VCAP快速入门手册 -rebuild

1.VCAP 简介

VCAP(vivo Computation Acceleration Platform)是vivo自研的移动端AI计算加速平台，该平台面向AI应用开发人员，助力开发者将AI算法在移动端快速部署、高效运行。

vivo会持续努力将VCAP打造成全能力、高性能、跨平台的移动端AI计算加速平台。

2.快速入门

本节以mobilenet_v1 tensorflow模型为例，展示如何使用VCAP快速集成。

1.下载mobilenet_v1 tensorflow模型

进入网址 http://download.tensorflow.org/models/mobilenet_v1_2018_02_22/mobilenet_v1_1.0_224.tgz 下载mobilenet_v1_1.0_224 包，解压后得到模型mobilenet_v1_1.0_224_frozen.pb。

此模型是一个1001分类模型，基本信息如下：

```
Model: mobilenet_v1_1.0_224
Input: input  224,224,3
Output: MobilenetV1/Predictions/Softmax  1,1001
```

2. 模型格式转换

下载VCAP Tools，详细阅读《VCAP工具使用手册》。开发者可自行搭建环境或者使用docker镜像，然后使用converter工具将tensorflow 模型转换成VCAP私有模型vaim。

```
$ python convert_to_vaim.py \
 --src_framework tf \
 --frozen_pb mobilenet_v1_1.0_224_frozen.pb \
 --input_shape 224 224 3 \
 --input_name input \
 --output_name MobilenetV1/Predictions/Reshape_1 \
 --dst_path mobilenet_v1_1.0_224_frozen.vaim \
 --fuse_activation \
 --fuse_bn \
 --reorder_weights
```

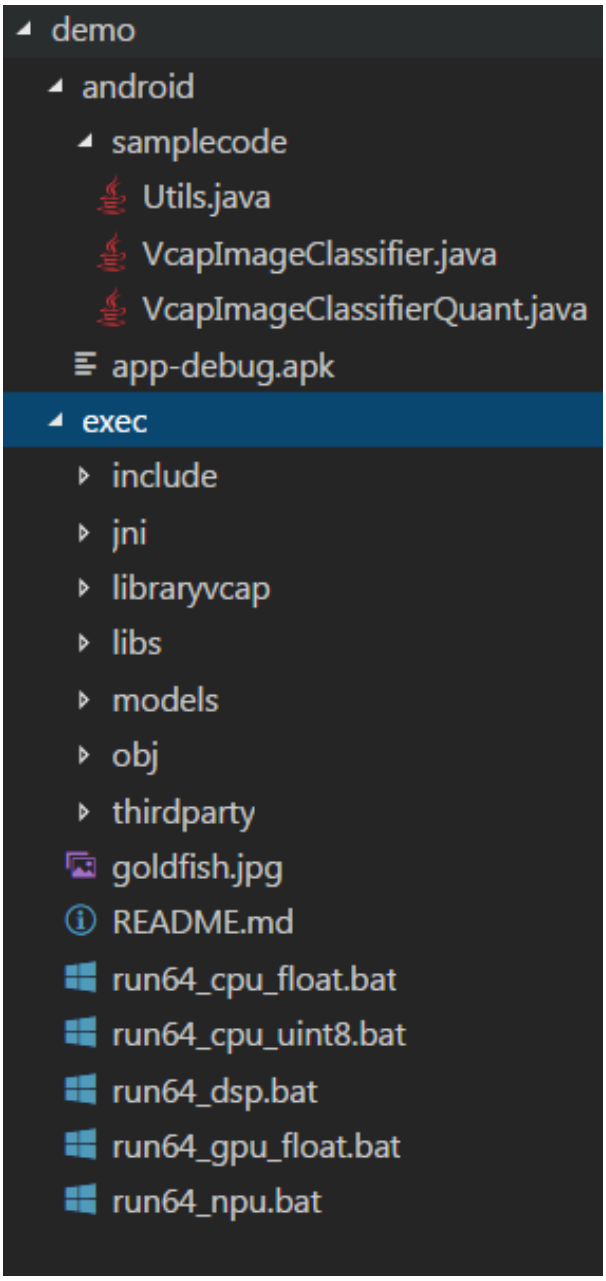
转换结束之后可以看到如下日志，显示了转换后vaim模型输入和输出名称：

New vaim input list : [u'input']
New vaim output list : [u'MobilenetV1/Predictions/Softmax']

3.导入VCAP java SDK ，设置编译依赖

进入VCAP SDK, 得到VCAP Android、C++ SDK和示例工程源代码，目录如下。

demo/android和demo/exec分别是VCAP Android和C++示例工程，开发者可以通过这两个工程更加快速地了解使用VCAP。



进入SDK/android目录，得到VCAP java SDK 。

在Android studio工程中，导入VCAP java SDK。其中jar导入libs目录下，so导入jniLibs目录。

(如果模型使用mmap加载方式，需要加非压缩标志)

```
dependencies {
    files('libs/vivo_vcap_V2.4.1.jar')
}
//mmap need to add  noCompress flag
android {
    compileSdkVersion xx
    defaultConfig {
                ...
    }
        //add this noCompress flag
    aaptOptions {
        noCompress "vaim"
    }
}
```

4. 使用VCAP API，集成网络解析、运行、释放等逻辑

VCAP API使用流程：

AI算法集成需要如下步骤（详细可以参考VCAP SDK/android/samplecode/VcapImageClassifier.java 文件）。

- 创建网络构建器
- 网络参数配置，创建网络
- 传入输入节点数据
- 网络前向推理
- 获取输出结果
- 资源释放

创建网络构建器：

```
VcapInstance mVcapNet = new VcapInstance();
```

网络参数配置，创建网络：

下面代码指导了如何创建网络实例。网络创建和释放是一一对应的，且推荐使用全局实例，同一网络不推荐频繁的创建和释放。

```
// model path
String modelPath = ...;
mVcapNet = mVcapNet .setRuntime(runtime )
                                        .setModelFile
(modelPath)
//byte array
InputStream mModelStream = mAssetManger.open(model);// open vaim file as a stream
mVcapNet = mVcapNet .setRuntime(runtime )
                                        .setModelFile(mModelStream) // model stream
//mmap
mVcapNet = mVcapNet .setRuntime(mVcapRuntime)
                                        .setModelFile(mapModelFromAssets(context,
mModelAssetsName))
//
mVcapNet .build();//create a network instance
```

传人输入节点数据：

```
 //
Utils.bitmapToFloatArray(bitmap, mImageMean, mImageStd, mInputarr, mInputSizeH, mInputSizeW);
 //tensor
 mVcapNet.setInput(mInputNode, mInputarr, mInputBatch, mInputChanels, mInputSizeH, mInputSizeW,
mInputByteSize);
```

网络前向推理：

```
mVcapNet.forward();// excute network
```

获取输出结果：

```
mNetOutputData = new float[mNumClasses];
 ......
mVcapNet.getOutput(mOutputNode, mNetOutputData);
```

资源释放：

```
mVcapNet.release();
```

示例工程：

本节提供一个简单的示例工程，直观展示了如何利用VCAP集成AI算法。

应用程序源代码见路径 VCAP SDK/demo/android/vcapclassify 。

下面是APP 模型运行的效果图：

（注意:demo展示的是实际应用场景的时间，详细的模型执行时间见《VCAP加速平台介绍》性能一章）

附录：

VCAP API接口详细说明文档：《VCAP API reference.pdf》