**通用分类网络应用**

本Application支持运行在Atlas 200 DK或者AI加速云服务器上，实现了对常见的分类网络的推理功能并输出前n个推理结果。

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-classification>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-classification*。

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/* googlenet

。

通用分类网络应用使用模型

| 模型名称 | 模型说明 | 模型下载路径 |
| --- | --- | --- |
| densenet | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的DenseNet121模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/densenet目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| googlenet | 图片分类推理模型。  此模型为**此**应用中使用的模型。  是基于Caffe的GoogLeNet模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/googlenet目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| inception\_v2 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的Inception V2模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/inception\_v2目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| mobilenet\_v1 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的MobileNet V1模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/mobilenet\_v1目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| mobilenet\_v2 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的MobileNet V2模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/mobilenet\_v2目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| resnet18 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的ResNet 18模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/resnet18目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| resnet50 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的ResNet 50模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/resnet50目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| resnet101 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的ResNet 101模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/resnet101目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| resnet152 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的ResNet 152模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/resnet152目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| vgg16 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的VGG16模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/vgg16目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| vgg19 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的VGG19模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/vgg19目录中README.md下载原始网络模型文件及其对应的权重文件。 |
| dpn98 | 图片分类推理模型。  此模型为此应用中使用的模型。  是基于Caffe的dpn98模型。 | 请参考https://github.com/Ascend/models/tree/master/computer\_vision/classification/dpn98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“Tools > Model Convert”，进入模型转换界面。

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。

googlenet、inception\_v2模型中网络贯通应用一次处理一张图片，所以转换时需要将Input Shape的N修改为1

2.点击OK开始转换模型。

模型转换成功后，存放地址为$HOME/modelzoo/ googlenet。

将转换好的模型文件（.om文件）上传到“sample-classification /src/model”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的部署脚本func\_deploy.sh , build\_ezdvpp.sh.

执行部署脚本:

bash build\_ezdvpp.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



* 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，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- classification 工程。

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。

将需要推理的图片上传至Host侧任一属组为HwHiAiUser用户的目录。

图片要求如下：

格式：jpg、png、bmp。

输入图片宽度：16px~4096px之间的整数。

输入图片高度：16px~4096px之间的整数。

**----结束**

运行

在工具栏找到Run按钮，单击。

如下图1.1所示，可执行程序已经在开发板运行。

图1.1所示



**提示：报错信息忽略，因为IDE无法为可执行程序传参，上述步骤是将可执行程序与依赖的库文件部署到开发板，需要ssh登录到开发板至相应的目录文件下手动执行，具体请参考以下步骤。**

在Mind Studio所在Ubuntu服务器中，以HwHiAiUser用户SSH登录到Host侧。

**ssh HwHiAiUser@***host\_ip*

对于Atlas 200 DK，host\_ip默认为192.168.1.2（USB连接）或者192.168.0.2（NIC连接）。

对于AI加速云服务器，host\_ip即为当前Mind Studio所在服务器的IP地址。

进入贯通网络的可执行文件所在路径。

例如：

**cd ~/HIAI\_PROJECTS/workspace\_mind\_studio/sample\_classification\_8cd0be4f/out**

执行应用程序。

执行**run\_classification.py**脚本会将推理结果在执行终端直接打印显示。

命令示例如下所示：

**python3 run\_classification.py -w *224* -h *224* -i *./example.jpg* -n *10***

-w/model\_width：模型的输入图片宽度，为16~4096之间的整数。

-h/model\_height：模型的输入图片高度，为16~4096之间的整数。

-i/input\_path：输入图片的路径，可以是目录，表示当前目录下的所有图片都作为输入（可以指定多个输入）。

-n/top\_n：输出前n个推理结果。

其他详细参数请执行python3 run\_classification.py --help命令参见帮助信息。

**----结束**

公共代码库下载

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