如何测试模型推理时，模型使用内存情况：

目前的理解及问题：

1. 模型已优化为tensorrt模型
2. 已将模型部署到nvidia平台，但是‘所测试的结果与预期不符合

模型进行推理时，所做的步骤：

1. 将序列化的tensorrt模型进行反序列化，
2. 创建context
3. 创建输入、输出数据缓存
4. 推理

从2项开始获取当前内存使用情况（use1）-----4项结束内存使用情况（use2），求差值

当前获取到的内存使用情况如下：

1、 32精度 700M

16精度 30M

8精度的模型 30M

2022-04-24 星期日

TensorRT优化改进方向：

1. 增加从.pt转化为uff格式、uff转trt、onnx转trt的说明；
2. 增加模型解密代码；
3. 增加tensorRT模型的直接加载；
4. 增加tensorrt进行推理的测试用例；
5. 增加转换后的trt模型使用测试样例；
6. 如何适配不同模型的推理前处理；

2022-04-27 星期三

16G预览界面问题汇总

172.97.0.1

/data/home/zwhu/pupil/MNN/build/MNNConvert -f ONNX

--modelFile ./CNN\_Classified\_ColorJitter140\_cv.onnx --MNNModel classify\_v12.mnn --bizCode MNN

---tf to onnx

python3 -m tf2onnx.convert --input ./mobilenet\_v2\_1.4\_224\_frozen.pb --inputs input:0[1,224,224,3] --outputs MobilenetV2/Predictions/Reshape\_1:0 --output mobile\_v2.onnx

trtexec --onnx=mobile\_v2.onnx --saveEngine=mobile\_v2\_fp16.trt --fp16

**2022-05-12**

获取远程分支，然后进行提交的步骤

1. git clone url
2. git checkout branch\_name(远程分支名，不用加origin)
3. 修改完成之后
4. git add -A
5. git commit -m “xx”
6. git push

**2022-05-13**

**RAC会议**

tensorrt前处理（提供全面的处理），后处理（提供简单的）。

**RM20A椭圆信息**

opencv中的椭圆通过RotatedRect来定义，其center(块中心(x,y)), size(宽和高,宽小于高), angle（旋转角）分别对应椭圆的中心，短轴和长轴，旋转角度。

1. 在Matlab验证的基础上，完成算法的C++化； 2. 建议通过一个PPT，给大家展示一下你当前整个算法的流程，以及每个中间流程的结果

FD16G---预览界面，后续计划：7

Visual studio安装编译器

将已有界面封装，报告查看；

Qt控件写界面，获取信息；

**RM20A：**

柱镜-

+10D、-10D不一致需要针对处理

从理论分析计算屈光值的方法：

1. 首先根据在零焦面得到的人眼反射光线得到圆环

可能的误差：是否在零焦面上

1. 根据圆环内径计算屈光值

圆环内径容易产生误差（计算方式是以假眼零焦面零屈光为基准进行拟合得出），若测试时眼睛未在零屈光位置，圆环内径误差变大。

1. 马达移动到屈光值对应位置

却决于屈光值计算得是否准确，如果不准，可能采集到得图片不是需要的，无法达到最清晰得位置。

1. 采集马达附近圆环数据
2. 拟合（马达—清晰度）曲线，求得曲线中清晰度最高值对应的马达步数

需要对采集数据进行处理，识别一组数据中椭圆，以置信度最高得椭圆为基准，对其余图片进行旋转，裁剪长轴方向和短轴方向得图片，分别进行拟合。

误差：椭圆识别精度，裁剪基准，清晰度识别算法

1. 根据马达步数与屈光值得出（马达步数—屈光值）曲线。

2022年6月2日 星期四

1. 将模块进行划分（椭圆检测、屈光值计算），提交代码
2. 找出opencv多项式拟合接近matlab拟合的方法
3. 将马达步数细分为原来的10倍

2022年6月7日 星期二

1、对圆环图片进行分类，-10D~+10D -20D~-10D 10D~20D

如果该图片1类和3类为不可计算状态，返回结果，2类可计算，进行计算，返回结果

2022年6月8日 星期三

1. 先搞准-15~15D图片的精度
2. 对椭圆进行细分能够识别15D~20D范围内的圆环

2022年6月10日 星期五

1. 完成rac算法的大致流程跑通；（信息）
2. 进入测试（信息）

2022年6月11日 星期六

1、15-20D解决wd时无法计算的情况；

2、解决精度的问题；

3、整理代码，保证逻辑和结构；

2022年6月13日 星期一

本周任务

1. 继续测量rm20a项目，优化结构和代码，多对其它数据进行测试，保证不产生问题；
2. 对rtsp进行编译及测试64位；
3. 学习tensorrt（暂定）。

2022年6月20日 星期一

1. 确定rm20a散光和轴位角值是否满足要求；
2. 进行rtsp进行编译及测试64位；
3. 编写第三季度okr和第二季度okr；

2022年6月24日星期五

Rm20A设备操作指令

algofc

lson l1on l2on xf xb yf yb

2022年6月30日 星期四

1、光学实验数据拟合处理；

2、模型单独分离，模型加密，解密、加配置文件

8MP设备连接

登录：root，1

sz将数据传输到主机

rz将数据从主机拉到开发板

ifconfig eth0 192.168.0.101

2022年7月6日 星期三

1. 优化屈光值计算时间；
2. 解决人眼拍摄不清晰、10柱镜拍摄不清晰的问题；
3. 优化

提前推动20A相关处理；

Ffmpeg gstream,技术细节、usb协议、

Nxp、v5平台nvidia平台

2022年7月9日 星期六

尝试学习Android的开发，进行技术储备；

2022年7月11日 星期一

Ffmpeg进行拉流显示处理

2022年7月20日 星期三

对收集到的副摄图像根据瞳孔识别结果进行裁剪，直接显示；

裁剪上下两部分，进行拼接；使用ffmpeg对视频进行处理。

2022年8月6日 星期六

EVT-3硬件，算法适配，人眼拍摄；

2022年8月12日 星期五

1. 柱镜度数适配；
2. Aie接口修改；

下周计划；

AIE接口测试，发布版本；

2022年8月23日 星期二

1. 将流获取封装为库：传入url、句柄、类型（主摄、副摄）
2. 增加异常信息输出

2022年8月26日 星期五

1. 将ffmpeg获取rtsp流的程序封装为动态库；
2. 获取rtmp流，进行显示；
3. 熟悉v4l2框架信息
4. 下周支持立体视盘的工作。