

# 1、任务概述

## 1.1、目标

该项目设计一个人脸表情识别小程序：该程序分为三个模块

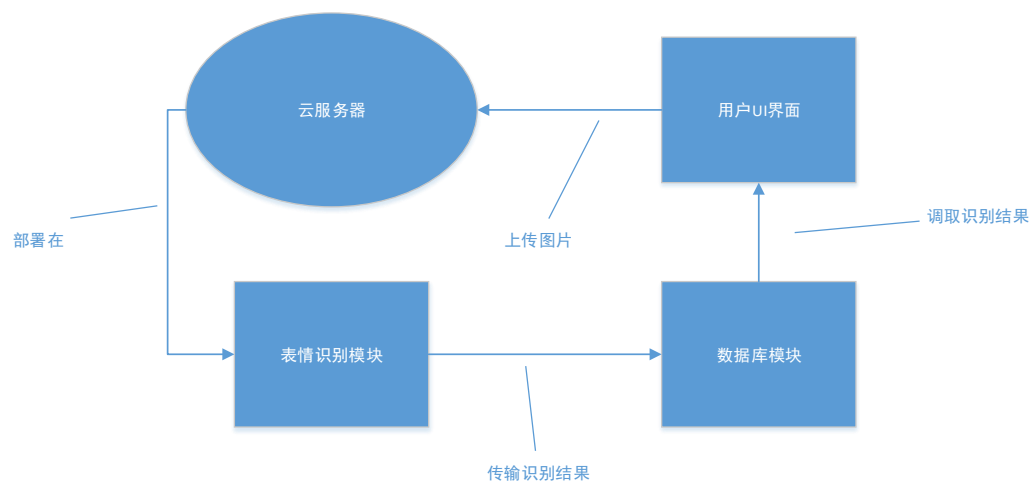
人脸识别模块：对图片进行人脸表情的识别

数据库模块：存储识别后的表情数据

用户界面：进行拍摄照片，查看表情识别的历史。

最终实现用户在小程序上可以上传照片来判断表情及当前表情推荐的音乐和文章，并且查看自己表情的历史记录图，以及诸如挑战的生气表情等的一些附加功能。

模块联系图如下：



## 1.2、系统（或用户）的特点

软件特点：

1. 免下载可以直接使用
2. 人脸表情识别模块部署在云端，小程序免于臃肿
3. 可以记录用户使用历史，可以随时调取历史数据，并且生成图表

软件预期使用频度：

1. 平均在每天两次左右，用以记录用户每日情绪
2. 特定时间，用户可以进行表情挑战，扮演出 100%属性的某一种情感。

用户特点：

1. 微信的使用者
2. 喜欢记录日常的人

## 1.3、假定和约束

系统反应时间：保证结果反应时间在 5s 以内。

开发期限：本学期第 13 周前完成全部开发和 Debug 工作。

经费限制：300 元以内。用以购买云服务器，云数据库等。

## 2、需求规定

### 2.1、软件功能说明

主要功能：识别并记录表情

用户在小程序端上传或拍摄后上传一张人的表情，上传到云服务器中，并传入人脸表情识别程序中，程序识别人脸表情并返回判断数据（高兴，难过，生气，惊讶，厌恶及中性的百分比），该数据传回小程序端，并且显示出来，同时，用户可以选择是否记录在历史中，如果是此数据会上传至数据库连同时间保存。

细分功能：

- 1、识别
  - a) 输入量：图片
  - b) 处理过程：人脸表情识别
  - c) 输出：表情属性占比
  - d) 并行操作用户数：10
- 2、读取历史数据
  - a) 输入量：读取交互按钮
  - b) 处理过程：读取历史记录
  - c) 输出：历史表情数据
  - d) 并行操作用户数：50
- 3、生成历史数据图像
  - a) 输入量：历史数据
  - b) 处理过程：图像化处理
  - c) 输出：生成柱形统计图和历史折线图
  - d) 并行操作用户：50
- 4、表情挑战
  - a) 输入量：挑战选择及识别功能结果
  - b) 处理过程：判断表情参数
  - c) 输出：挑战是否成功
  - d) 并行操作用户：10
- 5、生成挑战排行榜（选做）
  - a) 输入量：挑战结果
  - b) 处理过程：不同用户挑战结果榜单
  - c) 输出：挑战排行榜
  - d) 并行操作用户：5
- 6、文章及歌曲推荐（选做）
  - a) 输入量：表情识别结果
  - b) 处理过程：搜索对应表情的文章与音乐
  - c) 输出：文章及音乐的链接
  - d) 并行操作用户：50

### 2.2、对功能的一般性规定

- 1、交互操作尽量流畅
- 2、UI 界面简洁，操作明显的要求下，尽量美观

## 2.3、 用户界面

打开界面：

有动画加载条

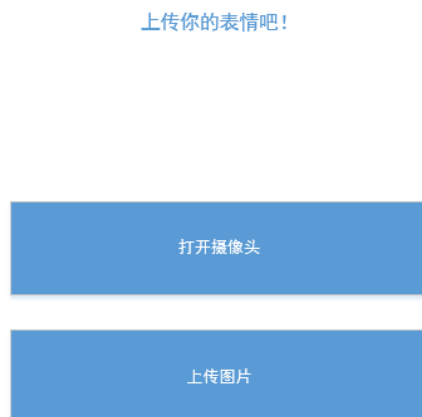
主页面：

识别功能按钮，查询历史按钮，表情挑战按钮，背景美观简约如（无背景）：



识别界面：

调取摄像头按钮，上传按钮，背景美观简约



等待界面：

有动画加载条

结果显示界面：

表情识别结果柱状图，推荐音乐示例，推荐文章及音乐链接，背景美观简约

## 2.4、对性能的一般性规定

### 2.4.1、精度

输入数据：

数据名称：输入图片（picture\_in）

精度：可识别，不可过大（超过 10Mb）

输出数据 1：

数据名称：输出图片（picture\_out）

精度：无

输出数据 2：

数据名称：表情（emotions）

精度：0 到 100 内，可自定义，常规为 3 位有效数字

### 2.4.2、时间特性要求

总时间要求不超过 6 秒

1、响应时间：1 秒内（考虑网络传输因素）

2、数据的转换和传送时间：100 毫秒左右（此处基本不会消耗过多的时间）

3、解题时间：3 秒内（处理时间未知，因此会花时间）

4、更新处理时间：无要求（不在总时间内，后期更新次数和可能性低）

### 2.4.3、灵活性

1、操作方式变化：

由于操作方式为我们在小程序上定义的，所以可更改范围很大，只要不影响用户体验均可修改。

2、运行环境变化：

由于我们需要用到微信的平台和他提供的好友数据，因此前端无法修改；数据库模块无特定需求，只需满足可在微信与表情识别模块中间传输即可；表情识别模块已经定下来由 python 环境来执行。

3、同其它软件接口发生变化：

该产品软件接口包括：

i. 微信小程序实时录制/播放音视频流；难以变更

ii. 腾讯云服务器公/内 IP；随服务器而转换

iii. TCP/IP 协议；难以变更

4、精度和有效时限变化：

精度变化对系统影响不大，图片精度过高可能会增加传输时间。

### 2.4.4、输入输出要求

输入数据：

数据名称：输入图片（picture\_in）

实际含义：通过小程序输入的待识别图片

数据类型：jpg、jpeg、gif、png、bmp

精度：可识别，不可过大（超过 10Mb）

输出数据 1：

数据名称：输出图片（picture\_out）  
实际含义：经过识别返回到小程序的图片  
数据类型：jpg、jpeg、gif、png、bmp  
精度：无  
输出数据 2：  
数据名称：表情（emotions）  
实际含义：经过识别返回到小程序具体表情数据组  
数据类型：float  
精度：0 到 100 内，可自定义，常规为 3 位有效数字

## 2.5、 数据管理能力要求

1. 需管理的全部数据样本
  - a) 照片：目前手机像素判断自拍照片大小预估为 1M 左右
  - b) 照片分析结果数据：包含各种情绪占比的数值。（愤怒、快乐、平静、悲痛、疑惑、厌恶、惊讶、恐惧）
  - c) 用户信息：每一次情绪检测结果与用户信息相关联，从而实现用户检索情绪曲线图的功能。
    - A. 数据库表形式基本为：

用户 id	性别	年龄	实时（最新）表情检测结果	全部上传自拍	全部情绪数据	近期情绪预测

- 1) 用户 id：存储用户的微信 id
- 2) 性别：男、女
- 3) 年龄：年龄
- 4) 实时（最新）表情检测结果：存储上传照片表情的检测结果，用于微信小程序前端 wx.request 调用，获取表情、情绪的检测结果
- 5) 全部上传自拍以及全部情绪数据：数据库子表存储全部上传自拍，与用户仍然相关联
- 6) 存储照片的列使用 mediumblob 数据类型，每行可以保存 16M 大小的数据用于存储照片  
其子表形式基本为：

日期	上传照片	情绪检测结果

- i. 日期：上传照片对应的日期
  - ii. 上传照片：用于检测表情情绪的照片
  - iii. 情绪检测结果：根据照片得出的情绪结果
- 7) 近期情绪预测：调用全部情绪数据，根据日期给予权重，智能分析近期的整体情绪，得出较为准确的结果，并存储。
  - 8) 文章歌曲链接：用户的情绪检测结果推送的相关文章歌曲，以链接形式进行数据存储。

B. 排行榜数据：

排行榜中参与挑战的用户上传的照片不存储进前面所讲的个人用户信息中，进行排行榜单独存储。排行榜实时更新存储数据，只保留存储上榜用户的挑战项目、用户信息、上榜表情照片以及成绩信息。

挑战表情	用户信息	上传照片	分数

- 1) 挑战表情：愤怒，惊讶，快乐，厌恶，恐惧等
- 2) 用户信息：用户微信 id
- 3) 上传照片：用于挑战的表情照片
- 4) 分数：根据表情照片分析出的情绪强烈指数

## 2. 用户使用量以及增长速率预估

- a) 日均表情检测上传照片次数：不超过 1000，增长速率每天不超过 10
- b) 单个用户上传照片总量：不超过 100，增长速率不超过（300/年）
- c) 单个用户日均上传照片数量：不超过 10，增长速率几乎为 0
- d) 排行榜日均挑战次数：不超过 1000，排行榜只需实时检测，更新存储数据，所以初步忽略数据大幅增长，如果后期出现大量数据上传分析，再作考虑。
- e) 单个用户相关文章歌曲推送获取次数：不超过 10，增长速率为 0

## 3. 数据总量预估

- a) 用户信息存储表行不超过 1000
- b) 排行榜信息存储表行不超过 10
- c) 子表行均不超过 1000
- d) 表总量预估不超过 1000，存在大幅增长可能，预留存储大量表数据的空间，为后期解决处理提前想好措施。

## 2.6、故障处理要求

### 1. 数据库和微信小程序下衔接故障

使用 `wx.request()` 函数对服务器数据进行调用时，固定的数据通过地址传输，但是当需要进行实时查询或者在线对云数据库操作时出现无法正常传输通信。通过完善后端服务器和数据库的相应程序，保证微信小程序和数据库的正常传输。如果实际难度较大，则尽量减少对于查询和实时修改云数据库的操作，并通过更多地址数据传输实现功能。

### 2. 存储数据过大或增长速率过快导致的数据库问题

数据库存储数据量过大会引发如下问题：

- a) 新数据插入以及旧数据修改删除速度变低
- b) 数据查询速度变慢
- c) 用于后端照片数据处理以及数据结果分支的实时处理性能降低
- d) 影响微信小程序前端调用时的数据库响应速度以及数据传输速度

初步设想解决方法：

- a) 优化代码性能，通过更优质的算法或者处理步骤，让数据的插入，删除，查询等操作更加节约时间和性能。
- b) 适当删除不重要的数据库存储数据

数据重要程度分级：

- i. 近期用户检测数据以及排行榜数据

- ii. 时间较为远久的检测数据
  - iii. 非实时的全部分支数据，如歌曲文章推送，排行榜挑战参与
- c) 扩大数据库存储容量，增加表的数量以及表的行数
- 3. 服务器端运行人脸表情识别程序无透明接口

由于微信小程序需要 `wx.uploadFile` 上传图片至服务器中的人脸表情识别程序，仅仅使用 `python` 很难设置方便的接口让微信小程序进行访问。

初步设想解决方法：

- a) 服务器端设置一个单独文件夹，将图片上传至此文件夹中，`python` 定时对此文件夹读取，如果读取成功则删除此文件夹，但此方法难以支持多个用户同时并行使用。
- b) 服务器端采用 `nodejs` 来运行 `python`，通过独立接口传送到 `python` 中，通过多个识别程序同时运行，达成多用户并行使用，识别后同样通过 `nodejs` 的接口将数据传回小程序

## 2.7、其他专门要求

### 1. 用户信息保密性

如果用户对于自身的情绪结果以及上传照片选择保密，可以选择不同模式进行处理（对所有人不可见，仅对微信好友可见，对自己可见），我们将直接在微信小程序前端调用时进行条件判断，从而判断是否调用相关的数据结果，实现用户信息的保密性。

### 2. 高度自由拓展性

我们希望小程序可以更加丰富，让用户的想法更多得以实现。不光通过传统手段，收集采纳评论区的意见，同时也直接提供适当的接口，让用户可以在线使用我们的后端情绪分析技术，实现更多的自由功能，在满足用户个人需求的同时，也创造了用户反馈、提升小程序质量的开放平台。

### 3. 可维护性

定期备份最为关键的数据库数据信息，并且定期对关键的数据进行检查，对照。定期检测数据库的存储量，空间占用一旦出现问题，立刻采取措施。实时检测数据库是否受到干扰，攻击，如果情况属实，立刻停止数据任何形式的交互，进行排查，维修。

## 3、运行环境规定

### 3.1、设备

成品程序：手机

表情识别程序：PC 主机

连接设备：腾讯云服务器，云数据库

### 3.2、支撑软件

PC 端：Windows 操作系统，`python3` 语言，PyCharm IDE，Navicat For MySQL 数据库管理工具，微信 web 开发者工具

手机端：iOS/Android 操作系统，微信程序。

### 3.3、接口

微信小程序实时录制/播放音视频流

腾讯云服务器公/内 IP  
TCP/IP 协议  
MySQL ODBC 接口

### **3.4、控制**

说明控制该产品的运行的方法和控制信号，并说明这些制信号的来源。

## **4、尚需解决的问题**

列出在需求分析阶段必须解决但尚未解决的问题。