**集创赛池**

10.129.238.204

**少林派的sail版本号是3.3.0**

sophonSDK参考指南：https://doc.sophgo.com/sdk-docs/v24.04.01/docs\_latest\_release/docs/SophonSDK\_doc/zh/html/index.html

sophonSAIL参考指南：https://doc.sophgo.com/sdk-docs/v24.04.01/docs\_latest\_release/docs/sophon-sail/docs/zh/html/index.html

去年国奖项目：https://www.sophgo.com/case-center/detail.html?id=55

PPT: https://www.canva.cn/design/DAGE6vihxLo/qveff7abo9mofO5nWxWG-A/edit?utm\_content=DAGE6vihxLo&utm\_campaign=designshare&utm\_medium=link2&utm\_source=sharebutton

**待办：**

* ~~语音播报模块调试~~
* ~~心率检测数据流串口通信~~
* ~~红外眼睛检测数据集采集+模型训练~~
* ~~少林派前端 （用python-Gtk库吗）~~
* ~~安装显示屏~~
* ~~呼吸检测测不到的问题解决~~
* ~~心率->疾病检测模型~~
* ~~语音识别模型（搞还是不搞？）~~
* ~~危险行为识别，如是否在打电话等（对标去年国奖报告的行为识别）~~
* ~~在实车上部署的演示视频：考虑如何在实车上部署~~
* 远程通信从局域网通信升级为不同局域网的通信（云服务器使用socketserver库）更高级的情况下采用也需要用到服务器的4G通信（为了实车部署的时候答辩能够说明白，也为了试车部署视频能录明白）
* ~~系统程序逻辑设计~~
* ~~程序多线程优化~~
* ~~健康数据通过蓝牙模块发给手机显示(搞）~~
* ~~关键模型（如68点）int8和fp32模型的速度和精度对比（余弦相似度等）~~
* 模型细粒度优化（参考去年国赛报告，能做吗？）
* 小车靠边确保不撞车（还得微调一下）小车电机转速不稳定
* ~~方向盘震动提醒司机~~

8.18日讨论新定待办

* 1 前端控制程序
* 2 振动
* 3 语音模块并发接入（多外设
* ~~4 PPT硬件展示页~~
* ~~5 第一个模型找更好的~~
* ~~6 ppt 并发页~~
* ~~7 ppt int8 fp32表格对比页~~
* ~~8 ppt 模型之间表格对比页~~
* ~~9 ppt 体现出我们用的是世界最强模型~~
* ~~12 视频补新增硬件介绍~~
* 13 调小车
* 14 拼盒子
* ~~15 安装电源~~
* ~~16 报告实车展示~~
* ~~17 ppt实车展示~~
* 18 脚本 接收程序之后启动其他程序

比赛要带：

1 白色队服 黑裤子

2 母母头 公母头 公公头杜邦线若干

3 热熔胶（用于防止松动）

**系统pipeline：**

数据获取：opencv实时捕获摄像头视频流

模型1：检测并定位驾驶员人脸框https://cjmcv.github.io/deeplearning-paper-notes/fdetect/2017/11/18/SFD.html

模型2：驾驶员人脸68关键点二维定位https://arxiv.org/pdf/1703.07332

模型3：驾驶员人脸68关键点三维重建https://arxiv.org/pdf/1703.07332和上一篇是一样的，在后面部分有

模型4：睁眼闭眼二分类器

疲劳检测逻辑整合：

通过三维68点重建口部形状实现哈欠检测

通过睁眼闭眼二分类器实现闭眼检测

在检测到哈欠或长时间闭眼之后进行疲劳检测预警，严重时基于超声波进行缓慢停车规避事故

**通信方式：**

通过USB串口通信实现上下位机信号传输，在检测到疲劳后发出警报及缓停

**硬件设备：**

通过Arduino单片机接收信号，模拟实际汽车行为



添加：**超声波模块**

**前端socket通信等**

**7.17**

考虑替换人脸检测模型

https://github.com/1adrianb/face-alignment

https://github.com/deepinsight/insightface/

**7.18**

考虑打印各个函数的运行时间，排查运行速度慢的代码并进行针对性优化

函数分为7个模块：推理时长约120-150ms为固定时长，难以优化**（第一个模型或许可以）**

|  |
| --- |
| Python module1 44 ms # pre1  module2 88 ms # inf of 1---------------------- module3 24 ms # post1 + pre2  module4 36 ms # inf of 2---------------------- module5 185 ms # post2 + pre3  module6 23 ms # inf of 3---------------------- module7 7 ms # post3  total 接近400 ms |

去除第三个模型的推理和前后处理：

|  |
| --- |
| Python module1 44 ms # pre1  module2 88 ms # inf of 1---------------------- module3 24 ms # post1 + pre2  module4 36 ms # inf of 2---------------------- module5 18 ms # post2 + pre3  module6 0 ms # inf of 3---------------------- module7 7 ms # post3  total 接近200 ms |

部署在摄像头上：

|  |
| --- |
| Python module1 13 ms # pre1  module2 88 ms # inf of 1---------------------- module3 14 ms # post1 + pre2  module4 36 ms # inf of 2---------------------- module5 18 ms # post2 + pre3  module6 0 ms # inf of 3---------------------- module7 1 ms # post3  total 接近 170 ms |

实际部署下延迟已经可以接受

**7.25国赛器件准备与方向确定**

**考虑添加生理测量方法：毫米波雷达无感式测量心率+呼吸，进而判定疲劳（蓝海市场）**

基于心率和呼吸的驾驶疲劳检测论文https://aammt.tmmu.edu.cn/Upload/rhtml/202203057.htm

通过TI毫米波模块实现心率和呼吸测量

https://blog.csdn.net/Poulen/article/details/127502203

https://blog.csdn.net/m0\_61934621/article/details/132047756

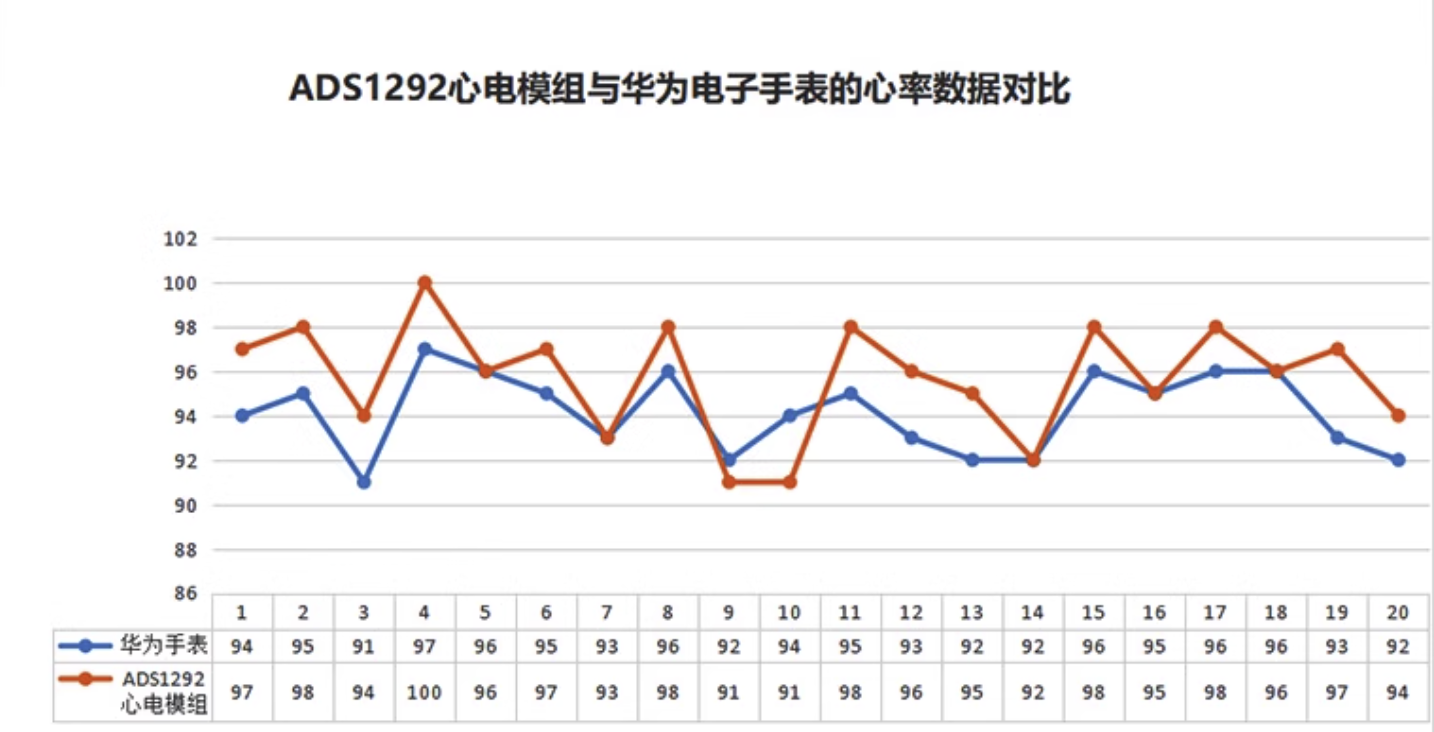
**规划器材：**

DCA1000 评估模块可实现实时数据捕获和流式传输https://www.ti.com.cn/tool/cn/DCA1000EVM

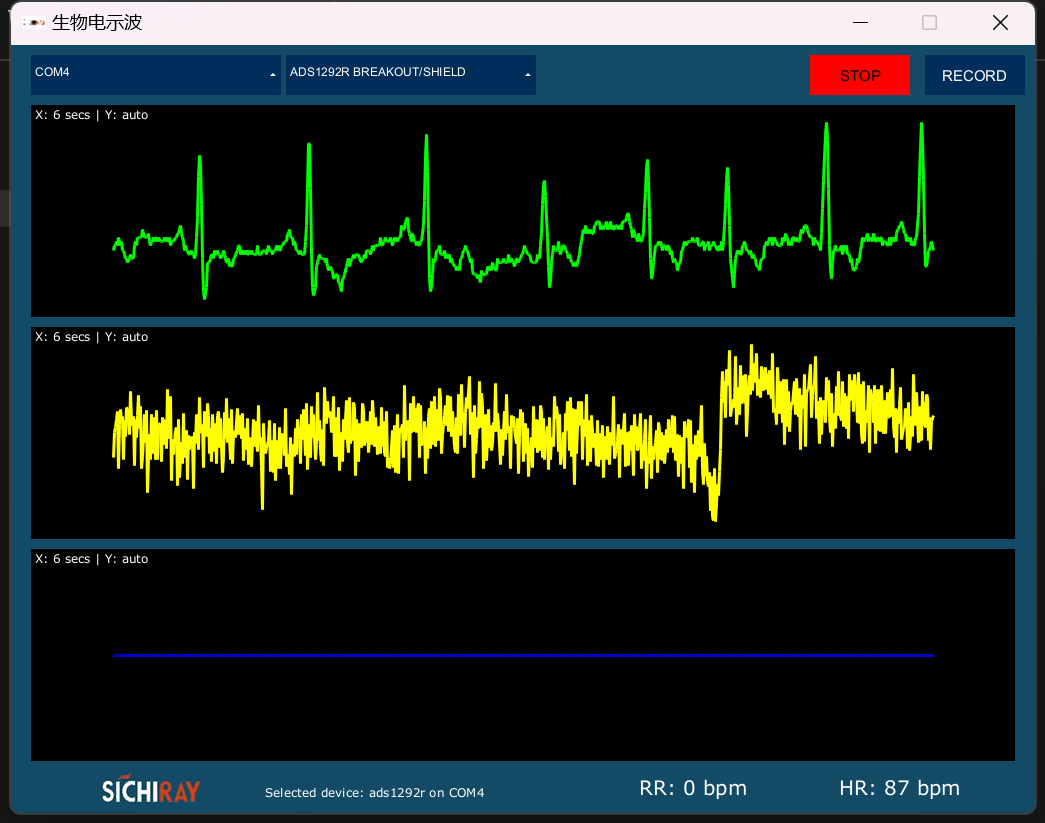
AWR1843 集成 DSP、MCU 和雷达加速器的单芯片 76GHz 至 81GHz 汽车雷达传感器https://www.ti.com.cn/product/cn/AWR1843?bm-verify=AAQAAAAJ\_\_\_\_\_zY68gqCVy9U2gYRjDHRaWQP6KGzFG5Zy80narzlSysB2To5bQtSuUEk3raxgl8jSW1YFM1wTaryGyfaHEcBCyTOUZrciHiHmFm6v8f6saln4BB1h-06cJyGZTAZ1K1WsE6xld3SLhvKrRg49YLl48xwHLQg3UpG6QIHz7QoO39eTTe-7BOJggyZ48tYR\_drOY53RvyerZpVXfMiNqqnlu\_eHLvHyaT\_Mh20SSYu8N1KMOmWm043NX7q8inJ\_Z28TeDMpvC6Xmhf2z8lZnx3LL6QqlW9ihlA0NF\_3zDQ7iBt3bStoIbOBYYi

IWR6843 单芯片 60GHz 远距离天线毫米波传感器评估模块https://www.ti.com.cn/tool/cn/IWR6843ISK

ADS1292



这张图可以放在报告里（？）



**外接显示器不显示问题**

https://blog.csdn.net/liu520888/article/details/128814161

https://blog.csdn.net/cylinc/article/details/121311586

https://blog.csdn.net/qq\_38502736/article/details/103734105

https://blog.csdn.net/myopengl0616/article/details/134544933

https://blog.csdn.net/qq\_44940689/article/details/132508548?spm=1001.2014.3001.5506

https://www.cnblogs.com/imzhi/p/15747434.html

**内存清理**

https://blog.csdn.net/Gavinmiaoc/article/details/80527717

https://www.sysgeek.cn/ubuntu-free-up-space/

一个可以找到所有占内存较大文件的指令

du -sh /\* 2>/dev/null

内存情况：

linaro@bm1684:/home$ du -sh /media/root-rw/overlay/\* 2>/dev/null

1.9M /media/root-rw/overlay/etc

8.0K /media/root-rw/overlay/factory

1.8G /media/root-rw/overlay/home

16K /media/root-rw/overlay/hostname

20K /media/root-rw/overlay/media

8.0K /media/root-rw/overlay/mnt

68K /media/root-rw/overlay/opt

8.0K /media/root-rw/overlay/root

56K /media/root-rw/overlay/tmp

2.4G /media/root-rw/overlay/usr

928M /media/root-rw/overlay/var

linaro@bm1684:/home$ du -sh /media/root-rw/overlay/usr/local/\* 2>/dev/null

383M /media/root-rw/overlay/usr/local/VSCode-linux-x64

64K /media/root-rw/overlay/usr/local/bin

408M /media/root-rw/overlay/usr/local/lib

254M /media/root-rw/overlay/usr/local/python3

2.1M /media/root-rw/overlay/usr/local/share

270M /media/root-rw/overlay/usr/local/trash:

于是删除了/media/root-rw/overlay/usr/local/trash:下所有有权限删的文件，腾出200M左右空间

vscode和python占内存较大。可以把其挂到/data或者/media/root-ro目录下。之后的模型文件建议放在这里

**系统版本号及内核信息（查询用）**

linaro@bm1684:~$ hostnamectl

Static hostname: bm1684

Icon name: computer

Machine ID: 1eef0b5f42d64d108c592bddfbc61a20

Boot ID: f04154fa53274a9daae62042d8f41276

Operating System: Ubuntu 20.04 LTS

Kernel: Linux 5.4.217-bm1684-g70a3e20060f6-dirty

Architecture: arm64（重要 nnd）

**检查串口代码**

计数目录下的所有文件数量

-1 /dev/ | wc -l

将目录下的所有文件名保存到一个文件夹

/dev/ > /data/current\_files.txt

对比两个文件夹的东西

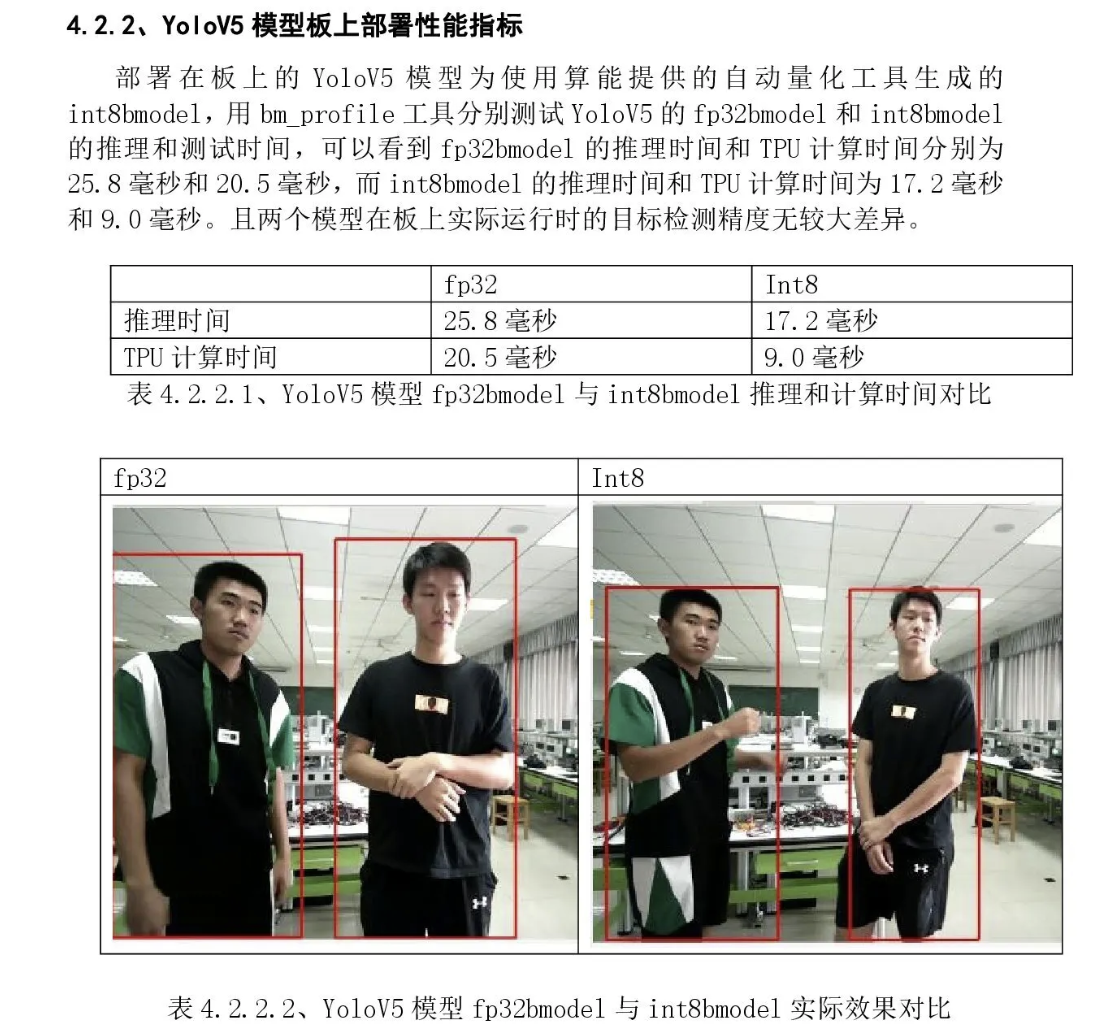
diff /data/previous\_files.txt /data/current\_files.txt | grep '^>'

**国一报告一些亮点**

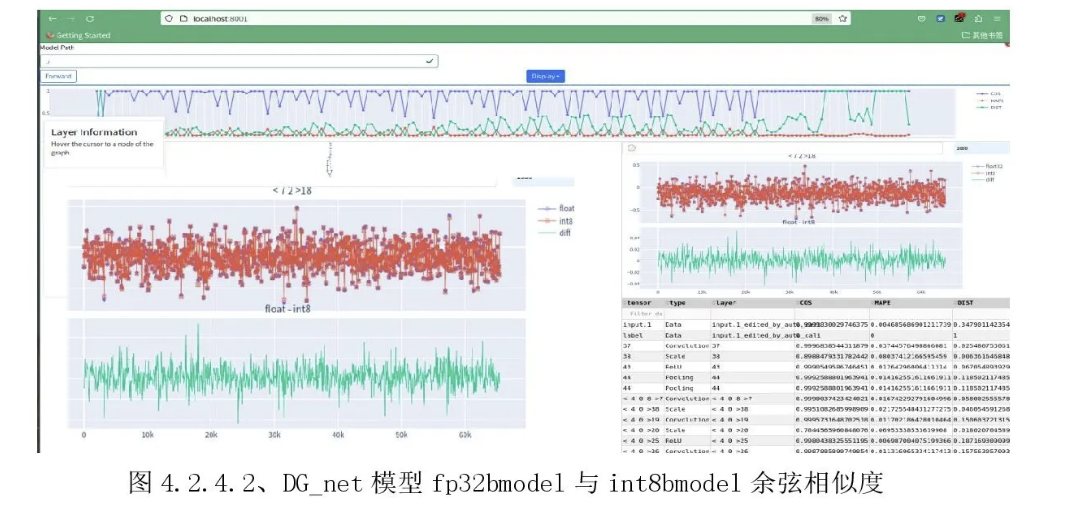
**我们报告中的架构图也可以加上数据库**



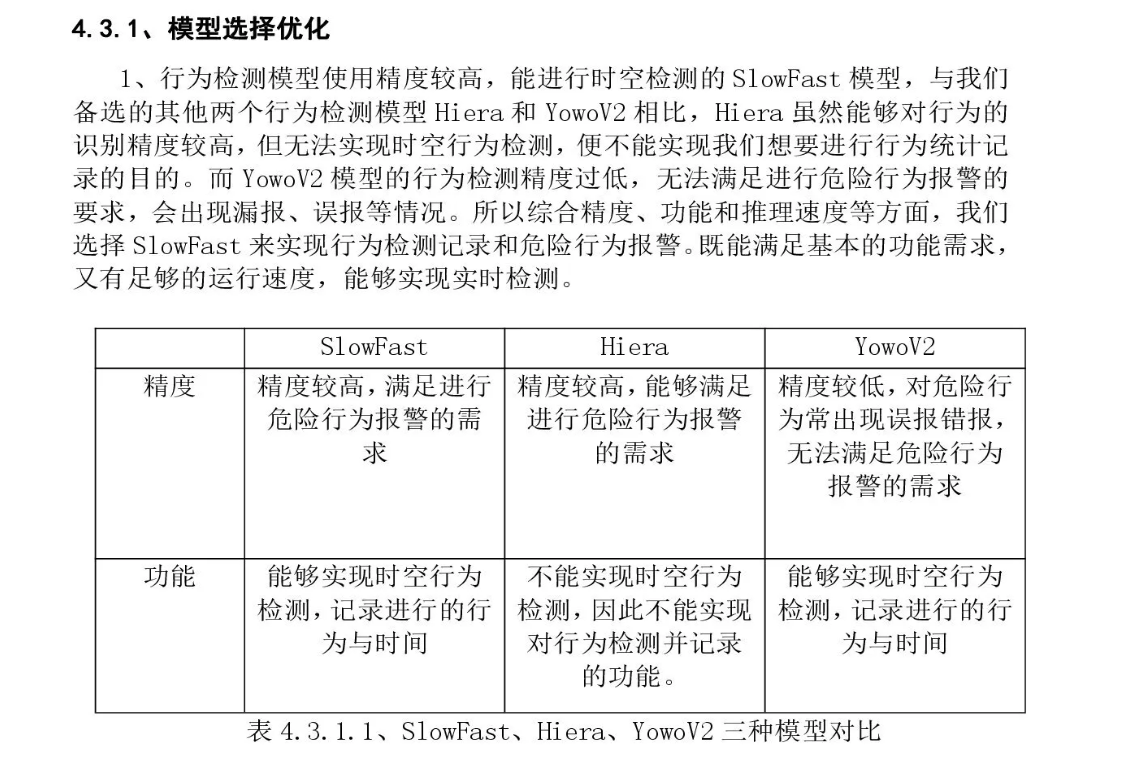
两个模型的对比：



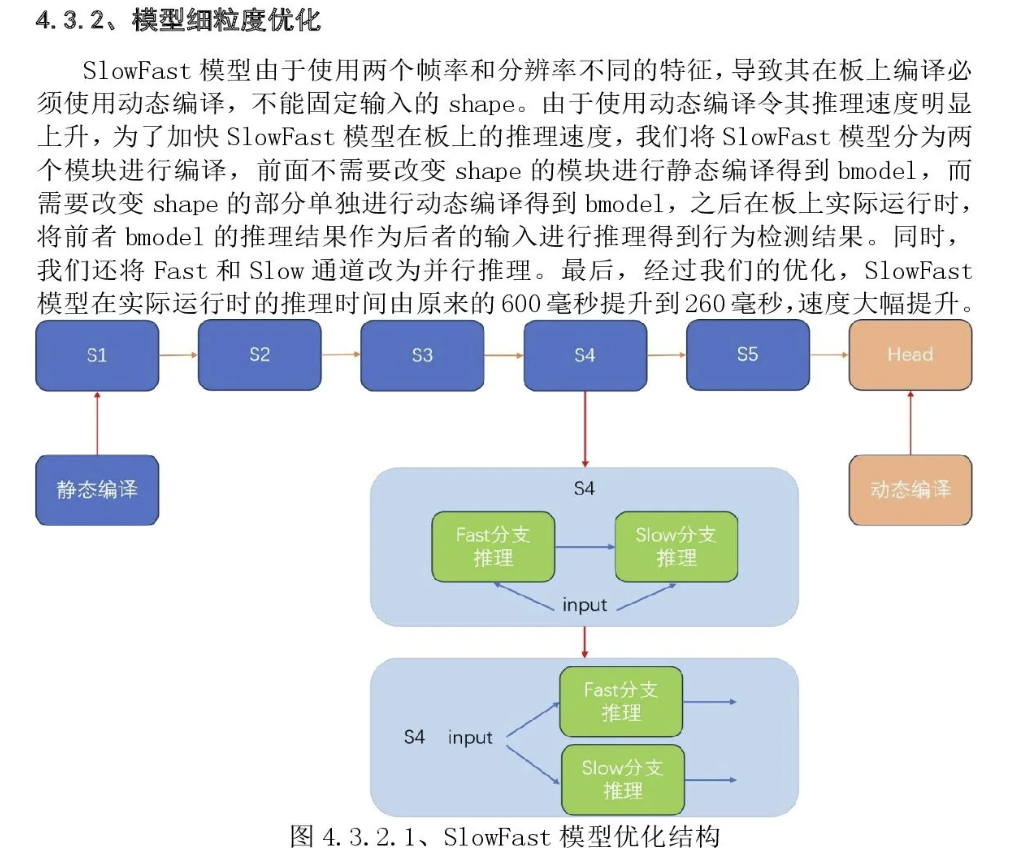
用余弦相似度来对比两个模型之间推理的区别



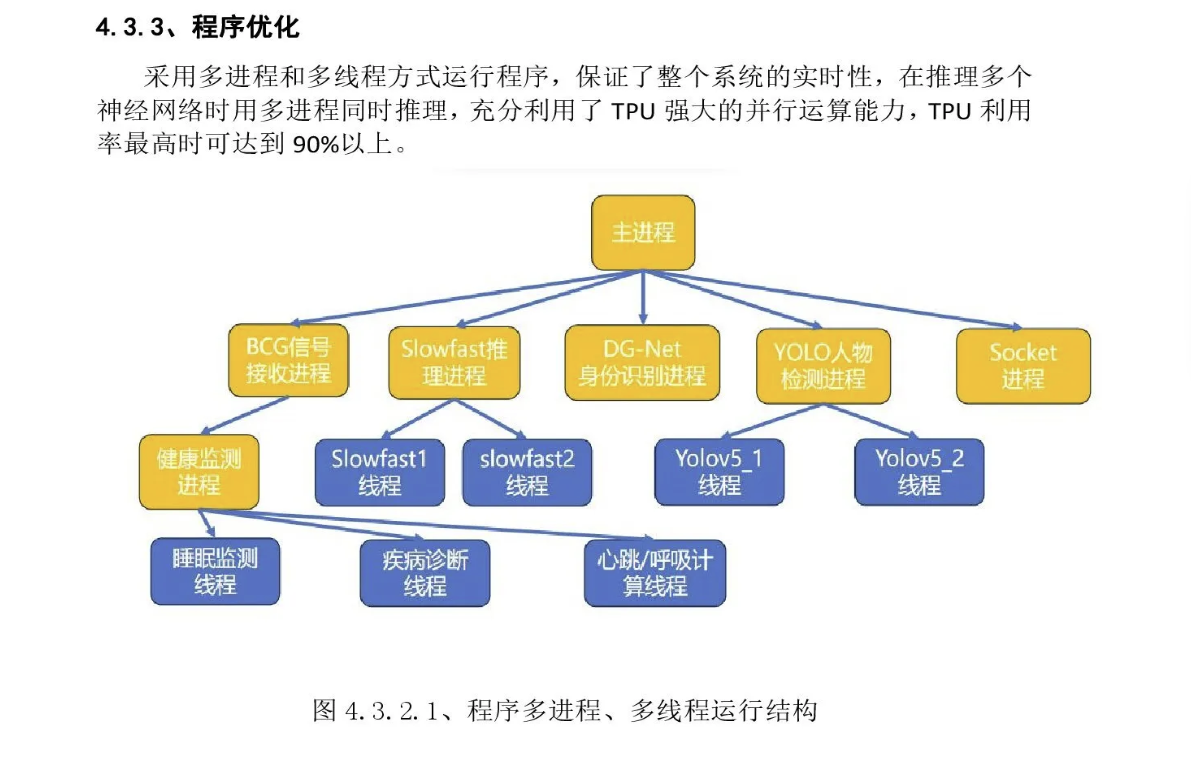
不同模型的选择对比：

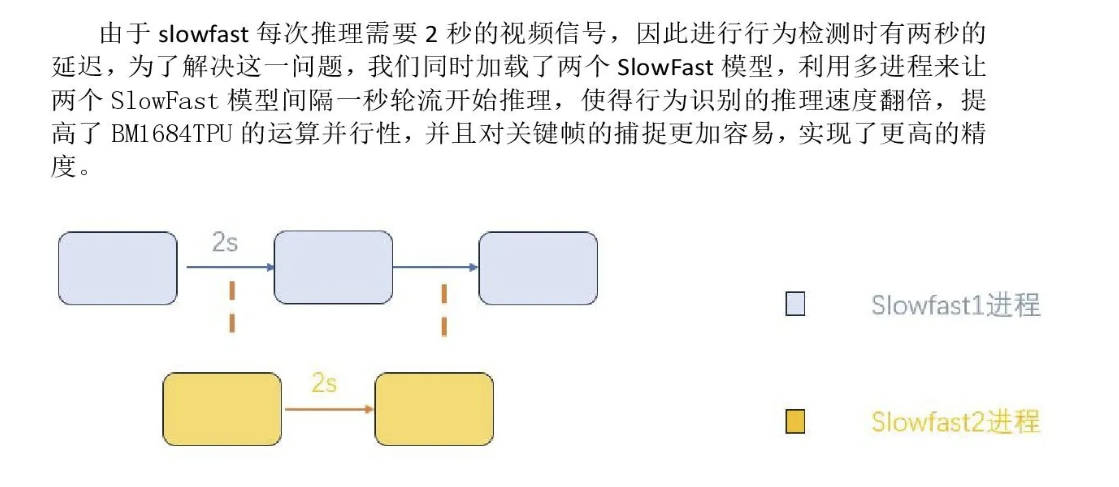


改变shape的部分？



多线程运行结构：





相较于省赛添加的模块及内容：

1. **更改RGB摄像头为红外摄像头：**  
   针对红外与RGB两种色彩空间对于视觉模型进行训练
2. **添加心率、呼吸等生命体征检测模块：**  
   利用生命体征检测驾驶员状态，利用时间序列模型可以预测心率不齐等特征，进行健康检测
3. **添加语音播报模块：**  
   将原先的蜂鸣器警报更改为AI合成的语音播报，提高用户体验
4. **完善前端页面和服务器通信：**  
   将边缘设备实时捕获到的视频流和驾驶员信息实时传输到服务器端进行显示和数据统计
5. **真车实地部署尝试：**  
   在真实车辆中进行设备部署，进行实地驾驶实验，记录设备表现（待定）

算能张老师建议：

张老师很切中肯綮的锐评，直言不讳很有参考价值，不过咱不用因此失去信心，加油~

* ~~1、注意一下汇报时间：前面的内容精简一些 后面的内容扩展一下 项目背景那一块有一个一页两页就够了。~~
* ~~产品介绍时间：前面缺少一个部分 就是项目概览那一块 有一个整体功能的展示和软硬件方案的展示。硬件方案是什么样的？软件方案是什么样的？前面得让老师知道我们做的是什么硬件。哪怕是第18页的图也不太清晰 把这些实物图换成框图就行了 外设其实不需要列的这么详细。18页右边这个表稍微有点割裂的感觉（张老师评价）。~~
* 2、18页有点乱 很confuse(草我觉得不乱啊）
* 3、确实要讲模型 不过不用一个模型分两页来讲 其实没太讲出来模型的特点是什么 参考文献和图对评委老师来说没有啥意义。只用展示算法的一些对比（cpu和少林派的对比）
* 4、有每个算法的速度 但是还没有给出整体的速度？加载多少毫秒？一秒钟能处理多少帧？
* 5、丢帧怎么办？用了什么并行的技术？
* 6、项目创新那一页ppt其实没太体现出创新点。张老师同意将心率模块的使用作为创新点（功能模块的创新）
* 7、TPU利用率优化没有运行时间的优化重要
* 8、竞品分析直接写市面上的产品 品牌 直接介绍有点泛
* ~~9、不要团队介绍了 不要最好不要看出来自己是什么学校的 也不能说~~
* ~~10、未来展望意义页不是很大，直接说现在达到的状态和未来想做什么。~~
* 11、注意演示时间 多留时间和评委互动 现场演示可以提前开始
* 12、往成熟工业化方向发展 也是很不错的方向
* 13、功能不需要堆数量，感觉这是之前的评委夸我们时间很短实现了很多东西的话外音。
* 14、重点考虑现场演示效果——高鹏

总结一下就是：少讲技术细节（虽然很想讲但是要少讲） 多讲创新点 工作量不是重点