1. **引言**
   1. **问题描述**

充分利用已有的开源软件资源，开发一个基于人脸识别技术的点名系统，可以自己定义系统的必要功能。

* 1. **开发使用工具**

Qt：qt-opensource-windows-x86-mingw492-5.6.2

Database：MySQL

C++ library：OpenCV 3.0.0 Library

Compile tools：cmake

* 1. **参考资料**

[1] Mastering OpenCV with Practical Computer Vision Projects, Shervin Emami (www.shervinemami.info)

[2] OpenCV 编程案例详解，李宗立

1. **全局数据结构说明**

timer\_recognition，类型：QTimer \*，用途：控制多次进行人脸识别的计时器

videoCapture，类型：cv::VideoCapture，用途：捕捉摄像头的图像信息

faceCascade，类型：cv:: CascadeClassifier，用途：对人的正脸进行识别

eyeCascade1，类型：cv:: CascadeClassifier，用途：识别眼睛（未佩戴眼镜）

eyeCascade2，类型：cv:: CascadeClassifier，用途：识别眼睛（佩戴眼镜）

m\_numPersons，类型：int，用途：保存已经训练好模型的学生数

m\_selectedPerson，类型：int，用途：保存正在训练模型的学生id值

1. **模块设计**
   1. **模块算法实现说明**

**3.1.1人脸检测**

对于本系统来说，检测的是正面的人脸，因此可使用LBP人脸检测器，它的识别速度较快。通过指定xml文件的文件名，就可加载LBP检测器。LBP检测器用来检测摄像机每帧中的人脸，但首先需专门针对人脸检测来预处理摄像机图像，步骤如下：

1) 灰度色彩转换：人脸检测只适用于灰度图像。因此，应转换彩色摄像机图像为灰度图像。

2) 收缩摄像机图像：人脸检测的速度取决于输人图像的大小(对大的图像很慢，而对小图像很快)，即使在低分辨率下，检测效果依然十分可靠。所以应该缩小摄像机图像，使其有一个比较合理的尺寸。

3) 直方图均衡化：在光线不足的情况下，人脸检测器并不可靠。因此，应该对图像进行直方图均衡化，来改善对比度和亮度。

如果用一个缩小图像进行人脸检测，得到的结果也是缩小的，因此要想得到人脸在原始图像中的区域，就需要放大此结果，还需要确定人脸在图像中的整个边界。

探测到人脸之后，要根据LBP检测器得到的信息在脸部区域周围画出矩形

框。与此同时，在眼睛周围画出圆圈表示。

**3.1.2人脸预处理**

人脸识别极易受光照条件、脸部朝向、面部表情等因素的影响，因此，尽量消除这些差异非常重要。否则在同样条件下，人脸识别算法经常会认为两个不同人脸比同一个人的两张脸更相似。人脸预处理的最简单形式就是使用equalizeHist()函数来做直方图均衡。但在现实环境中，为了让检测算法更可靠需要很多复杂的技术，包括面部特征检测(例如：检测眼睛、鼻子、嘴巴和眉毛)。

对于正面人脸而言，虽然面部表情、光照条件、摄像机属性、与相机的距离等都会变化，但总可假定人眼是水平的，并对称分部在人脸上。当人脸检测器将别的对象当成人脸时，可通过眼部检测来丢弃这种误判，这是一种很有效的方法。人脸检测器和眼部检测器基本上不会同时出错。

可将检测到的人脸和眼睛结合起来进行人脸预处理，其涉及的操作包括：

1. 几何变换和裁剪：这个过程包括缩放、旋转和平移图像，以使眼睛能被对齐，然后从人脸图像中删除额头、下巴、耳朵和背景。

2．对人脸左侧和右侧分别用直方图均衡：此过程会分别标准化左和右两侧的亮度和对比度。

3．平滑：使用双边滤波器来减少图像噪声。

4．椭圆掩码：该椭圆掩码会将一些剩余头发和人脸图像背景去掉。

预处理使得在不同亮度、人脸图像旋转、拍摄角度、背景、光照位置等条件下得到的图像用于人脸识别会更加一致。通过预处理的人脸图像将作为人脸识别阶段的输入，这样会在收集用于训练的人脸图像的同时，也可识别人脸。

**3.1.3收集并训练人脸**

收集人脸就是把从摄像机得到的人脸图像通过预处理后放到相应的数组中，也将其类标签放到类标签数组中，类标签用来指定拍摄的图像是哪个人的人脸。

一个好的数据集应包含人脸变化的各种情形。为了确保训练集中人脸不要太相似，应该在收集人脸图像之问有足够的延迟。例如，若摄像机每秒有30帧，则在短短几秒钟内就可收集100张人脸图像，这种情况下人还没有移动。因此，每秒采集一张人脸图像，或者只收集与前面差别较大的人脸图像。

人脸识别是一个有监督学习过程，首先利用训练集构造一个人脸模型，然后将测试集与训练集进行匹配，找到与之对应的训练集头像。在训练模型时需要首先进行降维，降维的一个方法即是PCA（主成分分析法）。

PCA算法的简单流程：

1）特征中心化，即每一维的数据都减去该维的均值；

2）（对维度）计算协方差矩阵；

3）计算协方差矩阵的特征值和特征向量；

4）选取最大的前几个特征值对应的特征向量，组成一个投影矩阵。

**3.1.4人脸识别**

为了验证预测是否可靠、或者说系统是否能对一个不认识的人进行正确地识别，这需要进行人脸验证，以此获得一个置信度，该置信度会说明用户想要找的人与某个人脸图像的相似度。在检测过程中，可以将输入图像与已有的多个人脸模型进行比较，找出置信度超过阈值的模型作为预测模型，输出其标签值。

**3.1.5人脸验证点名**

根据id提取学生信息。

**4. 数据库设计**

采用MySQL数据库，在数据库face\_recognition中建立表students。

表students结构如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 含义 |
| id | int | 学生的唯一模型ID |
| students\_name | Varchar(30) | 学生的姓名 |
| students\_id | Varchar(30) | 学生的学号 |
| attend\_class | Varchar(10) | 学生是否出勤 |

模型ID从0开始逐步增长。

attend\_class有unknow和yes两种状态，分别表示暂未签到和签到成功。

1. **UI界面设计**

整个系统的主窗口：

