# 车辆检测

开发者将本Application部署至Atlas 200 DK或者AI加速云服务器上实现对本地mp4文件或者RTSP视频流进行解码，对视频帧中的车辆并对其属性进行预测，生成结构化信息发送至Server端进行保存、展示的功能。

前提条件

部署此Sample前，需要准备好以下环境：

* 已完成Mind Studio的安装。
* 已完成Atlas 200 DK开发者板与Mind Studio的连接，交叉编译器的安装，SD卡的制作及基本信息的配置等。

软件准备

运行此Sample前，需要按照此章节获取源码包，并进行相关的环境配置。

获取源码包。

将<https://github.com/Ascend/sample-videoanalysiscar>仓中的代码以Mind Studio安装用户下载至Mind Studio所在Ubuntu服务器的任意目录，例如代码存放路径为：*/home/ascend/sample-videoanalysiscar*。

获取此应用中所需要的原始网络模型。

参考表1.1获取此应用中所用到的原始网络模型及其对应的权重文件，并将其存放到Mind Studio所在Ubuntu服务器的任意目录，例如$HOME*/ascend/models/car\_color*。

开源Application中使用模型

| 模型名称 | 模型说明 | 模型下载路径 |
| --- | --- | --- |
| car\_color | 车辆颜色识别模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/car\_color目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| car\_type | 车辆品牌识别模型。  基于Caffe的GoogleNet模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/car\_type目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| car\_plate\_detection | 车牌检测网络模型。  基于Caffe的Mobilenet-SSD模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/car\_plate\_detection目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| car\_plate\_recognition | 车牌号码识别网络模型。  基于Caffe的CNN模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/car\_plate\_recognition目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| vgg\_ssd | 目标检测网络模型。  基于Caffe的SSD512模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_visionobject\_detect/vgg\_ssd目录中README.md下载原始网络模型文件及其对应的权重文件。 |

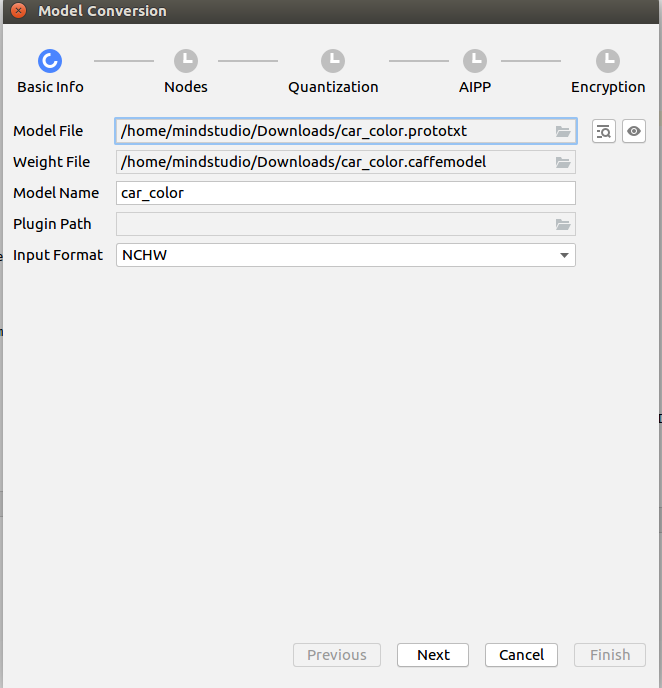
将原始网络模型转换为Davinci模型

1. 在Mind Studio操作界面的顶部菜单栏中选择“Tool > Convert Model”，进入模型转换界面。
2. 在弹出的**Convert Model**操作界面中，Model File与Weight File分别选择步骤 2中下载的模型文件和权重文件。

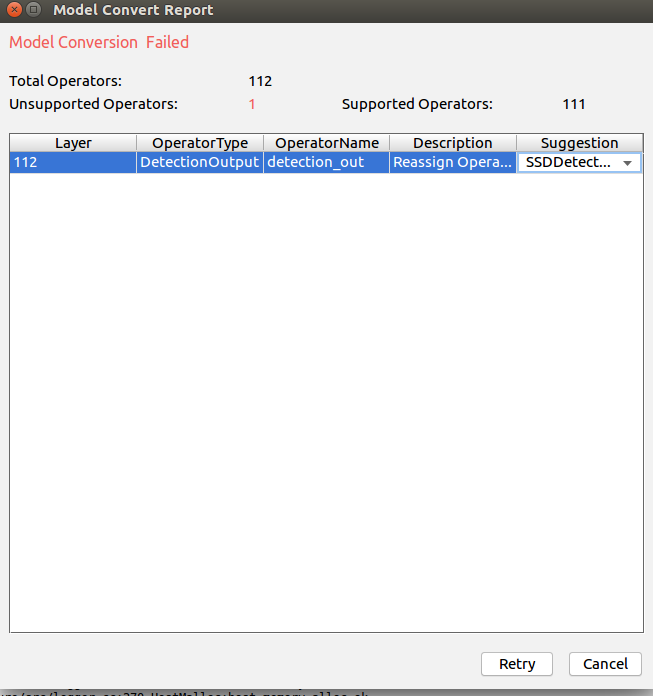
Model Name填写为表1.1对应的模型名称。

Car\_color模型中car\_color\_inference一次处理10张图片，所以转换时需要将Input Shape的N修改为10。

car\_type模型中car\_type\_inference一次处理10张图片，所以转换时需要将Input Shape的N修改为10。



1. 点击OK开始转换模型。

car\_plate\_detection模型在转换的时候，会有报错。报错信息如下所示

此时在DetectionOutput层的Suggestion中选择SSDDetectionOutput，并点击Retry。

模型转换成功后，存放地址$HOME/modelzoo/car\_plate\_detection。

将转换好的模型文件（.om文件）上传到“sample-videoanalysiscar/src/models”目录下。

以Mind Studio安装用户登录Mind Studio所在Ubuntu服务器，并设置环境变量DDK\_HOME, NPU\_DEVICE\_LIB和LD\_LIBRARY\_PATH。

**vim ~/.bashrc**

执行如下命令在最后一行添加DDK\_HOME及LD\_LIBRARY\_PATH的环境变量。

**export DDK\_HOME=$HOME/.mindstudio/huawei/ddk/1.31.T9.B090/ddk**

export NPU\_DEVICE\_LIB=$DDK\_HOME/../RC/host-aarch64\_Ubuntu16.04.3/lib

**export LD\_LIBRARY\_PATH=$DDK\_HOME/lib/x86\_64-linux-gcc5.4**



* 如果此环境变量已经添加，则此步骤可跳过。

输入:wq!保存退出。

执行如下命令使环境变量生效。

**source ~/.bashrc**

添加软连接

进入到NPU\_DEVICE\_LIB的目录，为动态库protobuf添加软连接

cd `echo $NPU\_DEVICE\_LIB`

ln -s libprotobuf.so.18 libprotobuf.so



* 如果此软连接已经添加，则此步骤可跳过。

准备公共依赖代码库

在用户目录下任一位置建立文件夹，如$HOME/commen/ 通过此链接下载本程序所依赖的代码库ezdvpp, presentagent和ffmpeg的部署脚本func\_deploy.sh ,build\_ezdvpp.sh，build\_presentagent.sh 和build\_ffmpeg.sh

执行两个部署脚本:

bash build\_ezdvpp.sh  *host\_ip*

bash build\_presentagent.sh *host\_ip*

bash build\_ffmpeg.sh host\_ip

* *host\_ip*：Atlas 200 DK开发者板的IP地址。

命令示例：

* bash build\_ezdvpp.sh 192.168.1.2
* bash build\_presentagent.sh 192.168.1.2
* bash build\_ffmpeg.sh 192.168.1.2



* 在执行上述脚本前，请确保宿主机已经连接开发板，并且宿主机已配置完交叉编译环境。
* 如部署过ezdvpp, presentagent和ffmpeg ，上述步骤可以跳过

----结束

编译

打开对应的工程。以Mind Studio安装用户进入Mind Studio 安装目录，如/home/ascend/mindide/MindStudio-ubuntu/bin。执行./MindStudio.sh &开启Mind Studio打开***sample-videoanalysiscar***工程。

在配置文件中填写相关参数。在工程中找到src目录，找到param\_configure.conf文件。该配置文件内容如下：

remote\_host=

presenter\_view\_app\_name=

video\_path\_of\_host=

rtsp\_video\_stream=

需要手动添加参数配置：

* *remote\_host*：Atlas 200 DK开发者板的IP地址。
* presenter\_view\_app\_name: 用户自定义的在PresenterServer界面展示的View Name，此View Name需要在Presenter Server展示界面唯一。
* video\_path\_of\_host：为HOST侧的视频文件的绝对路径。
* rtsp\_video\_stream：为RTSP视频流的URL。



* 参数*remote\_host和*presenter\_view\_app\_name:必须全部填写，否则无法通过build。
* 注意所填参数不用使用“”。
* 注意presenter\_view\_app\_name必须是大小写字母、数字、“\_”的组合，位数为3-20位
* 参数video\_path\_of\_host和rtsp\_video\_stream必须至少填写一项

开始build。下在工具栏中找到build 单击Build-Configuration。会在工程目录下生成build和run文件夹。

启动Presenter Server

其中Presenter Server用于接收Application发送过来的数据并通过浏览器进行结果展示。

切换到命令行模式，进入到src目录下，例如：”sample\_videoanalysiscar/src”。执行如下命令在后台启动Video Analysis应用的Presenter Server主程序:

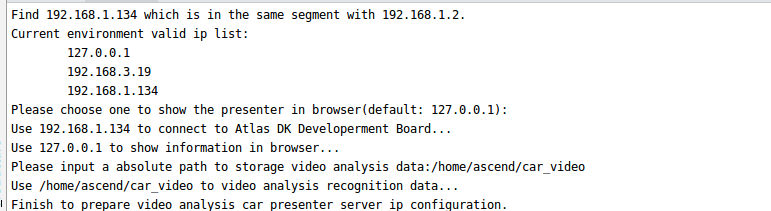
bash prepare\_present\_server.sh

当提示“Please choose one to show the presenter in browser(default: 127.0.0.1):”时，请输入在浏览器中访问Presenter Server服务所使用的IP地址（一般为访问Mind Studio的IP地址。）

当提示“Please input a absolute path to storage video analysis data:”时，请输入MindStudio中的绝对路径用于存储视频解析数据，此路径Mind Studio用户需要有读写权限，若此路径不存在，脚本会自动创建。

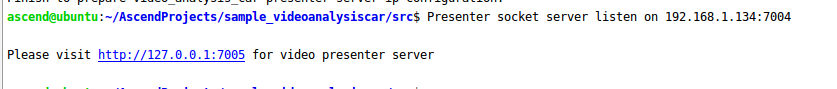
如图2.1所示，请在“Current environment valid ip list”中选择通过浏览器访问Presenter Server服务使用的IP地址，并输入存储视频解析数据的路径。

图4.1示意图



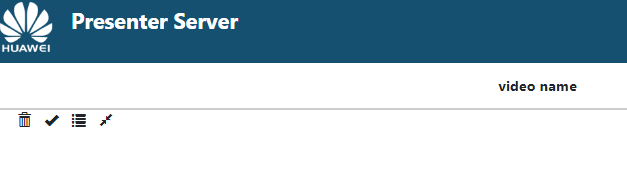
如下图所示，表示presenter\_server的服务启动成功。

图4.2Presenter Server进程启动



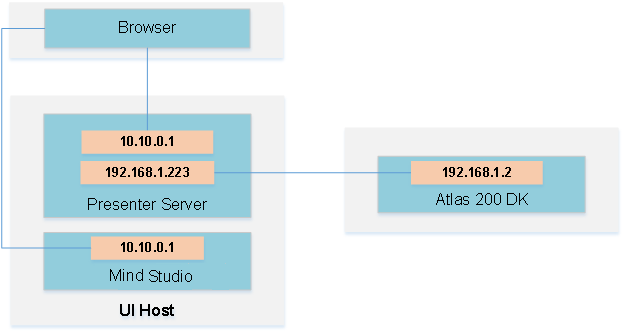
使用上图提示的URL登录Presenter Server，仅支持Chrome浏览器，IP地址为[步骤2](#d0e1864)中输入的IP地址，端口号默为7005，如下图所示，表示Presenter Server启动成功。

主页显示



Presenter Server、Mind Studio与Atlas 200 DK之间通信使用的IP地址示例如下图所示：

IP地址示例



* Atlas 200 DK开发者板使用的IP地址为192.168.1.2（USB方式连接）。
* Presenter Server与Atlas 200 DK通信的IP地址为UI Host服务器中与Atlas 200 DK在同一网段的IP地址，例如：192.168.1.223。
* 通过浏览器访问Presenter Server的IP地址本示例为：10.10.0.1，由于Presenter Server与Mind Studio部署在同一服务器，此IP地址也为通过浏览器访问Mind Studio的IP。

视频结构化应用支持解析本地视频和RTSP视频流。

* 如果需要解析本地视频，需要将视频文件传到Host侧。

例如将视频文件car.mp4上传到Host侧的“/home/HwHiAiUser/sample”目录下。



支持H264与H265格式的MP4文件，如果MP4文件需要剪辑，建议使用开源工具ffmpeg，使用其他工具剪辑的视频文件ffmpeg工具可能不支持解析。

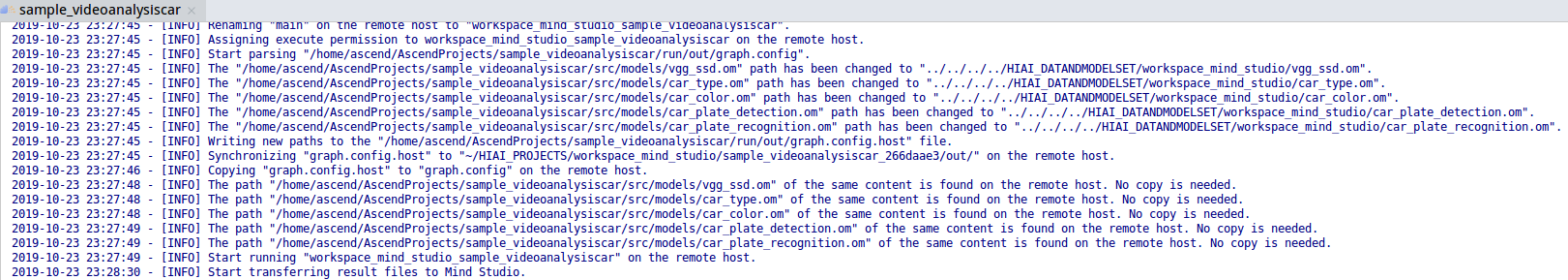
* 如果仅解析RTSP视频流，本步骤可跳过。

----结束

运行

在工具栏找到Run按钮，单击。

如下图所示，可执行程序已经在开发板运行。



使用启动Presenter Server服务时提示的URL登录 Presenter Server 网站（仅支持Chrome浏览器），详细可参考[步骤3](#d0e1924)。



页面左侧树结构列出了视频所属app name以及通道名，中间列出了抽取的视频帧大图以及检测出的目标小图，点击下方小图后会在右侧列出详细的推理结果、评分。

本应用支持车辆属性检测，包括车辆品牌、车辆颜色的识别和车牌号码识别。



车牌号码识别的网络模型，是通过程序自动生成的车牌作为训练集图片训练的，不是使用真实车牌图片训练的。所以该模型在识别真实车牌号码时准确度比较低，如果需要较高的准确度的模型，请自己搜集真实车牌图片作为训练集并训练。

----结束

后续处理

* **停止视频结构化应用**

视频程序分析完之后会自动退出



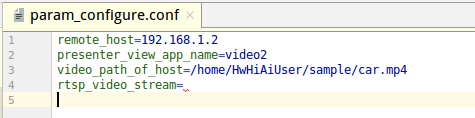
再次运行时请确保以下条件满足任意一个，否则会报错:

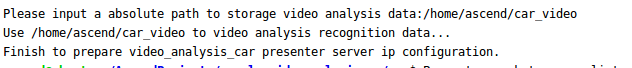
1.param\_configure.conf配置文件中参数presenter\_view\_app\_name的值对应的用于存储视频解析数据的路径中的数据内容已经清空。

2. param\_configure.conf配置文件中参数presenter\_view\_app\_name的值对应的用于存储视频解析数据的路径，在重新启动presenter server时需要修改为不同的路径用于存储视频解析的数据。

3.启动presenter server时填写的存储视频解析数据的路径下如果已有数据且不想删除，可以修改

.param\_configure.conf配置文件中参数presenter\_view\_app\_name的值，rebuild(重新编译)；之后再Run。





* **停止Presenter Server服务**

Presenter Server服务启动后会一直处于运行状态，若想停止视频结构化应用对应的Presenter Server服务，可执行如下操作。

以Mind Studio安装用户在Mind Studio所在服务器中执行如下命令查看视频结构化应用对应的Presenter Server服务的进程。

**ps -ef | grep presenter | grep video\_analysis\_car**

ascend@ascend-HP-ProDesk-600-G4-PCI-MT:~/sample-videoanalysiscar$ ps -ef | grep presenter | grep video\_analysis\_car   
ascend ***3655*** 20313 0 15:10 pts/24?? 00:00:00 python3 presenterserver/presenter\_server.py --app video\_analysis\_car

如上所示 *3655* 即为车辆检测应用对应的Presenter Server服务的进程ID。

若想停止此服务，执行如下命令：

**kill -9** *3655*

公共代码库下载

将依赖软件库下载到func\_deploy.sh, build\_ffmpeg.sh, build\_ezdvpp.sh，build\_presentagent.sh同一个目录下。

依赖代码库下载

| 模块名称 | 模块描述 | 下载地址 |
| --- | --- | --- |
| EZDVPP | 对DVPP接口进行了封装，提供对图片/视频的处理能力。 | <https://github.com/Ascend/sdk-ezdvpp>  下载后请保持文件夹名称为ezdvpp。 |
| Presenter Agent | 与Presenter Server进行交互的API接口。 | <https://github.com/Ascend/sdk-presenter/tree/master>  请获取此路径下的presenteragent文件夹，下载后请保持文件夹名称为presenteragent。 |
| 开源工具ffmpeg | 实现对视频文件的解封 | ffmpeg 4.0代码下载地址：<https://github.com/FFmpeg/FFmpeg/tree/release/4.0>。  下载后，目录名称请使用ffmpeg。 |
| tornado (5.1.0)  protobuf (3.5.1)  numpy (1.14.2) | Presenter Server依赖的Python库 | 可以在python官网<https://pypi.org/>上搜索相关包进行安装。  若使用pip3 install命令在线下载，可以使用如下命令指定相关版本进行下载，例如  pip3 install tornado==5.1.0 -i *指定库的安装源* --trusted-host *安装源的主机名* |