

程序设计综合实验

需求报告

刘志远 2020212174

刘威 2020212172

沈琪 2020212178

2022 年 3 月 26 日

目录

1 任务概述	3
1.1 目标	3
1.2 系统（或用户）的特点	3
1.3 假定和约束	4
2 需求规定	5
2.1 软件功能说明	5
2.2 对功能的一般性规定	5
2.3 用户界面	6
2.4 对性能的一般性规定	6
2.4.1 精度	6
2.4.2 时间特性要求	7
2.4.3 灵活性	7
2.4.4 输入输出要求	7
2.5 数据管理能力要求	7
2.6 故障处理要求	8
2.7 其他专门要求	8
3 运行环境规定	9
3.1 设备	9
3.2 支撑软件	9
3.3 接口	9

目 录	2
3.4 控制	9
3.4.1 GUI (图形界面)	9
3.4.2 核心算法	10
3.4.3 数据库	10
4 尚需解决的问题	10

1 任务概述

1.1 目标

1. 在 PC 端完成针对一个用户的实时情绪分析与疲劳识别，并整合为用户精神健康信息实时输出在 GUI 上。
2. 保存一段时间内用户的精神健康数据，并根据信息提供精神健康的检测结果与改善建议。
3. 进阶目标：可以同时识别并记录多人的精神健康情况。
4. 进阶目标：添加神态识别模块（如皱眉）以及相关心理微表情的检测分析。

1.2 系统（或用户）的特点

系统特点：

可以提供专业的精神健康情况检测结果与改善建议，结合情绪分析与疲劳识别服务，让用户更了解自己的精神状态。

对于集体单位，可以获得并追踪员工、学生的具体精神状况，以便于关心员工、学生的心理状态与工作、学习效率，以便施以及时的补救措施。

用户特点：

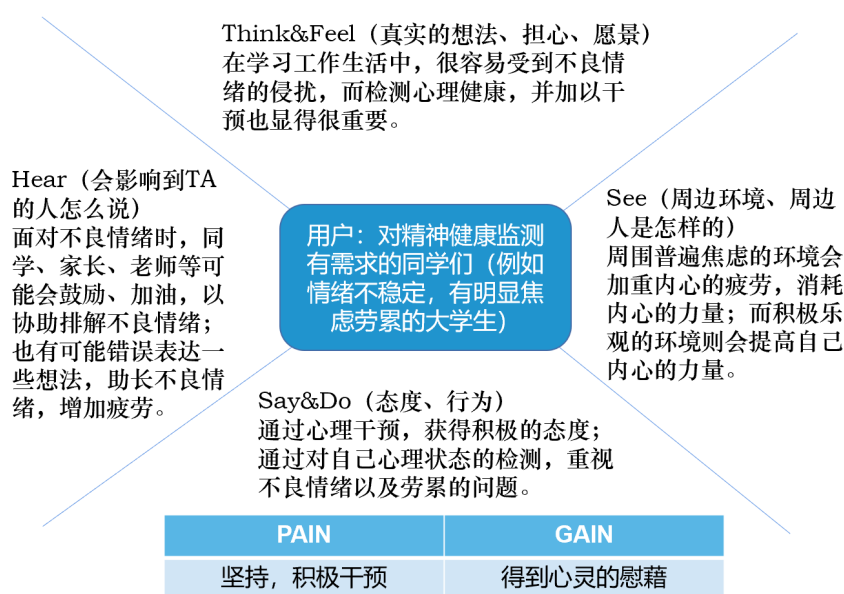


图 1：用户画像/共感图（以学生用户为例）

产品的用户主要分为一下四类：

1. 从事长时间面对屏幕的职业者（文字工作者、程序员等）。此类用户精神状态往往较为低迷，需要在心理问题上得到更多帮助。
2. 长期进行重复性工作的职业者。此类用户人际交往活动不多，渴望得到关心等正反馈来调节心理状态。
3. 心理状态尚未稳定成熟的学生。此类用户的情绪波动较大，使用特点可能不规律而急促。
4. 对自己心理健康状态关注较多的人。此类用户可以使用产品长期规律记录个人心理状况，使用特点为较规律且持续。

1.3 假定和约束

开发期限假定：

用两周时间进行概要设计，两周时间进行详细设计，六周时间实现调试和集成，在教学周第十六周完成所有内容。

开发约束：

Python 的实现：运行效率较慢，倚赖库较多，对平台的适应性较差。

识别检测模块：可能出现检测模型过大，检测时间较长，从而导致识别帧率降低，精度下降，用户体验下降。

数据库的结构及最终实现方式需要在编写过程中进行设计。

2 需求规定

2.1 软件功能说明

功能要求	输入量	处理	输出
情绪识别	从用户获得的实时视频流	通过训练好的 CNN 网络进行分类	蕴含情绪信息的列表 (int list)
倦态检测	从用户获得的实时视频流	通过 68 个人脸特征点进行分析计算出哈欠及眼动信息	疲劳程度 (int)
汇总精神信息	蕴含精神状态信息的列表 (int list)	分类汇总存储到数据库中	含有精神状态和时间的 json 文件
精神信息数据分析	含有精神状态和时间的 json 文件	进行数据分析	用户一段时间的精神状态信息图表与相关建议
* 人脸识别	从用户获得的实时视频流	根据 68 个人脸特征点提取的特征向量	——

表 1: 软件功能说明表 (* 为进阶任务)

产品容量：一次仅接受一个人脸输入，并且系统一次仅在一个终端上运行，不存在在一次运行中并行操作的用户。

2.2 对功能的一般性规定

- 界面格式统一为简约风格。
- 在镜头中没有出现人脸的情况下，应用将显示出提示通知，以引导用户正确使用功能。
- 进阶目标：在镜头中出现多张人脸的情况下，可以依据人脸识别，实时追踪、定位、精准储存每个人脸的精神状态。

2.3 用户界面



图 2: 用户界面概览

程序界面以简洁直观、亲切和谐为主要特点,主要包括检测界面和历史数据查看界面:检测界面主要以视频监控框为主体,右侧为实时检测参数(情感数据与疲劳指数);数据查看界面主要包括数据折线图与精神状态贴士。

2.4 对性能的一般性规定

2.4.1 精度

输入:实时视频流

计算过程中会出现 float 类型的数据。

输出:情绪信息(int list 0-100 的整数列表);疲劳程度(int 0-100 的整数);精神健康信息图表

2.4.2 时间特性要求

要求：系统响应较为快速，实时性强。

使用 python 的 keras 框架与 dlib 库、OpenCV 库等，并协同神经网络运行视频流的实时处理，预计精神信息的处理延迟将 $<0.1s$ 。

2.4.3 灵活性

操作方式上的变化：用户的输入仅可能为视频流文件，没有其他输入，操作方式不会发生很大改变。

运行环境的变化：在使用 python 时有可能会因为电脑没有安装对应版本的 python 环境导致运行不成功。

同其他软件的接口的变化：无同其他软件的接口的变化

精度和有效时限的变化：对任意分辨率的视频流文件，程序均将其转换为同一格式的文件，所以对精度无影响；神经网络内部模型较为固定，不会有精度或有效时间的变化。

2.4.4 输入输出要求

摄像头输入：视频流文件。

实时精神状态信息输出：百分制精神状态信息共八种，为 0-100 的 int 数据。

状态信息图表与相关建议输出：图表信息。

2.5 数据管理能力要求

一组精神状态信息包含 8 个百分制精神状态信息，分别为 1 个疲劳指数信息和 7 个不同情绪信息 (angry、sad、neutral、happy、surprised、disgust、scared)，均为 0-100 的 int 类型数据，所以一组精神状态信息大小为 $8 * 4B = 32B$ 。

一天内每隔 5 分钟，核心算法由原始视频流文件生成一组精神状态信息，并将其保存到数据库内，所以一天最多需要保存 $24 * 60 / 5 = 288$ 组精神状态信息，大小为 $288 * 32B = 9216B$ 。

数据库中最多保存 7 天的每日详细精神状态信息，大小为 $9216 * 7B = 64512B = 63KB$ 。此外，数据库中還保存一个月 (31 天) 的每日平均精神状态信息，大小为 $32 * 31 = 992B < 1KB$ 。所以数据库中最多管理 64KB 的精神状态信息。

2.6 故障处理要求

软件故障: python 未安装对应的版本, python 中配套 requirements.txt 文件, 辅助安装环境。

逻辑及操作问题: 在镜头中没有出现人脸的情况下, 应用将显示出提示通知, 以引导用户正确使用功能。在镜头中出现两个及以上的人脸, 目前按照没有出现人脸的情况不处理数据, 显示出提示通知, 以引导用户正确使用功能。在进阶目标中将结合人脸识别分析设备主人的情绪。

2.7 其他专门要求

信息安全性: 数据库最多存储 64KB 的信息, 信息量较小, 科研在用户本地存储而不需要传输至服务器进行处理, 故用户不需要担心隐私数据安全问题。

可维护性: 本程序由多个不同组件构成, 应用模块化编程, 封装功能组件, 以提高程序的可维护性。

代码易读性: 代码模块化封装, 每部分接口有相应注释, 内部函数注释丰富, 易读性好。

运行环境可转化性: 可通过 pyinstaller 生成对应 exe 文件, 以提高程序的适应性。

3 运行环境规定

3.1 设备

安装有 python 3 运行环境且有摄像头的电脑一台。

3.2 支撑软件

Anaconda、QT designer、Pycharm、SQLite

3.3 接口

本程序是独立系统，无接口

3.4 控制

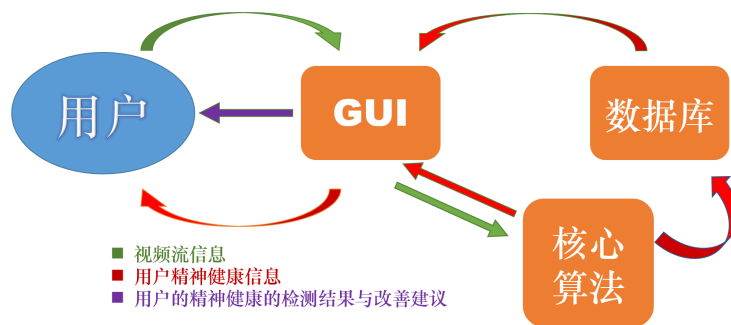


图 3: 系统数据流简图

3.4.1 GUI (图形界面)

该部分主要完成以下任务：

- 1、给予用户更优质的程序交互体验
- 2、实时传输来自用户的视频流信息至核心算法部分
- 2、实时输出经核心算法分析后的用户精神健康信息
- 3、获取数据库中存储的用户精神健康信息以提供检测结果与改善建议

代码由 python 的 PyQt 库、QT designer 完成

3.4.2 核心算法

该部分主要完成以下任务：

- 1、获取来自 GUI 的视频流信息
- 2、利用核心算法（情绪分析与疲劳识别）实时分析视频流信息
- 3、将实时分析的用户精神健康信息在 GUI 输出给用户并保存部分信息于数据库

3.4.3 数据库

该部分主要完成以下任务：

- 1、获取并保存来自核心算法部分的的精神健康信息

代码由 SQLite 完成

4 尚需解决的问题

1. 在拓展到多人识别时，尚不清楚识别效率是否符合实时监测的要求。
2. 对识别载体的拓展上，除了使用已有的打开摄像头进行识别外，如果使用上传视频文件时，尚不清楚如何进行识别的加速。
3. 绘制一定时期的精神健康状态曲线需要存储相应时期的数据，可以选择使用云数据库或是搭建本地数据库，使用云数据库的后续实用性更高，但是需要考虑网络的稳定性等问题。