人脸面部表情识别设计文档

**课程名称： 软件工程课程设计**

**指导老师： 李毅**

**项目名称： 人脸面部表情识别**

**项目组长：**

**项目成员：**

**2019年11月4日星期一**

目录

[人脸面部表情识别设计文档 1](#_Toc23760662)

[一、 引言 3](#_Toc23760663)

[1.1编写目的 3](#_Toc23760664)

[1.2定义 3](#_Toc23760665)

[1.3参考资料 3](#_Toc23760666)

[二、 任务概述 5](#_Toc23760667)

[2.1 系统功能概述 5](#_Toc23760668)

[2.2 运行环境 5](#_Toc23760669)

[2.3 开发工具 5](#_Toc23760670)

[2.4 条件与限制 5](#_Toc23760671)

[2.4.1技术环境约束 5](#_Toc23760672)

[2.4.2业务环境约束 6](#_Toc23760673)

[三、 总体设计 7](#_Toc23760674)

[3.1总体架构 7](#_Toc23760675)

[3.2系统主要功能和模块设计 8](#_Toc23760676)

[四、 数据设计 12](#_Toc23760677)

[4.1 数据结构（数据表、数据库）设计 12](#_Toc23760678)

[4.2 OO-类的设计 12](#_Toc23760679)

[4.3 物理结构设计 15](#_Toc23760680)

[4.4 数据结构与程序的关系 15](#_Toc23760681)

[五、 接口设计（用户界面） 16](#_Toc23760682)

# 一、 引言

## 1.1编写目的

本软件计划开发出一个基于计算机视觉，通过上传照片、摄像头捕捉等途径进行识别人脸面部表情的程序。

## 1.2定义

在该概要设计说明书中的专门术语有：

**人脸识别**：人脸识别，是基于人的脸部特征信息进行身份识别的一种生物识别技术。用摄像机或摄像头采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行脸部识别的一系列相关技术，通常也叫做人像识别、面部识别[1]。

**识别面部表情**：通过对五官的位置、人体轮廓、不同角度的人脸等描述脸部情况和属性的识别，来进行相应的表情判断，如愤怒、恐惧、高兴、伤心、惊讶、无情绪等[2]。

## 1.3参考资料

[1] 《人脸识别辨身份 未来支付请“刷脸”》 人民网[引用日期2014-03-28]

[2] 《马航事件驱动 人脸识别概念或受关注》 新华网[引用日期2014-03-28]

# 二、 任务概述

## 2.1 系统功能概述

我们打算通过实现人脸面部表情识别技术（Face Recognition）和检测面部关键点，开发出一个能够实时在线地识别人脸面部表情的程序。

对于人脸的识别的关键点在于对五官的位置、人体轮廓、不同角度的人脸等描述脸部情况和属性，来进行相应的表情判断，如愤怒、恐惧、高兴、伤心、惊讶、无情绪等，此项目可运用于不同场景，帮助人们对当事人进行心理情绪判断。

## 2.2 运行环境

我们的程序计划在64位Windows10操作系统上运行。

## 2.3 开发工具

开发软件：基于64位Windows系统下的PyCharm和VS code

开发硬件：PC、附带摄像头或相机

## 2.4 条件与限制

### 2.4.1技术环境约束

在进行开发时，脸部识别主要由判断人脸位置和相应的五官位置来进行标准化设计，但是由于拍照角度不同，每个人的五官差别非常大，而且在某些极端情况下表情管理失控会造成五官扭曲从而难以判断相应情绪。由于很难每次都找到类似于证件照的标准人脸照片，所以需要软件的判断力足够应付大多数情况，这对于现在我们的开发能力是一个很大的挑战。

### 2.4.2业务环境约束

开发时间有限，需要在较短时间内做出一定成果，颇为困难。

由于各种现实因素的影响，本软件开发过程可能还会遇到各种困难，这些都是隐性存在的约束。

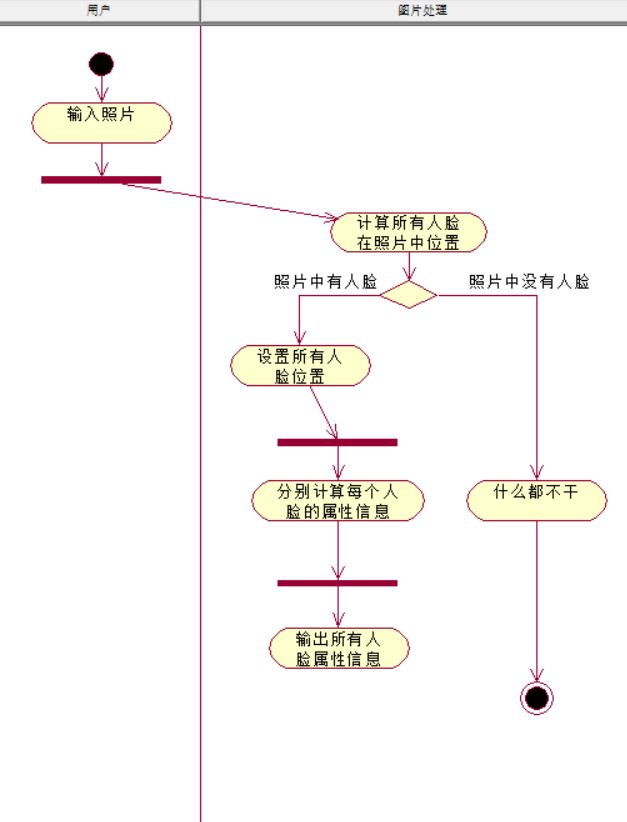
# 三、 总体设计

## 3.1总体架构

系统总体结构图

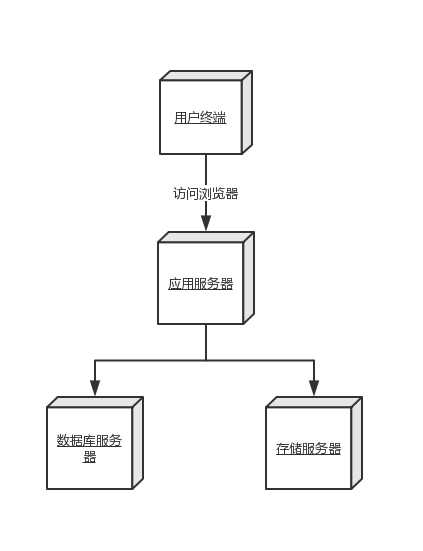
主要步骤有：输入照片、处理图片、计算人脸位置、进行判断、输出信息

总体结构可以通过活动图显示：



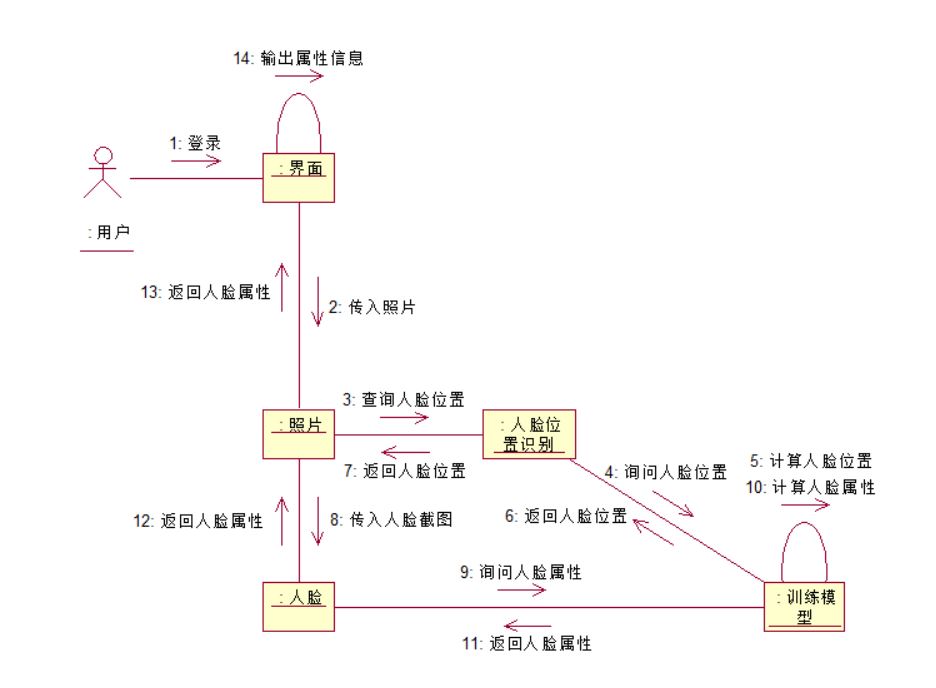
## 3.2系统主要功能和模块设计

部署图如下：

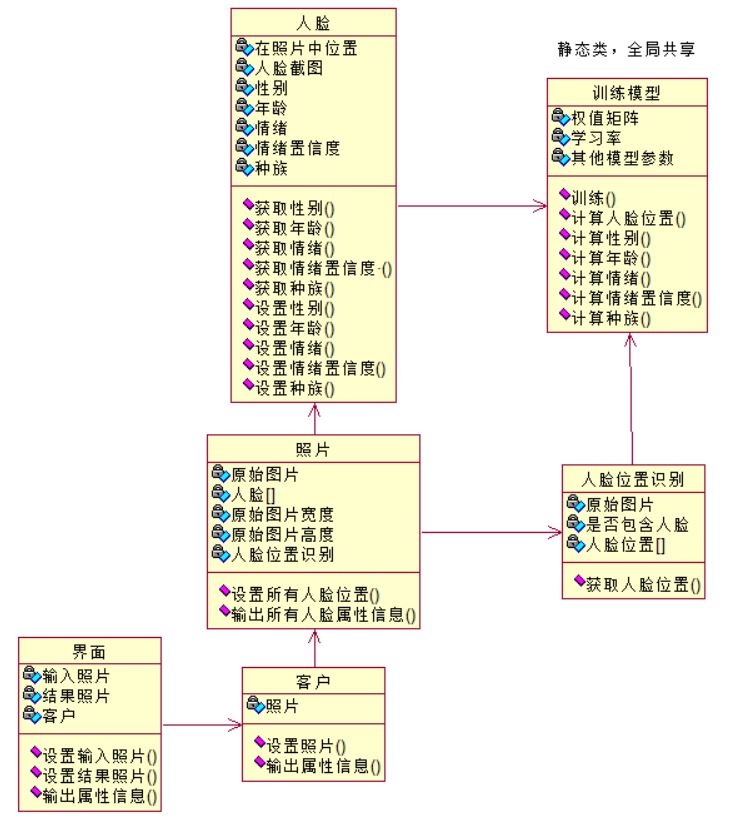


主要分为两个模块，客户端模块进行总体识别和分析，并将识别的信息传输至网络服务端模块并借助可视化界面显示在web上。

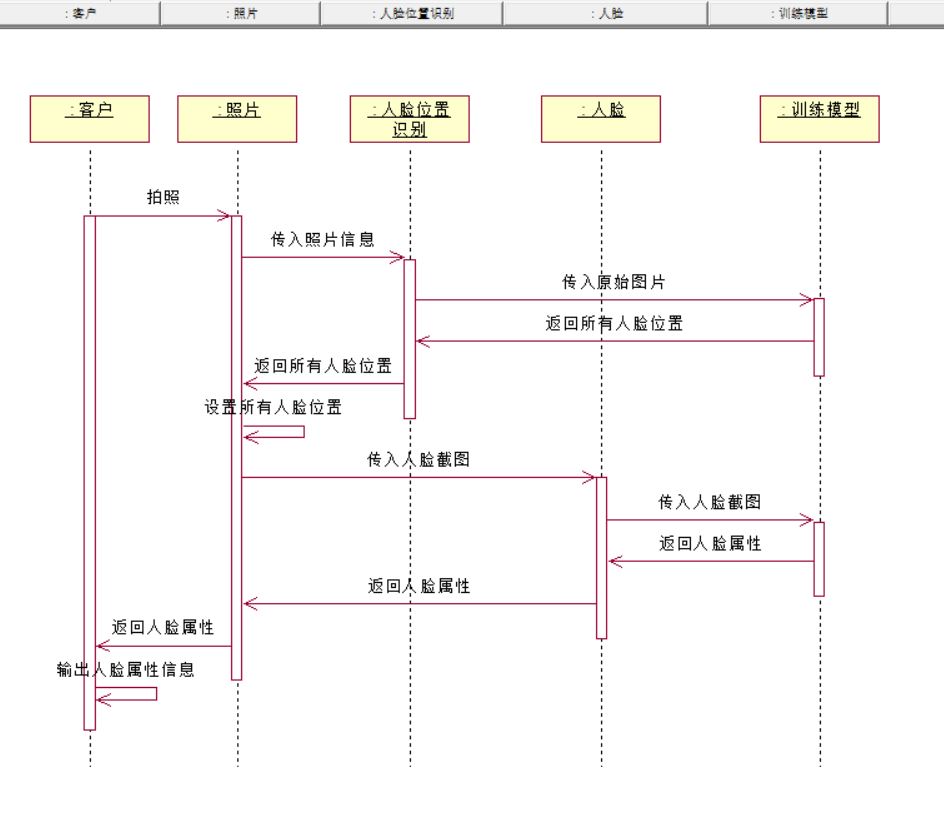
程序算法流程：



在该人脸面部表情识别系统中，我们需要至少6个类：界面类、客户类、照片类、人脸类、人脸位置识别类和一个训练模型。界面类，提供三个主要行为：输入照片、输出照片和输出属性。照片类、客户类来投影现实世界中的客观对象：要处理和要得到的照片和使用软件的用户。人脸类、人脸位置识别类分别是照片类属性域中的相关类。而人脸类中属性的获取和人脸位置识别类的功能都是基于静态类训练模型的。类图如下：



活动流中，用户输入照片，进入图片处理活动流。先计算出所有人脸在照片中的位置，进行判断，有人脸则设置所有人脸位置。若无人脸则不处理。设置所有人脸位置后，分别计算每个人脸的属性信息，最后输出所有人脸属性信息。序列图如下：



# 四、 数据设计

## 4.1 数据结构（数据表、数据库）设计

用户信息表：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 型类 |
| 1 | 用户名 | char(20) |
| 2 | 用户号 | int |
| 3 | 密码 | char(20) |

## 4.2 OO-类的设计

界面类数据成员：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **型类** | **长度** | **说明** | **备注** |
| 1 | User | 类 |  |  | 不可空 |
| 2 | Input\_Image | 字符 | 10M | 用户传入的图片 |  |
| 3 | Output\_Image | 图像 | 10M | 输出给用户的结果 |  |

界面类功能：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **返回值** | **参数表** | **说明** | **备注** |
| 1 | Set\_Input\_Image() | Boolean | Input\_Image URL |  | 不可空 |
| 2 | Set\_Output\_Image() | Boolean | Input\_Image |  |  |
| 3 | Print\_Output\_Info() | 字符串（30） |  |  |  |

用户类数据成员：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **型类** | **长度** | **说明** | **备注** |
| 1 | Photo | 类 |  |  | 不可空 |

用户类功能：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **返回值** | **参数表** | **说明** | **备注** |
| 1 | Set\_Image() | Boolean | Input\_Image URL |  | 不可空 |
| 2 | Print\_Characteristic\_Info() | 字符串（30） | Photo类 |  |  |

照片类数据成员：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **型类** | **长度** | **说明** | **备注** |
| 1 | Ori\_Image | 图像 | 10M |  | 不可空 |
| 2 | Face[] | 类数组 |  |  |  |
| 3 | Height | Int | 32bit |  |  |
| 4 | Width | Int | 32bit |  |  |

照片类功能：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **返回值** | **参数表** | **说明** | **备注** |
| 1 | Set\_Faces\_pos() | Int | Face[] | 设置Face[]的人脸截图 | 不可空 |
| 3 | Print\_All\_Info() | 字符串（30） | Face[] |  |  |

人脸类数据成员：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **型类** | **长度** | **说明** | **备注** |
| 1 | Pos | pair<int> | 64bit | 截图在照片中中点位置 | 不可空 |
| 2 | Face\_Screanshoot | 图像 | 10M | 人脸部分截图的图片 |  |
| 3 | Sex | 字符串 | （10） |  |  |
| 4 | Age | Int | 32bit | 小于100 |  |
| 5 | Emotion | 枚举类 | 20 |  |  |
| 6 | Emotion\_Score | int | 32bit | 小于100 |  |
| 7 | Ration | 枚举类 | 10 |  |  |

人脸类功能：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **返回值** | **参数表** | **说明** | **备注** |
| 1 | Get\_Sex() | 字符串 |  |  | 不可空 |
| 2 | Get\_Age() | Boolean |  |  |  |
| 3 | Get\_Emotion() | 字符串（30） |  |  |  |
| 4 | Get\_Emotion\_Score() | int | 32bit |  |  |
| 5 | Get\_Ration | 枚举类 |  |  |  |
| 6 | Set\_Sex() |  | Face\_Screanshoot |  |  |
| 7 | Set\_Age() |  | Face\_Screanshoot |  |  |
| 8 | Set\_Emotion() |  | Face\_Screanshoot |  |  |
| 9 | Set\_Emotion\_Score() |  | Face\_Screanshoot |  |  |
| 10 | Set\_Ration |  | Face\_Screanshoot |  |  |

人脸位置识别类数据成员：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **型类** | **长度** | **说明** | **备注** |
| 1 | Ori\_Image | 图像 | 10M |  | 不可空 |
| 2 | Has\_Face | Boolean | 1 bit |  |  |
| 3 | Face\_pos[] | Int[] | 1000bit |  |  |

人脸位置识别类功能：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **返回值** | **参数表** | **说明** | **备注** |
| 1 | Get\_Face\_Pos() | Int[] | Ori\_Image |  | 不可空 |

## 4.3 物理结构设计

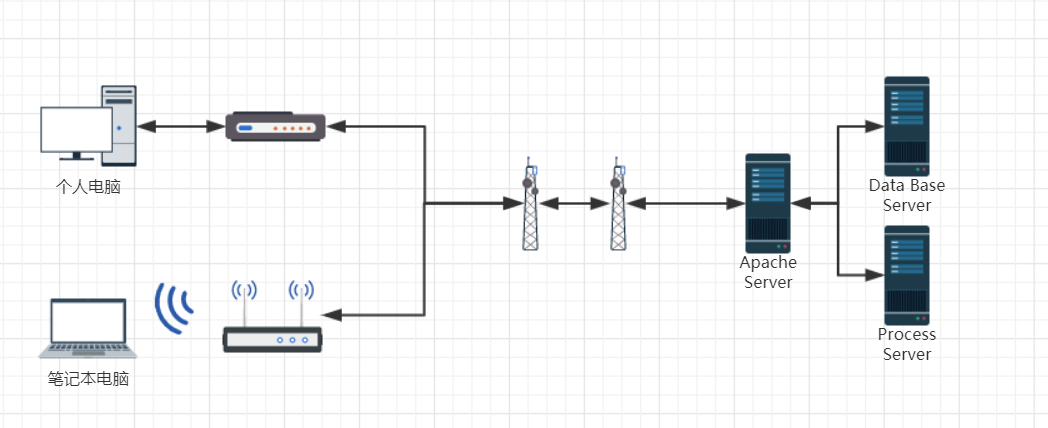
客户端：带有浏览器的PC，支持flash并且带有摄像头。

应用服务器：Apache服务器上部署面部属性识别程序，提供访问接口

数据库服务器：提供有关用户信息的访问。

处理服务器：具体实现面部表情识别的功能。

通信：服务器通过TCP socket与客户端通信。



## 4.4 数据结构与程序的关系

界面数据结构：提供交互功能，向程序输入用户传入的图片。

用户数据结构：标识用户信息，方便操作。

照片数据结构：提供人脸的多目标检测接口。

人脸数据结构：提供识别属性的接口。

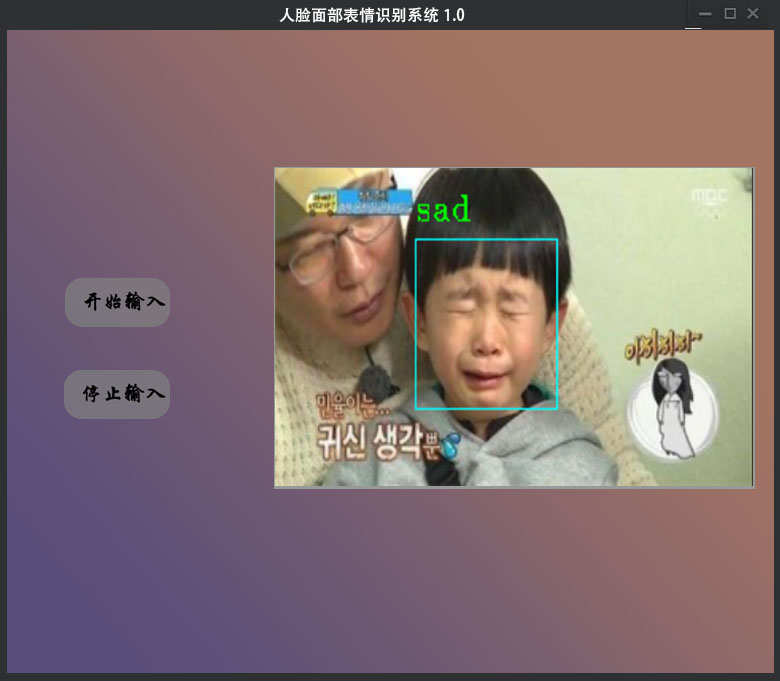
人脸位置识别数据结构：实现对于给定照片的人脸多目标检测功能。

训练模型数据结构：实现人脸面部表情属性的识别功能。

# 五、 接口设计（用户界面）



首页界面，当我们打开该人脸面部表情识别系统时，我们将看到这个界面。该界面有标题栏：人脸面部表情识别系统1.0，指示这是我们迭代版本的1.0版本。主界面是由几大元素构成，人脸面部表情识别是我们的大标题，下面有两个按钮，分别是开始与退出。点击开始，则进入下一个界面，点击退出则关闭程序。此外，我们还添加了制作人团队的成员信息（姓名），用以表示这是我们组所创建的



这是开始界面，仅有两个按钮：开始输入与停止输入。我们采用了视频流的方式，通过点击开始输入，我们将摄像头打开，输入视频，然后于右端显示视频内容，在捕捉到的人脸用框框出，左上角用鲜艳颜色的字体表示识别出来的人脸表情。在按下停止输入之后，摄像头关闭，退回到停留在最后一刻的画面。