

密级状态：绝密() 秘密() 内部() 公开(√)

RK3399_Linux_NN_SDK _V1.0_20180713

(技术部，图形显示平台中心)

文件状态： [] 正在修改 [√] 正式发布	当前版本：	V1.0
	作 者：	杜坤明
	完成日期：	2018-07-13
	审 核：	熊伟
	完成日期：	2018-07-13

福州瑞芯微电子股份有限公司

Fuzhou Rockchips Semiconductor Co., Ltd

(版本所有,翻版必究)

更新记录

版本	修改人	修改日期	修改说明	核定人
V1.0	杜坤明	2018-07-13	初始版本	熊伟

目 录

1	主要功能说明.....	3
2	系统依赖说明.....	3
2.1	LINUX 平台依赖	3
2.2	关于性能	4
3	SDK 使用说明.....	4
3.1	TENSORFLOWLITE 修改说明	4
3.2	LABEL_IMAGE DEMO 编译说明	5
3.3	LABEL_IMAGE DEMO 运行说明	6

1 主要功能说明

本 SDK 为基于 RK3399 Linux 的神经网络 GPU 加速方案，可为采用 TensorflowLite 开发的 AI 相关应用提供通用加速支持。

本 SDK 主要包含 3 个部分：

- 1) TensorflowLite 修改说明。
- 2) Label_image Demo 编译说明：在 TensorflowLite 中编译出 Linux 平台上使用 GPU 加速的 MobileNet 分类器 Demo 和 MobileNet-SSD 目标检测 Demo。
- 3) Label_image Demo 运行说明。

2 系统依赖说明

2.1 Linux 平台依赖

本 SDK 基于 RK3399 的 Linux 开发，要求系统具有 OpenCL 驱动。检查是否有 OpenCL 驱动的简单方法如下：

在 RK3399 目标板的 /lib/ 目录或 /usr/lib/ 目录下检查是否存在下列三个库的任意一个：

```
ls /lib |grep -E "libmali.so|libOpenCL.so|libGLES_mali.so"
```

```
ls /usr/lib |grep -E "libmali.so|libOpenCL.so|libGLES_mali.so"
```

如存在，则通过 objdump 检查这个库是否包含 “clCreateContext” 的符号表，如：

```
objdump -T libmali.so |grep clCreateContext
```

如果包含上述符号表，则可以认为 OpenCL 驱动存在。如果没有包含上述符号表或找不到上述库，则说明 OpenCL 驱动不存在，请更新至 RK3399 Linux 最新的 SDK。

同时，本 SDK 只适用于 RK3399 的 64 位系统，在 RK3399 的以下 Linux 固件上验证通过：

```
RK3399_Evb_Buildroot_V2.2_20180704
```

```
rootfs-debian9-20180713-64bit
```

2.2 关于性能

在 CPU 1.8G, GPU 800M 的情况下, 性能如下:

Model	Time (ms)
mobilenet_v1_1.0_224.tflite	43.9
mobilenet_ssd_v1_300.tflite	88.5
mobilenet_ssd_v2_300.tflite	104.1
squeezenet_2018_04_27.tflite	66.5
inception_resnet_v2.tflite	685.7
inception_v3_slim_2016.tflite	386.2
inception_v4.tflite	724.2
vgg16.tflite	686.1
densenet.tflite	1255.2

3 SDK 使用说明

在上述 Linux 平台依赖满足的情况下, 支持基于 Tensorflow Lite 开发应用, Tensorflow Lite 自动调用 GPU 加速接口。

目前提供基于 Tensorflow Lite 自带的 label_image 进行修改的两个 Demo:

一是使用 MobileNet 模型图像分类器 Demo。

二是使用 MobileNet-SSD 模型的目标检测 Demo。

3.1 TensorflowLite 修改说明

为使相关优化生效, 需要对原始 Tensorflow Lite 框架 (v1.9.0 版本) 打上 SDK 包中的 0001-patch1.0-NNAPI-Squeeze-label_image-ssd.patch 补丁。

补丁使用方法如下:

- 1) 下载 tensorflow, 将补丁放入 tensorflow 目录, 然后 checkout 到 v1.9.0 分支, 再打上补丁:

```
git clone https://github.com/tensorflow/tensorflow.git
cp 0001-patch1.0-NNAPI-Squeeze-label_image-ssd.patch tensorflow/
cd tensorflow
git checkout v1.9.0 -b 1.9.0
git apply 0001-patch1.0-NNAPI-Squeeze-label_image-ssd.patch
```

- 2) 执行下述命令下载相关依赖库和安装交叉编译工具:

```
./tensorflow/contrib/lite/download_dependencies.sh
chmod +x tensorflow/contrib/lite/build_rpi_lib.sh
sudo apt install gcc-aarch64-linux-gnu
sudo apt install g++-aarch64-linux-gnu
```

3.2 Label_image Demo 编译说明

在打完 Tensroflow Lite 补丁, 并下载完相关依赖库以及安装完交叉编译工具, 可以执行:

```
./tensorflow/contrib/lite/build_rpi_lib.sh
```

生成可执行二进制程序, 生成路径如下:

```
tensorflow\contrib\lite\gen\bin\rpi_arm64\label_image
```

该 label_image 默认为支持 MobileNet 模型图像分类器的 Demo, 如需要编译成支持 MobileNet-SSD 模型的目标检测 Demo, 可以将 tensorflow/contrib/lite/examples/label_image/label_image.cc 里的 FOR_SSD 宏定义改为 1, 即可编译出支持 MobileNet-SSD 模型的目标检测 Demo。

注: 本 SDK 包内 tmp 目录下包含编译好的两个 Demo:

label_image_mobilenet 为支持 MobileNet 模型图像分类器的 Demo。

label_image_mobilenet_ssd 为支持 MobileNet-SSD 模型的目标检测 Demo

3.3 Label_image Demo 运行说明

label_image 的运行需要将相关依赖库拷贝至/usr/lib/下，同时将相关资源文件拷贝至/tmp目录下，具体步骤如下：

- 1) 将 SDK 包里的 usr/lib 目录下的内容拷贝至 RK3399 目标板的/usr/lib/目录下。
- 2) 将 SDK 包里的 tmp 目录下的内容拷贝至 RK3399 目标板的/tmp/目录下。
- 3) 将上述编译生成的 label_image 也拷贝至 RK3399 目标板的/tmp/目录下。
- 4) 如果 label_image 为编译成支持 MobileNet 模型图像分类器的 Demo，则进入 RK3399 目标板的/tmp 目录执行：

```
./label_image -m mobilenet_v1_1.0_224.tflite -l labels.txt -i cat.bmp -a 1 -c 100
```

执行成功后会有如下耗时打印（average time 为平均执行时间）：

```
first invoked time: 1395.19 ms
invoked
average time: 43.9085 ms
0.405029: 283 tiger cat
0.245605: 286 Egyptian cat
0.175537: 282 tabby
0.0319519: 288 lynx
0.0282135: 285 Siamese cat
```

- 5) 如果 label_image 为编译成支持 MobileNet-SSD 模型的目标检测 Demo，则需要先将 tensorflow/contrib/lite/examples/label_image/3rdparty/opencv/lib 目录下的 opencv 库拷贝至 RK3399 目标板的/usr/lib/目录下，然后再进入 RK3399 目标板的/tmp 目录执行：

```
./label_image -m mobilenet_ssd.tflite -i test.bmp -a 1 -c 100
```

执行成功后会有如下耗时和检测结果打印（average time 为平均执行时间）：

```
first invoked time: 2064.61 ms
invoked
average time: 88.5672 ms
validCount: 26
car @ (546, 501) (661, 586)
```

```
car      @ (1, 549) (51, 618)
person   @ (56, 501) (239, 854)
person   @ (332, 530) (368, 627)
person   @ (391, 541) (434, 652)
person   @ (418, 477) (538, 767)
person   @ (456, 487) (602, 764)
car      @ (589, 523) (858, 687)
person   @ (826, 463) (1034, 873)
bicycle @ (698, 644) (1128, 925)

write out.jpg succ!
```

同时还会在 RK3399 目标板的/tmp 目录下生成包含检测结果的图像 out.jpg，可以导出 out.jpg 查看检测结果。