

# 基于表情识别的智能歌曲推荐应用

## 详细设计说明书

项目名称：\_\_\_\_基于表情识别的智能歌曲推荐\_\_\_\_

小组成员：\_\_\_\_姚宇婕 201830310206\_\_\_\_

\_\_\_\_朱佳莹 201830310230\_\_\_\_

\_\_\_\_董绘芳 201830310236\_\_\_\_

\_\_\_\_周玲玲 201830310256\_\_\_\_

2019 年 3 月 15 日

# 1 引言

## 1.1 文档目的

本文档主要为《基于表情识别的智能歌曲推荐》提供详细设计说明。

面部表情不仅是表达情感的途径，也是交往的重要手段。在人表露的情感中，言辞占据 7%，声音占据 38%，面部表情占据了大部分 55%。由此可见，面部表情在人与人的交往中，包含了大部分的情感信息。对人的面部表情进行研究，可以进一步了解人对应的心理状态。

随着互联网的普及以及音乐资源的电子化使得人们可以更方便地获得音乐资源。但音乐库变得越来越大、资源越来越丰富，人们已经很难准确及时地找到自己喜欢的音乐。基于表情识别的智能歌曲推荐正是为满足这一需求而设计的。

## 1.2 文档范围

本文档主要包括软件的需求分析、模块划分以及功能模块的实现设计。

## 1.3 读者对象

本文档主要为软件测试人员、开发人员、应用维护人员等相关人员提供查阅。

## 1.4 术语与缩写解释

专门术语：

- (1) TensorFlow：是一个基于数据流编程的符号数学系统，被广泛应用于各类机器学习算法的编程实现，其前身是谷歌的神经网络算法库 DistBelief。TensorFlow 提供 Python 语言下的四个不同版本：CPU 版本（tensorflow）、包含 GPU 加速的版本（tensorflow-gpu），以及它们的每日编译版本（tf-nightly、tf-nightly-gpu）。
- (2) Keras：基于 Theano 和 TensorFlow 的深度学习库。Keras 是一个高层神经网络 API，Keras 由纯 Python 编写而成并基 Tensorflow、Theano 以及 CNTK 后端。Keras 为支持快速实验而生，能够把你的 idea 迅速转换为结果。

- (3) 深度学习：概念源于人工神经网络的研究。含多隐层的多层感知器就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表示属性类别或特征，以发现数据的分布式特征表示。

缩写：

- (1) CNN：卷积神经网络（Convolutional Neural Networks, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络，是深度学习的代表算法之一。
- (2) SQL：Structured Query Language(结构化查询语言)。
- (3) UML：统一建模语言、是一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言。

2 概述

2.1 系统目标

本项目的目的是开发一个可以识别人脸情绪并且智能推荐歌曲的播放器。设计的主要实现功能是播放 MP3 等格式的音乐文件，并且能控制播放，暂停，停止，切换歌曲，管理歌曲列表文件等操作。界面简明，操作简单。 软件系统检测到错误行为时，报告错误，并提示处理操作。

2.2 系统环境

2.2.1 开发环境

产品开发和维护所需的配置如下表：

分类	名称	版本
操作系统	Windows	Windows 10
数据库		
编程软件	Python	3.4
GUI 设计	tkinter	/
开发环境 IDE	PyCharm	2018.1.2

2.2.2 运行环境

软件运行所需要的条件支持如下表：

分类	名称	备注
操作系统	Windows	暂定为 Windows，其他 OS 看需求开发
硬件	CPU、硬盘等	默认最低配置

3 需求分析

3.1 功能需求

3.1.1 需求设计

本产品不仅要实现对一些常见的音频格式文件的播放，友好的用户界面，还要根据用户的情绪为用户推荐歌曲。其最终设计目标是：

- (1) 根据用户拍照上传的实时情绪，为用户推荐相关歌曲
- (2) 实现对多种音频格式文件的播放
- (3) 实现对当前播放曲目长度显示
- (4) 显示歌曲文件信息
- (5) 实现静音和音量控制
- (6) 实现歌曲模糊查找

根据功能需求可以将产品主要划分为以下四个功能模块，每个功能模块的作用描述如下：

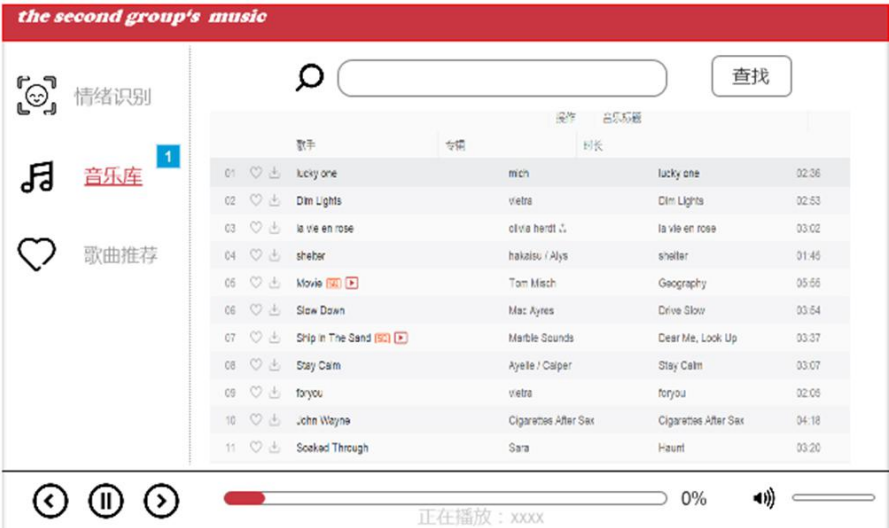
- (1) 情绪采集识别：采集用户的表情，判断出用户当前情绪。
- (2) 播放控制：控制歌曲的播放状态，如：播放、暂停、上一曲、下一曲。播放过的歌曲按播放顺序排列在播放列表中，每次退出程序、重启播放器后播放列表中的播放记录会自动保存不会清空。
- (3) 音乐库：存放产品提供的音乐，可根据歌名或歌手模糊查询歌曲。用户也在音乐库模块打开本地歌曲进行播放。
- (4) 歌曲推荐：产品的主页上随机或者按照分类推荐歌曲。

3.1.2 原型图

● 播放器



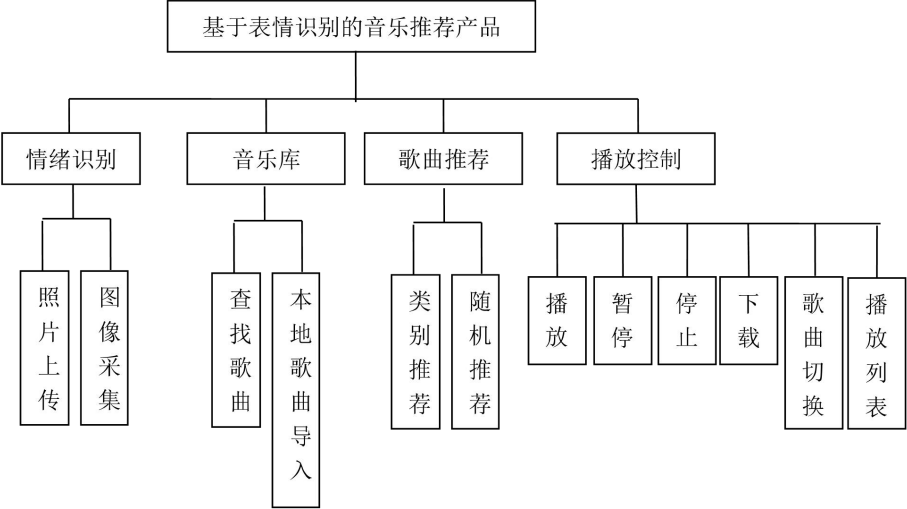
● 音乐库



● 情绪识别



3.1.3 功能结构图



3.2 非功能性需求

非功能性需求	详细要求
安全性	a. 为保证安全，不致使产品遭到意外事故的损害，系统因该能防止火，盗或其他形式的人为破坏 b. 产品要能重建 c. 产品应该是可审查的 d. 产品应能进行有效控制，抗干扰能力强 e. 产品使用者的使用权限是可识别的
可靠性	播放器软件占用系统资源少、运行平台要求不高
性能及效率	响应时间快，更新内容快
易用性	操作简单、能满足用户的常见需求
兼容性	兼容性应尽可能的强一些

3.3 UML

3.3.1 用例表

- 播放控制模块用例表：

用例名称	播放
参与者	用户
目标	使得用户可以播放在播放列表中选中的歌曲
前置条件	播放器正在运行
基本事件流	1、用户单击“播放”按钮； 2、播放器将歌曲和播放信息发送到播放 Service 中； 3、播放歌曲。

用例名称	暂停
参与者	用户
目标	使得用户可以暂停正在播放的歌曲
前置条件	歌曲正在播放且未暂停或停止
基本事件流	1、用户单击“暂停”按钮； 2、播放器将暂停歌曲信息发送到播放的 Service 中； 3、播放器暂停当前播放的歌曲。

用例名称	停止
参与者	用户
目标	使得用户可以停止正在播放的歌曲
前置条件	歌曲处于播放或暂停状态
基本事件流	1、用户单击“停止”按钮； 2、播放器将停止歌曲信息发送到播放的 Service 中； 3、播放器停止当前播放的歌曲。

用例名称	歌曲切换
参与者	用户
目标	使得用户可以听上一首或下一首歌曲
前置条件	播放列表有 2 首或 2 首以上的歌曲
基本事件流	1、用户单击“上一首”或“下一首”按钮 2、播放器停止当前播放的歌曲，并开始播放上一首或下一首歌曲。

用例名称	下载
参与者	用户
目标	使得用户可以下载歌曲至本地
前置条件	播放列表有歌曲且本地内存足够
基本事件流	1、用户单击“下载歌曲”按钮 2、在播放列表中选择需要下载的歌曲 3、选择下载路径 4、点击“确认下载”



用例名称	播放列表
参与者	用户
目标	用户可以查看播放清单
前置条件	产品处于运行状态
基本事件流	1、用户单击“清单”按钮； 2、播放器进入播放清单页面（歌曲不停止播放）

● 情绪采集识别模块用例表：

用例名称	情绪识别
参与者	用户
目标	采集用户当前表情并判断出情绪
前置条件	产品处于运行状态且开通相关权限
基本事件流	1、用户单击“情绪识别”按钮 2、用户上传照片或拍摄照片

● 音乐库模块用例表：

用例名称	音乐库
参与者	用户
目标	使用户可以查看本地歌曲
前置条件	产品处于运行状态且开通相关权限
基本事件流	1、用户单击“音乐库”按钮

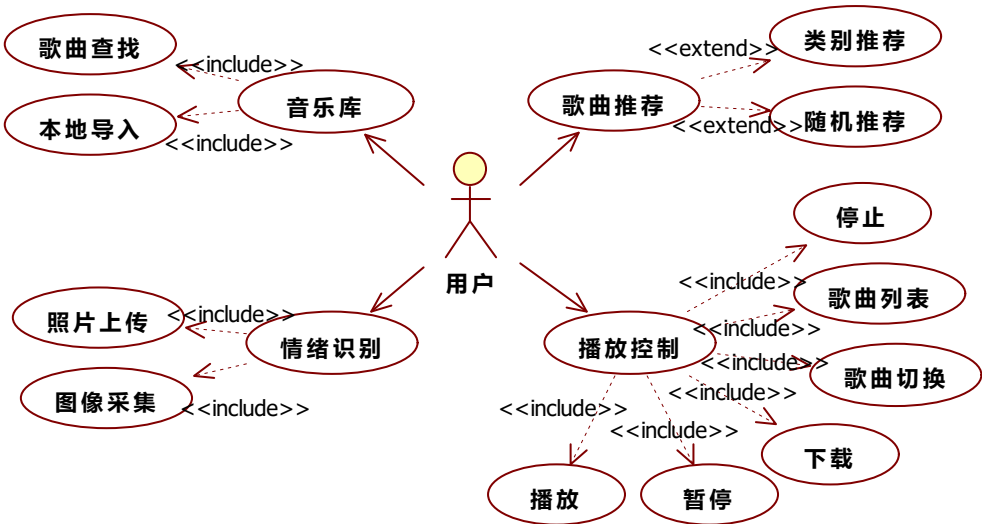
用例名称	查找歌曲
参与者	用户
目标	根据歌名等歌曲信息模糊查找歌曲并播放
前置条件	产品处于运行状态
基本事件流	1、用户在文本框内输入相关信息 2、单击“查找”按钮

用例名称	歌曲导入
参与者	用户
目标	导入本地的歌曲至播放器
前置条件	产品处于运行状态
基本事件流	1、用户单击“歌曲导入”按钮 2、用户选择导入歌曲的路径

● 歌曲推荐用例表：

用例名称	歌曲推荐
参与者	用户
目标	导入本地的歌曲至播放器
前置条件	产品处于运行状态
基本事件流	1、用户直接选择主页上的推荐歌曲进行播放 2、用户可以按照分类选择推荐歌曲进行播放

3.3.2 用例图



3.4 约束和假定

3.4.1 假定

本系统的开发经费为 0，同时在一个月的时间内完成设计，保证系统的按时交付，完成全部需求，同时应发掘出的隐藏用户需求，并予以实现。

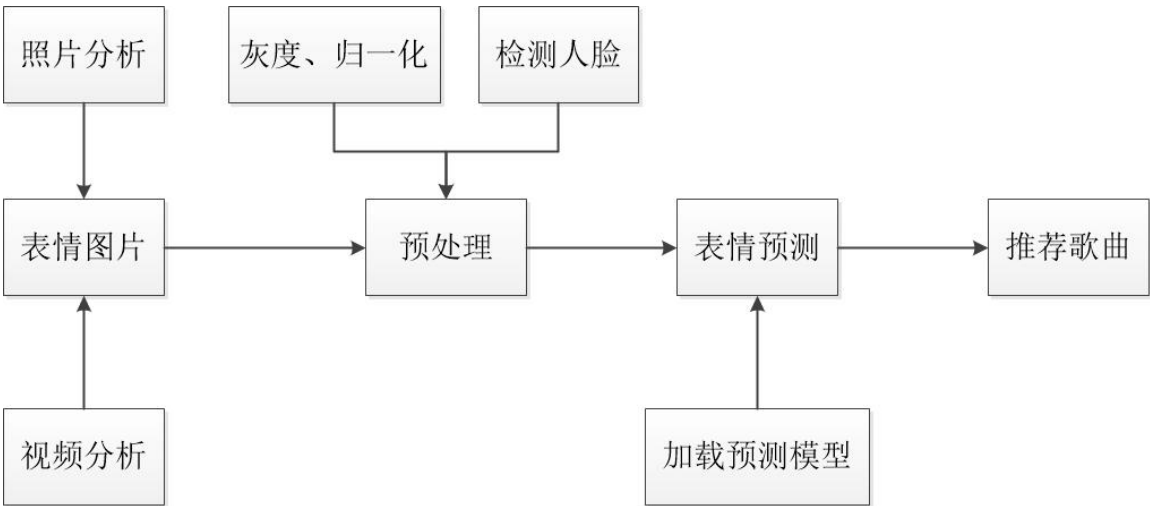
在开发这套软件时，我们假设用户已经熟悉基本的 Windows 操作。

3.4.2 约定

约束类型	约束条件
对编程语言的约束	Python
工具约束	开源或免费工具
性能约束	以产品能满足需求为首，暂不考虑优化性能
特殊场景约束	/
代码体积	/

4 架构设计

4.1 基本设计概念和处理流程



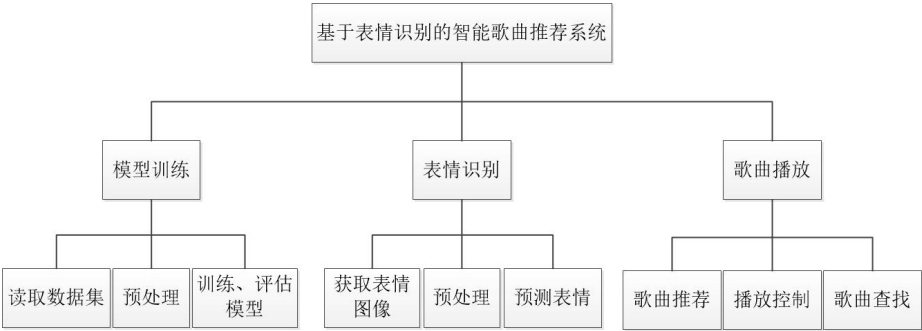
首先用户需要上传表情图像或视频以获得歌曲推荐。系统获得了表情图像后，先对该图像预处理（归一化并转换成灰度图像），检测出其中的人脸部分。然后系统根据训练好的预测模型对该人脸的表情进行预测，返回一个表情标签。最后，系统根据表情标签为用户做出歌曲推荐。

4.2 软件模块

4.2.1 模块命名规则

- 模型训练模块：每个功能性函数均以 dataset\_标识。
- 表情识别模块：每个功能性函数均以 emotion\_recognition\_标识。
- 歌曲播放模块：每个功能性函数均以 music\_标识。

4.2.2 模块汇总



模块关系如下：

模型训练模块：读取数据集，导出训练好的模型文件。

表情识别模块：导入训练好的模型文件，返回 `emotion` 标签。

歌曲播放模块：基于表情推荐歌曲时，可以读取 `emotion` 标签，返回推荐的歌曲。

4.2.3 各模块函数功能描述

● 模型训练模块：

方法名称	dataset_read
功能描述	读取人脸图像数据集，将数据集中 <code>pixels</code> 列和 <code>emotion</code> 列分开存储。
输入参数	file_path<人脸图像数据集的完整路径>
返回值	img_data; label
调用函数	read_csv ()
补充说明	无

方法名称	dataset_preprocessing
功能描述	将人脸图像的像素值归一化，将 <code>label</code> 映射为二值列别矩阵。
输入参数	img_data; label
返回值	Face_data; Face_label
调用函数	无
补充说明	无

方法名称	dataset_split
功能描述	将预处理后的人脸图像及其标签划分为训练集和测试集。
输入参数	Face_data; Face_label
返回值	train_x; train_y; test_x; test_y
调用函数	无
补充说明	无

方法名称	dataset_modeling
功能描述	训练卷积神经网络并保存模型。
输入参数	batch_size; epochs; train_x; train_y; test_x; test_y
返回值	model
调用函数	TensorFlow 函数
补充说明	无

● 表情识别模块：

方法名称	emotion_recognition_read
功能描述	打开要识别表情的图片路径。
输入参数	str<人脸图像的完整路径>
返回值	img
调用函数	文件读取函数
补充说明	无

方法名称	emotion_recognition_preprocessing
功能描述	将要识别的图像归一化、转化为灰度图像，并检测人脸。
输入参数	img
返回值	img_gray; faces
调用函数	cvtColor (); detectMultiScale ()
补充说明	无

方法名称	emotion_recognition_predict
功能描述	根据之前训练的 model，对人脸灰度图像预测表情。
输入参数	img_gray; faces
返回值	emotion
调用函数	无
补充说明	无

● 歌曲播放模块：

方法名称	music_recommend
功能描述	根据之前识别的 emotion，根据某种的算法推荐歌曲。
输入参数	emotion
返回值	mfile_path<音乐的完整路径>
调用函数	无
补充说明	无

4.3 功能需求与系统模块的关系

情绪识别需求：通过模型训练模块和表情识别模块实现。

歌曲的播放、歌曲搜索和根据表情推荐歌曲的需求：通过歌曲播放模块实现。

## 5 数据结构设计

### 5.1 逻辑结构设计

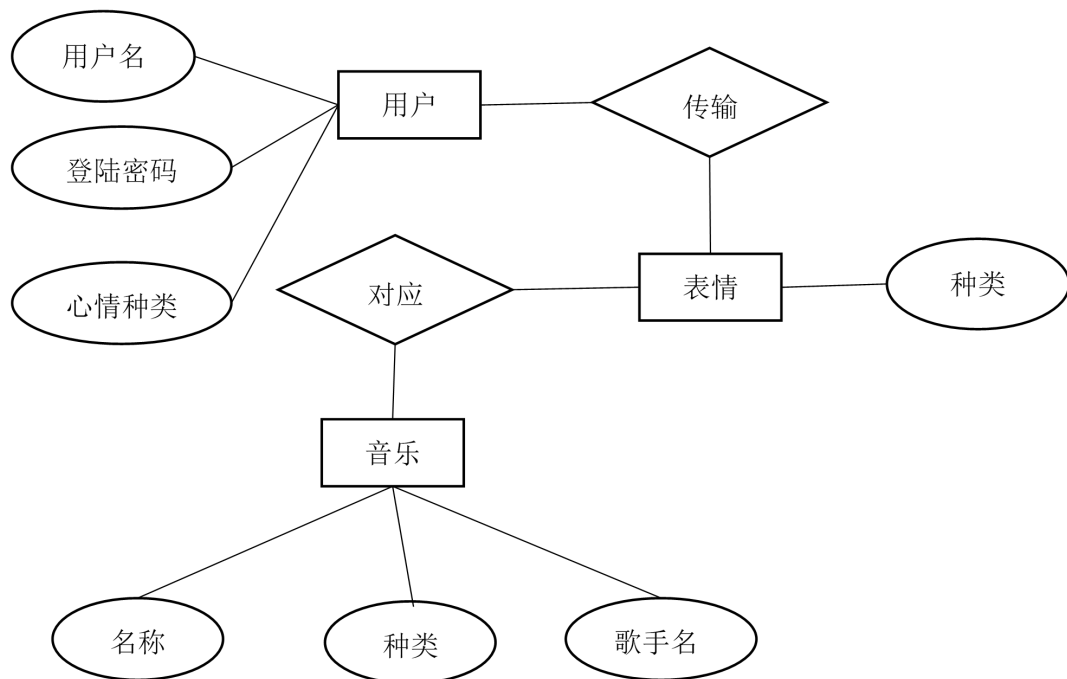
逻辑结构设计任务就是要将概念结构设计阶段设计好的基本 E-R 图转换为与选用 DBMS 产品所支持的数据模型相符合的逻辑结构

逻辑结构设计的步骤，如下：

- (1) 将概念结构转化为一般的关系、网状、层次模型
- (2) 将转换来的关系、网状、层次模型向特定 DBMS 支持下的数据模型转换
- (3) 对数据模型进行优化

### 5.2 E-R 图

本系统的 E-R 图如下：



由 E-R 图转换为关系模式，如下：

用户（用户名，用户登录密码，用户心情）

表情（种类）

音乐（音乐名，歌手名，音乐种类）



### 5.3 物理结构设计要点

物理结构是指数据库在存储介质上的存取方法、存储结构和存放位置。物理结构设计：指根据逻辑结构设计的结果，设计逻辑结构的最佳存取方法、存储结构和存放位置以及合理选择存储介质等，从而设计出适合逻辑结构的最佳物理环境即存储模式的过程。

- (1) 选择存取方法。存取方法是指用户存取数据库数据的方法和技术。常用存储方法包括平衡树、聚簇和散列索引。
- (2) 设计存储结构。存储结构是指逻辑结构的指标以及 DBMS 支持的数据类型，所确定的数据项的存储类型和长度以及元组的存储结构等，即数据文件及其数据项在介质上的具体存储结构。
- (3) 确定存放位置。存放位置是指数据库文件和索引文件等在介质上的具体存储位置。
- (4) 选取存储介质。存储介质是指用于存储文件的物理存储设备。包括，磁盘、光盘等。

6 系统出错处理设计

系统出错最严重的就是系统没有能够识别出用户提供的表情信息，为了从根本上解决这个问题，首先要确认用户给出的是有人脸表情的图片，其次在表情图片处理中，要注意图片的预处理的的过程中，准确有效的对表情图片进行灰度和归一化预处理，由此可以避免在一开始表情识别时产生的错误。

6.1 出错信息

用一览表的方式说明每一种可能出错的情况出现时，系统输出信息的形式、含义级处理的方法。

- 由于输入信息不符合规范，称之为软错误；
- 由于硬件方面的错误（如：网络传输超时、硬件出错等），称之为硬错误；
- 对于一些关键的操作，应该提供提示确认机制；
- 对于数据、测试文档，都要提供相应的保密措施设置。

错误类型	子项	错误原因
数据库错误	连接	连接超时
		连接断开
	数据库本身	数据库代码错误
		数据库溢出
系统部分自定义错误	输入错误	用户上传图片错误或为空
		用户没有上传于情绪有关的表情图片
	识别错误	用户上传的表情图片中的情绪不够明显
	查找错误	未找到符合情绪的记录
连接错误	内部连接错误	歌曲连接错误
		歌曲导入错误
	外部连接错误	页面休整或其他错误打不开

## 6.2 补救措施

说明故障出现后可能采取的变通措施，主要包括：

- (1) 对于软错误，需要先对用户提供的表情图片信息进行验证，图片预处理后，检测不出人脸表情或是表情识别不明显，要给出相应的错误提示语句，并返回给用户。
- (2) 对于硬错误，错误类型不会很多并且比较明确，所以可以在可能出错的地方中输出相应的出错语句，并将程序重置，最后返回图片提供阶段。
- (3) 后备技术说明准备采用的后备技术，当原始数据丢失时启用的副本的建立和启动的技术，例如周期性的把磁盘信息记录在案。
- (4) 恢复在启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或是软件从头开始重新开始运行的方法。

## 7 系统维护设计

设计的系统较小，设计时间比较紧张，没有外加维护模块。本系统维护比较简单，表情识别功能完成后，基本没有什么大问题，主要是对接音乐库的基本维护就可以了。为了便于维护，应该设计三种日志，系统运行日志、操作日志和出错日志。三种日志根据不同的严重程度采取存放在文件和数据库的方式，系统维护和管理人员可以很轻松的监控这个系统的运行情况。

## 8 尚未解决的问题

具体音乐库实现的数据库是以什么样的方式呈现，是直接使用自己建立的数据库，还是对接第三方音乐播放软件，尚且还没有完全确定，这是一个急需探讨和解决的问题。