一.硬件系统



二.软件系统



1.引擎

a.负责维护相机输入数据。

b.负责将可见光数据送入算法模块，同时维护算法输出数据。

c.根据算法输出数据中的时间戳，找到红外数据中时间戳相近的红外图像，和算法的输出结果一并送入测温模块，进行温度测试，同时得到测温结果。

d.负责最终结果发送到用户界面显示，如果有网络，也需要发送到网络。

2.通用数据结构

Class image\_info{

cv::mat visibleimg;//可见光图像

cv::mat infrareimg;//红外图像

std::chrono::steady\_clock::time\_point m\_capturetime;//时间戳

std::vector< face\_info > faceinfos;//这个图像中所有人脸信息

}

Class face\_info{

int id; //唯一ID

int x; //人脸在图片中的水平偏移

int y; //人脸在图片中的垂直偏移

int w; //人脸在图片中的宽度

int h; //人脸在图片中的高度

float temperature; //温度

bool bfever;是否发热

bool bmasked;//是否佩戴口罩

}

2.相机模块

相机模块主要负责相机的管理，由于可能使用不同的相机，统一接口定义：

*int initialize()*

*int deinitialize()*

*int Registercallback(std::function<void(*cap\_info *&)> callback)*

*int Getparams()*

*int Setparams()*

*int start()*

*int stop()*

在调用回调函数时，需要传回如下信息：

Class cap\_info{

cv::mat img;//图像

std::chrono::steady\_clock::time\_point m\_capturetime;//时间戳

}

3.算法

算法模块主要实现人脸检测，跟踪，后续还可以增加是否佩戴口罩等。

算法模块拿到Class cap\_info信息后，做算法处理后，返回Class image\_info需要填充如下信息：

*cv::mat visibleimg;//可见光图像*

*std::vector<* *face\_info > faceinfos;//这个图像中所有人脸信息*

以及*face\_info中*的如下信息：

*int id; //唯一ID*

*int x; //人脸在图片中的水平偏移*

*int y; //人脸在图片中的垂直偏移*

*int w; //人脸在图片中的宽度*

*int h; //人脸在图片中的高度*

*bool bmasked;//是否佩戴口罩*

帧率控制建议在这里做，如果不需要处理，就直接回传。

4.测温模块

测温模块主要是根据检测出来的人脸框位置，找到对应红外图像中的区域，取最大值或者平均值，得到人脸的温度信息。同时，是否使用黑体，也在这里考虑。

测温模块拿到Class image\_info信息后，测量温度，填充如下信息

*float temperature; //温度*

*bool bfever;是否发热*

同时，将温度，人脸框画在图像上。

如果没有检测到人脸，则直接返回。

5.用户界面

用户界面通过注册回调函数的方式，把自己的处理函数注册到引擎中。引擎拿到测温结果后，就可以通过回调函数发送给用户界面。

用户界面需要把可见光图像，红外图像，显示在界面上，同时缓存N个人脸小图信息，以及M个异常人脸小图信息，显示在界面上（要显示的小图可以在这里crop，或者加在Class image\_infox信息中）。

同时如果此次有异常，则发出声光报警。

6.数据库

存储每个人脸信息。

7网络

和云服务器交互。