## 3.3关键技术

### 3.3.1 人脸信息比对

本项目采用三种模型集成测试的思路，确保人脸识别的精确性。

1、首先采用dlib模型和感知hash模型进行人脸比对。使用dlib中提供的人脸检测方法（使用HOG特征或卷积神经网方法），并使用提供的深度残差网络（ResNet）实现实时人脸识别，不过本项目的目的不是构建深度残差网络，而是利用已经训练好的模型进行实时人脸识别，实时性要求一秒钟达到10帧以上的速率，并且保证不错的精度。

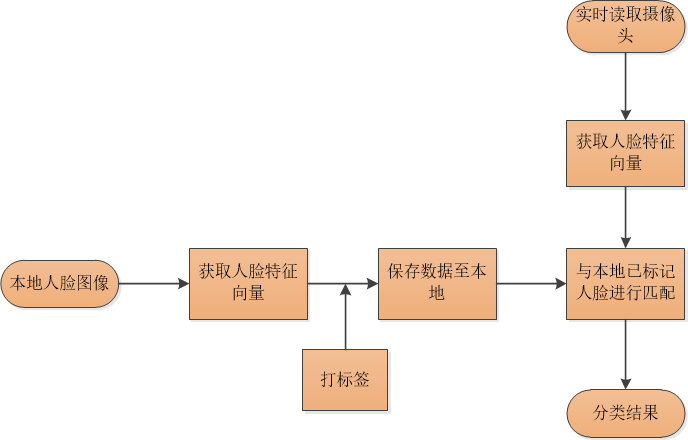
人脸识别分为人脸检测和识别两个阶段，人脸检测会找到人脸区域的矩形窗口，识别则通过ResNet返回人脸特征向量，并进行匹配。

1. 人脸检测阶段。人脸检测算法需要用大小位置不同的窗口在图像中进行滑动，然后判断窗口中是否存在人脸。dlib中使用的是HOG（histogram of oriented gradient）+ 回归树的方法，使用dlib训练好的模型进行检测效果要好很多。dlib也使用了卷积神经网络来进行人脸检测，效果好于HOG的集成学习方法。
2. 识别阶段。识别也就是我们常说的“分类”，摄像头采集到这个人脸时，让机器判断是张三还是其他人。分类分为两个部分：

• 特征向量抽取。本文用到的是dlib中已经训练好的ResNet模型的接口，此接口会返回一个128维的人脸特征向量。

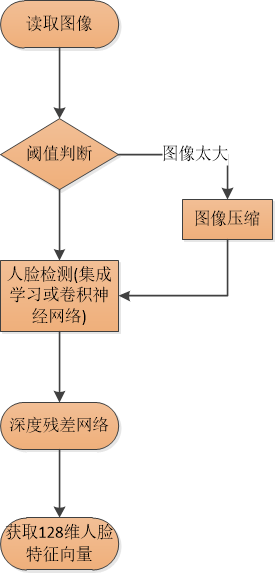
• 距离匹配。在获取特征向量之后可以使用欧式距离和本地的人脸特征向量进行匹配，使用最近邻分类器返回样本的标签。

根据上述内容，识别的大致过程如图1所示：



**图1.人脸识别分类过程**

对于图1中的获取人脸特征向量，其过程如图2所示：



**图2.获取人脸特征向量过程**

1. 采用百度api人脸比对，将摄像头所拍摄的照片与数据库中黑名单人员进行人脸比对，缩短检测运行时间。

• 两张人脸图片相似度对比：比对两张图片中人脸的相似度，并返回相似度分值；

• 多种图片类型：支持生活照、证件照、身份证芯片照、带网纹照四种类型的人脸对比；

• 活体检测控制：基于图片中的破绽分析，判断其中的人脸是否为二次翻拍（举例：如用户A用手机拍摄了一张包含人脸的图片一，用户B翻拍了图片一得到了图片二，并用图片二伪造成用户A去进行识别操作，这种情况普遍发生在金融开户、实名认证等环节。）；

• 质量检测控制：分析图片的中人脸的模糊度、角度、光照强度等特征，判断图片质量；

### 3.3.2地图路径规划

路径规划API是一套以HTTP形式提供的步行、公交、驾车查询及行驶距离计算接口，返回JSON 或 XML格式的查询数据，用于实现路径规划功能的开发。 由于道路/数据/算法的变更，很可能存在间隔一段时间后请求相同起终点的经纬度返回不同结果。

第一步，申请”Web服务API”密钥（Key）；

第二步，拼接HTTP请求URL，第一步申请的Key需作为必填参数一同发送；

第三步，接收HTTP请求返回的数据（JSON或XML格式），解析数据。

如无特殊声明，接口的输入参数和输出数据编码全部统一为UTF-8。

针对步行、公交、驾车这三种路径规划的接口，实现批量请求，使用批量请求接口（https://lbs.amap.com/api/webservice/guide/api/batchrequest）。

### 3.3.3摄像头调用

由于本项目需要将网站部署至云服务器上，所以此处摄像头的的调用不能直接使用eclipse中的jsf依赖包，而是需要使用webcam-capture访问网络摄像头，通过用户所使用的网络摄像头来实现在线拍照功能。

webcam-capture可以通过JavaScript直接使用小型API访问网络摄像头- 或者更确切地说是jQuery。因此，可以将图像置于Canvas（回调模式），将图像存储在服务器上（保存模式），并在Canvas（流模式）上流式传输Flash元素的实时图像。

### 3.3.4 ECharts可视化图表

ECharts一款基于HTML5的图形库，图形种类丰富，风格搭配美观。

引用ECharts首先需要下载ECharts.js文件。ECharts代码的编写基于JavaScript,本项目通过ajax异步请求获取数据库数据作为可视化图表的数据来源，从而创建可视化图表统计医院一些相关数据在某一段时间内的变化，同时当数据库中数据更新的时候，图表中数据也能随之动态变化。

### 3.3.5网站部署

本项目将javaweb端配置在华为云的云服务器ubuntu16.04系统下，同时申请域名uscrzwj.cn将域名与公网ip地址绑定，实现网站的域名访问，构造安全链接https形式。将后代源代码通过Eclipse打包为war形式上传至tomcat8服务器环境下的webapps文件夹中，实现0.0.0.0:8080端口可直接访问网站。其中所需要涉及的环境包括但不限于：JDK8环境安装，mysql数据库安装，python相关版本依赖包以及工具安装。

### 3.3.6真实数据爬取

本项目所有医生数据均由某医院信息库爬取得到，其中所用到的技术点包含但不限于：

• 浏览器分析ajax请求

• requests模拟请求

• pyquery解析

• pymysql存储数据库

• Pool进程池多线程爬取

开始爬取前安装好相应的依赖包，确定抓取目标的地址，选取所需要爬取的信息。首先抓取第一页的内容。我们实现了get\_one\_page()方法，并给它传入参数。然后将抓取的页面返回，通过main()方法调用，将所提取的结果写入文件，这里直接写入到一个文本文件中，通过JSON库的dumps()方法实现字典的序列化。最后将文本文件中的医生信息数据导入数据库，如前端代码相结合，实现前后端连接。