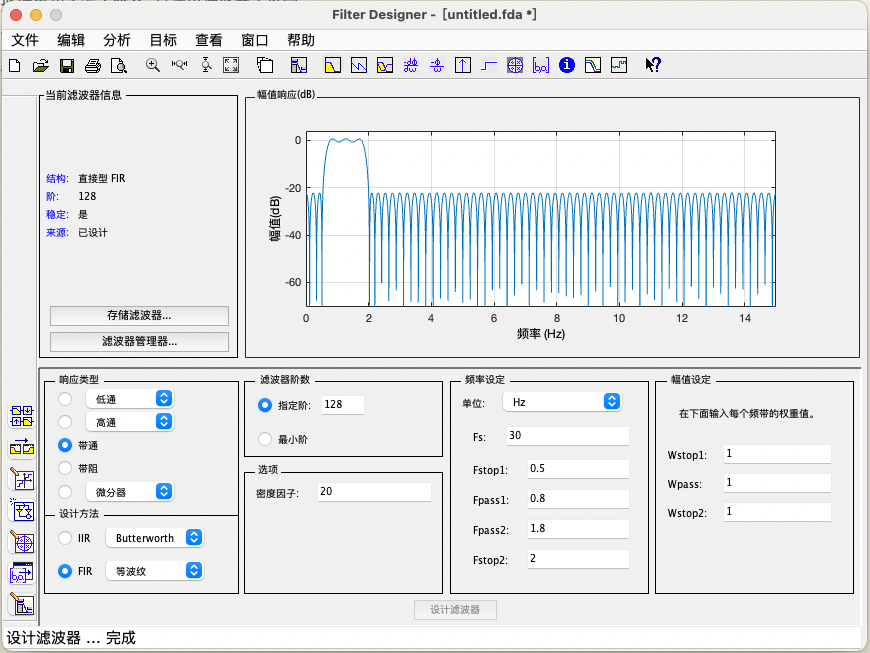
**第十四周进展报告**

**李浩伟:**

1. 完成视频心率在Mac笔记本上的完整构架
   1. 第一版尝试使用5-15Hz滤波器,滤波器长度32,差分,平方,均值,而后进行波峰检测,发现最后的波形结果和心率不符合,且滤出波形并不稳定
   2. 第二次尝试带宽更窄,到1-6Hz,32位,去除差分,平方处理,依旧不理想
   3. 第三次使用更窄的滤波器,128阶,只保留滤波和均值处理
   4. 最终结果的频谱可以在1Hz左右看到稳定峰值
   5. 最后采用定长度,带不应期的一段心率上的峰值检测
   6. 可以稳定输出波形,心率在cmd窗口输出



2. 将在笔记本上运行成功的视频心率的代码部署到了kinect端,进行了适应性的代码重构

3. 解决了kinect在存在人体骨骼时出现的帧率下降的问题,问题产生的原因为此前将骨架点的获取和转换等功能放在input线程中,导致在出现骨架时需要耗费一段时间进行处理数据。解决方法为将映射器和重建器分别抽出来形成一个新的线程,同时用pygame.clock.tick替代time.sleep增强时间的均匀性。在这边由于将重构和映射抽象成一个新的线程,不可避免得降低了效率的降低和速度的下降

4. 测试了各个组件的帧率状态。在正常状态下,视频获取为60fps+,当出现骨骼时需要视频和映射器信息,骨架点信息,速率降到约35fps+,因此直接将整体帧率降到30fps,限制最高帧率,从而获得了稳定的帧率状态。同时契合了原先用笔记本实现的视频心率的帧率

5. 在程序中增加在图片上进行文字显示的模块

6. 部分解决了子线程不能通过ctrl+c的方式kill的问题,设置线程为daemon

7. 研究了window和mac上的主/子线程的关系。在windows端,子线程可以进行ui,但是在mac端,ui相关功能只能放在主线程,给代码转移增加了难度

8. 研究了多个不同的滤波器对以后视频心率的效果的影响。在现在的版本中,如果指定方框中不存在人脸相近颜色的物体,则生成一个阈值内的mask,用以获取数据。但是不可避免的,在没有人存在,但是有相似颜色物体时,会有信号存在。但是滤波器本身的带宽较窄,最后滤波结果的信号和真实的心率信号实际相似,但难以分辨。因此我认为在这个时候我们不应该关注这个问题。我们需要的是在希望获得心率的时候给出较为准确的判断,但是如果环境本身就是异常的,如何的判断实际上没有意义。当然我们也考虑过通过人脸特征点识别的方法来部分上消除这个问题,但是一方面这会拖累整个进程,另一方面,没有太多实际的需求,我们应该将注意力放在在需求环境上的优化,而不是花太多精力对抗数不清的异常情况。

**江柔蓝**

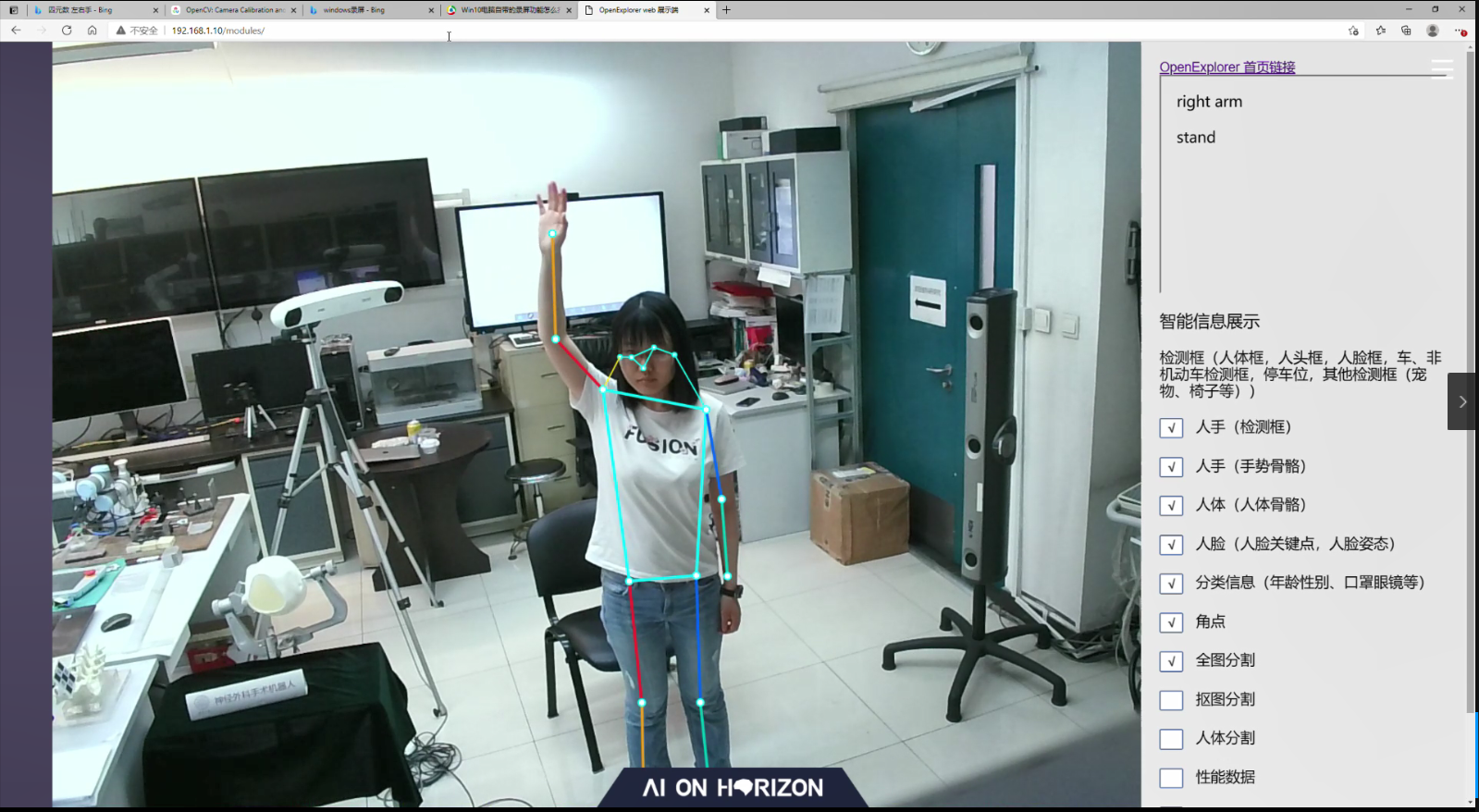
1、在地平线开发板上，优化了基于点坐标的姿态识别的流程。原先程序会串行进行“举手”“站立”“蹲下”三种姿态的检测，并给出三个输出结果，仔细梳理后，我发现“站立”和“蹲下”的检测有重复的部分，所以将“蹲下”检测删去。经过测试，我发现程序测试“举手”的效果较好，故将“举手”分为了“举起左手”“举起右手”“举起双手”三种结果输出；而“站立/蹲下”的检测结果较差，主要与人面对摄像机的角度有关，暂时计划设定为“站立”“坐下”两种输出，并且后续会进一步优化这里的算法和输出效果。

2、为基于点坐标的姿态识别程序输出增加了滑动平均的功能，对于“站立/坐下”，用大小为10的滑动窗口储存最近10帧的识别结果，并且输出该十帧中数量最多的类作为最终结果。经测试，这样程序的最终输出更为稳定。

3、重新设计了识别结果输出到网页显示的流程。原先“举手”输出和“站立/坐下”输出均在同一文件中。由于网页刷新和程序写入结果并不同步，显示并不稳定。修改后，“举手”和“站立/坐下”分别输出到“data1.html”和“data2.html”两个文件中，再用“data\_pre.html”用iframe加载两个文件，这样输出就不再会出现冲突。

4、协助李浩伟完成了kinect训练/测试集数据的录制（站，坐，行走）

5、录制了地平线开发板部分最终演示视频的素材，准备演示视频剪辑。



**构建动作判断的深度学习模型——李子涵：**

**工作概览：**

本周主要是针对采用我们的设备采集的数据进行迁移学习的尝试，同时也针对实验结果进行了分析，通过与数据输入端的同学进行了沟通交流来优化我们的整个工作流程。

首先，我们直接采用上周采集的坐着、站着和躺着的数据进行了模型的迁移学习，迁移学习过程中采用的是使用小学习率的策略，并没有将前面的层全部freeze。训练40个epoch的效果如下。

表1 未进行数据规范等操作时的迁移学习正确率

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 动作 | 站 | 走 | 坐 | 总 |
| 测试集正确率 | 33.4% | 37.3% | 100.0% | 60.1% |

**问题分析：**

由表1所示，可以看到”坐”这个动作非常好辨别，网络可以完全识别。然而对”站”和”走”这两个动作网络的识别效果则相当的不尽如人意，几乎是瞎猜。而”站”和”走”这两个动作的本质区别是位置上的运动，然而由于在迁移学习过程中，输入的是1s的数据，在这1s内位置的变化可能不太明显；另一方面，上周数据输入端同学曾报告采集的数据存在帧率跳变的问题，每帧数据之间的间隔不一样可能会给网络引入非常谬误的信息，因此效果不好。出于以上两点考虑，我们将数据采集的过程规范化，并将每一步的考虑进行详细说明，见附录1。

**优化后结果与后续期望：**

经过采集流程规范化与机位架设的优化，同时在数据预处理端删除了人体主轴与z轴对齐的先验之后，我们在加入了“躺”这个动作，目前迁移学习的最佳正确率如表2所示。

表2 规范化与加入躺的迁移学习正确率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 动作 | 站 | 走 | 坐 | 躺 | 总 |
| 测试集正确率 | 59.6% | 98.3% | 100.0% | 100.0% | 94.2% |

可以看到，在具体的体位上，我们的模型已经有了非常好的判断能力。目前还区分得不太好得动作就是“站”和“走”，推测问题应该出在“站”和“走”两个类的类内区分。由于模型仍在处在训练过程中，在这里暂时还没有进一步的去分析精确率，计划模型训练完成后对每一步训练模型进行补充精确率测试，最后进行模型的选择。同时如果确实是站和走区分度不强的话，我计划在输出的分类结果是“站”或“走”时，计算两秒内受试者的质心位移，如果执行位移足够大，则判定为“走”，反之判定为“站”。

**附录1：**

声明：本数据采集流程应当不涉及伦理问题，所有受试者仅保存骨架数据，并不会保存图像数据，故未向伦理委员会报备。

受试者：匿名

采集流程与主要注意事项：

1. 受试者需要采取行走、站立、躺、坐四种行为。各行为采集中受试者注意事项如下所示：

（1）行走：受试者在行走过程中，仅需保持双脚反复离地，需要尝试上半身静止、旋转、前后摇动。

（2）站立：受试者在站立过程中，仅需保持双脚不离地，需要尝试上半身静止、旋转、前后摇动。

（3）躺：受试者在躺的过程中，仅需保持背部与臀部不离开躺椅，需要尝试腿部进行抬起，蜷缩等复杂运动；上肢尝试张手，抬手等复杂运动。

（4）坐：受试者在坐的过程中，仅需保持背部与臀部不离开座椅，需要尝试腿部进行抬起，蜷缩等复杂运动；上肢尝试张手，抬手等复杂运动。

2. 采集机位保持固定，采集帧率保持30fps固定。机位摆放如下附录图1所示。



附录图1 成设

红圈为采集设备，紫圈为受试者，棕圈为显示设备。此时受试者正在进行“躺”