密级状态：绝密（　　）　秘密（　　）　内部（　　）　公开（ √ ）

RK3399_SSD_优化库_V1.0_20180522

（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：	卓鸿添
	完成日期：	2018-05-22
	审　　核：	熊伟
	完成日期：	2018-05-22

福州瑞芯微电子股份有限公司

Fuzhou　Rockchips　Semiconductor　Co . , Ltd

更新记录

版本	修改人	修改日期	修改说明	核定人
V1.0	卓鸿添	2018-05-22	初始版本	熊伟

目 录

1 主要功能说明

本优化包为 RK3399 平台的基于深度学习的目标检测（SSD）方案提供优化，可为 AI 人工智能行业提供准 Turnkey 解决方案，可同时支持 Android 及 Linux 系统。

本次优化包主要包含 3 个部分:

1) SSD 优化库。 Rockchip 针对 SSD 不同的 class 及 alpha 提供了不同的优化库，可通过在线网站（http://ai.rock-chips.com/mobilenet-ssd.php）下载。

2) rkl 模型转换工具，支持将 tflite 模型转成 rkl 模型

3) Android/Linux 优化库 Demo

2 系统依赖说明

2.1 Android 平台依赖

本次优化库基于 Android Nougat 开发，只提供 64 位版本，要求系统具有 OpenCL 驱动。并且在 Mali 版本为 r14p0 上通过验证。（Note： 可通过如下命令查看 mali 驱动版本: getprop | grep mali）

2.2 Linux 平台依赖

本次优化库基于 Ubuntu 16.04 ARM 版本开发，只提供 64 位版本，要求系统具有 OpenCL 驱动。优化库将自动搜索系统中的 libOpenCL.so。如果 libOpenCL.so 不在搜索路径中，请设置环境变量 LIBOPENCL_SO_PATH，优化库将优先使用该环境变量指定的 OpenCL。本优化库在 Mali 版本为 r13p0 上通过验证。

2.3 关于性能

为了达到最好的性能，需要将 CPU 及 GPU 定频，特别是 GPU。

CPU 定频方法参考：

```
cat /sys/devices/system/cpu/cpu4/cpufreq/scaling_available_frequencies
echo "userspace" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
echo    "1416000" > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
echo "userspace" > /sys/devices/system/cpu/cpu1/cpufreq/scaling_governor
echo    "1416000" > /sys/devices/system/cpu/cpu1/cpufreq/scaling_setspeed
echo "userspace" > /sys/devices/system/cpu/cpu2/cpufreq/scaling_governor
echo    "1416000" > /sys/devices/system/cpu/cpu2/cpufreq/scaling_setspeed
echo "userspace" > /sys/devices/system/cpu/cpu3/cpufreq/scaling_governor
echo    "1416000" > /sys/devices/system/cpu/cpu3/cpufreq/scaling_setspeed
echo "userspace" > /sys/devices/system/cpu/cpu4/cpufreq/scaling_governor
echo    "1800000" > /sys/devices/system/cpu/cpu4/cpufreq/scaling_setspeed
echo "userspace" > /sys/devices/system/cpu/cpu5/cpufreq/scaling_governor
echo    "1800000" > /sys/devices/system/cpu/cpu5/cpufreq/scaling_setspeed
```

GPU 定频方法参考：

```
echo "userspace"    >/sys/devices/platform/ff9a0000.gpu/devfreq/ff9a0000.gpu/governor
echo "800000000"    >/sys/devices/platform/ff9a0000.gpu/devfreq/ff9a0000.gpu/userspace/set_freq
cat /sys/devices/platform/ff9a0000.gpu/devfreq/ff9a0000.gpu/cur_freq
```

3 优化库使用说明

在优化库中，将对输入数据做如下的预处理：

RGB Value/128.0f - 1.0f

所以 APP 中不需要做相应的处理。

对于优化库的输出的 locations 及 confidence 都需要做相应的后处理，具体参考相应的 demo。

优化库可通过在线网站（http://ai.rock-chips.com/mobilenet-ssd.php）下载。

3.1 Android 平台使用说明

本次优化库将为 Android 平台提供一个 JNI 文件。

需要说明的是 JNI 支持两种格式的输入：

1) OpenGL Texture：纹理只支持 GL_TEXTURE_2D，不支持 GL_TEXTURE_EXTERNAL_OES（Camera 输出），格式为 RGBA。

```
/*
*    descption:
*          SSD 检测, 只适用于 Android 平台
*    params:
*     textureId:          输入图像纹理 Id
*     outputLocations:       用于保存预测框位置（xmin, ymin, xmax, ymax）（需要后处理具体参考相应的 demo）
*     outputClasses:    用于保存 confidence, confidence 还需要做 expit 处理((float) (1. / (1. + Math.exp(-x)));)
* */
private native int native_run_ssd(int textureId, float[] outputLocations, float[] outputClasses);
```

2) RGB Raw Data：格式为 RGB/BGR（取决于用户自己训练的模型）。

```
/*
*   descption:
*          SSD 检测
*   params:
*      inData:              输入图像, 目前支持格式为 RGB/BGR（取决于用户自己训练的模型）
*      outputLocations:     用于保存预测框位置(xmin, ymin, xmax, ymax)（需要后处理具体参考相应的 demo）
*      outputClasses:   用于保存 confidence, confidence 还需要做 expit 处理((float) (1. / (1. + Math.exp(-x)));)
* */
private native int native_run_ssd(byte[] inData, float[] outputLocations, float[] outputClasses);
```

3.2 Linux 平台使用说明

Linux 平台只支持一种格式的输入：

1) RGB Raw Data：格式为 RGB/BGR（取决于用户自己训练的模型）。

```
/*
*   descption:
*          SSD 检测
*   params:
*          inData:              输入图像, 目前支持格式为 RGB/BGR（取决于用户自己训练的模型）
*          predictions:     用于保存预测框位置(xmin, ymin, xmax, ymax)（需要后处理，具体参考相应的 demo）
*          outputClasses:   用于保存 confidence, confidence 还需要做 expit 处理((float) (1. / (1. + Math.exp(-x)));)
* */
virtual void run_ssd(char *inData, float *predictions, float* outputClasses) = 0;
```

4 rkl 模型转换工具使用说明

4.1 环境依赖

模型转换工具基于 Python 2.7 开发，依赖如下：

```
numpy

tensorflow  =  1.8
```

4.2 转换步骤

```
Step 1:

bazel run -c opt tensorflow/python/tools/optimize_for_inference -- \

--input=5c.pb    --output=5c_opt.pb --frozen_graph=True \

--input_names=Preprocessor/sub --output_names=concat,concat_1 \

--alsologtostderr


Step 2:

bazel run tensorflow/contrib/lite/toco:toco -- \

--input_file=5c_opt.pb --output_file=5c.tflite \

--input_format=TENSORFLOW_GRAPHDEF --output_format=TFLITE \

--input_shapes=1,300,300,3 --input_arrays=Preprocessor/sub \

--output_arrays=concat,concat_1 --inference_type=FLOAT --logtostderr


Step 3:

./TFLiteMobileNetSSD300Convertor 5c.tflite 5c
```

运行模型转换工具后将得到一个 xxxx.rkl，这个文件就是 SSD 优化库可用的参数模型。

5 开源软件使用说明

RK3399 SSD 优化包参考并使用了一些第三方软件，其版权归属原作者所有，非常感谢所有相关的软件及其作者，由于篇幅所限无法一一列举，包括但不限于：

1) Tensorflow

Copyright 2018 The TensorFlow Authors. All rights reserved.

https://github.com/tensorflow/tensorflow/blob/master/LICENSE

2) ACL

Copyright (c) 2017 ARM Software

https://github.com/ARM-software/ComputeLibrary/blob/master/LICENSE

3) NNVM

Distributed (Deep) Machine Learning Community. Licensed under an Apache-2.0 license.

https://github.com/dmlc/nnvm/blob/master/LICENSE

4) NCNN

Copyright (C) 2017 THL A29 Limited, a Tencent company. All rights reserved.

https://github.com/Tencent/ncnn/blob/master/LICENSE.txt