底层操作，实现数据库的连接和添加记录的操作。

数据访问层提供数据库访问的底层操作。本文主要采用 MYSQL 的静态链接库。该库提供一向接口：初始化 MYSQL 对象 `mysql_init`，根据该对象连接数据库 `mysql_real_connect` 并执行 SQL 语句的操作 `mysql_real_query`。

4.3.2 服务端功能设计

服务端由界面层，网络层，业务层，数据访问层以及分布式系统五部分组成。

界面层主要提供可视化界面供管理员管理面像数据库，其操作包括面像数据库记录的查询、查看以及删除。由于人脸识别需要根据训练文件才能进行识别，因此界面还提供了训练的操作，该操作能自动从人脸库中获取所有人的面像信息进行训练，并将结果保存在系统的临时目录里。当系统启动时，系统自动检索临时目录是否存在训练文件。如果不存在，系统提示识别任务处理失败。如果存在，系统会定期从识别任务库中提取任务，并保存为识别文件以供分布式系统处理。

网络层主要实现客户端与服务端点对点的通信机制。它提供两个接口。其中一个接口使用面向流的 SOCKET 和面向数据报的 SOCKET 实现消息、文件的接受和传递，另外一个接口封装上述接口的操作以供系统调用。

业务层主要提供管理面像库的相关业务操作，它通过封装数据访问层的底层操作来实现面像库数据的添加、删除、查询。

服务端的数据访问层的功能和结构与监控端的相同，另外分布式系统设计可以参考 4.2 小节，在此省略不提。

4.3.3 客户端功能设计

服务端由界面层，网络层两部分组成。网络层的结构与服务端的结构相同。界面层主要提供查询界面、结果显示界面以及查询操作。查询界面为用户提供查询的类型（姓名/身份 ID/图片）、时间（这个是必填的）以及地点。当用户点击查询时，系统会提供查询界面供用户输入查询内容。当用户点击查询完成时，系统会封装查询内容，并通过网络层向服务端发送查询命令和查询内容，服务端接受结果后向网络层发送查询结果。

4.3.4 图像采集功能设计

图像采集模块主要由人员注册、图像预处理以及人脸库管理三大部分组成。人员注册主要采集负责用户面像和信息，并保存到数据库中。图像预处理主要提供人脸扶正、人脸提取、光线补偿、平滑处理的技术。在视频采集过程中，采集的图片很容易受到光线的影响，需要对图像进行光线补偿，另外人的姿势、表情或动作很容易导致采集的图片人脸弯曲等情况，所以需要对面脸图像标准化。标准化过程大概分为两个步骤：采用基于灰度模型的人脸检测分割人脸区域，采用垂直投影的方法定位人眼，通过人眼对人脸进行扶正。基于灰度模型的人脸检测虽然受光照影响比较大，但它可以帮助选择光照比较均匀的图片，所以本文在标准化过程主要采用这种方法。人脸库管理主要提供管理界面以及查看、删除和更新人脸库信息的操作。

4.3.5 人脸识别算法分析功能设计

人脸识别模块主要负责对几种技术进行分析。它由信息配置以及算法分析两部分组成。信息配置主要完成选择哪种特征值和数据库、图像的大小等基本信息设置。算法分析主要集成几种主流的算法，并提供界面显示算法运行结果。考虑到目前存在

多种人脸识别算法，本文将流行的几种人脸识别算法做了对比，选择效果较好的识别算法作为人脸识别模块的一部分。另外为了便于日后算法的扩展和更改，人脸识别算法应该设计成通用的模块。

4.4 本章总结

本章对分布式人脸检索系统的总体设计过程进行了详细地叙述。通过对消息中间件的总体结构、分布式系统设计、模块划分进行介绍，本章描述了系统总体设计的过程。在系统总体设计中，本章根据 MapReduce 思想提出了分布式系统的设计方案，并将点对点的通信模式应用于客户端与服务端的通信上。

第五章 应用实例-考勤系统

考勤系统是在分布式人脸检索系统设计方案基础上设计和实现的应用系统。其系统结构和原理大致与分布式人脸检索系统相同,可以看成是分布式人脸检索系统的一个应用实例。通过这个应用实例的实现,可以发现分布式人脸检索系统在实现过程中存在的各种问题,从而验证方案的可行性。由于考勤系统需要应用在特定的场合,因此考勤系统部分功能的设计有别与分布式人脸检索系统,但是不影响分布式人脸检索系统的可行性的分析。下面将会介绍考勤系统的系统设计。

5.1 考勤系统功能分析

一个完整的考勤系统应该能够记录特定区域内所有人的出入情况,根据这些记录应该能够识别出哪些人属于这个小区,哪些人不属于。而分布式人脸检索系统只是记录曾经出现在某个地区的人的情况,并根据统一的人脸库进行识别,也就是这个人脸库并不一定针对某个小区,它可能是全市、全省甚至全国的人,同时它不反应其进出情况。因此考勤系统需要在分布式人脸检索系统的监控端的配置上增加监控端出入的属性,识别任务也需要添加状态信息,表明这个监控端采集的人脸是出的情况还是入的情况。另外考勤系统不提供图像检索的功能,因此客户端也不具备该项的查询。最后在服务端的设计上,由于作者实现该系统的时间有限,作者只能采用多线程技术完成识别任务,提高识别任务处理的速度。虽然其处理的速度比 MapReduce 低,但不影响对分布式人脸检索系统的方案可行性的分析,分布式人脸检索系统的设计的关键是算法的准确性,这些算法在考勤系统中都有应用。在图像采集方面,系统需要为每个特定区域的人采集人脸,在识别过程中,根据人出现的区域选择适当的训练文件。在硬件配置方面,考勤系统必须严格要求有两个摄像头,能够同时监控人进和入的情况。

5.2 考勤系统软件结构

根据上述功能分析,将考勤系统的功能模块划分为监控端、服务端和客户端、图像采集模块以及人脸识别算法分析模块。

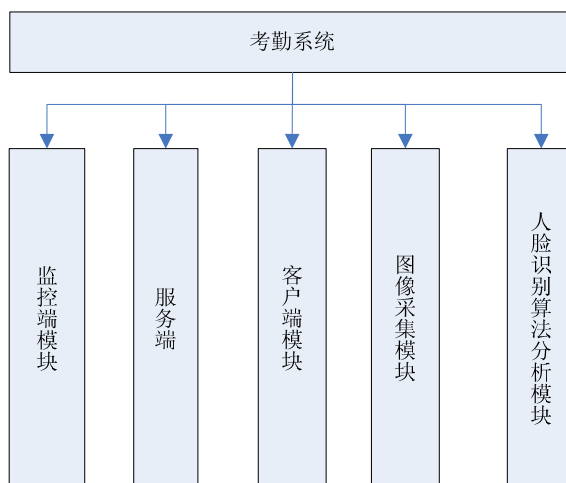


图 5-1 系统软件结构图

1. 监控端：负责从监控摄像头或者视频文件捕捉人脸,添加人脸识别任务，同时对该人脸进行实时跟踪。
2. 服务端：负责提取识别任务并通过多线程处理识别任务。它还通过 TCP/IP 方式与客户端建立连接，响应客户端查询请求，发送查询结果。最后它还提供管理界面供管理员查看、删除、查询面像库记录。
3. 客户端：提供用户界面供用户远程查询查看面像库存储记录。另外它通过采用 TCP/IP 方式向中心处理模块发送查询请求。
4. 图像采集模块：负责登记人员信息，并采集该人的面像。另外它提供管理界面供管理员添加、删除、更新和查询查看人脸信息库。最后它还提供图像预处理功能包括人脸扶正，人脸提取，光线补偿和平滑处理。
5. 人脸识别算法分析模块：负责分析目前几个主流算法的性能情况

5.3 考勤系统硬件结构

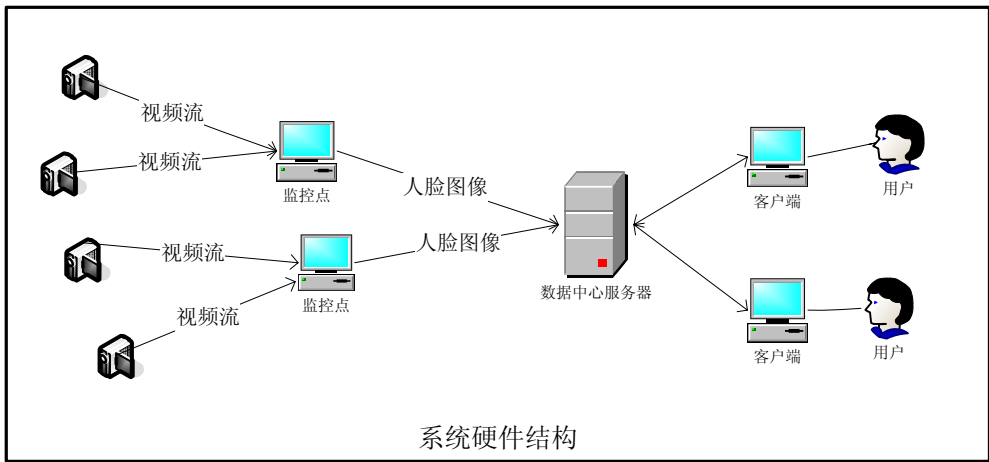


图 5-2 系统硬件结构图

由图 5-2 可知，系统基本上由三部主机，2 个摄像头构成。这三部分主机分布部署在监控点，服务中心以及客户端。每个监控点连接 2 个摄像头，一个摄像头监控人进入的情况，一个摄像头监控人出去的情况。另外服务中心部署的服务器要求有较高的性能、较强的扩容能力。

5.4 考勤系统用例分析

由考勤系统的分析可知，分布式人脸检索系统的图像采集和人脸识别算法分析模块与考勤系统的功能相同，因此下面只是针对不同的功能模块的用例进行分析。

5.4.1 监控端

表 5-1 添加识别任务用例分析

主要参与者	用户、行人
前置条件	启动监控端服务程序
后置条件	监控端与服务器的中心数据库成功建立数据连接，识别任务库更新记录结果
主成功场景	

10. 用户发出建立数据连接请求
11. 系统提供数据连接界面。界面显示系统配置的监控端属性，包括摄像头属于前摄像头还是后摄像头，监控地点以及连接服务器的 IP 地址。
12. 用户点击连接
13. 系统提示数据库连接成功
14. 用户开启摄像头或导入视频文件，系统展示从摄像头或视频源采集图像的界面
15. 系统自动完成对采集图像的人脸检测，并将检测结果封装成识别任务，添加到中心数据库。
16. 系统自动为检测结果建立跟踪对象，并显示跟踪结果。对不同的跟踪目标，系统用不同的颜色框表示。
17. 用户关闭摄像头或视频，系统停止进行图像采集，关闭数据库连接。
18. 用例结束。
扩展
d. 用户提出建立数据库连接请求后，系统连接等待超时，系统提示失败
e. 用户在开启摄像头过程中，尝试导入视频文件，反之亦然
1. 系统提示关闭摄像头或者关闭视频文件
2. 用户关闭视频源
3. 用户可以选择其他视频源
f. 在添加识别任务过程中，数据库突然断开连接
1. 系统提示数据库连接断开，系统停止采集图像

5.4.2 服务端

表 5-2 识别任务处理用例分析

主要参与者	无
前置条件	启动服务端
后置条件	系统成功连接中心数据库；系统已完成训练各地人脸库图像，并

	保持训练结果；识别任务库删除已处理的任务，面像库成功添加记录结果
主成功场景	
6. 系统启动多个人脸识别线程 7. 系统连接中心数据库 8. 系统定期从识别任务库中提取任务，系统根据识别任务所在的地点，选择相应的训练结果，进行识别处理。 9. 识别线程处理识别任务后，向面像库添加识别结果，并从识别任务库中删除识别任务。 10. 系统重复上述过程，直到识别任务库为空。	
扩展	
c. 用户重启系统或系统出现故障，系统重启后恢复上次处理结果 d. 数据库突然断开连接或数据库操作失败，系统提示错误信息，系统停止执行任务	

表 5-3 查询请求处理用例分析

主要参与者	无
前置条件	启动服务端，系统建立网络连接
后置条件	系统成功连接中心数据库，系统成功接收查询请求，系统与客户端成功建立 tcp 连接，客户端成功接收查询结果文件。
主成功场景	
8. 系统接收查询请求 9. 系统根据查询内容从数据库中检索所有符合查询内容的结果 10. 系统将结果写入文件中 11. 系统与客户端建立 TCP 连接 12. 系统发送查询结果文件给客户端 13. 发送结束 14. 用例结束	
扩展	

c.	系统连接不成功，等待超时，系统放弃连接，等待下一个请求
d.	系统发送过程中，客户端断开连接，系统提示发送失败，断开连接，等待下一个请求

表 5-4 面像库管理用例分析

主要参与者	用户
前置条件	启动服务端
后置条件	系统成功连接中心数据库，数据库更新用户处理结果
主成功场景	
6 用户提出管理面像数据库请求	
7 系统提供面像数据库管理界面	
8 用户查询和查看面像数据库	
3.1.用户选择查看全部，系统显示所有查询记录结果	
3.2 用户选择查询具体符合条件的记录，系统提供查询界面，用户填写查询内容，系统查找所有符合用户请求的记录并显示	
9 系统显示查询结果后，系统还提供删除选项，用户可以任意删除一行或多行记录，系统保存用户处理结果。	
10 用例结束	
扩展	
c. 数据库连接失败，系统提示连接失败信息	
d. 系统突然重启，系统恢复上一次处理结果	

5.4.3 客户端

表 5-5 查询用例分析

主要参与者	用户
前置条件	无
后置条件	客户端与服务端建立网络连接，服务端响应请求，客户端接收查

	询结果。
主成功场景	
<p>9. 用户发出查询请求</p> <p>10. 系统提供用户查询界面，该界面提供选择检索类型按钮，如姓名或学号等，用户输入检索类型对应的内容。该界面还提供时间段和地点选项，用户可以输入相应的内容。</p> <p>11. 用户点击查询确认</p> <p>12. 系统向服务端发出查询请求</p> <p>13. 系统接收服务端发过来的查询结果</p> <p>14. 系统列出所有符合条件（时间地点和姓名）的查询结果。如果结果中含有陌生人的记录，系统显式显示出来。</p> <p>15. 系统提供查询结果保存选项。如果用户希望保存查询结果，用户可以根据系统的提示保存文件路径。</p> <p>16. 用例结束。</p>	
扩展	
<p>e. 用户没有填写查询内容，却点击查询确认，系统默认查询当天的所有地点所有人的出入记录。</p> <p>f. 用户填写查询内容后，点击取消查询，系统提示是否确定取消。如果确认取消，退出查询界面；如果不是，继续填写查询内容。</p> <p>g. 用户只填写查询选项的一条或者多项内容，系统自动组合查询结果。</p> <p>h. 系统发出查询结果，服务端没有响应请求，系统等待超时时，向用户提示无法连接服务器信息。</p>	

5.5 考勤系统领域模型分析

图 5-5 为系统的领域模型，它以可视化的形式描述系统中的各个实体及其之间的关系。监控端抽象为实体 **Monitor**，它产生识别任务 **TaskData**，并通过识别任务管理器 **TaskDataManger** 保存识别任务。识别任务管理器管理识别任务库 **TaskDataBase**。

客户端抽象为实体 Client，它通过 Socket 向主机 Master 发出查询请求，主机 Master 通过面像管理器 FaceDataManager 查询所有符合条件的内容。Server 通过识别任务管理器 TaskDataManger 提取识别任务，并将任务分配给 worker，并通过 worker 处理每个任务 job。线程池 ThreadPool 存放很多线程，ThreadPoolImp 从线程池中分配线程给 worker 对象，完成 worker 的任务。主机 Master 根据人脸信息库 ImageDataBase 对保存的人脸进行训练，产生训练文件 TrainFile。Worker 根据训练文件完成识别任务。

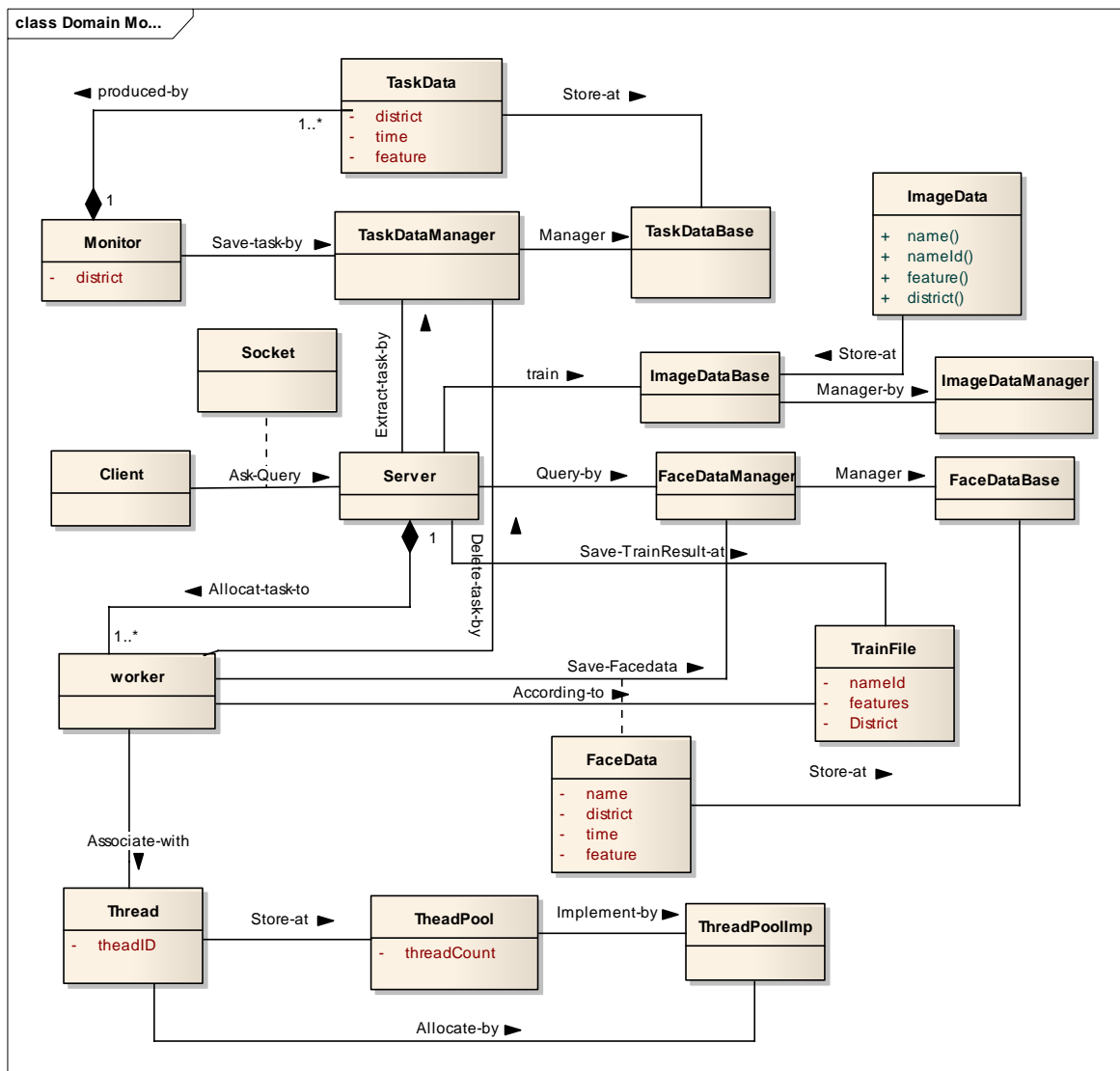


图 5-5 考勤系统领域模型

5.6 考勤系统数据流分析

本节将使用数据流图对考勤系统主要组成模块进行分析,以便更好地捕获用户需求。由于图像采集模块和人脸识别算法分析模块只是系统提供的额外辅助工具,而系

统关键部分在于如何获取人脸，人脸获取后如何处理、识别任务如何设置和处理，如何记录识别结果、如何处理用户查询请求等，而這些部分的实现主要依赖监控端、中心处理模块以及客户端，所以下面主要对监控端、中心处理模块以及客户端的数据流进行分析，最后得出整个系统的数据流图。数据流图包含四种基本的组成成分：数据的源点或终点、数据处理、数据存储、数据流。为了确定考勤系统的数据流，首先从考勤系统的需求出发描述数据流的四种基本成分，然后根据这四种成分画出系统的基本系统流图，最后得出整个系统数据流图。

根据考勤系统的需求描述，可以将考勤系统处理的消息数据抽象出四个方向的数据流：

- 1. 监控端从摄像头采集图像，并提取其人脸
- 2. 监控端提取人脸后，设置识别任务，并通过网络连接把识别任务保存在中心服务器的识别任务数据库中。
- 3. 服务端从数据库中提取识别任务，并启动多线程进行识别处理，最后把识别结果保存在面像库中。
- 4. 客户端发出查询请求，通过 TCP/IP 发送给中心处理模块，中心处理模块将查询的结果返回给客户端。

(1) 首先根据不同的流程确定该数据流的四种成分。

- 1. 数据从摄像头/视频源流到监控端

表 5-6 从摄像头/视频文件到监控端的数据流四大成分

数据源点： 摄像头 数据终点： 监控端	
数据流	视频流，图像，显示消息
数据处理	人脸提取，
数据存储	人脸图片

- 2. 数据从监控端流到中心服务器数据库

表 5-7 从监控端到中心服务器数据库数据流四大成分

数据源点： 监控端 数据终点： 中心服务器数据库	
数据流	人脸图像、人脸特征、任务消息
数据处理	提取人脸特征、设置识别任务
数据存储	记录识别任务

3. 数据从中心处理模块流到中心服务器数据库

表 5-8 从服务端到中心服务器数据库数据流四大成分

数据源点： 服务端 数据终点： 中心服务器数据库	
数据流	任务消息
数据处理	提取识别任务、识别处理
数据存储	记录识别结果

4. 数据从客户端流到客户端

表 5-9 从客户端到客户端数据流四大成分

数据源点： 客户端 数据终点： 客户端	
数据流	查询请求、查询结果
数据处理	请求处理、发送请求结果
数据存储	查询的记录

(2) 通过四大成分确定基本数据流图

根据前面的描述，可以将四种情况下的数据流图描述如图 5-6、图 5-7、图 5-8 和图 5-9。

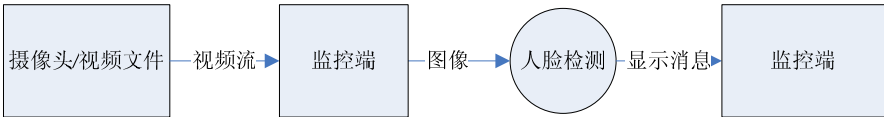


图 5-6 从摄像头/视频文件到监控端的数据流图

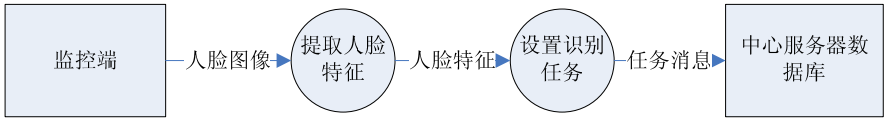


图 5-7 从监控端流到中心服务器数据库的数据流图

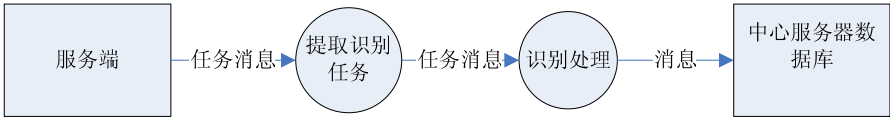


图 5-8 从服务端流到中心服务器数据库的数据流图

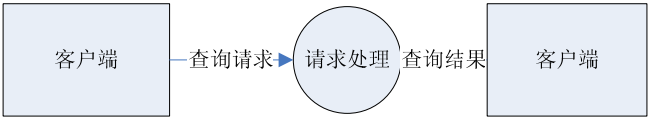


图 5-9 从客户端流到客户端的顶层数据流图

(3) 最后得出整个系统的数据流图

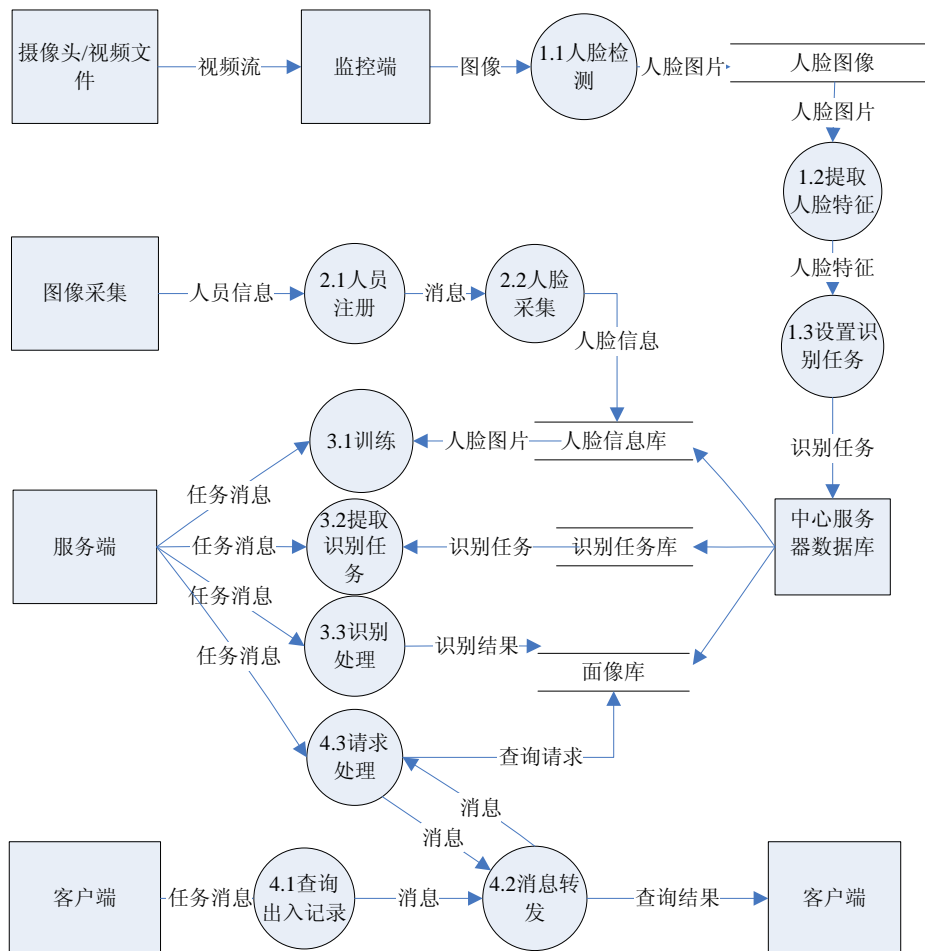


图 5-10 系统的数据流图

5.7 本章小结

本章主要介绍了基于分布式人脸检索系统设计方案的具体应用系统-考勤系统。作者根据实际应用场景修改分布式人脸检索系统的功能使其符合实际需求。另外作者还根据系统需求划分系统的软件和硬件组成结构,接着作者为有别于分布式人脸检索系统进行用例分析,这些用例细化系统功能,例如系统提示、故障修复等。最后通过对系统的领域建模和数据图的分析系统外部和内部数据的流入流出情况,为考勤系统的接口设计提供依据

第六章 考勤系统设计

由于考勤系统与分布式人脸检索系统的系统架构大致相同，所以在这一章，作者省略系统架构和功能的设计。作者主要根据用例分析，实现用例的设计。分布式系统设计里着重介绍线程池的设计。

6.1 线程池的设计

为了提高识别任务处理的速度，本文采用线程池技术实现识别任务的处理。线程池的数据流图如图所示：创建多个线程对象，并将这些对象放在线程队列中。当用户请求时，系统从线程队列中选择空闲的线程完成该任务，并设置线程状态为忙。当任务完成后，系统设置该线程为空闲。系统定时检查线程池的状态，当线程池有 80%的线程都处于忙碌时，系统认为线程池快要满了，系统根据用户设置的最大线程数，增加线程的对象，扩大线程池。当线程池只有 20%的线程处于忙碌时，系统认为线程池过于空闲，系统根据用户设置的最小线程数，从线程队列中删除部分空闲的线程，缩小线程池。

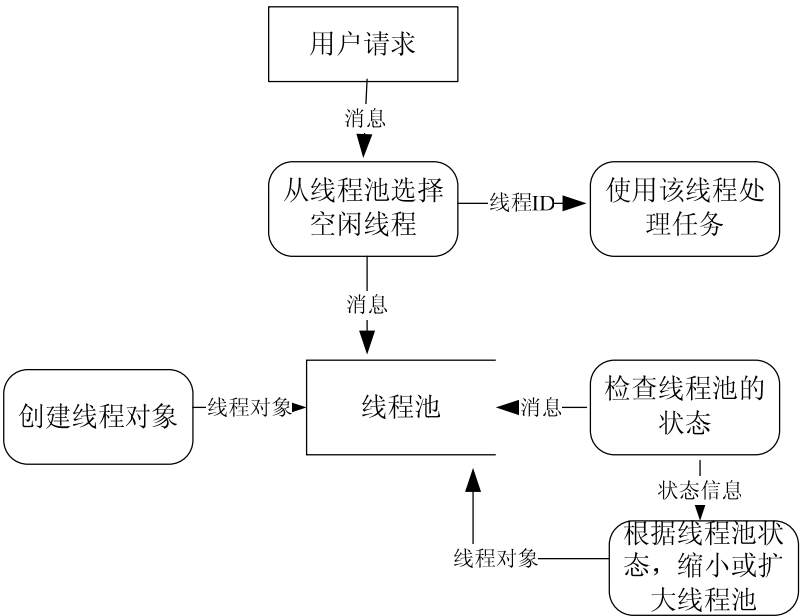


图 6-1 线程池数据流图

根据线程池的数据流图，对线程池的接口进行如下设计：

1. 抽象所有任务对象统一接口 `IJob`, 每个任务都必须实现该接口, 以供线程执行调度任务.
2. 抽象所有工作线程为统一接口 `IWork`, 每个工作线程都必须实现该接口。
3. `ThreadPool` 提供所有类型的线程池的公共操作。包括创建线程 `Start`, 处理任务 `ProcessJob`, 撤销线程 `Stop`
4. `ThreadPoolImp` 完成特定线程池模型的操作, 包括线程池大小的控制, 分配线程等。它需要一个线程队列管理线程, 还需要一个容器存放线程与其状态的映射。

6.2 用例实现

6.2.1 监控端模块

(1) 添加识别任务用例实现

1. 用户在监控端界面点击连接数据库 `OnConnect`
2. 监控端连接数据库 `Connect`, 并连接成功
3. 用户打开摄像头 `OnOpenVideo`
4. 监控端采集图像 `CallBackProcess`, 并返回采集的图像
5. 监控端对图像进行人脸检测 `Detectface`, 并返回人脸图片
6. 监控端根据人脸图像创建目标跟踪实例 `CreateTrackObj`, 在下一帧进行跟踪
7. 监控端根据人脸图像创建识别任务实例 `CreateTraskObj`
8. 监控端将该识别任务实例添加到数据库 `AddTaskData`

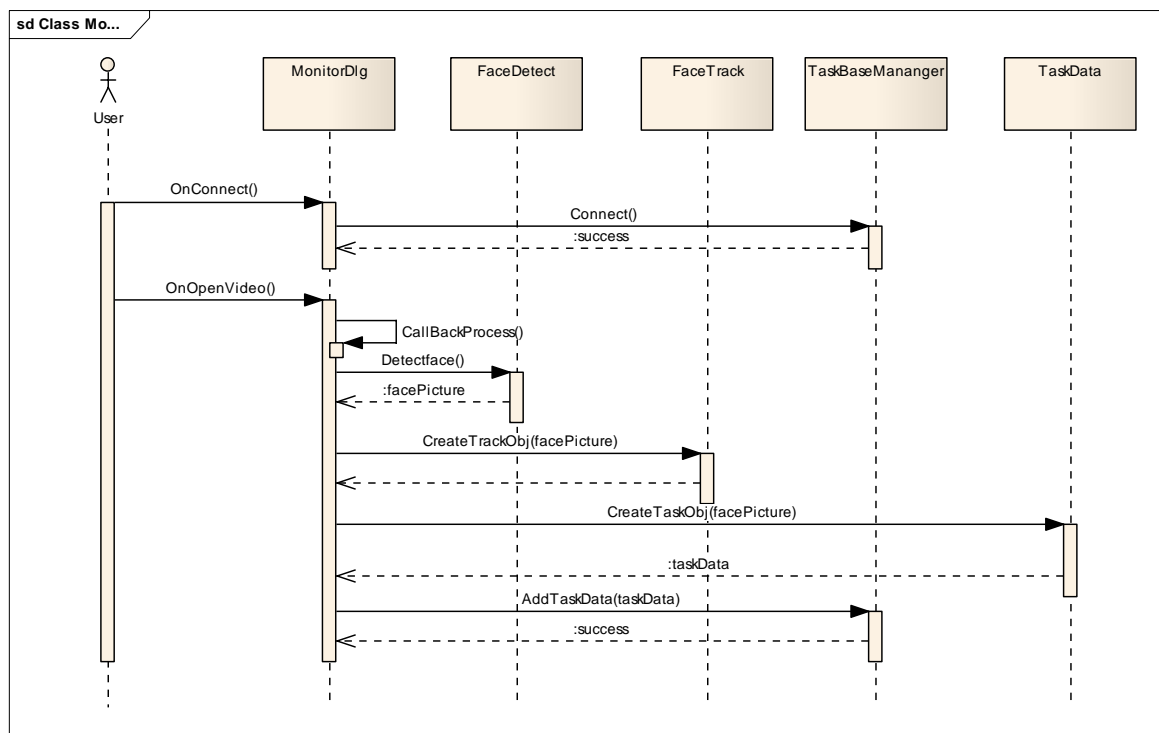


图 6-2 添加识别任务用例实现

6.2.2 服务端

(1) 识别任务处理用例实现

1. 服务端启动多线程人脸识别进程 **startThread**
2. 服务端创建多线程工作实例 **CreateWorker**
3. 服务端连接数据库 **Connect**，并连接成功
4. 服务端提取所有识别任务 **QueryTaskData**，并返回识别任务
5. 服务端为每个识别任务创建 **Job** 实例 **CreateJob**
6. 服务端调用线程的 **ProcessWorker** 处理每个 **Job**
7. 每个线程从队列中弹出每个 **Job**，并进行识别处理 **faceReg**
8. 线程根据识别结果创建识别记录 **CreateObj**，并连接面像数据库 **Connect**，将识别记录插入数据库 **Add**

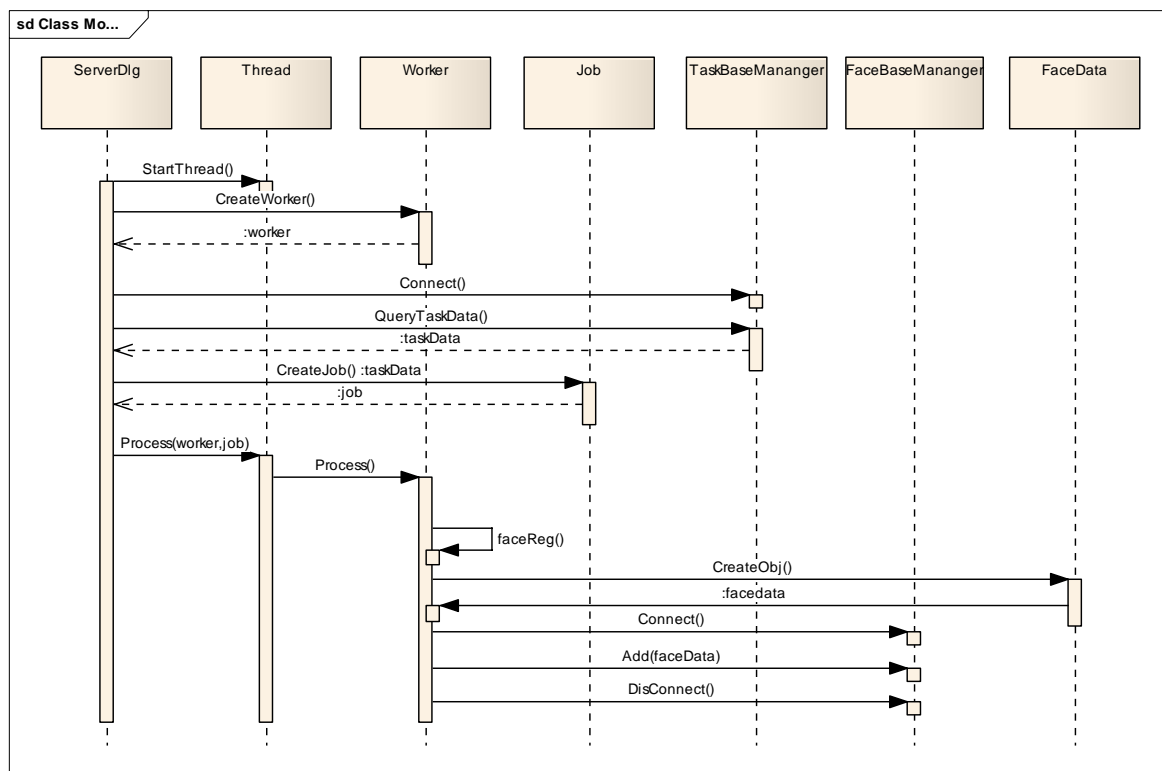


图 6-3 识别任务处理用例实现

(2) 查询请求处理用例实现

1. 网络控制模块接受查询请求消息,将消息传递给中心处理程序 **SendMessage**
2. 服务端接受消息后,解析消息命令和内容 **DecodeMessage**
3. 服务端连接数据库 **Connect**,并连接成功
4. 服务端根据查询请求的内容调用 **FaceBaseManager** 实例查询所有符合条件的内容,并返回所有结果
5. 服务端接受结果后将结果写入文件,并成功建立 **tcp** 连接 **ConnectTcp**
6. 服务端发送文件 **sendFile**

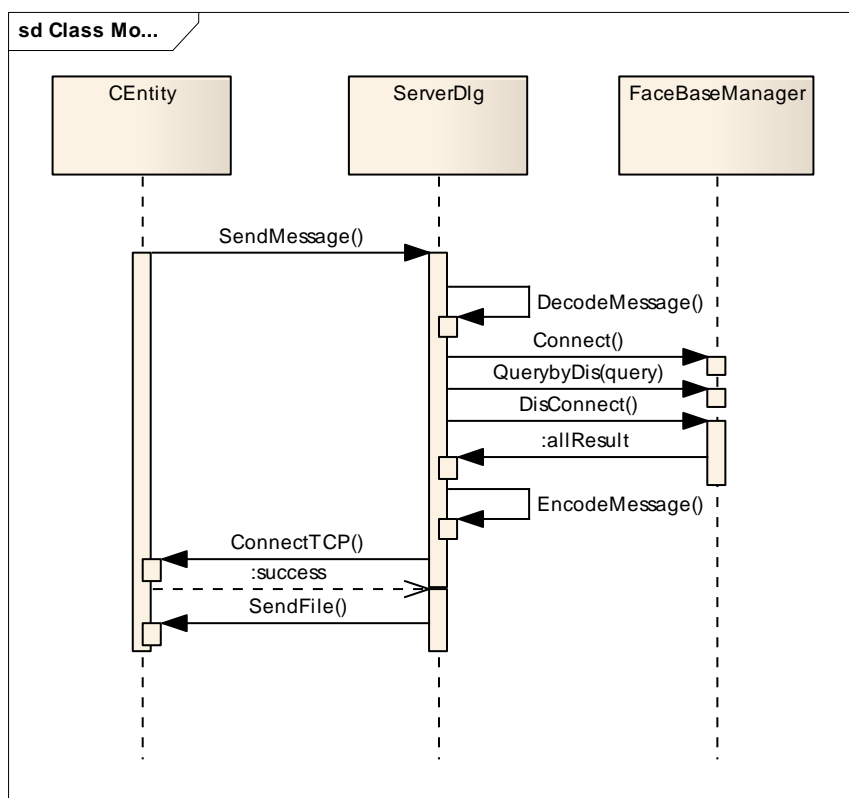


图 6-4 查询请求处理用例实现

(3) 面像库管理用例实现

面像库管理用例由通过面像库的查询、查看以及删除用例完成。下面介绍这三个用例具体实现过程。

➤ 面像库查看用例实现

1. 用户点击面像库管理界面 **OnManager**
2. 中心处理程序创建面像库管理界面实例 **CreateObj**
3. 用户点击查看所有记录 **OnShowAll**
4. 面像库管理界面调用面像管理进程完成查询任务 **QueryAll**，并返回结果
5. 面像库管理界面显示所有结果 **ShowResult**

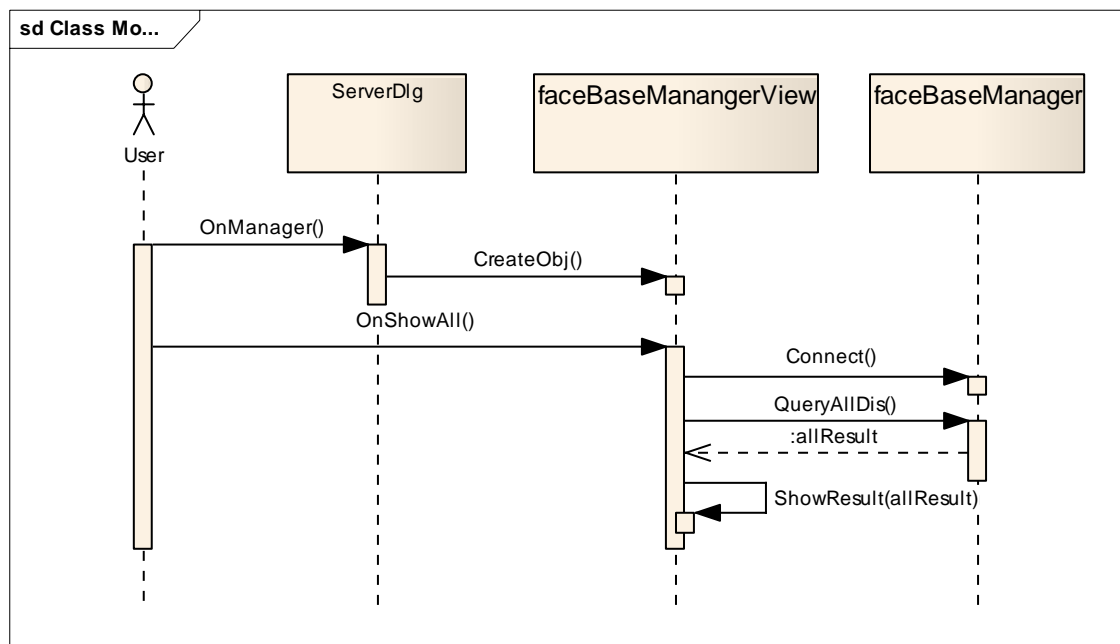


图 6-5 面像库查看用例实现

➤ 面像库查询用例实现

1. 用户点击面像库管理界面 OnManager
2. 中心处理程序创建面像库管理界面实例 CreateObj
3. 用户点击查询 OnQuery
4. 面像库管理界面创建查询框实例 CreateObj
5. 用户填写内容后，点击确认 OnOk
6. 面像管理库根据查询内容调用 FaceBaseManager 实例查询所有符合条件的内容，并返回所有结果
7. 面像管理库界面显示所有结果 ShowResult

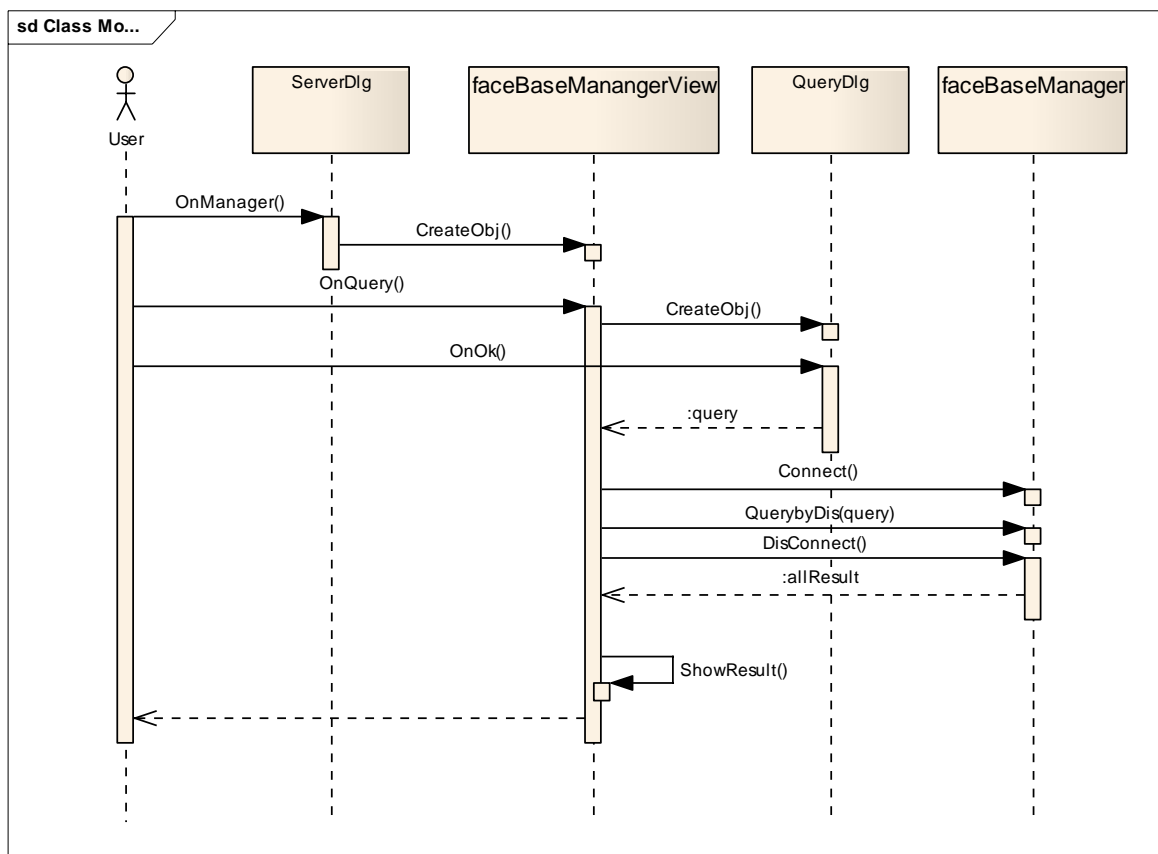


图 6-6 面像查询用例实现

➤ 面像库删除用例实现

面像库删除某一行或多行的操作的前提是显示查询或查看的结果

1. 用户在面像库管理界面显示的查询结果中选择一行或多行 **OnSelect**
2. 用户点击删除 **OnDelete**
3. 面像管理库根据删除记录的 ID 调用 **FaceBaseManager** 实例删除所有符合条件的记录 **DelDataById**，并更新面像库管理界面显示结果

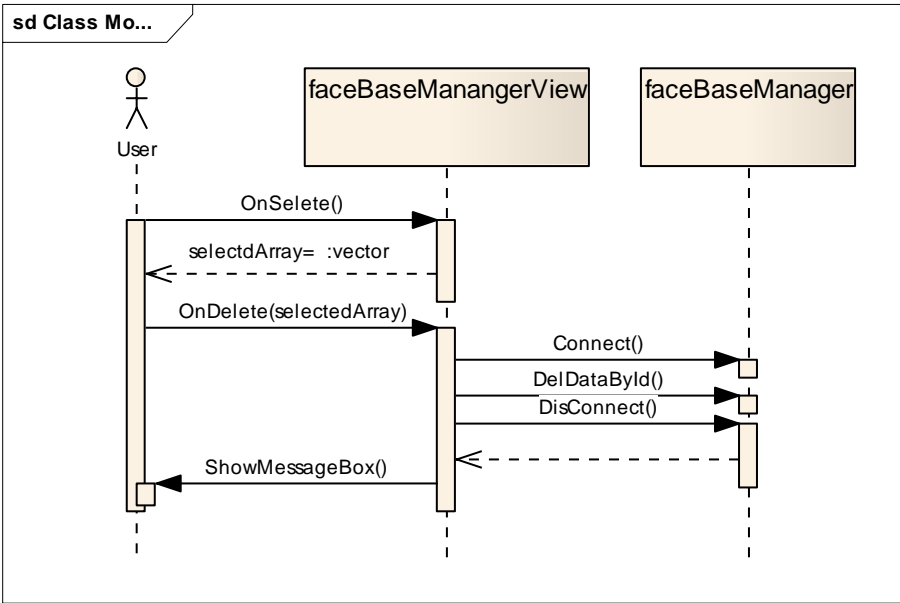


图 6-7 面像库删除用例实现

6.2.3 客户端模块

(1) 查询用例实现

- 1. 用户在客户端界面点击查询 OnQuery
- 2. 客户端创建查询对话框 CreateObj
- 3. 用户在查询对话框填写查询内容，并点击确认 OnOk
- 4. 客户端创建消息实例，封装查询命令和内容 CreateObj
- 5. 客户端发送查询消息 SendMessage

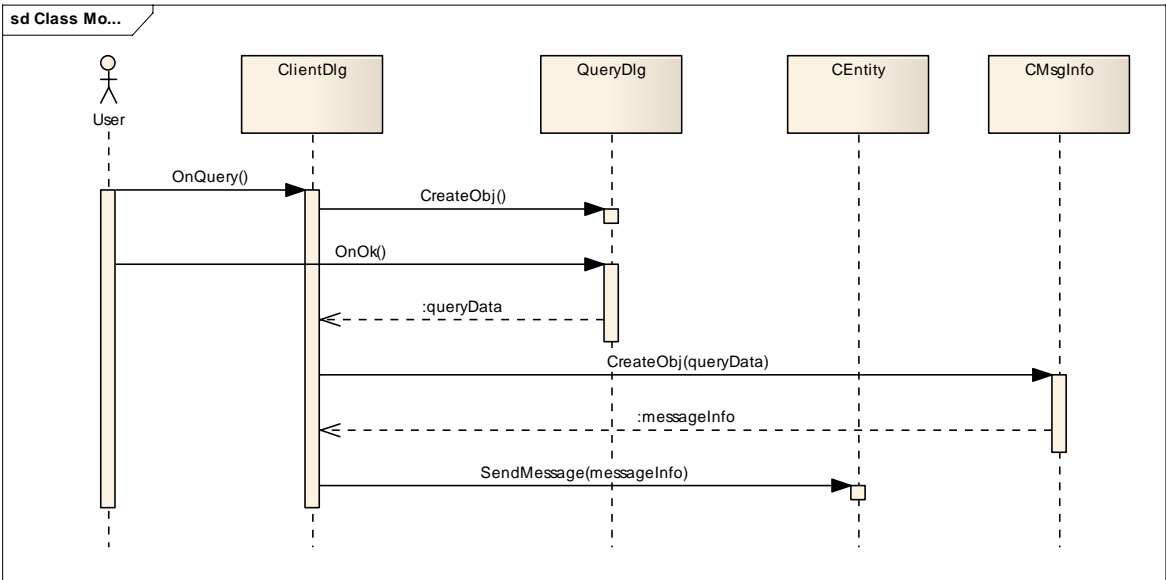


图 6-8 查询用例实现

6.2.4 图像采集模块

(1) 人员注册用例实现

1. 用户打开摄像头 OnOpenCamera
2. 用户点击注册 OnAddObject
3. 图像采集模块创建基本信息对话框实例 CreateObj
4. 用户填写对话框内容，并确认 OnOk，并调用 imageBaseManager 插入信息内容
5. 图像采集模块创建保存图片目录 CreateDirectory
6. 用户添加图片 OnNewPicture，直到结束

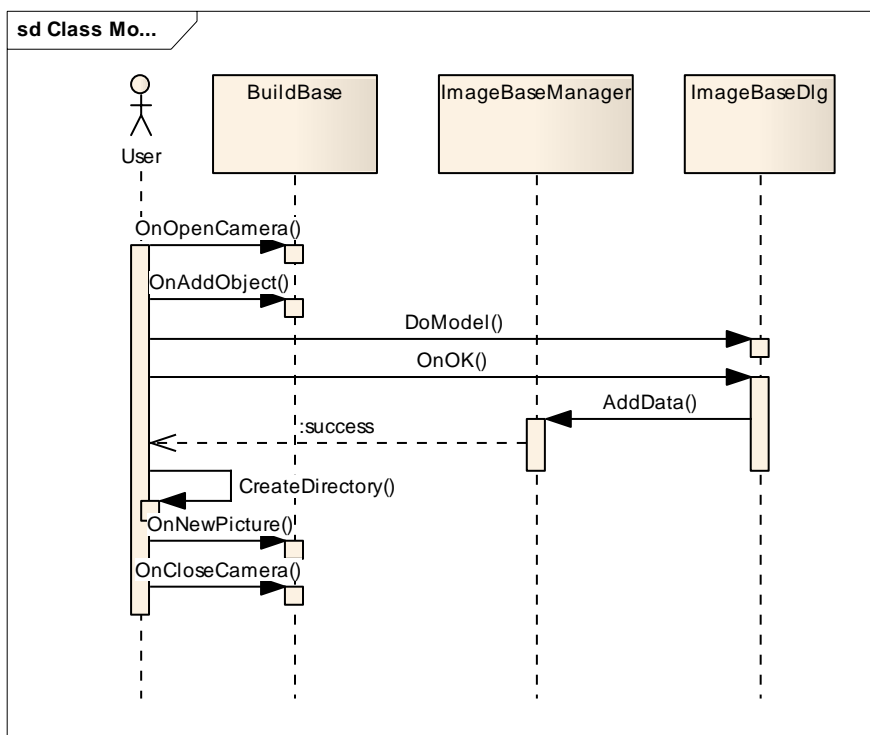


图 6-9 人员注册用例实现

(2) 人脸信息库管理用例实现

人脸信息库管理用例由通过面像库的查询、查看、添加、更新、删除用例完成。下面介绍这三个用例具体实现过程。

➤ 人脸信息库查看用例实现

1. 用户在图像采集界面点击人脸信息库管理 **OnManager**
2. 图像采集程序创建人脸信息库管理界面实例 **CreateObj**
3. 用户点击查看所有记录 **OnShowAll**
4. 人脸信息库管理界面调用人脸信息管理进程完成查询任务 **QueryAll**，并返回结果
5. 人脸信息库管理界面显示所有结果 **ShowResult**

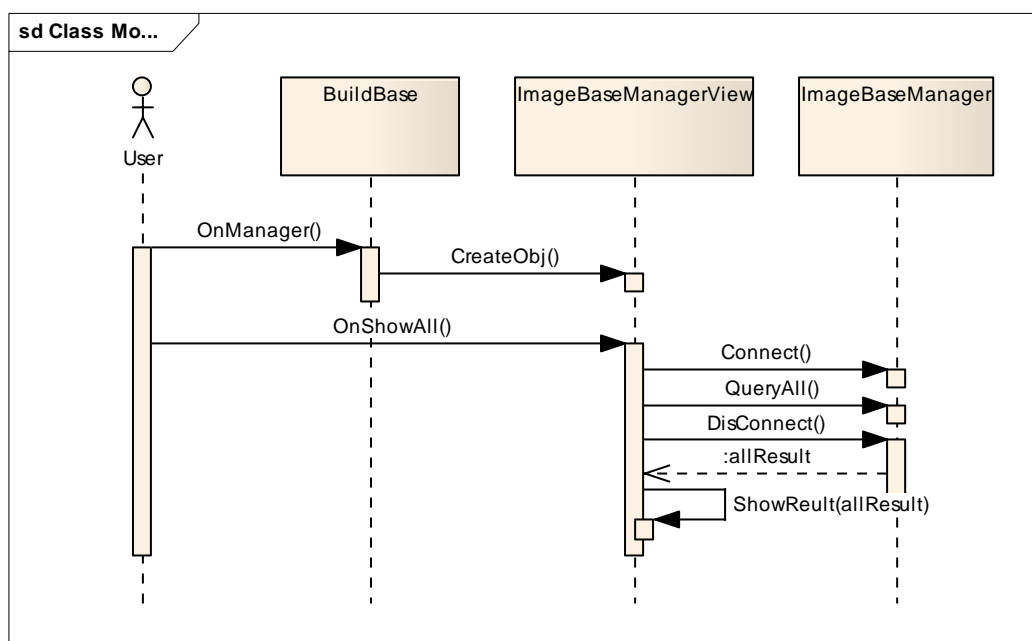


图 6-10 人脸信息库查看用例实现

➤ 人脸信息库查询用例实现

1. 用户在图像采集界面点击人脸信息库管理 **OnManager**
2. 图像采集程序创建人脸信息库管理界面实例 **CreateObj**
3. 用户点击查询 **OnQuery**
4. 人脸信息库管理界面创建查询框实例 **CreateObj**
5. 用户填写内容后，点击确认 **OnOk**
6. 人脸信息库管理根据查询内容调用 **ImageBaseManager** 实例查询所有符合条件的内容，并返回所有结果
7. 人脸信息库管理界面显示所有查询结果 **ShowResult**

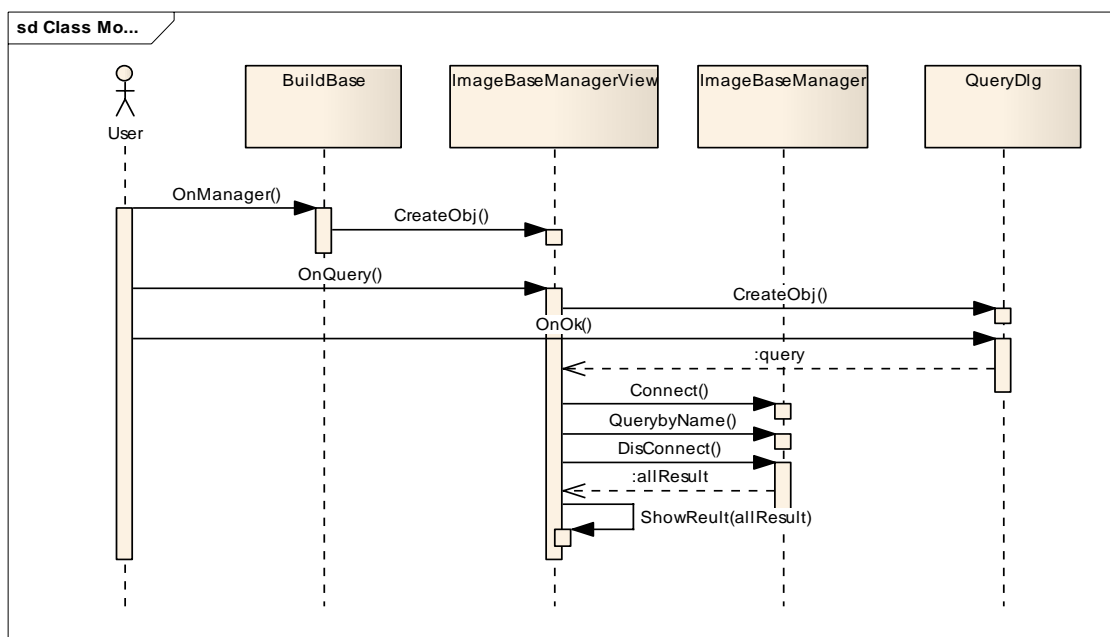


图 6-11 人脸信息库查询用例实现

➤ 人脸信息库添加用例实现

1. 用户在图像采集界面点击人脸信息库管理 OnManager
2. 图像采集程序创建人脸信息库管理界面实例 CreateObj
3. 用户点击添加 OnAddObj
4. 人脸信息库管理界面创建基本信息框实例 CreateObj
5. 用户填写内容后，点击确认 OnOk
6. 人脸信息库管理界面根据用户填写的内容创建信息实例 CreateObj，并调用 ImageBaseManager 实例将消息实例添加到数据库中

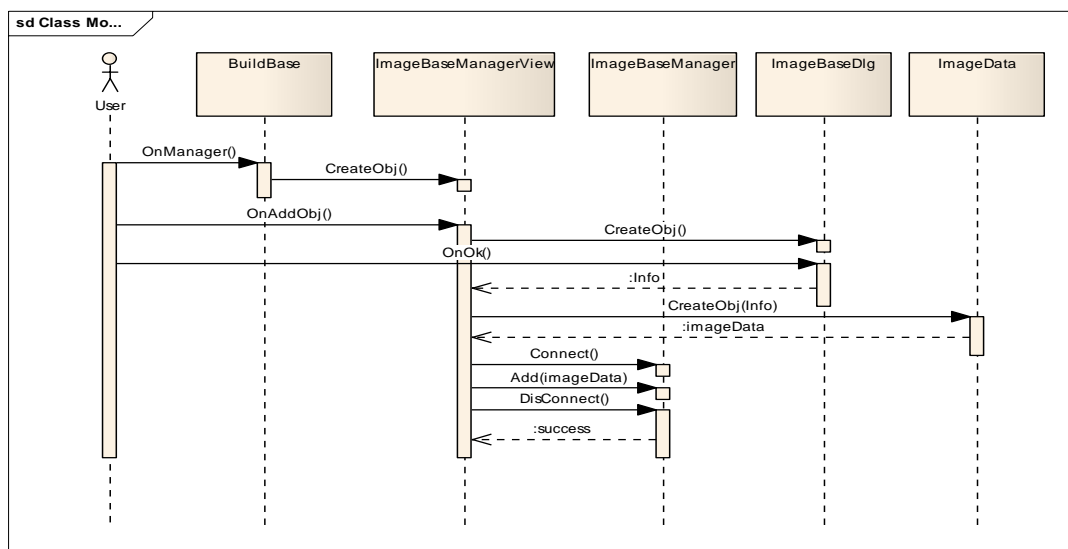


图 6-12 人脸信息库添加用例实现

➤ 人脸信息库更新用例实现

人脸信息库更新某一行记录操作的前提是显示查询或查看的结果

1. 用户在人脸信息库管理界面选择需要更新的记录 **OnSelect**
2. 用户点击更新 **OnUpData**
3. 人脸信息库管理界面调用人脸库管理进行查询该记录的信息，并返回信息内容
4. 人脸信息库创建基本信息对话框显示信息内容
5. 用户更新内容，并点击确认 **OnOk**
6. 人脸信息库管理界面根据用户填息的内容创建信息实例 **CreateObj**，并调用 **ImageBaseManager** 实例将消息实例添加到数据库中

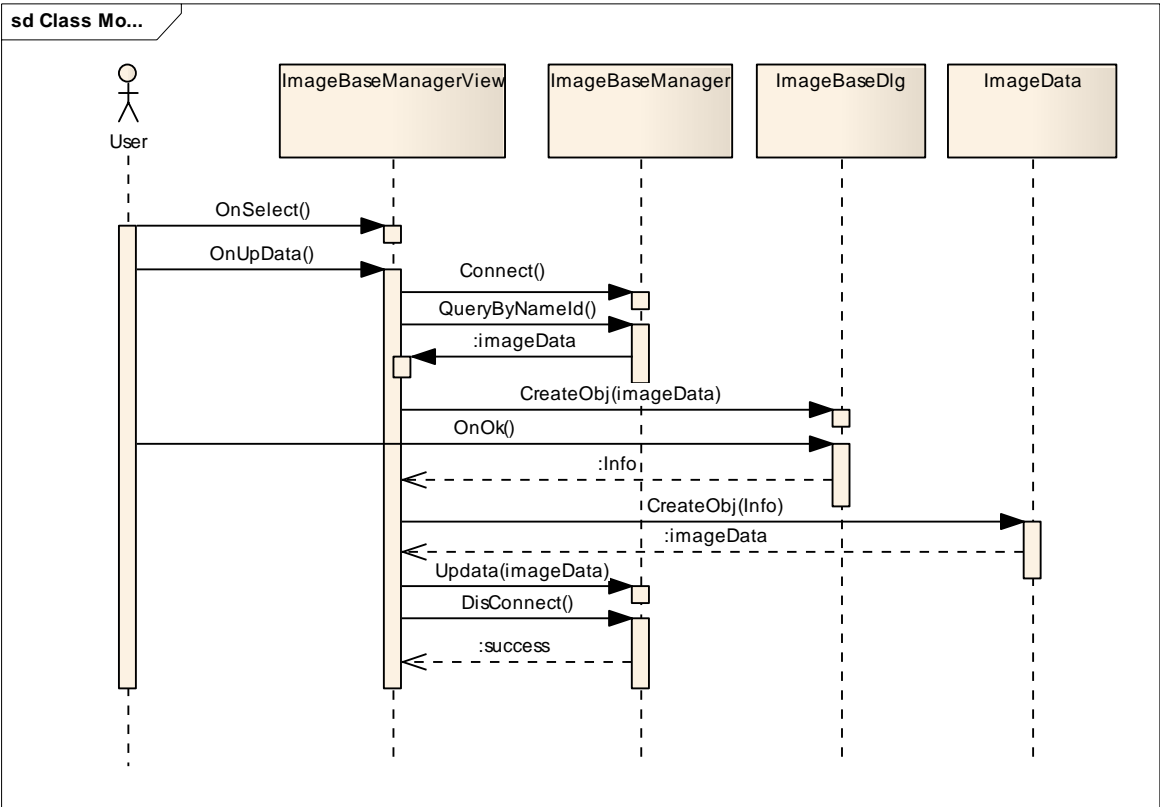


图 6-13 人脸信息库更新用例实现

➤ 人脸信息库删除用例实现

人脸信息库删除某一行或多行操作的前提是显示查询或查看的结果

- 1. 用户在人脸信息库管理界面显示的查询结果中选择一行或多行 **OnSelect**
- 2. 用户点击删除 **OnDelete**
- 3. 人脸信息库管理界面根据删除记录的 **ID** 调用人脸信息管理实例删除所有符合条件的记录 **DelDataById**，并更新人脸信息库管理界面显示结果

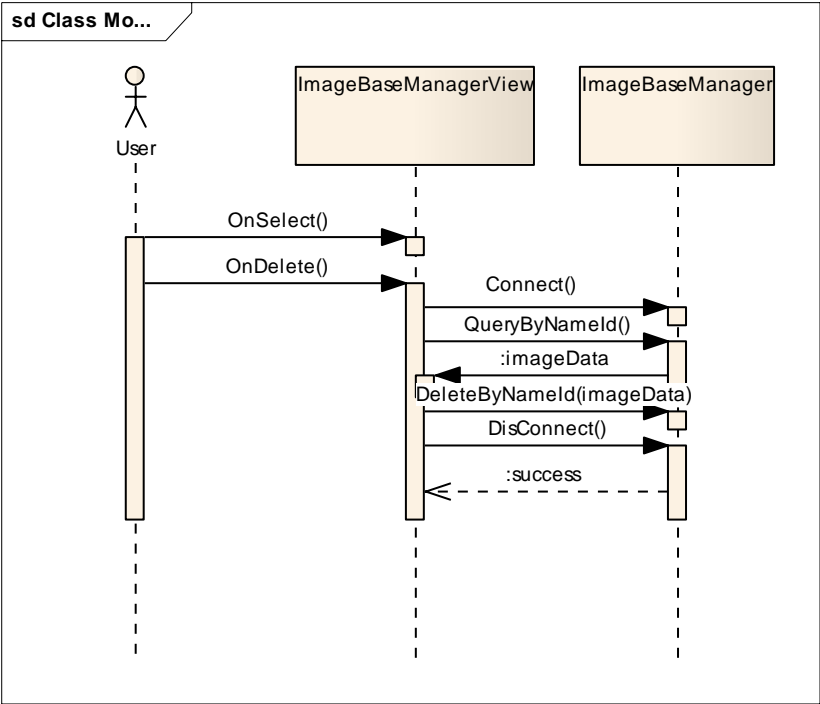


图 6-14 人脸信息库删除用例实现

6.3 系统数据库的设计

为了给特定的数据以及活动提供持久化的服务，系统中以数据库作为数据的存储介质。

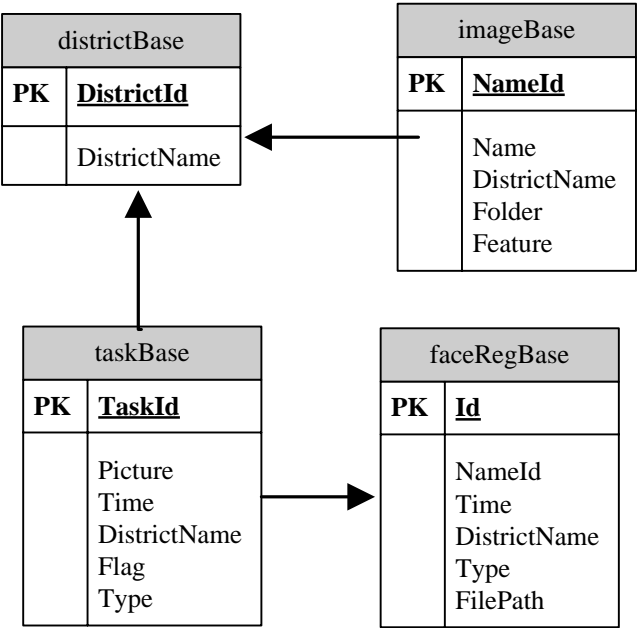


图 6-15 数据库设计图

上图以 E-R 图的形式描述了系统中主要的表以及它们之间结构和关系。DistrictBase 为地方数据表，存储监控地点对应的 ID 以及名称。imageBase 为人脸信息表，存储注册人员的基本信息以及面像索引，其主键为 NameID，Name 表示人员名称，其外键 DistrictId 表示监控地点对应的 ID，对应为 DistrictBase 的主键。TaskBase 为识别任务表，其余字节为任务表所需的参数，如图像索引，时间等。其中主键 taskId 作为标志任务的唯一主键。外键 DistrictName 分布对应表 districtBase 的 DistrictName。flag 标志该任务是否完成。faceRegBase 为面像表，其存储识别后的基本信息。NameId 表示识别的人员的 Id，如果 NameId 为-1，那么表示该 Id 不存在。Time 表示人脸出现的时间，Type 表示出或入。DistrictName 表示图像的来源。

6.4 本章小结

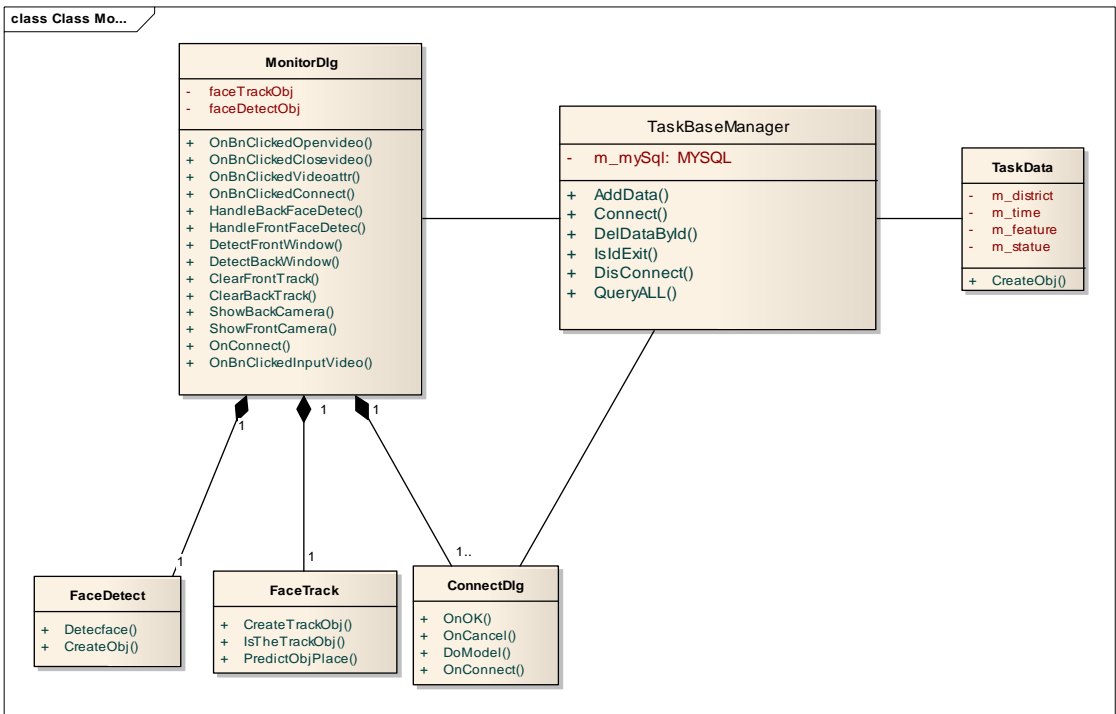
本章根据前面的需求分析，对系统功能作了设计，为下面设计和实现各个模块，进而完成各模块的详细设计与实现做准备。

第七章 系统详细设计

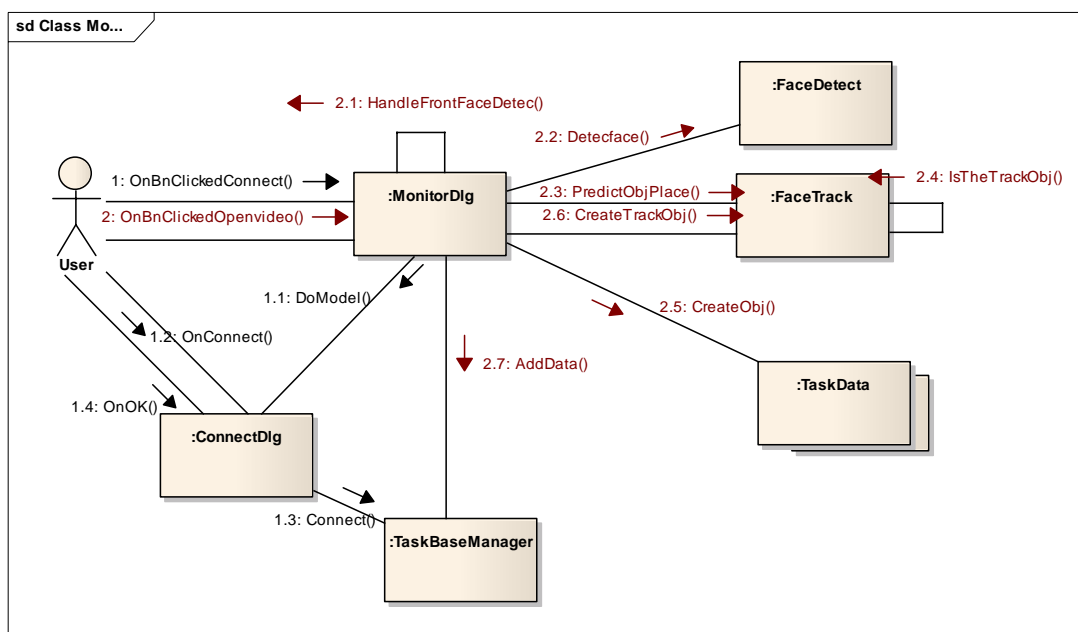
本章主要介绍每个模块类之间的关系以及类的设计。并针对部分用例的时序图设计其类之间的协作图。

7.1 监控端模块详细设计

如图 7-1 为监控端模块类图设计。为了方便日后算法的扩展，本文将人脸跟踪和人脸检测算法的操作抽象为一个独立的类。人脸跟踪算法提供公共接口 createTrackObj、IsTheTrackObj 和 PriectObjPlace。MonitorDlg 用 DetectTrackObject、DetectWindow、ClearTrackObj 操作封装 FaceTrack 类对应的接口。另外它还提供 DetectFace 调用人脸检测算法的 DetecFace 操作。MonitorDlg 提供 OnBnClickedConnect 操作创建 ConnectDlg 实例，ConnectDlg 提供 OnConnect 操作创建 TaskBaseMananger 实例，并调用 TaskBaseManager 的 Connect 操作连接数据库。MonitorDlg 提供 AddFaceObj 添加 TaskData 实例。如图 6-2 为监控端添加识别任务的类之间协作图，它描述了整个添加识别过程的类之间的函数时序过程。



如图 7-1 监控端模块类图



如图 7-2 监控端添加识别任务的类协作图

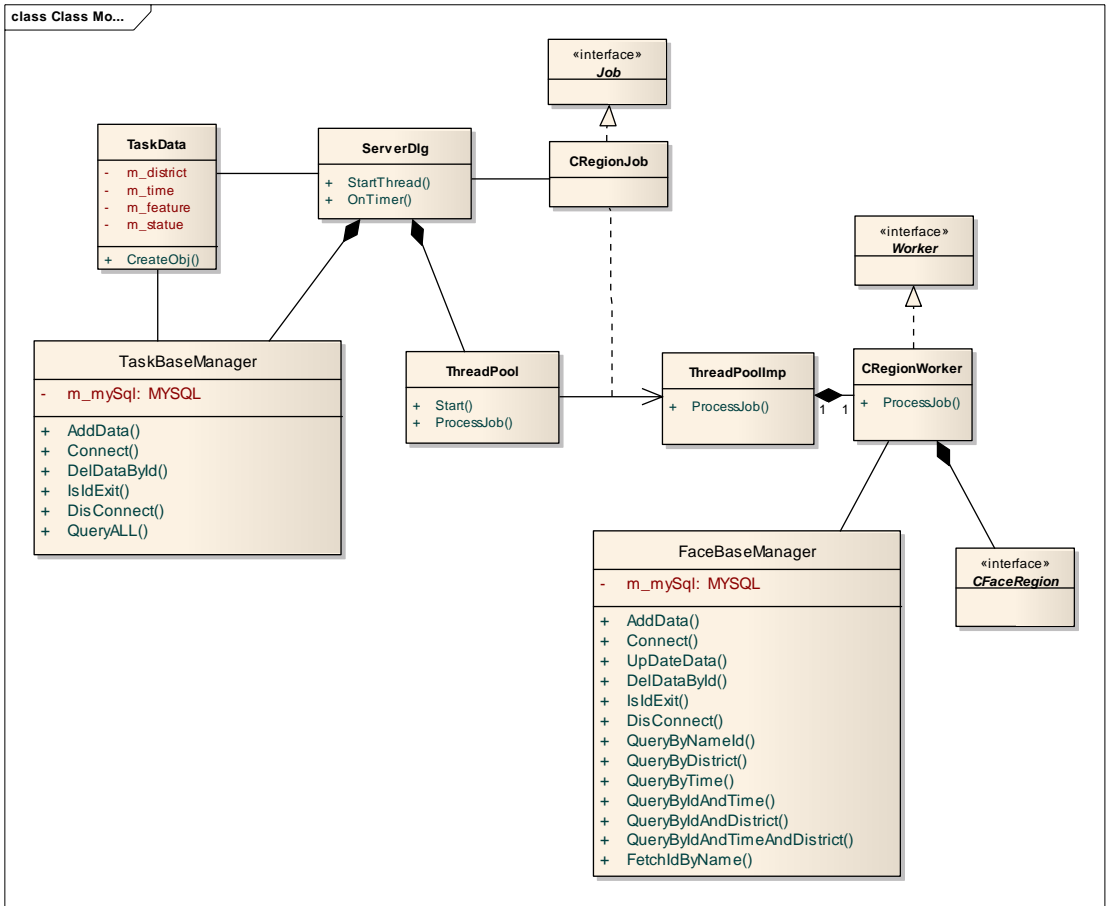
7.2 服务端详细设计

根据服务端的功能设计，服务端模块分为识别任务处理、网络控制、面像库管理三部分。以下根据每个功能设计相应的类图。

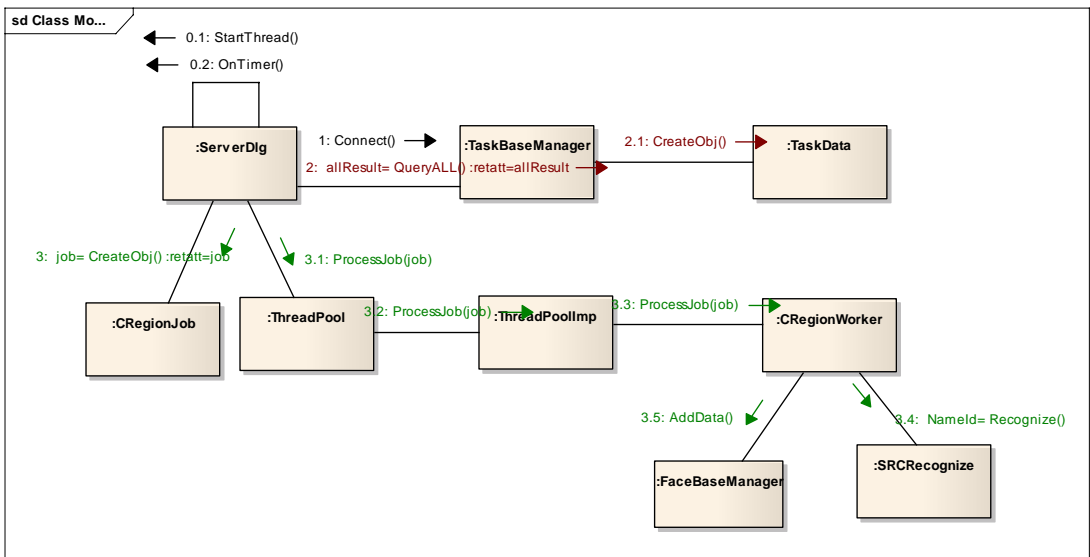
(1) 识别任务处理类图设计

如图 7-3 为服务端识别任务处理类图设计。ServerDlg 提供 OnTimer 和 StartThread 接口完成识别任务处理。OnTimer 主要负责定期从数据库中提取识别任务记录，并封装成识别任务，将任务添加到线程池中。StartThread 完成线程池的启动，用户可以在该操作中设定线程池的大小和最多启动线程的数目。ThreadPool 启动线程，并通过 ProcessJob 接口将识别任务 Job 传递给 ThreadPoolImp。ThreadPoolImp 主要负责将识别任务添加到任务队列中，并从线程池中选择线程完成识别任务。另外每个线程都调用 Woker 的 ProcessJob 操作完成识别处理。Worker 的 ProcessJob 调用通用模块 FaceRegion 的 Regionze 识别操作，接着将返回的结果封装为 FaceData 类型的数据，最后调用 FaceBaseMananger 向数据库插入类型为 FaceData 的数据。ThreadPoolImp 还负责设置线程的状态和分配线程工作。TaskBaseMananger 提供所有识别任务库的数据库操作，ServerDlg 的 OnTime 调用它的 QueryAll 接口，提取所有识别任务，并返回 TaskData 类型的数据。图 7-4 为识别任务处理协作图，它描述整个识别处理过程

中参数的传递以及函数操作的时序过程。



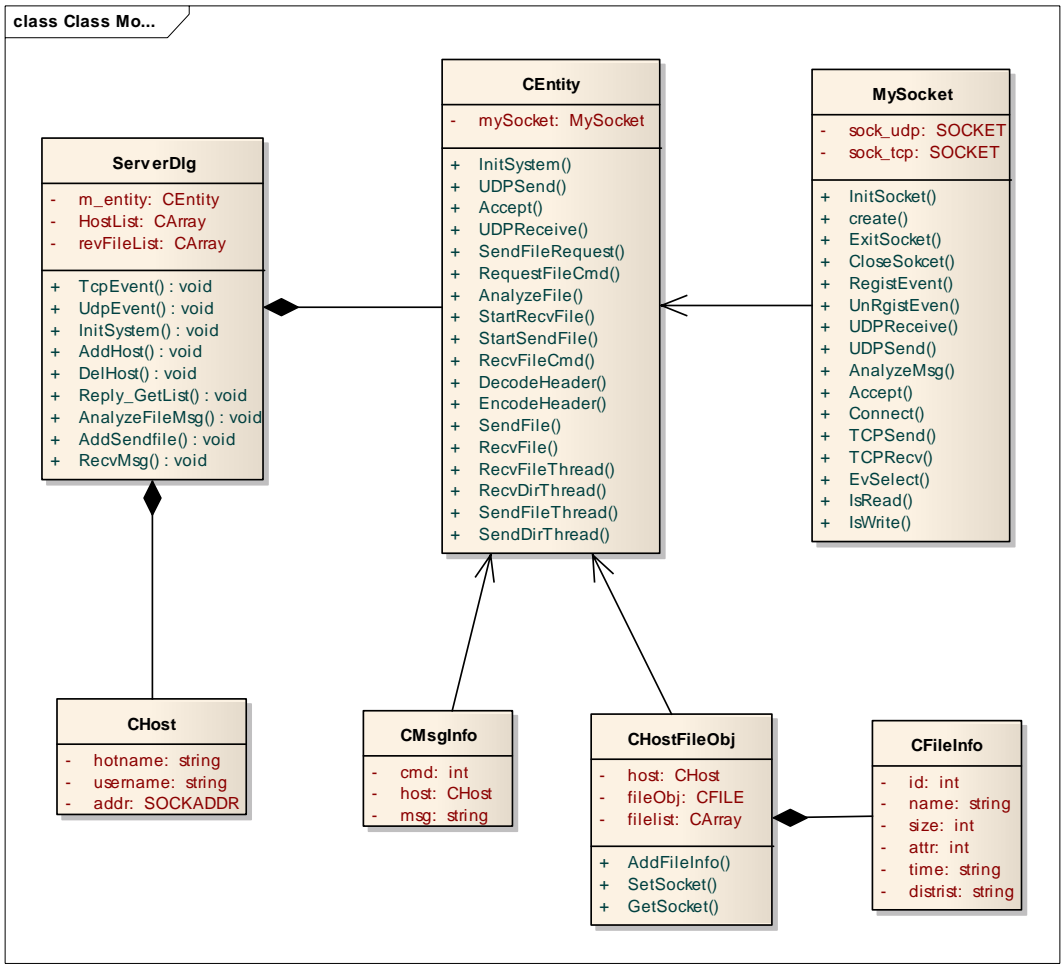
如图 7-3 识别任务处理模块类图



如图 7-4 识别任务处理类协作图

(2) 网络控制的类图设计

如图 7-5 为服务端网络控制的类图设计。类 **ServerDlg** 代表所有服务程序。**CEntity** 封装 **MySocket** 的所有操作，并把 **CMsgInfo** 作为与 **MySocket** 参数传递和接受的对象。对于 **CEntity** 发送的 **CMsgInfo** 对象，**MySocket** 解析其对象，并发送相应信息；对于外面接受的消息，**MySocket** 解析信息，并封装为 **CMsgInfo** 对象，传送给 **CEntity**。类 **CHost** 保存主机信息（主机名，地址等）。**CFileInfo** 封装文件信息。**CHostFileObj** 维护所有需要发送或接受的文件列表。**MySocket** 提供与底层 **socket** 通信接口。表格 7-1 主要介绍对话框类的函数功能，它提供整个通信模块与系统其他模块通信的接口。表格 7-2 介绍网络控制消息命令的设计，服务端模块与客户端通过发送和接受消息命令，实现相应的操作。



如图 7-5 网络控制模块类图

表 7-1 ServerDlg 类函数功能列表

函数名称	函数功能
------	------

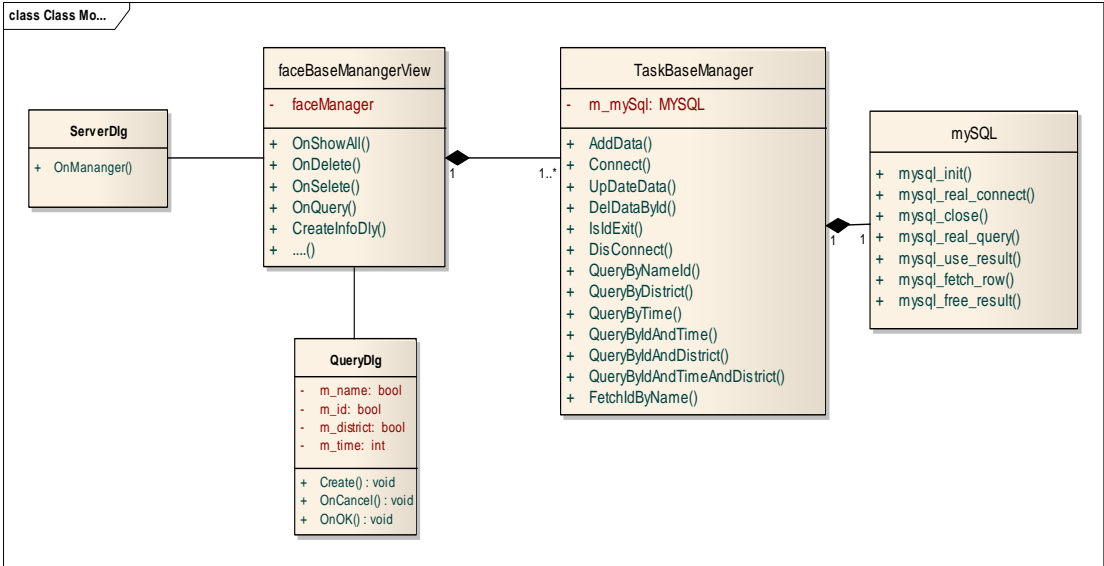
TcpEvent	注册 Tcp 响应事件。每个服务程序要响应来自 tcp 端口的事件，都需要注册 tcp 事件，它是整个通信模块的核心函数。
UdpEvent	注册 Udp 响应事件。每个服务程序要响应来自 udp 端口的事件，都要注册 udp 事件，它是整个通信模块的核心函数。
InitSystem	绑定 tcp 端口，并获取本机和其他连接的主机名字，地址等
AddHost	每个服务程序都维护一个主机列表，AddHost 添加每个主机对象
DelHost	当其他主机离开时候，DelHost 删除其对应的主机信息
AnalyzeFileMsg	解析每个接收的 CHostFileObj 对象
RecvMsg	对 CEntity 传递的 CMsgInfo 对象，RecvMsg 解析其命令，并激发相应事件。

表 7-2 消息响应命令设计

命令	功能
MSG_GETMONITORLIST	要求服务端把监控点信息添加到监控点映射表
MSG_GETCLIENTLIST	要求服务端把客户端主机信息添加到客户端主机映射表
MSG_AWAY	要求服务端删除发送方的主机信息
MSG_FILEDATA	要求接受到该信息的主机接受文件
MSG_FILEDIR	要求接受到该信息的主机接受文件目录
MSG_GETLOCAL	添加本地主机信息
MSG_GETREQUEST	要求服务端接受查询请求

(3) 面像库管理类图设计

图 7-6 为面像库管理类图设计。面像库管理模块大致分为三层：界面层，业务逻辑层以及数据访问层。ServerDlg 提供管理面像库操作的接口 OnManager。类 faceBaseManagerView 主要完成数据的展示和人机交互等工作。类 FaceBaseManager 通过封装 mySQL 的操作，提供查询、查看、删除数据等业务。类 mySql 直接访问数据库，对数据库实现相应的操作。类 QueryDlg 实现查询条件的设置。



如图 7-6 面像库管理模块类图

7.3 客户端模块详细设计

图 7-7 为客户端类图设计。客户端由查询模块和网络控制模块这两部分组成。客户端的网络控制模块的类图结构与中心处理模块的网络控制模块的类图结构一样，在此省略描述。这样设计的目的是为了设计网络控制通用组件，它可以应用到其他模块中。ClientDlg 提供 OnQuery 和 PrintResult 以及 ShowResult 操作。当用户发出查询请求时，ClientDlg 创建 QueryDlg 实例，用户填写查询内容。接着 ClientDlg 将查询的内容封装为消息格式（消息命令 MSG_GETREQUEST +内容），最后通过 CEntity 发送查询请求。CEntity 封装所有网络通信的基本操作。

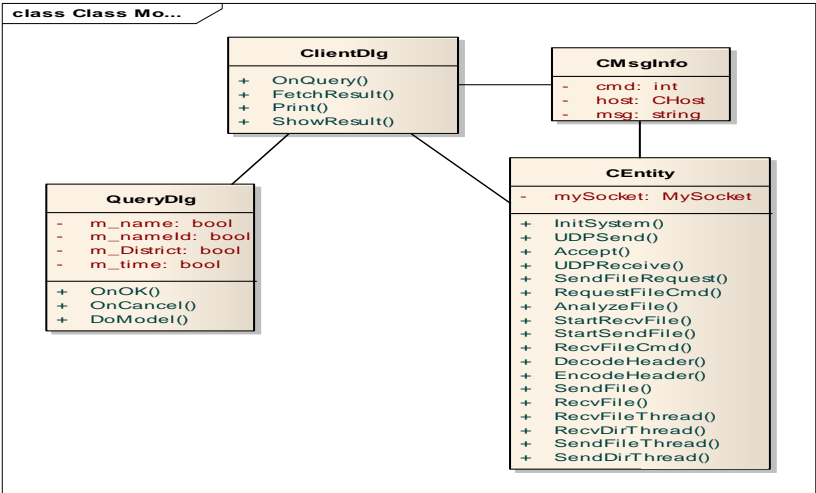


图 7-7 客户端模块类图

7.4 图像采集模块详细设计

图 7-8 为图像采集模块的类图设计。BuildBase 提供 OnManager 和 OnAddObj 接口。OnManager 创建类 ImageBaseManangerView 的一个实例。ImageBaseManagerView 实现人脸信息库的管理，它提供数据的展示和人机交互等工作。类 ImageBaseManager 通过封装 mySQL 的操作，提供查询、查看、更新、删除数据等业务。类 mySql 直接访问数据库，对数据库实现相应的操作。类 QueryDlg 实现查询条件的设置。类 CFaceBase、类 CPerson、CPersonImage 实现人脸图像的采集。类 PictureProcess 实现人脸图像的预处理和标准化过程，其中预处理过程包括光线补偿、平滑处理以及直方图均衡。类 ImageBaceDlg 实现人员基本信息的设置。

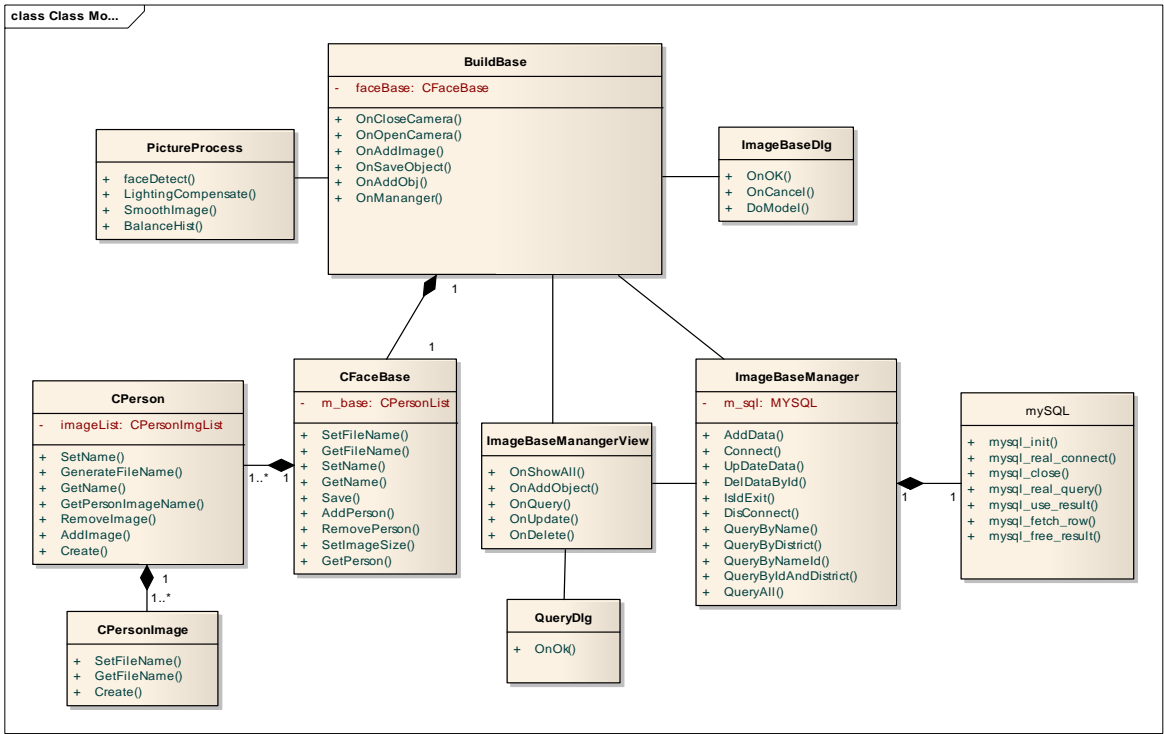


图 7-8 图像采集模块类图

7.5 人脸识别算法分析模块详细设计

图 7-9 为人脸识别算法分析模块类图设计。**FaceRecogAnalyzeDlg** 提供算法分析界面。**BaseInfoPanel** 属于 **FaceRecogAnalyzeDlg** 的一部分，它提供基本信息的配置。**FaceRecognize** 提供所有算法的接口操作，它提供训练 **train**，测试 **testOne** 或者批量测试 **testBatch** 的操作接口，其他算法实现这些接口。本系统由基于压缩感知的人脸识别算法、**SVM** 的人脸识别算法、最近邻接法的人脸识别算法以及基于隐式马克夫模型的人脸识别算法组成。用户可以选择任意一种人脸识别算法作为整个系统的主要人脸识别算法。另外除了基于隐式马克夫模型的人脸识别算法外，其他算法都要提取图像特征进行训练，所以人脸识别模块还提供 **fisher**，**eigen** 的特征脸抽取，并采用 **PCA** 进行特征空间的降维。

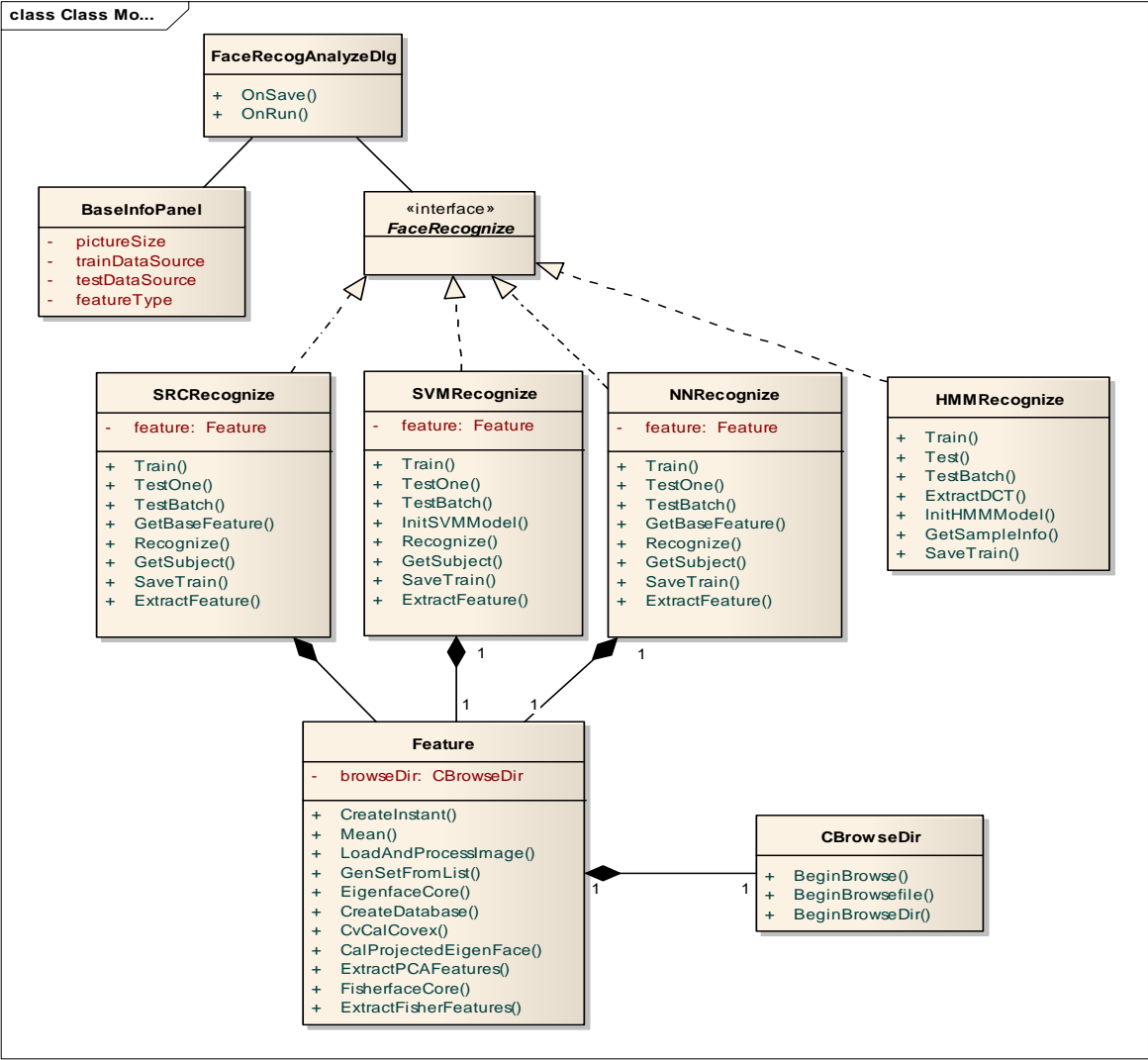


图 7-9 人脸识别算法类图

人脸识别一般流程如下：

1. 训练样本图片-Train
2. 保持训练结果
3. 选择测试图片
4. 加载训练结果，并提取测试图像特征值
5. 根据特征值和训练集的特征值进行人脸识别
6. 输出识别结果。

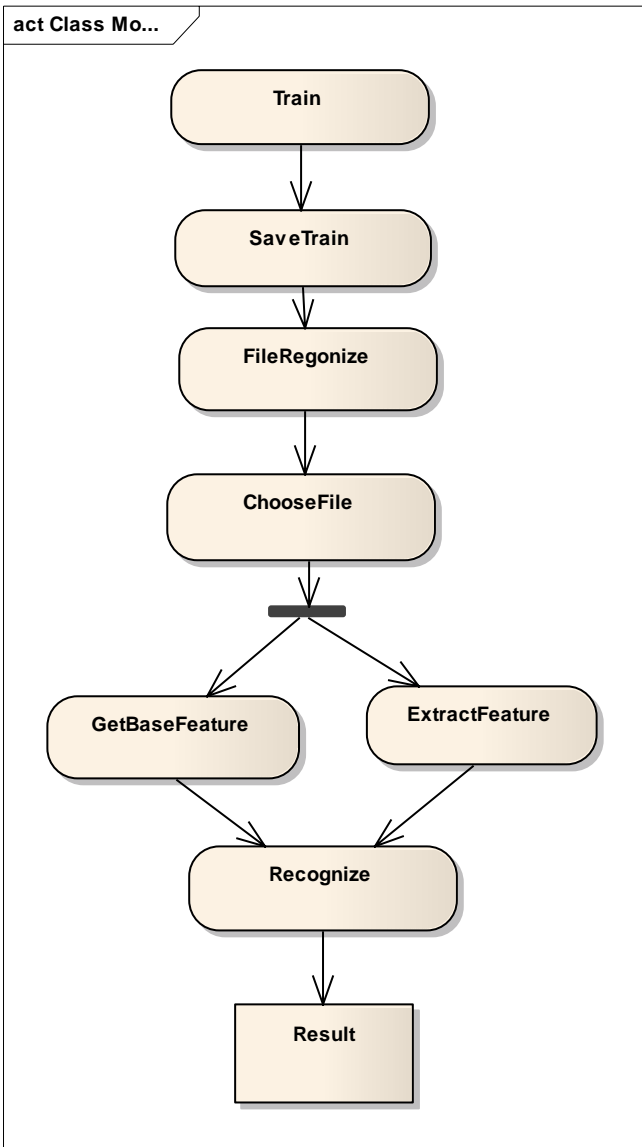


图 7-10 人脸识别流程图

7.6 本章小结

考勤系统由监控端、服务端、客户端、图像采集和人脸识别算法分析模块五大系统模块组成，本章主要介绍了这五大模块的具体实现方式，对实现模块的主要类以及它们提供的接口进行了详细的介绍。

第八章 实现及测试结果分析

本章主要介绍了系统的开发工具、运行环境以及测试过程。**8.1** 节主要介绍开发工具及运行环境。**8.2** 节主要介绍测试的前期工作、测试用例以及测试结果。测试的前期工作包括图像采集、系统配置。系统测试过程分为三个部分：第一部分根据系统功能设计，测试系统是否实现所有的功能；第二部分根据系统的性能设计，测试系统运行的速度、系统稳定性是否达到系统要求；第三部分根据系统技术设计，测试系统算法在正常情况和不正常情况下这些算法的性能情况。根据这三个测试过程，分别设计不同的测试用例，并根据测试结果进行分析。

8.1 开发工具及运行环境介绍

表 8-1 测试工具和运行环境介绍

开发工具	
VS2005.NET	
MySQL 2005	
Opencv1.0	
运行环境	
数据库服务器	MySQL Server5.1
中间服务器	操作系统：Windows xp 内存：2G 以上 CPU:1.86GZ 以上
监控端	操作系统：Windows xp 内存：1G 以上 CPU:0.96GZ 以上
客户端	操作系统：windows xp 内存和 CPU 无要求
摄像头	两个
部署环境	
中间服务器	服务端服务程序
监控端	监控端服务程序
客户端	客户端服务程序

8.2 软件测试

8.2.1 测试前期工作

（1）测试环境

表 8-2 描述了系统测试环境的情况。其中，服务器运行中心处理程序以及 MySQL5.0 数据库。监控端的 PC 机运行监控程序，由于普通的 PC 机只有一个视频采集卡，要实时监控某个区域的出入情况，必须有两台 PC 机。客户机运行客户端程序，中心处理程序启动以后，客户端的程序与服务端进行 Socket 连接，负责向服务端发送查询请求和接收查询结果，并显示。摄像头主要实现视频流的实时采集，它们分别部署在不同监控端的 PC 机上.图 8-1 为系统测试的实验环境。其中一个摄像头监控门口人进来实验室的情况，另外一个摄像头监控人离开实验室的情况。本文的测试数据来源该测试环境采集的数据。

表 8-2 测试设备

设备	数量	平台
客户机	6	Windows XP SP2 专业版
服务器	1	Windows XP SP2
监控端的 PC 机	2	Windows XP SP2
摄像头	2	分辨率为 320*240
视频文件	2	分辨率为 320*240



图 8-1 测试环境

（2）采集人脸信息

由于系统根据人脸库的图片进行训练和识别，因此采集人脸信息是系统运行的前提。另外图片的质量直接影响识别算法的准确性，所以在采集过程中，必须根据算法要求采集合适的图像。本文采用基于稀疏表示的人脸识别算法。根据第二章对该算法原理的分析，得出人脸库的图片应该尽可能满足多样性和均匀性的要求。多样性是指同一个人的图片其拍摄的表情、角度、光照应该不同。均匀性是指不同的人的照片其拍摄光照、角度、表情应该相同，也就是说当有某个人侧面 45 度的图片时，其他人的图片也要有 45 度的图片。最后图片采集的工具、拍摄环境应该尽可能一致。由于在测试过程中缺乏专业的拍摄器材，因此作者只能选用相机进行拍摄。下面是具体的采集过程。

1. 随机选择实验室 6 个人进行人脸采集。为了保证图片的质量，本文采用 SONY 的相机进行拍摄。同时为了保证图片来自不同的角度。作者在拍摄过程中，要求每个拍摄者做上下左右的动作。通过相机快照功能，拍下不同人的角度的图片。

2. 应用图像采集模块的人脸提取功能提取采集后图像的人脸，选择符合多样性和均匀性原则的图片。



图 8-2 人脸图片

3. 注册人脸信息。应用图像采集模块的人脸信息库管理功能将人脸信息输入到数据库中。具体步骤如图 8-3 所示：点击图像采集模块的人脸库管理查看按钮，弹出人脸管理库界面，点击插入按钮，系统弹出人脸信息框，用户输入 ID（身份唯一标志）、姓名、所在地以及图像存放目录。



图 8-3 注册人脸信息过程


另外值得一提的是图像采集模块还提供了人脸采集功能，此功能可选择摄像头作为辅助设备，但是由于摄像头的拍照功能比较弱，拍摄效果较差，采集的图像不清晰，因此不建议使用摄像头进行图像采集。但是在器材缺乏的情况下，使用摄像头也不失为一个选择。以下只是介绍该功能的使用。首先开启摄像头，然后点击新建对象，系统弹出人脸信息框，用户输入 ID（身份唯一标志）、姓名、所在地以及图像存放目录（系统默认的情况下根据该用户 ID 新建一个目录）。然后点击图片 ，拍摄下一张图片，其拍摄效果如图 8-4 所示，最后保存人脸图片。另外用户在拍摄过程中，还可以通过右键删除照片。



图 8-4 采集过程

(3) 配置系统文件

表 8-3 监控端配置文件信息

名称	意义
BufferFolder	临时文件的存放目录（E:\\temp）
Place	监控端所在地点（实验室）
Server	数据库的 IP 地址（本机：127.0.0.1）
Camera	摄像头的属性（出去还是进来）

表 8-4 服务端配置文件信息

名称	意义
BufferFolder	临时文件的存放目录（E:\\temp）
Database	训练文件存放目录（E:\\database）

（4）训练人脸库图片

中心处理程序在启动时，首先检测是否存在训练文件，如果不存在，系统提示缺少训练文件的信息，停止进行人脸识别处理。因此在系统启动前，需要对图片进行训练。如图 6-3 是中心处理模块界面，选择文件菜单的训练按钮，系统将会根据不同的地区训练不同的照片，并将训练文件存放在不同的目录中，最后系统提示训练完成。



图 8-5 中心处理模块界面

8.2.2 测试用例

监控端在启动时需要建立与中心数据库的连接，在摄像头开启或视频播放后还要实时检测人脸，对检测后的人脸进行跟踪，定期还要添加识别任务。服务端需要处理客户端的连接，并需要根据客户端的请求回复消息，另外服务端还要定期处理识别任务。因此，在对考勤系统的测试中，我们为这些需求分别设计了测试用例，用来测试整个系统的性能。

a. 监控程序/中心数据库连接测试

该用例主要测试监控程序和中心数据库的连接响应速度

测试步骤：

1. 启动 4 个监控程序
2. 同时启动与中心数据库的连接
3. 计算每个程序成功连接中心数据库的时间

连接成功后，中断连接，重复步骤 1，2，3

b. 监控程序向数据库添加识别任务测试

该用例主要测试监控程序向中心数据库添加识别任务的时间

测试步骤：

1. 启动 4 个监控程序，它们都与中心数据库成功建立连接
2. 向这 4 个监控程序导入同一个视频文件
3. 同时点击播放
4. 计算监控程序在每一帧检测到人脸后向数据库添加识别任务的时间

c. 监控程序人脸检测测试

该用例主要测试监控程序进行人脸检测所需时间、正确率以及漏检率

测试步骤：

1. 启动 1 个监控程序，它与中心数据库成功建立连接
2. 向监控程序导入视频文件，同时点击播放
3. 计算监控程序在每一帧检测到人脸的时间以及数目
4. 向监控程序导入另外一个视频文件，重复 1-3 过程

d. 监控程序人脸跟踪测试

该用例主要测试监控程序进行人脸跟踪所需时间和正确率

1. 启动 1 个监控程序，它与中心数据库成功建立连接
2. 向监控程序导入视频文件，同时点击播放
3. 计算监控程序在检测到人脸后进行人脸跟踪所需的时间以及准确跟踪目标的数目
4. 向监控程序导入另外一个视频文件，重复 1-3 过程

e. 服务端识别任务测试

该用例主要测试服务端处理识别任务所需时间以及识别的准确率

测试步骤:

1. 启动服务端，它与中心数据库成功建立连接
2. 中心处理程序同时处理 10 个、50 个识别任务
3. 计算每条识别任务处理的时间以及识别结果

f. 客户端/服务端序查询测试

该用例主要测试客户端与服务端请求/回复查询所需时间

测试步骤:

1. 启动 6 个客户端和服务端
2. 客户端与服务端成功建立连接
3. 服务端在分别处理 0 个、10 个、50 个识别任务时，所有客户端同时发送查询请求
4. 计算每个客户端从请求查询到显示查询结果的时间

8.2.3 测试结果分析

表 8-5 测试结果

测试用例 ID	测试用例	测试结果
A	监控程序/数据库连接测试	3000ms
B	监控程序向数据库添加识别任务 测试	84ms
C	监控程序人脸检测测试	平均时间: 78ms/帧, 检测 正 确 率 :100%, 漏 检 率 : 43.1%
D	监控程序人脸跟踪测试	平均时间:1125ms/帧, 跟踪 正 确 率 : 90%
E	服务端处理识别任务测试	平均时间: 578ms, 识别正 确率: 61%

F	客户端/服务端查询测试	服务端在处理 0 个识别任务时， 查询回复平均时间为 109ms。 服务端在处理 10 个识别任务时， 查询回复平均时间为 2107ms。 服务端在处理 30 个识别任务时， 查询回复平均时间为 6078ms ms。
---	-------------	--

见表 8-5 的测试数据， 监控程序/数据库连接时间平均为 3000ms。这个连接时间比较长，主要是监控程序与数据库的远程连接需要经过服务器防火墙的许可和数据库访问权限的验证。一旦建立连接后，监控程序向数据库添加任务的时间就会比较短。由测试用例 b 的测试数据可以看出，监控程序向数据库添加识别任务的平均时间仅为 84ms。

监控程序进行人脸检测平均时间为 78ms/帧，检测正确率为 100%，漏检率为 43.1%。从实验数据可以看出，监控程序漏检的情况比较严重。由测试数据集可得，漏检的原因主要有以下几种：一种是人脸的几何特征因为光照等原因而无法突出，所以无法检测出来；一种是侧脸如图 8-6，由于检测算法是使用矩形特征来描述人脸特征。人脸的摆动各种各样，在训练集里难以涵盖各种摆情况，因此无法检测到部分的侧脸；一种是人脸的大小小于 20*20。检测算法检测的人脸最小窗口为 20*20，低于这个范围的人脸无法检测；一种是人脸由于光照的原因，肤色特征变得不明显，被肤色检测算法丢弃。

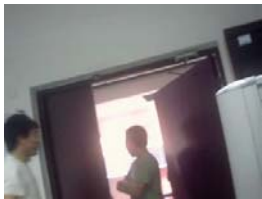


图 8-6 侧脸照片

监控程序在检测人脸后,在下一帧对目标对象进行跟踪,跟踪算法平均处理时间为 1125ms/帧。随着目标对象的增多,跟踪算法处理时间越来越长。这主要是因为获得目标对象的跟踪结果和检测结果后,要对每一个检测对象和所有跟踪对象进行目标确认,最后还要对每一个检测对象重新建立跟踪。另外跟踪的正确率为 90%。这主要是因为人脸相互遮挡的情形下,误跟踪情况比较严重。应用跟踪算法后,识别任务的数量大大减小。例如有一个视频其检测的人脸数目总共为 65 个,应用跟踪算法后,人脸的数目为 15 个。另外一个视频检测的人脸数目为 20 个,应用跟踪算法后,人脸的数目为 3 个。可见,跟踪算法的好坏直接影响到整个系统识别任务处理的效率。

服务端处理识别任务的平均时间为 578ms,这个主要与识别算法本身时间复杂度较高有关。识别正确率仅为 61%,识别率较低。这主要是因为人脸检测后的图像质量由于受到拍摄设备、场景以及运动失真的影响,人脸变得模糊,容易产生识别错误。图 8-7 是实验室 A 同学采集的人脸照片。图 8-8、8-9 是实验室 B 和 C 同学的图片。图 8-10 分别为 B、C 同学的测试图片,他们都误识别为 A。图 8-11 为 B、C 同学的另一组测试图片,他们都可以正确识别为本人。可见人脸图片的质量越好,识别率就越高。测试图片的质量严重影响识别的好坏。然而在实际应用中,硬件设备以及人体运动导致的图片失真都是无法控制的问题,因此在下一步,应该着重研究图片的恢复问题。



图 8-7 实验室 A 同学照片



图 8-8 实验室 B 同学照片



图 8-9 实验室 C 同学照片



图 8-10 错误识别图片



图 8-11 正确识别图片

8.3 本章小结

本章首先介绍了系统的开发工具与运行环境，针对系统的测试环境设计了测试用例。最后通过对测试结果的分析，测试系统的性能指标是否符合系统需求设计，同时对系统存在的问题进行分析，得出下一步研究的重点。

第九章 总结与展望

总结

本文介绍的分布式人脸检索系统来源对视频智能化技术的研究,作者希望提出一种视频整合的方案,提高视频资源的利用率,为日后数据的挖掘提供信息平台。作者参与整个分布式人脸检索平台的设计到具体应用的设计与实现过程。目前该应用系统虽然能够满足基本设计要求,但是离实际应用还有一段距离。

本人的主要工作总结如下:

- (1) 实现目前主流的几种人脸检测、人脸跟踪、人脸识别算法,并根据实验结果进行分析和比较,最后得出基于 Adaboost 人脸检测、Ensemble track 的人脸跟踪算法以及基于稀疏表示的人脸识别算法比较能够满足系统的准确性和实时性的要求。
- (2) 由于基于 Adaboost 的人脸检测算法在实际应用中存在误测情况,Ensemble track 原本只是为单目标跟踪设计的算法,因此本人结合肤色模型改善人脸检测误测的情况,对 Ensemble 的算法流程也进行修改使其满足多目标跟踪的要求。
- (3) 提出了分布式人脸检索系统的设计方案,并根据该方案设计和实现具体应用系统-考勤系统。
- (4) 为了提高数据处理的速度,作者根据 MapReduce 分布式程序设计模型设计分布式人脸检索系统的分布式系统,采用多线程池技术提高考勤系统任务的处理速度。
- (5) 提供人脸库从采集到人脸标准化完整实施方案。
- (6) 根据客户端和服务端请求和处理的方式,选择点对点的通信模型。
- (7) 对考勤系统进行测试,验证系统的性能

本文最后设计了测试用例,对系统进行了测试,根据测试数据和测试结果,可以看出系统人脸检测的正确率基本上达到预期目标,跟踪效果勉强达到实时和正确性要求,识别率的速度和正确率基本上满足系统设计要求。

展望

本论文所介绍的考勤系统的实现除了验证分布式人脸检索系统方案设计的可行性外,还希望能够满足具体应用环境的需要,尽可能发挥它的性能,以期其设计与实现的整个过程能给同类应用和研究带来一些有用的积累和值得借鉴的经验。

同时,在实际应用过程中,随着系统规模的扩大,系统在对大容量通信的支持方面存在不足,在考勤系统的响应时间等方面仍然需要进一步改进。另外系统的扩大还进一步影响了识别任务的处理,目前识别任务主要由服务器的线程组完成,然而这并不能满足大规模数据的处理速度要求,因此识别任务的处理日后要扩展到其他机器共同完成。另外识别任务主要由监控端提交到中心服务器的数据库,当监控端越来越多时,数据库的并发处理等要求也越来越高,数据库的处理性能也需要进一步优化。另外在多目标跟踪算法的实现上,虽然目前国内外提出的很多种跟踪算法,然而这些算法要么无法实现多目标跟踪,要么计算复杂,无法满足实时要求。

因此,今后的工作和研究将主要集中在两方面:

- 优化服务端的消息处理和线程管理的方式、扩大对大容量通信、任务和并发处理的支持等方面,对消息中间件的性能进行进一步的改进、完善和提高。
- 进一步加深对多目标跟踪算法的研究,希望能够提出既能够满足实时要求,又能够正确分割目标的跟踪算法。

参考文献

- [1] 刘富强, 卢赤班, 数字视频监控系统及其应用, 工矿自动化, 3(31): 2003
- [2] R. Collins ,A. Lipton, Introduction to the Special Section on Video Surveillance. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):745-766, 2000
- [3] Haritao glu I. ,Harwood D. and Davis L. W4:real-time surveillance of people and their activities,IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8): 809-830,2000
- [4] Papageorgiou, C. , M. Oren, and T. Poggio,A General Framework for Object Detection, International Conference on Computer Vision, 555-562, 1998
- [5] R L Hsu, M Abdel-Mottaleb, A K Jain .Face detection in color images.IEEE Transactions on Pattern Analysis and Machine Intelligence, 696-706, 2002
- [6] Y. T. Pai, S.J.Ruan, M.C.Shie and Y.C.Liu,A Simple and Accurate Color Face Detection Algorithm in Complex Background,IEEE International Conference on Multimedia and Expo, 1545-1548,2006
- [7] Yullie A, Hallinan P .Feature exaction from faces using deformable templates. International Journal of Computer Vision, 8(2): 99-111,1992
- [8] Paul Viola, Michael Jones ,Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 511-518, 2001
- [9] W. Kienzle, G. Bakir, M. Franz and B. Scholkopf,Face Detection-Efficient and Rank Deficient,Advances in Neural Information Processing Systems,673-680, 2005
- [10] 艾海舟,王栓,何克忠,基于差分图像的人脸检测,中国图形图像学报,3(12), 987-992,1998
- [11] P. Anandan,J. Bergen,K.Hanna and R.Hingorani,Hierarchical Model-baded Motion Estimation, Motion Analysis and Image Sequence Processing , 237-252 ,1993
- [12] Jebara T S, Pentland A ., Parameterized structure from motion for 3D adaptive feedback tracking of faces, IEEE Conference on Computer Vision and Pattern, 1997

- [13] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Process*,50 (2): 174–188, 2002
- [14] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,27 (11): 1805–1819, 2005
- [15] Z. Khan, T. Balch, F. Dellaert, MCMC data association and sparse factorization updating for real time multi-target tracking with merged and multiple measurements, *IEEE Transactions on Pattern Analysis and Machine*
- [16] D. Comaniciu, V.Ramesh, P. Meer, Kernel-based object tracking, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,25 (5): 564–577, 2003
- [17] Bradski G. , Computer vision face tracking for use in a perceptual user interface,*Interl.Technology Journal*,2(2):12-21,1998
- [18] Shai Avidan, Ensemble Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):261-271, 2007
- [19] W. Bledsoe,Man-machine facial recognition,Panoramic Research Inc,Palo Alto,CA, RepPRI:22, 1966
- [20] Brunelli R. and Poggio T., Face recognition: features versus templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15 (10):1042 - 1052, 1993
- [21] A. L. Yuille, P. W. Hallinan and D. S. Cohen, Feature extraction from faces using deformable templates, *International journal of computer vision* ,8(2): 99 – 111,1992
- [22] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,24 (5): 603–619, 2002
- [23] Levi K., Weiss Y., Learning Object Detection from a Small Number of Examples: The Importance of Good Features, *Computer Vision and Pattern Recognition*, 53-60, 2004.
- [24] P. N. Belhumeur, J.P.Hespanha and D.J. Kriegman, Eigenfaces vs. fisherfaces:

- recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 711-720, 1997.
- [25] X. He and S. Yan et al., Face recognition using laplacianfaces, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (3): 328–340, 2005.
- [26] J. A. K. Suykens, T. Van Gestel, A Support Vector Machine Formulation to PCA Analysis and Its Kernel Version, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2): 447-450, 2003.
- [27] Dean J. , Ghemawat S. MapReduce: MapReduce: Simplified Data Processing on Large Clusters, 2004
- [28] 曹羽中,用 Hadoop 进行分布式并行编程.
<http://www.ibm.com/developerworks/cn/opensource/os-cn-hadoop1/#N10059>
- [29] W. X. LIN, G PAN, Z. H. WU, Y. H. PAN, A Survey on Facial Features, Pattern Recognition and Machine Intelligence, 24(5):696-706, 2002
- [30] Paul Viola, Michael Jones, Robust Real-Time Face Detection, International Journal of Computer Vision, 57(2):137–154,2004.
- [31] Remagnino P, Tan T, Baker K. Multi-agent visual surveillance of dynamic scenes, Image and Vision Computing, 16(8): 529-532, 1998
- [32] Jinfeng Yang, Zhouyu Fu, Tieniu Tan and Weiming Hu, Skin Color Detection Using Multiple Cues, International Conference on Pattern Recognition , 632-635,2004
- [33] Yang G.Z, Huang T.S., Human face detection in a complex background, Pattern Recognition, 27(1).53-63, 1994
- [34] Yullie A, Hallinan P .Feature extraction from faces using deformable templates. International Journal of Computer Vision, 8(2): 99-111,1992
- [35] S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking, IEEE Transactions on Signal Process, 50 (2): 174–188, 2002
- [36] Huang C.L, Chen C W, Human facial feature extraction for face interpretation and recognition, Pattern Recognition, 25(12):1435-1444, 1992
- [37] B. Georgescu, I. Shimshoni, P. Meer, Mean shift based clustering in high dimensions:

- a texture classification example, Computer Vision and Pattern Recognition, 456–463, 2003
- [38] R. Collins, Mean shift blob tracking through scale space, Computer Vision and Pattern Recognition, 234–240, 2003
- [39] W. Bledsoe, Man-machine facial recognition, Panoramic Research Inc, Palo Alto, CA, RepPRI:22, 1966
- [40] A. J. O'Toole, K. A. Deffenbacher, H. Abdi and J. C. Bartlett, Classifying faces by race and sex using an autoassociative memory trained for recognition, Cognition Science Society, 847-851, 1991

致谢

首先，衷心感谢我的导师朝红阳教授能够在忙碌的教学工作中挤出时间来审查、修改我的论文，其次还要感谢她在学术方面给予我悉心的指导。另外我要特别感谢伍沛然同学，与他在学术上的探讨和交流，使得我对论文的有更深入的了解。同时我还要感谢实验室的段晓华师兄、蓝振杰、杨铭、高鲁鑫等同学，他们给予我工作积极的配合和无私的帮助，使我的工作能顺利完成。最后我要感谢我的家人在精神上对我的鼓舞与激励。

