

Rockchip

RKNN_DEMO 开发指南

发布版本:**0.1**

日期:**2018.12**

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

版权所有 © 2018 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：www.rock-chips.com

前言

概述

本文档主要介绍 Rockchip 处理器内 RKNN_DEMO 的使用方法

产品版本

芯片名称	内核版本
RK3399PRO	4.4

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2018-12-8	0.1	lhp	提交初始文档

目录

1	RKNN_DEMO 运行	1-1
1.1	概述.....	1-1
1.2	内核配置	1-1
1.3	NPU 相关	1-1
1.4	编译和运行	1-1
2	RKNN_DEMO 开发	2-2
2.1	文件目录介绍.....	2-2

1 RKNN_DEMO 运行

1.1 概述

rknn_demo 模块代码位于<SDK>/buildroot/package/rockchip/rknn_demo 目录下。主要通过 usb camera 采集图像,送到 NPU 进行处理,并通过 minigui 显示结果。当前支持的模型为 mobilenet_ssd。

1.2 内核配置

SDK 中默认已将需要配置使能,主要依赖的有 rga 和 usbcamera。如果未开启,请到内核中查看相关 config 的历史修改。

1.3 NPU 相关

SDK 中,相关模型文件已经默认编译到板子中。对应的文件宏和目录为:

```
#define MODEL_NAME          "/tmp/mobilenet_ssd.rknn"  
#define BOX_PRIORS_TXT_PATH " /tmp/box_priors.txt"  
#define LABEL_NALE_TXT_PATH " /tmp/coco_labels_list.txt"
```

模型运行起来前,请确保相关文件存在。

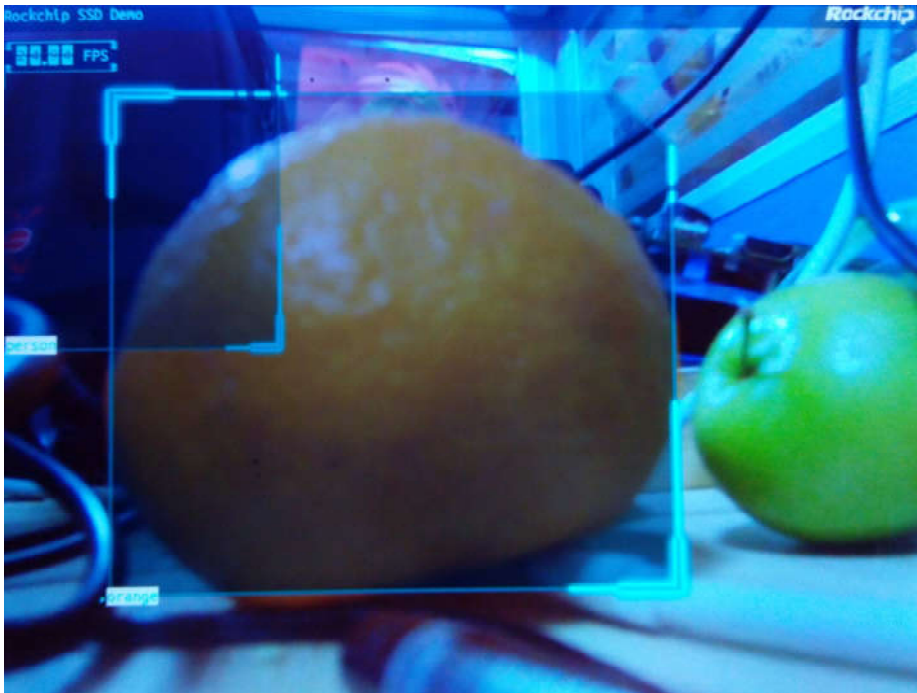
1.4 编译和运行

可以在 SDK 目录中,通过命令./build.sh rknn_demo 进行模块编译,会生成 rknn_demo 执行文件。拷贝到板子上,确保 USB Camera 已经插入,直接运行 rknn_demo 命令即可。

注意:请不要与其他的 UI 共存,在启动前,请把相关 UI 启动命令删除。板子默认有带 QT,可以运行命令: /etc/init.d/S50launcher stop。

正常运行帧率在 25~30fps 左右,如果帧率不够,可能是 usbcamera 输入帧率不够,建议对着强光或者更换 usbcamera。Usbcamera 连接不稳定会导致运行异常,请保持稳定连接。

运行结果如下图:



2 RKNN_DEMO 开发

2.1 文件目录介绍

Config.in 为配置文件，rknn_demo.mk 为基本编译文件，资源的拷贝就是在这做的。具体命令可以查看 RKNN_DEMO_INSTALL_TARGET_CMDS。

src/为代码目录，在 src 目录下，CMakeLists.txt 为编译文件。可以在 RKNN_DEMO_SRC 中添加自己的文件进行编译。

rknn_camera.c 为主文件，主要是用于启动 minigui 的主窗口和初始化模块。MiniGUIMain 为主函数入口。rknn_ui_show 为 minigui 创建主窗口。rknn_demo_init 会进行启动两个线程：post 和 run。run 主要进行图像的获取和 npu 的处理，并将结果发送到 post 线程中；post 接收到 npu 的处理结果，进行后处理，并输出结果送给显示。

src/rknn/ssd 为 ssd 相关处理文件。ssd.c 中，ssd_run 函数进行模型加载，通过 cameraRun 获取 usbcamera 的 buf，并输出到注册函数 ssd_camera_callback 中。在 ssd_camera_callback 函数中，函数 yuv_draw 送视频数据到 minigui 层中，进行将视频数据和 ui 数据进行 rga 合成。

YUV420toRGB24_RGA 进行视频数据转换，将 640*480 的 nv12 格式转换成 300*300 的 rgb888 格式。然后送到函数 ssd_rknn_process 进行处理。

src/ui/ssd 为 ssd 的 ui 显示文件。函数 caption_create 绘制标题栏，并在 caption_wnd_proc 中进行显示；函数 fps_create 绘制帧率栏，并在 fps_wnd_proc 中进行显示；ssd_paint_object 为物体框绘制栏，ssd 的处理结果会送到这里进行显示。详细 minigui 的开发处理，可以参考相关开源资料。