PPGDetector需求规格说明

“吼”组

1. 综合描述
2. 产品前景

如今，人们对医疗、安全与健康的需求日益增长。心率监测是分析人体生理与心理健康状况的重要手段，通过对心率的监测，可以实现对用户的生理和心理健康状态的分析，从而实现在医疗、安全等领域的应用。汽车上，通过心率监测及健康状态估计可以掌握驾驶员的疲劳状况，从而适时提醒，避免疲劳驾驶及安全问题。部分监控场合，也可通过测量心率获取被监控对象的心理生理健康状况，从而及时进行救助。

我们的软件可以通过获取视频中的人脸反映出的心率信息，结合对心率信息的经典分析方法，实现对人的心率监测和简单的健康状况估计，力图为公众的安全健康保障添砖加瓦。

1. 产品功能

目前，医院最普遍的方法是采用心电图机（EEG）检查患者的心脏功能，但监测时间较短且人力物力成本较高。大多消费类电子设备如穿戴式手环、手表通常是监测用户的运动心率，且受限于功耗，通常五分钟或十分钟才测一次，医学参考价值不大，且应用场景受限（需要佩戴手环）。

DARMA是一家创立于美国硅谷的创新医疗健康科技公司，旗下的一款监测垫可以非接触式的测量心率，无需穿戴、无需直接接触便可以进行心率、呼吸频率等生命体征的监测。但仅能监测睡眠时的健康状况，对普通民众的应用场景有限。

综上可知，目前的心率监测方式存在使用场景受限的问题。因此，我们希望实现一款非接触式的心率监测软件，能够通过摄像机对目标区域的拍摄获取被监测者的面部图像，分析得到对应的PPG信号，并结合对指夹器信号的学习，实现对心率的监测。

1. 预期用户

我们的软件能够在非接触、甚至不知情的情况下对人员的心率进行监测。因此，我们的软件及设备可以用于机动车中，用来监测司机的心率以确保驾驶的安全等。此外，在某些重大的刑事案件、或关乎国家安全等重要事件的调查中，可以在被监测方不知情的情况下检测其心率，并从而获知其基本的心理健康等状态。

1. 外部接口需求
2. 用户界面

通过多平台兼容的Qt开发用户界面，实现基本的用户交互、结果显示和参数调整等功能。

1. 硬件接口

视频模式下，无需特殊的硬件接口；

相机模式下，需要通过串口连接相机完成图像采集及保存功能。

1. 软件接口

图像处理：OpenCV库

1. 通信接口

视频模式下，无需特殊的通信接口；

相机模式下，需要调用串口或网络端口GIGBYTE。

1. 功能需求
2. 视频模式
3. 访问文件浏览器，获取视频路径；
4. 读取视频，通过前后连续多帧图像分析视频中被测者的心率信息；
5. 通过文本、图像等方式显示结果。
6. 相机模式
7. 访问串口，与相机建立连接；
8. 从相机获取连续的视频流，通过单帧图像分析视频中被测者的心率信息；
9. 通过文本、图像等方式显示结果。
10. 其他非功能需求
11. 性能需求

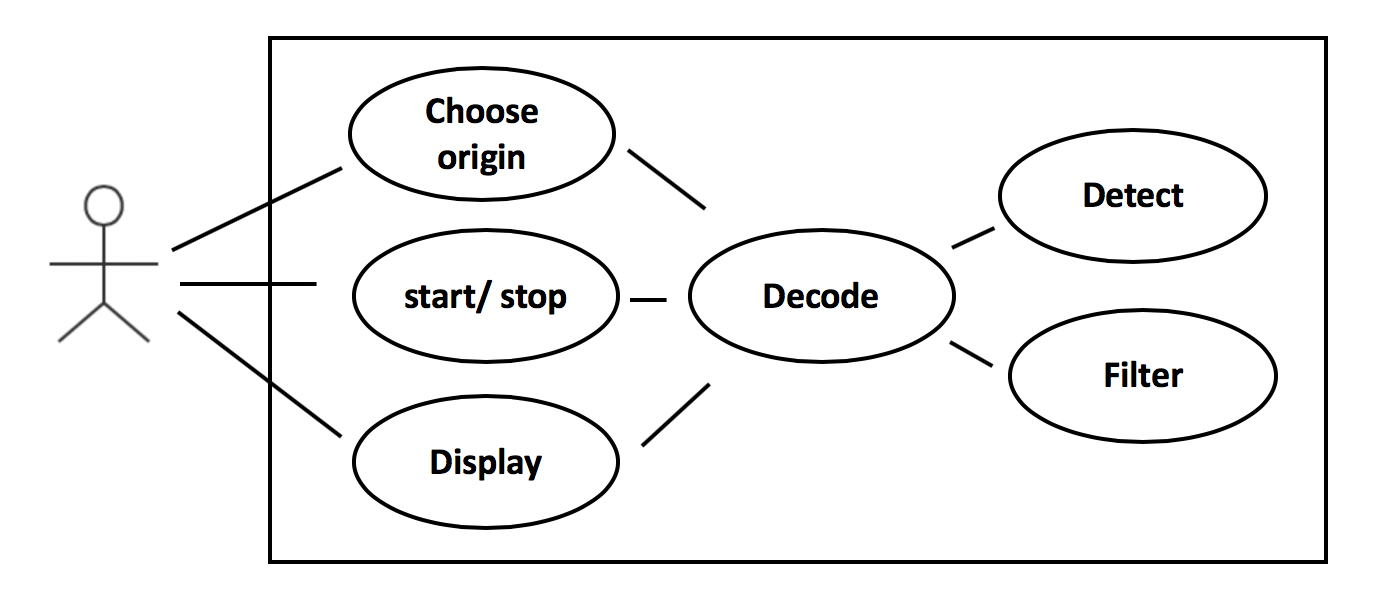
对单张人脸的情况，算法效率越高越好，目标处理速度为30 Hz，可实现相机实时采集并处理。(普通CPU单进程运算)

暂不考虑多张人脸的情况。

1. 安全性需求

一般情况下，未经被测者允许不得对其进行心率分析，需要对被测者进行详细的隐私安全说明。

1. 用例分析
2. 用例图



1. 用例说明
2. Detect:

|  |  |  |
| --- | --- | --- |
| 名称 | | Detect |
| 用例描述 | | 检测视频帧，识别人脸区域，选取计算域，获得信号 |
| 前置条件 | | 1.用户已经点击start  2.当前帧输入数据正常  3.Detector对象已正常创建 |
| 事件流 | 基本事件流 | 1.将原图片缩放至合适大小，转为灰度图  2.对灰度图调用dlib库识别人脸区域  3.将人脸区域位置转换为原图坐标  4.对原图调用dlib库识别人脸标志点  5.选取计算域，获得信号值  6.在原图上画出人脸区域及标志点，用于界面演示  7.存储当前帧的人脸区域及标志点位置 |
| 扩展事件流 | 1.若当前帧不是第一帧，且人脸区域与前一帧距离较小，对人脸区域进行平滑  2.若当前帧不是第一帧，且人脸区域与前一帧距离较小，对标志点位置进行平滑 |
| 异常事件流 | 1.当前帧找不到人脸，返回空值，提示错误信息  2.当前帧存在多于1个人脸，返回空值，提示错误信息 |
| 后置条件 | | 1.当前帧信号值通过Filter::filterInput()传入Filter对象  2.若当前帧数据有效，则人脸区域及标志点位置被正常存储 |

1. Filter:

|  |  |  |
| --- | --- | --- |
| 名称 | | Filter |
| 用例描述 | | 对视频检测获取的各个颜色通道的数据进行滤波，获得较为平滑的波形，进行心动周期分割 |
| 前置条件 | | 1.用户已经点击start  2.Detector::detect()正常运行完毕  3.当前帧输入数据正常  4.Filter对象已正常创建 |
| 事件流 | 基本事件流 | 1.存储新的数据点并将其加入计算窗口  2.窗口数据足够时进行滤波计算  3.分割周期  4.计算心率  5.计算需要输出的数据范围  6.设置outflag为True |
| 扩展事件流 | 1.如果计算出的心率值超过正常范围，认为当前周期分割点识别错误，舍弃当前分割点 |
| 后置条件 | | 1.若进行计算，则outflag为True  2.原波形、处理过的波形、心率和周期分割点都被正确存储 |

1. Decode

|  |  |  |
| --- | --- | --- |
| 名称 | | Decode |
| 用例描述 | | 视频通过解码器逐帧提取数据，经由Detector与Filter获得输出信号 |
| 前置条件 | | 1.用户已经点击start  2.在文件模式下，输入路径有效  3.在相机模式下，相机流获取正常 |
| 事件流 | 基本事件流 | 1.解析视频，获取当前帧数据  2.将当前帧传入Detector，获取信号值  3.将信号值转入Filter  4.查询计算结果是否更新，更新则获取之  5.将计算结果与视频帧传输至界面，用于显示 |
| 后置条件 | | 无 |

1. Choose Origin

|  |  |  |
| --- | --- | --- |
| 名称 | | Choose Origin |
| 用例描述 | | 用户选择数据来源 |
| 前置条件 | | 1.界面正常初始化  2.运算进程并未开始 |
| 事件流 | 基本事件流 | 1.若选择相机模式，调用串口打开相机  2.若选择文件模式，弹出选择页面，选取视频文件 |
| 后置条件 | | 1.使start按钮可用 |

1. start/stop

|  |  |  |
| --- | --- | --- |
| 名称 | | start/stop |
| 用例描述 | | 用户控制ppg检测的开始或停止 |
| 前置条件 | | 1.界面已正常初始化 |
| 事件流 | 基本事件流 | 1.若按钮被点击时并未进行计算，则开始计算，将按钮文字更换为stop  2.若按钮被点击时正在进行计算，则停止计算，将按钮文字更换为start |
| 后置条件 | | 1.开始计算后，使数据来源选择按钮不可用  2.停止计算后，使数据来源选择按钮可用 |

1. Display

|  |  |  |
| --- | --- | --- |
| 名称 | | Display |
| 用例描述 | | 将后端计算结果显示在界面上 |
| 前置条件 | | 1.界面已正常初始化  2.已进行有效计算，存在计算结果 |
| 事件流 | 基本事件流 | 1.将后端传来的视频帧与计算结果显示在相应控件上 |
| 后置条件 | | 无 |