# 人脸检测

开发者可以将本application部署至Atlas 200DK上实现对摄像头数据的实时采集、并对视频中的人脸信息进行预测的功能。

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-facedetection>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-facedetection*。

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/ face\_detection*

。

Face Detection中使用模型

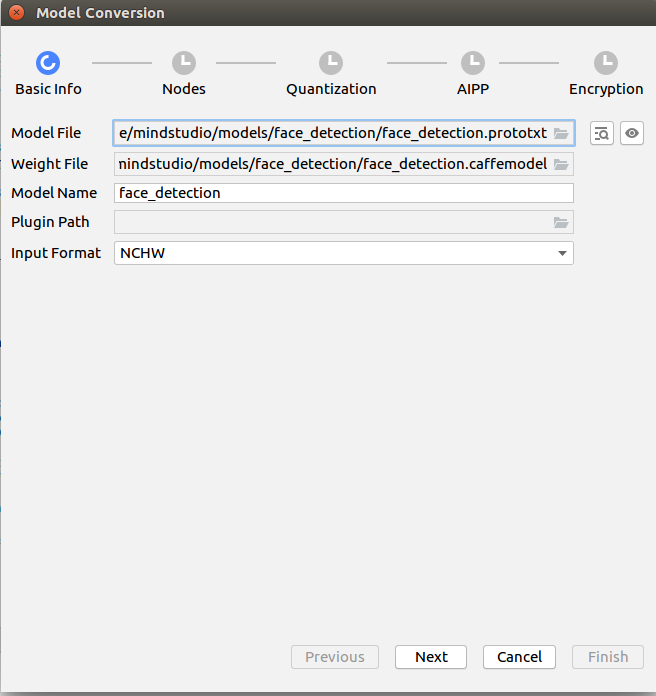
| 模型名称 | 模型说明 | 模型下载路径 |
| --- | --- | --- |
| face\_detection | 人脸检测网络模型。  此模型是基于Caffe的Resnet10-SSD300模型转换后的网络模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/object\_detect/face\_detection目录中README.md下载原始网络模型文件及其对应的权重文件。 |

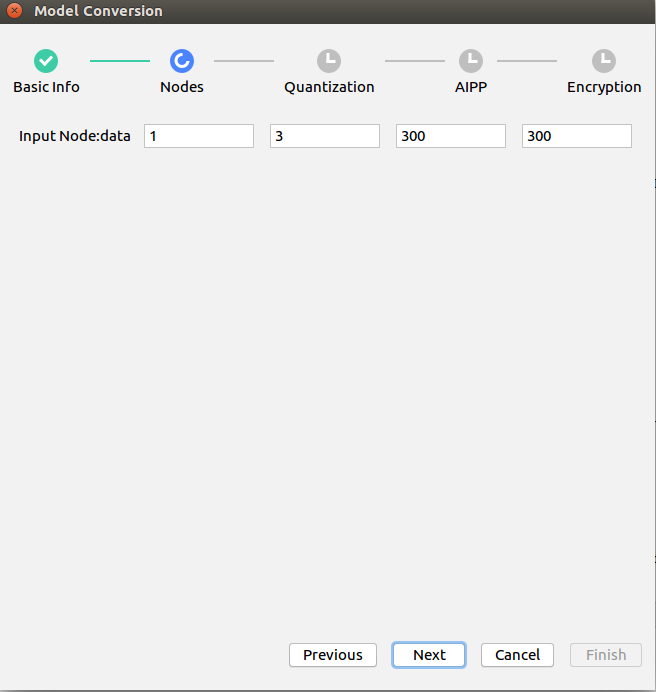
将原始网络模型转换为Davinci模型。

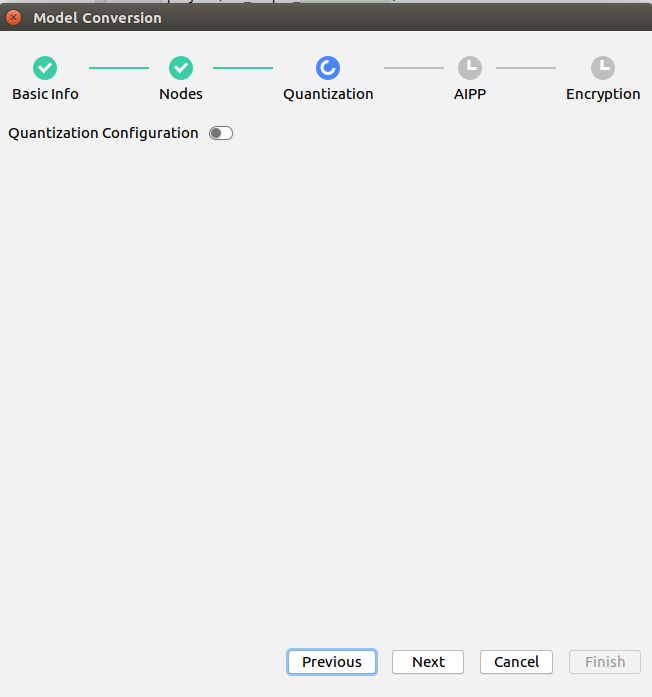
在Mind Studio操作界面的顶部菜单栏中选择“Tools > Model Convert”，进入模型转换界面。

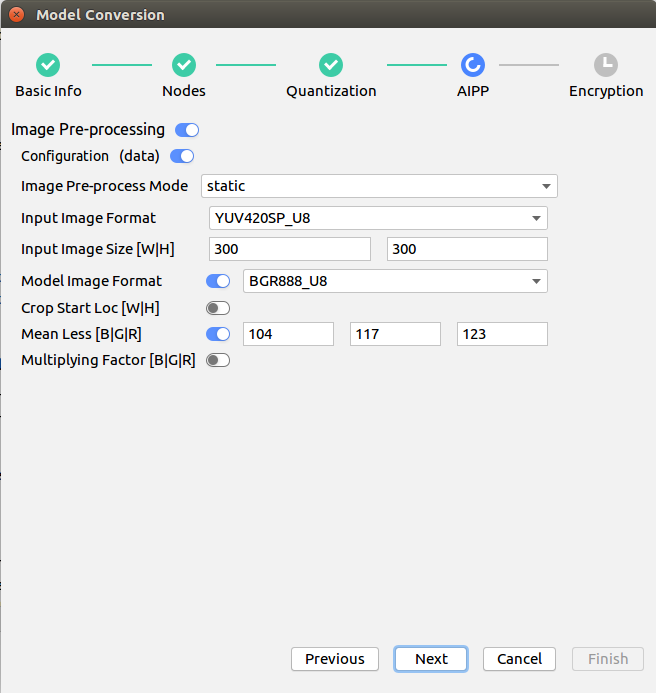
在弹出的**Model Conversion**操作界面中，Model File与Weight File分别选择步骤 2中下载的模型文件和权重文件。

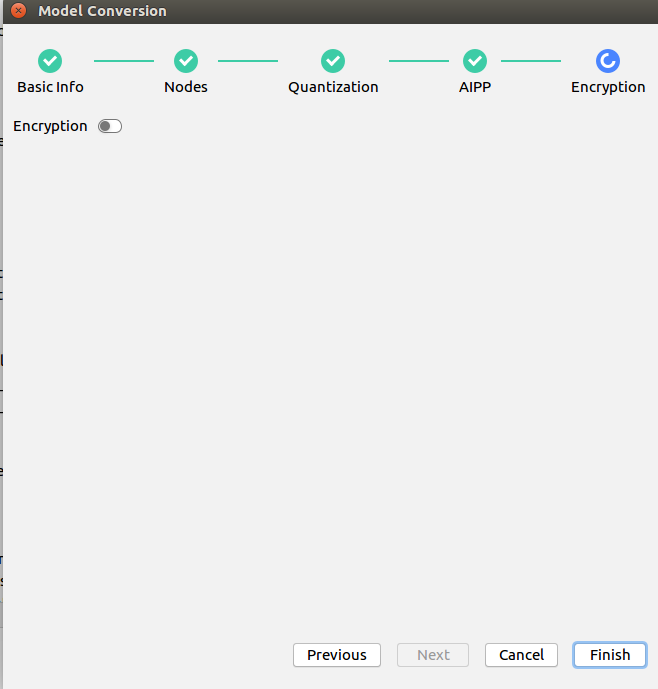
**Model Name**填写为表1.1对应的模型名称：face\_detection。







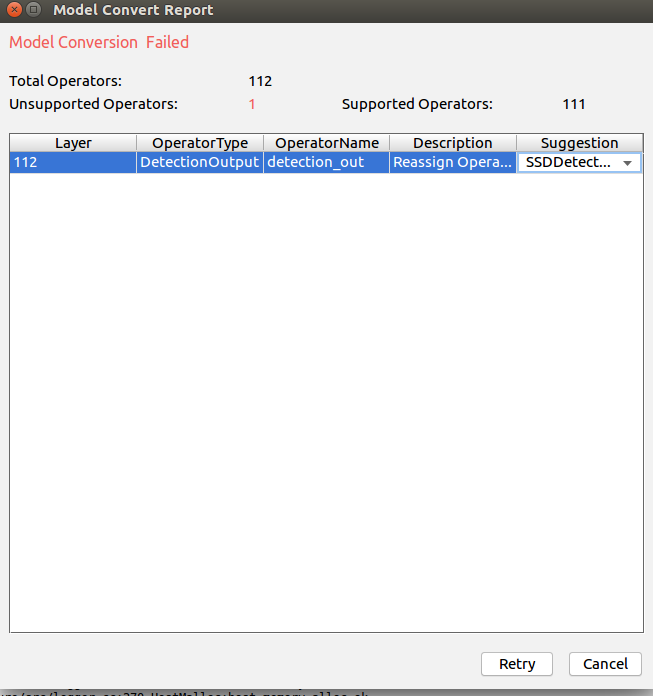




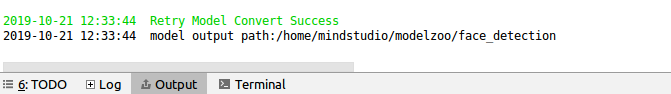
其他均保持默认值。

点击**Finish**开始转换模型。

face\_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，存放地址如下图所示$HOME/modelzoo/face\_detection。

将转换好的模型文件（.om文件）上传到“sample-facedetection/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和presentagent的部署脚本func\_deploy.sh , build\_ezdvpp.sh和build\_presentagent.sh

执行两个部署脚本:

bash build\_ezdvpp.sh  *host\_ip*

bash build\_presentagent.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



* 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和presentagent ，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-facedetection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=

data\_source=

presenter\_view\_app\_name=

需要手动添加参数配置：

* *remote\_host*：Atlas 200 DK开发者板的IP地址。
* data\_source: 摄像头所属Channel，取值为Channel-1或者Channel-2，查询摄像头所属Channel的方法请参考[Atlas 200 DK使用指南](https://ascend.huawei.com/documentation)中的“如何查看摄像头所属Channel”。
* 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。



* 三个参数必须全部填写，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，位数至少为1位

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。

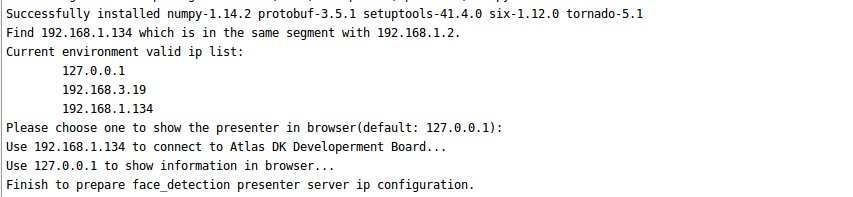
切换到命令行模式，进入到src目录下，例如：”sample\_facedetection/src”。执行如下命令在后台启动Face Detection应用的Presenter Server主程序。

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。）

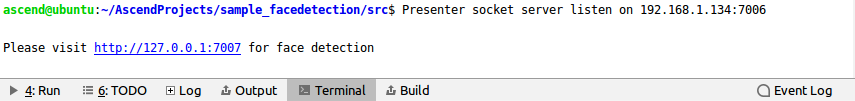
如图4.1所示，请在“Current environment valid ip list”中选择通过浏览器访问Presenter Server服务使用的IP地址。

示意图



如图4.2所示，表示presenter\_server的服务启动成功。

Presenter Server进程启动



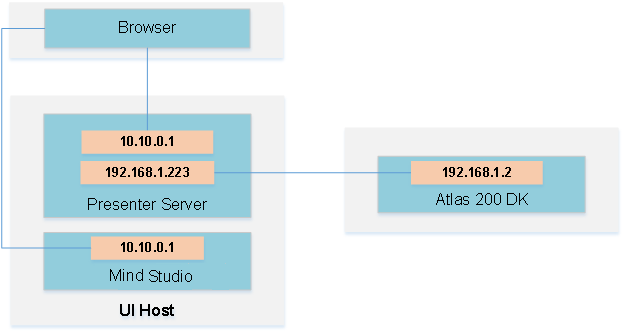
使用上图提示的URL登录Presenter Server，仅支持Chrome浏览器。IP地址为[步骤4](#a1)中输入的IP地址，端口号默为7007，如下图所示，表示Presenter Server启动成功。

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



其中：

* 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。

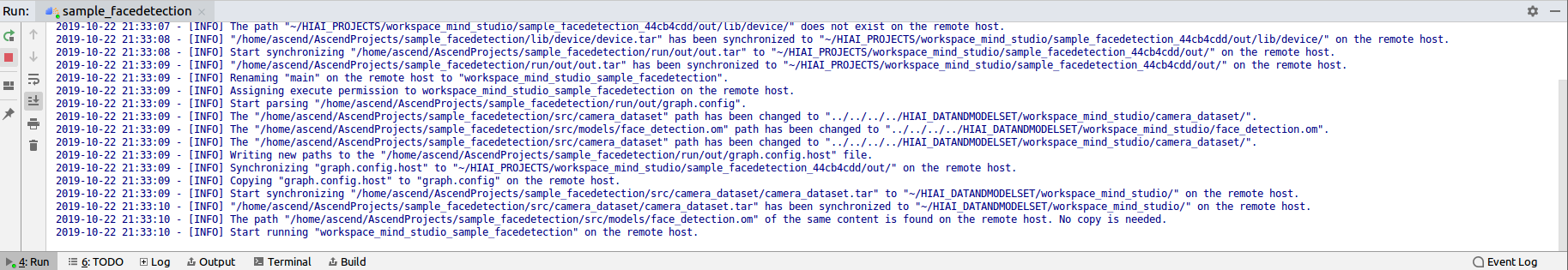
----结束

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。

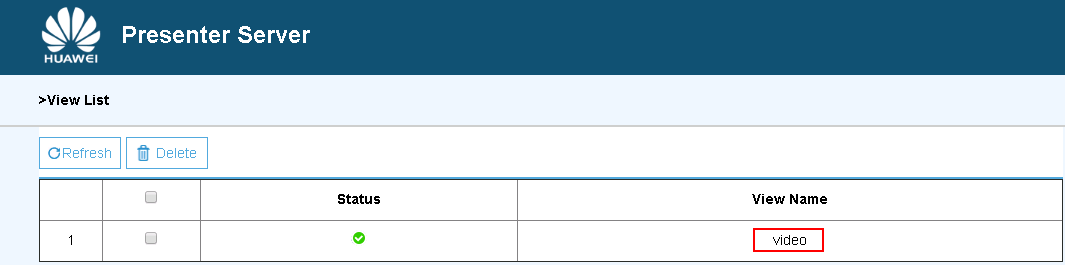
图1.1所示



使用启动Presenter Server服务时提示的URL登录 Presenter Server 网站，详细可参考[步骤3](#a2)。

等待Presenter Agent传输数据给服务端，单击“Refresh”刷新，当有数据时相应的Channel 的Status变成绿色，如图3.1所示。

Presenter Server界面





* Face Detection的Presenter Server最多支持10路Channel同时显示，每个 *presenter\_view\_app\_name* 对应一路Channel。
* 由于硬件的限制，每一路支持的最大帧率是20fps，受限于网络带宽的影响，帧率会自动适配为较低的帧率进行展示。

单击右侧对应的View Name链接，比如上图的“video”，查看结果，对于检测到的人脸，会给出置信度的标注。

----结束

后续处理

* **停止Face Detection应用**

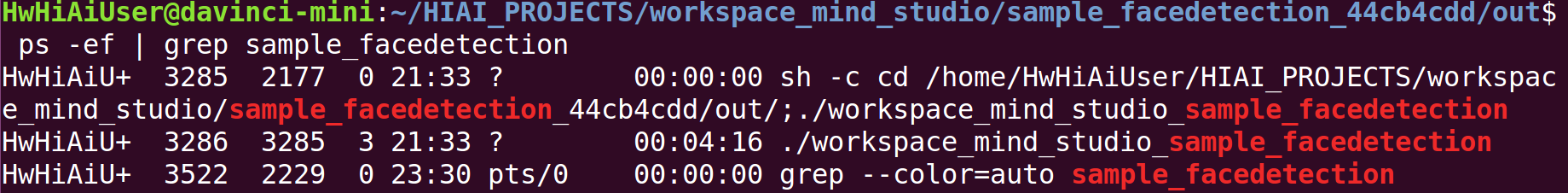
Face Detection应用执行后会处于持续运行状态，若要停止Face Detection应用程序，可执行如下操作。

单击如下图所示的停止按钮：



之后ssh登录到开发板执行如下命令查看facedetection应用对应的服务进程

**ps -ef | grep sample\_facedetection**



如上图所示3286即为face\_detection应用所对应的进程ID。

若想停止此项应用，执行如下命令：

Kill -9 3286

* **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Face Detection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Face Detection应用对应的Presenter Server服务的进程。

**ps -ef | grep presenter | grep face\_detection**

ascend@ascend-HP-ProDesk-600-G4-PCI-MT:~/sample-facedetection$ ps -ef | grep presenter | grep face\_detection   
ascend ***7701*** 1615 0 14:21 pts/8 00:00:00 python3 presenterserver/presenter\_server.py --app face\_detection

如上所示 *7701* 即为face\_detection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** *7701*

公共代码库下载

将依赖软件库下载到同func\_deploy.sh,build\_ezdvpp.sh和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。 |
| 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* |