



EyecLOUD OpenNCC AI Model Integration

User Guide

August 2021

Revision 1.0.0

技术支持

扫描下方二维码“eyecloud 小助手”二维码添加微信好友，成功后小助手会邀请您入群。

申请请备注“官方文档”

eyecloud小助手



联系方式

电话：0571-8535-2616

邮箱：info@eyecloud.tech

版权声明

本手册版权归杭州眼云智家科技有限公司所有。未经本公司书面许可，任何单位及个人无权以任何形式复制、传播、转载本手册的任何内容，违者将被追究法律责任。

修订历史

Version	Date	Editor	Description
1.0.0	Aug. 2021	Zed	

目录

技术支持.....	2
联系方式.....	2
版权声明.....	2
修订历史.....	3
目录.....	4
第一章 介绍.....	5
第二章 基于已有解析函数的集成.....	5
2.1 解析方法及模型类别.....	5
2.2 代码修改及编译.....	5
第三章 基于新的解析函数集成.....	8
常见问题.....	8

第一章 介绍

新模型集成分为两种情况：

- 基于已有解析函数。
- 基于新的解析函数。

第二章 基于已有解析函数的集成

2.1 解析方法及模型类别

目前具备的 metadata 解析方法包括以下几种：

- 物体分类：classification
- 目标识别：人脸识别、行人识别、车牌的一级模型
- 文字分类：车牌二级模型
- 概率图分割：road-segmentation

2.2 代码修改及编译

在 widget.cpp 第 438 左右的 blobVecs 容器的字符容器末尾，加上新模型的地址。(你可以把接下来要集成的模型放在 Configuration/blob/目录下，与其他模型统一管理。)

```
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455

static std::vector<QString> blobVecs={
    "",
    "/Configuration/blob/classification-fp16.blob",
    "/Configuration/blob/face-detection-retail-0004-fp16.blob",
    "/Configuration/blob/face-detection-adas-0001-fp16.blob",

    "/Configuration/blob/person-detection-retail-0013-fp16.blob",
    "/Configuration/blob/person-vehicle-bike-detection-crossroad-0078-fp16.blob",
    "/Configuration/blob/face-person-detection-retail-0002-fp16.blob",
    "/Configuration/blob/pedestrian-detection-adas-0002-fp16.blob",
    "/Configuration/blob/vehicle-detection-adas-0002-fp16.blob",
    "/Configuration/blob/vehicle-license-plate-detection-barrier-0106-fp16.blob",

    "/Configuration/blob/pedestrian-and-vehicle-detector-adas-0001-fp16.blob",
    "/Configuration/blob/road-segmentation-adas-0001.blob"
};
```

然后在 493 行左右，加入新模型序号的判断，使 fun 函数指针指向对应的解析函数。(例如上图中 road-segmentation 模型的序号是 11，则添加一个判断让 index==11 时，fun 指向 segmentation_func。)

```

492
493
494
495
496
497
498
499
500
501
502

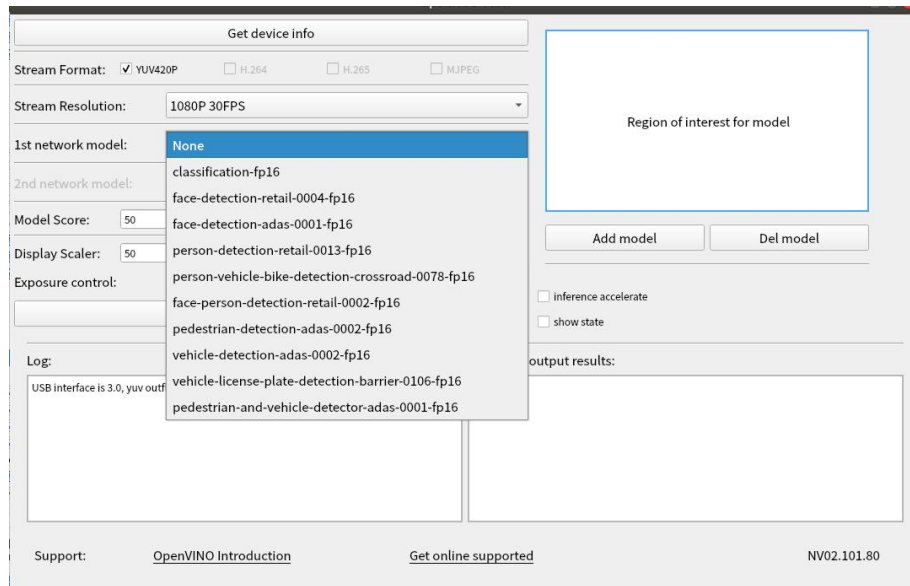
```

```

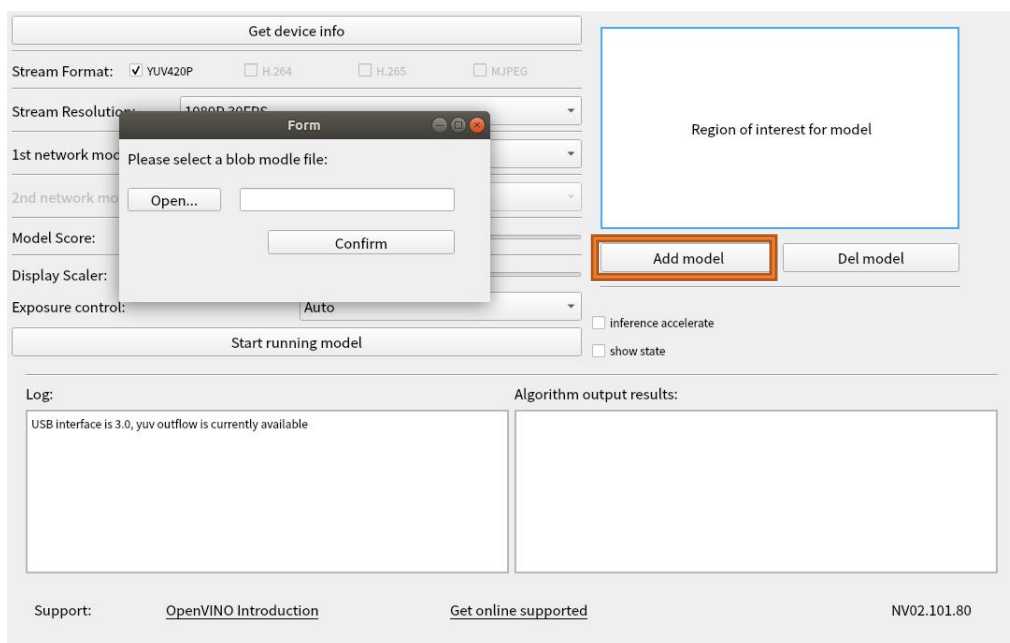
if(1 == index){
    fun = cls_show_img_func;
}else if(2 == index){
    fun = fd_show_img_func;
}else if(11 == index){
    fun = segmentation_func;
}else{
    fun = obj_show_img_func;
}

```

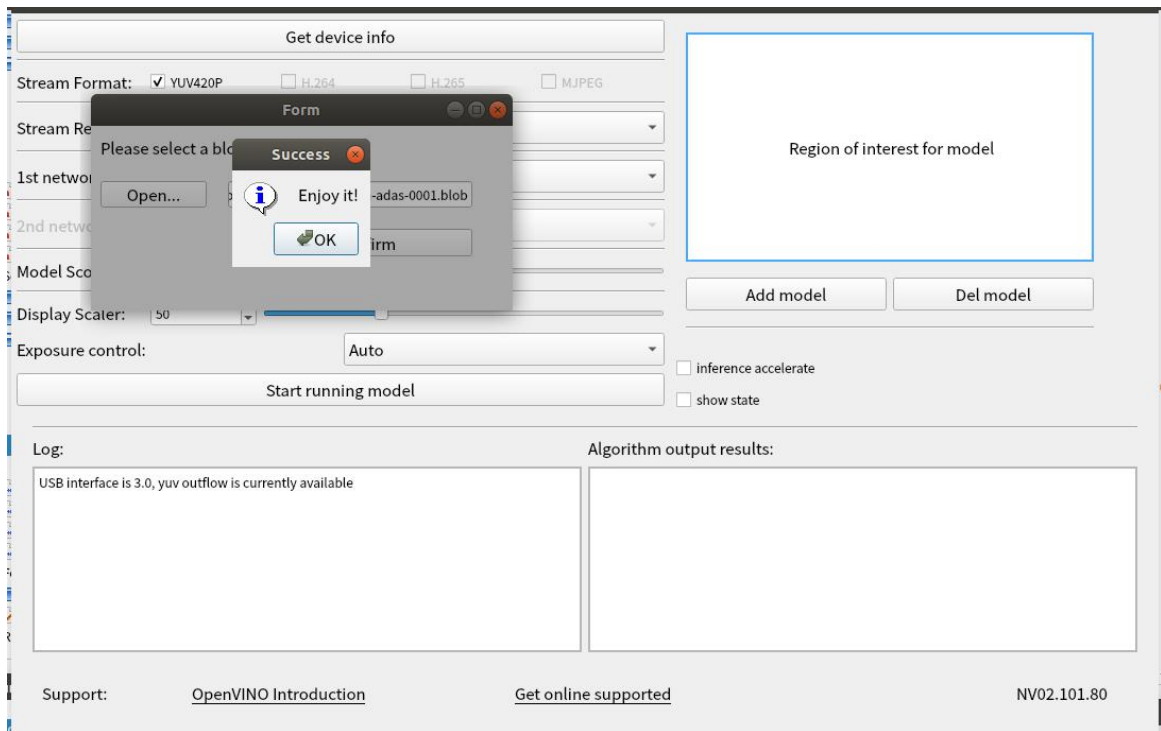
接着，构建运行程序。(此时的下拉菜单中不会出现新模型！)



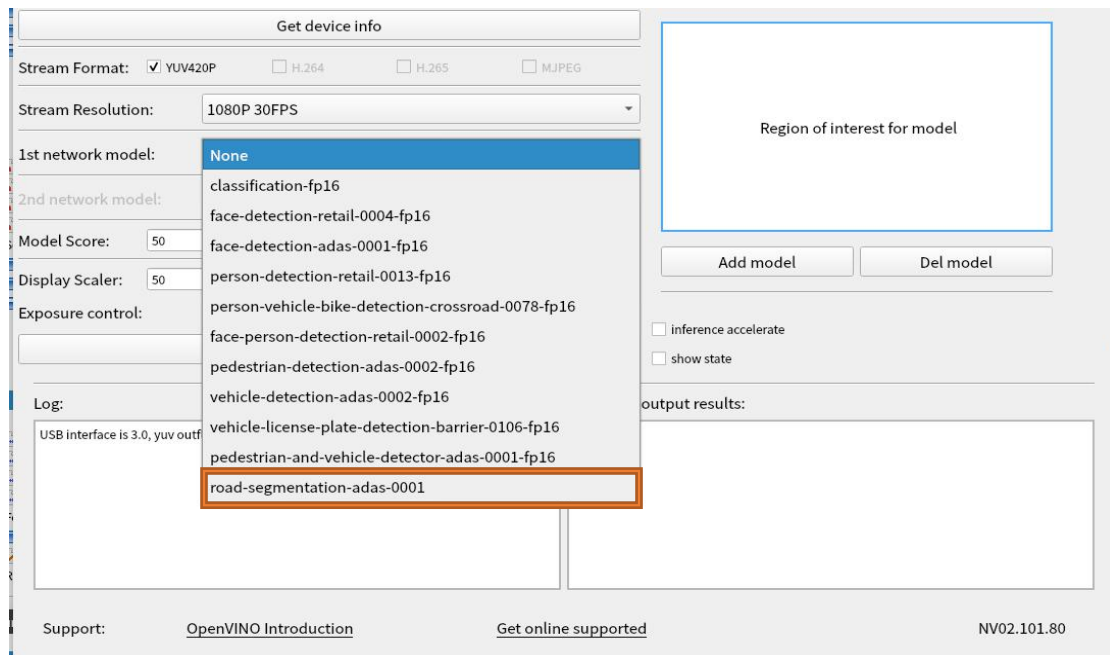
所以，你需要点击 view 界面右侧|Add model|按钮。



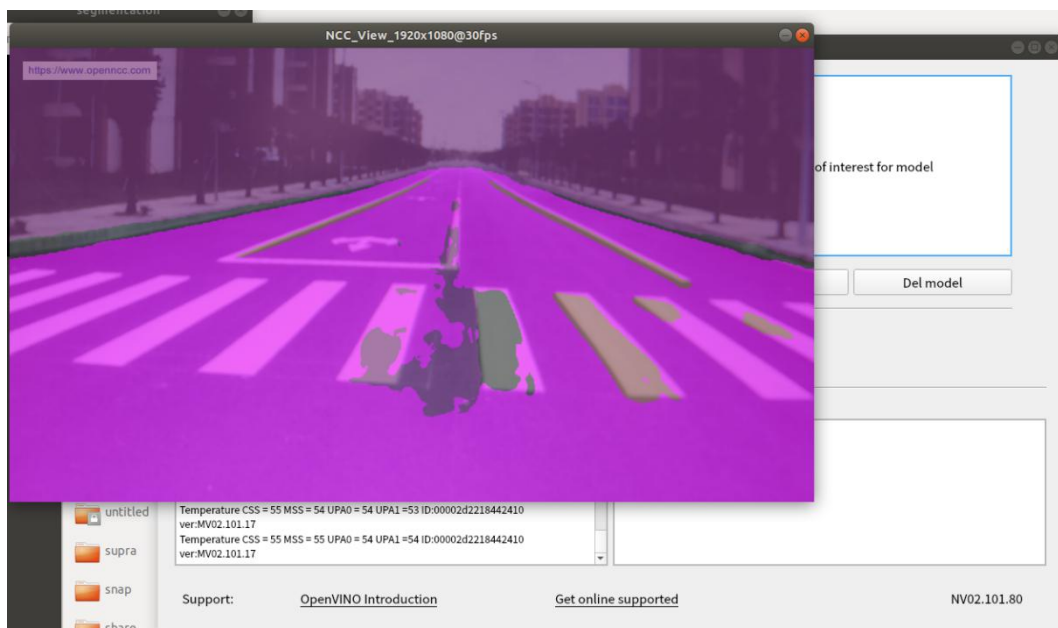
找到新模型，点击 **Confirm** 确定。



至此，你就可以在|1st network model|的下拉菜单中找到你刚刚添加的新模型并使用了。(可以看到，不算 None 的情况下，新模型排在菜单的自上而下第 11 位，这个数值必须和代码中模型的 index 一致。)



最后可以看一下模型的运行效果，确保新模型成功集成。



第三章 基于新的解析函数集成

注意在判断序号时，要让 fun 函数指向你自写的解析函数。

解析函数的参数需要符合指定模板：

void fun(void* yuv 图像数据, int 图像宽, int 图像高, float 图像缩放比例, char* 显示窗口名, int 算法有效区显示, Network1Par* 模型参数结构体, char* metadata 数据, float 算法置信度, int 算法帧时间, int 分辨率模式, char*设备 id, bool 状态信息标志, int 流帧时间)

常见问题

- sdk 参数配置和模型版本问题。当前 View 的 sdk 只支持 2020.3 版本的 blob 模型。以上两种问题均会倒是 sdk 初始化失败，打印提示 “failed to init the device,ret=-1”。
- metadata 导出错误或解析错误。存在某些特殊情况可能会造成其他格式模型转化为 blob 格式时，虽然转化成功，但模型无法正确运作，从而导出错误的 metadata 数据。另外也有可能该模型的结果数据超出了 metadata 缓存总长，比如列表形式结果通常较短，而概率图结果较长，这点需要注意。meatata 静态缓存的声明在 widget.cpp 第 76 行。至于解析问题，可参考 sdk 目录下已有的几种解析方法修改。