密级状态：绝密(　　)　　秘密(　　)　　内部(　　)　　公开(√)

RKNN SDK 快速上手指南

（技术部，图形计算平台中心）

文件状态： [　] 正在修改 [√] 正式发布	当前版本：	V1.7.5
	作　　者：	饶洪
	完成日期：	2023-07-31
	审　　核：	熊伟
	完成日期：	2023-07-31

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd

更新记录

版本	修改人	修改日期	修改说明	核定人
V0.9.9	饶洪	2019-03-25	初始版本	熊伟
V1.0.0	饶洪	2019-05-08	同步 RKNN-Toolkit-V1.0.0 修改内容	熊伟
V1.1.0	饶洪	2019-06-28	1. 同步 RKNN-Toolkit-V1.1.0 修改内容 2. 新增 Windows/MacOS/ARM64 等平台的快速上手指南	熊伟
V1.2.0	饶洪	2019-08-21	同步 RKNN-Toolkit-V1.2.0 修改内容	熊伟
V1.2.1	饶洪	2019-09-26	同步 RKNN-Toolkit-V1.2.1 修改内容	熊伟
V1.3.0	饶洪	2019-12-23	同步 RKNN-Toolkit-V1.3.0 修改内容	熊伟
V1.3.2	饶洪	2020-04-03	同步 RKNN-Toolkit-V1.3.2 修改内容	熊伟
V1.4.0	饶洪	2020-08-13	同步 RKNN-Toolkit-V1.4.0 修改内容	熊伟
V1.6.0	饶洪	2020-12-31	同步 RKNN-Toolkit-V1.6.0 修改内容	熊伟
V1.6.1	饶洪	2021-05-21	同步 RKNN-Toolkit-V1.6.1 修改内容	熊伟
v1.7.0	邢正	2021-08-06	同步 RKNN-Toolkit-V1.7.0 修改内容	熊伟
v1.7.1	饶洪	2021-11-20	同步 RKNN-Toolkit-V1.7.1 修改内容	熊伟
V1.7.3	饶洪	2022-08-07	1. Ubuntu 平台更新依赖的系统版本(18.04)和 Python 版本(3.6)； 2. 增加开发环境搭建相关内容。	熊伟
V1.7.5	饶洪	2023-07-31	1. 更新依赖说明；示例使用说明； 2. 增加模型部署快速上手指南； 3. 将主要功能说明替换成概述； 4. 文档重命名为 RKNN SDK 快速上述指南。	熊伟

目 录

1 概述 5

2 开发环境搭建 6

2.1 参考系统配置 6

2.2 系统依赖说明 6

2.3 连接 ROCKCHIP NPU 设备 8

2.3.1 通过 USB-OTG 口连接设备 8

2.3.2 通过 DEBUG 口连接设备 10

2.3.3 确认设备连接正确 12

2.3.4 常见问题 13

3 UBUNTU 平台快速上手 14

3.1 环境准备 14

3.2 安装 RKNN-TOOLKIT 14

3.3 运行安装包中附带的示例 15

3.3.1 在 RV1126 上运行示例 15

3.3.2 在 RK1808 上运行示例 16

4 WINDOWS 平台快速上手指南 18

4.1 环境准备 18

4.2 安装 RKNN-TOOLKIT 19

4.3 在 RV1126 或 RK1808 上运行示例 20

5 MAC OS X 平台快速上手指南 21

5.1 环境准备 21

5.2 安装 RKNN-TOOLKIT 21

5.3 在 RV1126 或 RK1808 上运行示例 22

6 ARM64 平台快速上手指南 23

6.1 环境准备 23

6.2 安装 RKNN-TOOLKIT 23

6.3 运行安装包中附带的示例 24

7 模型部署快速上手指南 26

7.1 下载 RKNPU 工程 26

7.2 准备编译工具 26

7.2.1 下载交叉编译工具 26

7.2.2 设置编译工具路径 27

7.3 更新 RKNN 模型 27

7.4 编译示例程序 27

7.5 在开发板上运行示例 27

8 附录 30

8.1 参考文档 30

8.2 问题反馈渠道 30

1 概述

此文档面向零基础用户，详细介绍如何快速上手使用 RKNN Toolkit 完成模型转换，并通过 RKNPU 提供的 C API 接口将模型部署到瑞芯微 EVB 开发板上进行推理。

适用芯片：RK1806 / RK1808 / RK3399Pro / RV1109 / RV1126。

RKNN Toolkit 工程下载地址：https://github.com/rockchip-linux/rknn-toolkit

RKNPU 工程下载地址：https://github.com/rockchip-linux/rknpu

2 开发环境搭建

本章节介绍在使用 RKNN Toolkit 前的开发环境准备工作。对于环境部署可能遇到的问题，请参考《Rockchip_Trouble_Shooting_RKNN_Toolkit_CN.pdf》文档第 3 章节。

2.1 参考系统配置

RKNN Toolkit 建议的系统配置如下：

表 2-1 推荐系统配置

硬件/操作系统	推荐配置
CPU	Intel Core i3 以上或同级别至强 / AMD 处理器
内存	16G 或以上
操作系统	Linux: Ubuntu 18.04 64bit 或 Ubuntu 20.04 64bit MacOS: 10.13.5 Windows: 10

注：模型转换阶段建议使用以上推荐配置的 PC 完成。在 RK3399Pro、RK1808 计算卡等设备上使用指定固件时也可以完成模型转换，但因为这些设备的 CPU 和内存资源有限，并不建议使用。

2.2 系统依赖说明

使用本开发套件时需要满足以下运行环境要求：

表 2-2 运行环境

操作系统版本	Ubuntu18.04（x64）及以上 Windows 7（x64）及以上 Mac OS X 10.13.5（x64）及以上 Debian 9.8（aarch64）及以上
Python 版本	3.5 / 3.6 / 3.7 / 3.8
Python 库依赖	'numpy == 1.19.5' 'scipy == 1.4.1' 'networkx == 1.11' 'flatbuffers == 1.10' 'protobuf == 3.13.0' 'ruamel.yaml == 0.15.81' 'psutil == 5.6.2' 'ply == 3.11' 'sklearn == 0.0' 'matplotlib == 3.3.4' 'tqdm == 4.63.0' 'flask == 2.0.2' 'Jinja2 == 3.0.0' 'requests == 2.22.0' 'onnxoptimizer >= 0.1.0' 'tensorflow == 1.14.0' 'onnx == 1.10.0' 'onnxruntime == 1.9.0' 'torch == 1.10.0' 'mxnet == 1.5.0'

注：

1. Windows 提供适配 Python3.6 的安装包；MacOS 提供适配 Python3.6 和 Python3.7 的安装包；ARM64 平台（安装 Debian 9 或 10 操作系统）提供适配 Python3.5（Debain 9）和适配 Python3.7（Debian10）的安装包。
2. Linux x86_64 平台从 RKNN Toolkit 1.7.3 版本开始，移除对 Python3.5 的支持，增加对 Python3.8 的支持。
3. 由于部分依赖在 Python3.8 环境中无法正常安装或使用，所以在 Linux x86_64 / Python3.8 环境中，将 TensorFlow 版本升级到 2.2.0。
4. 使用 pip 安装 RKNN Toolkit 时默认不会安装 tensorflow / torch / mxnet，请根据实际需

要安装相应的版本，以上表格中的版本为建议使用的版本。

5. ARM64 平台不需要依赖 flask, sklearn, requests。
6. Jinja2 只在使用自定义 OP 时用到。

2.3 连接 Rockchip NPU 设备

模型评估或部署时，需要连接 Rockchip NPU 设备。本节将给出 RK3399Pro, RK1808, RV1109, RV1126 开发板的连接方式，以及如何确认设备是否成功连接。

注：

1. 如果使用的是 Toybrick 开发板，请参考以下链接中的开机启动和串口调试内容：https://t.rock-chips.com/wiki.php?filename=%E7%BD%91%E7%AB%99%E5%AF%BC%E8%88%AA/%E9%A6%96%E9%A1%B5。

2.3.1 通过 USB-OTG 口连接设备

RKNN Toolkit 与 Rockchip NPU 开发板之间的通信是通过 USB-OTG 接口进行的。在模型评估或模型部署阶段，需要通过 USB-OTG 接口连接 PC 和 Rockchip NPU 设备，以完成相关工作。

RK3399Pro USB-OTG 接口位置如下图红圈所示，通过 USB Type-C 线与 PC 连接：

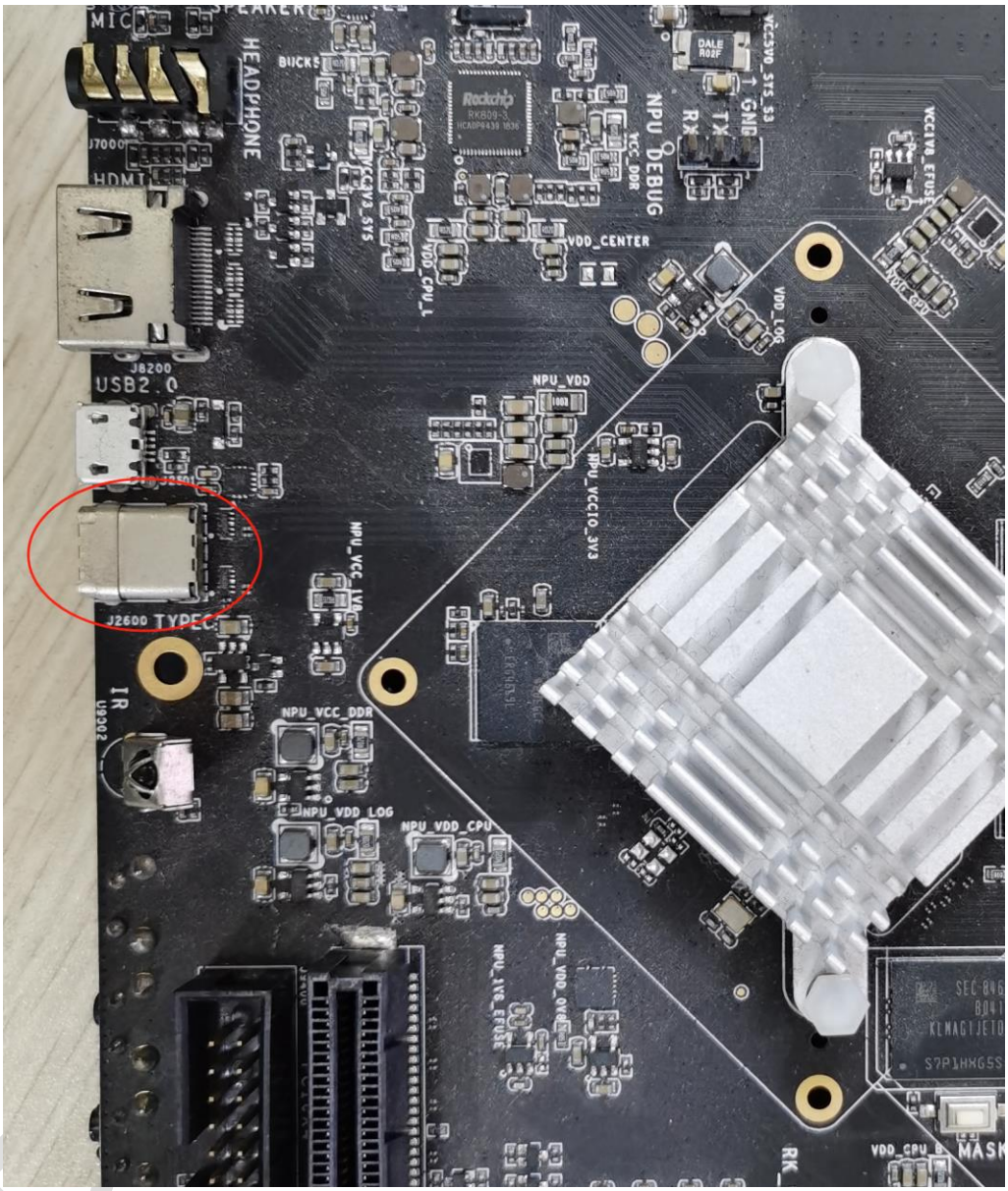

图 2-1 RK3399Pro USB-OTG 接口位置

RK1808 USB-OTG 接口位置如下图所示，通过 Micro-USB 线与 PC 连接：

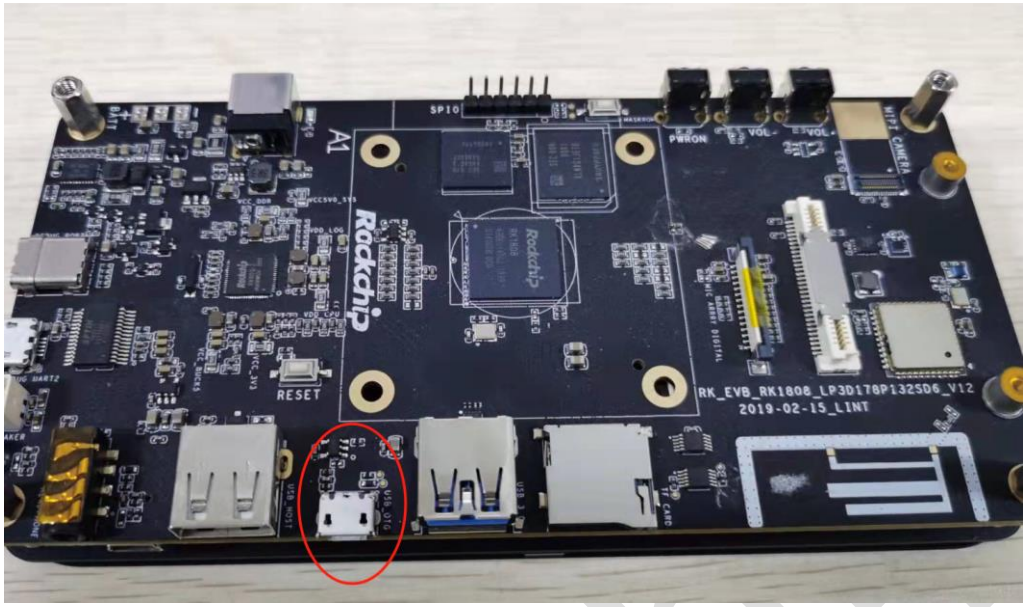

图 2-2 RK1808 USB-OTG 接口位置

RV1109/RV1126 的 USB-OTG 接口位置如下图红圈所示，通过 Micro-USB 线与 PC 连接：

图 2-3 RV1109/RV1126 USB-OTG 口位置

2.3.2 通过 DEBUG 口连接设备

在模型评估或模型部署阶段，如果遇到开发板上的错误，需要通过 DEBUG 口连接 NPU 设备，抓取上面的日志。

RK3399Pro NPU DEBUG 接口位置如下图所示，通过 USB-串口转换小板与 PC 连接，PC 端通过 Minicom 或 Putty 等软件与开发板通信：

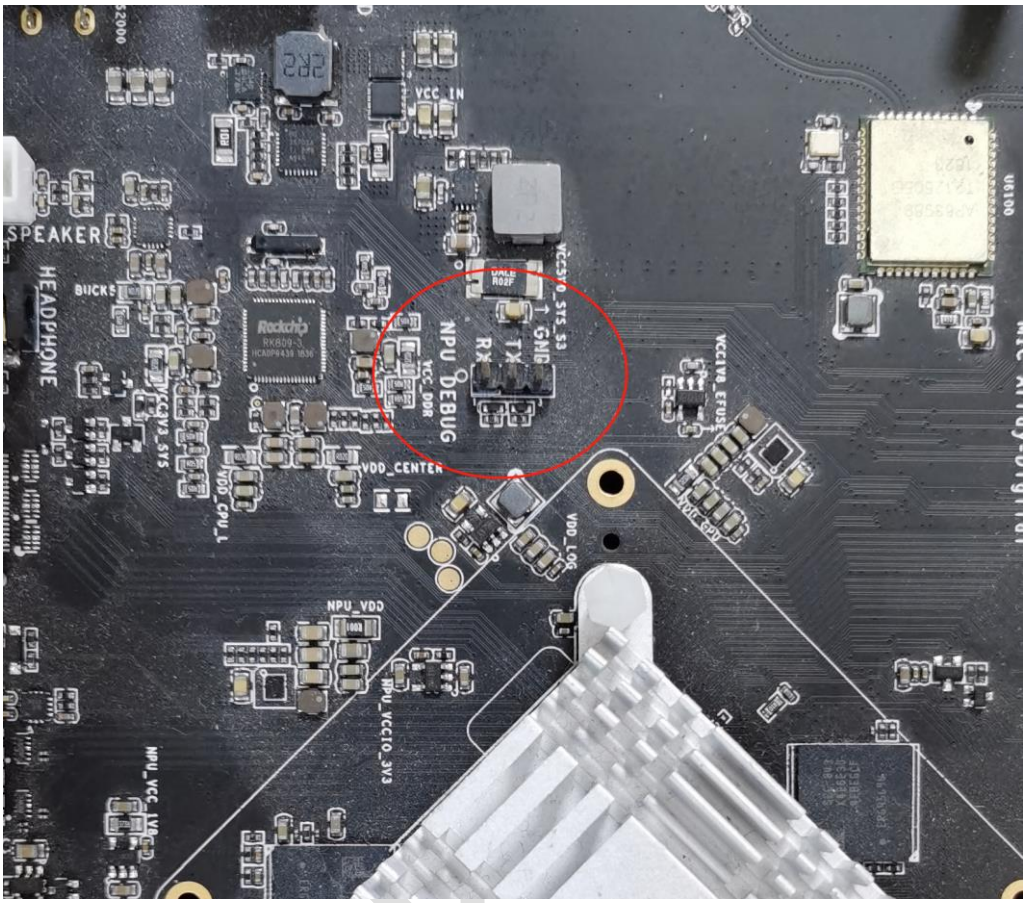

图 2-4 RK3399Pro DEBUG 接口位置

RK1808 DEBUG 接口位置如下图红圈所示，通过 Micro-USB 线与 PC 连接：

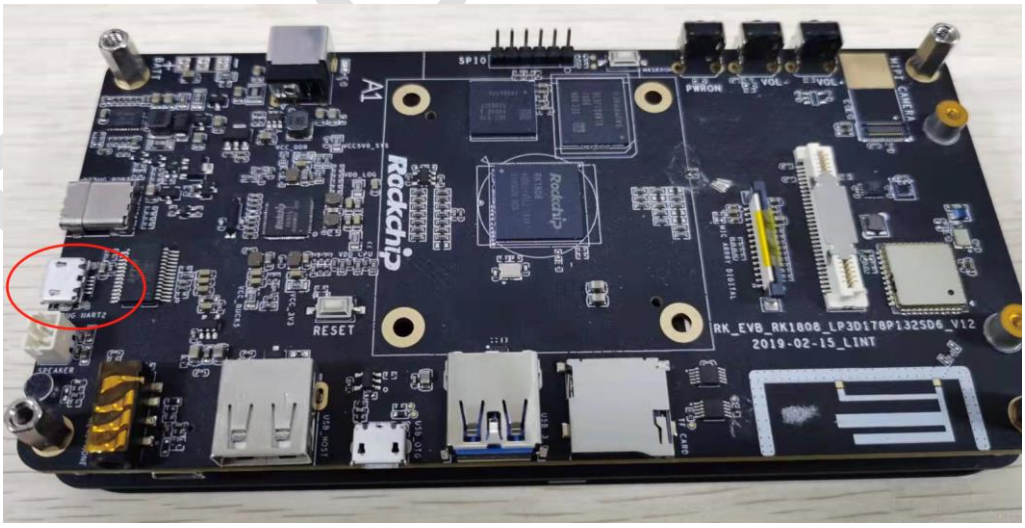

图 2-5 RK1808 DEBUG 接口位置

RV1109/RV1126 DEBUG 接口位置如下图红圈所示：

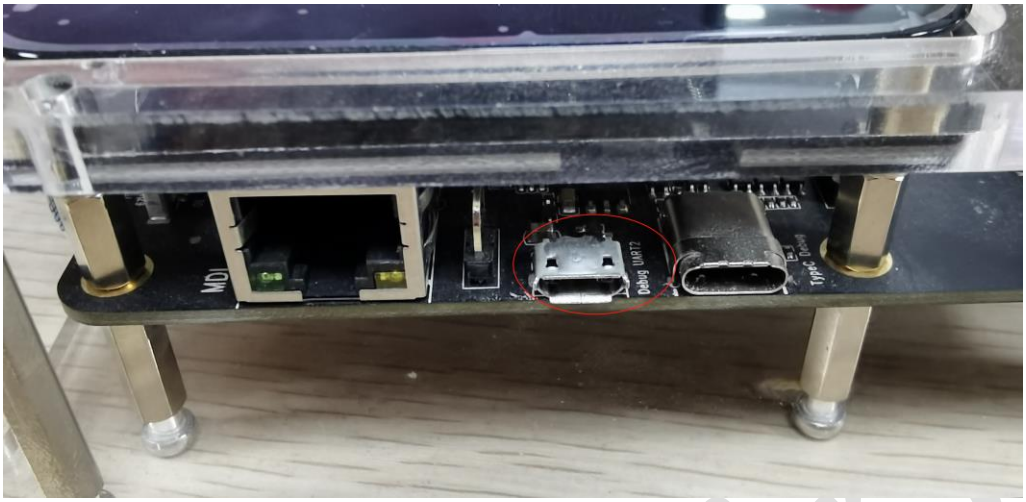

图 2-6 RV1109/RV1126 DEBUG 接口位置

2.3.3 确认设备连接正确

如果设备连接正确，在 PC 端的命令行窗口执行“python3 -m rknn.bin.list_devices”命令将列出所连接设备的 ID。参考输出如下：

```
**************************
all device(s) with adb mode:
cf8f01ae745b49ce
all device(s) with ntb mode:
1126
**************************
```

从这个输出可以看到，PC 目前连着两块开发板，第一块开发板的设备编号是 cf8f01ae745b49ce，第二块开发板的设备编号是 1126。

注：

1. 如果使用的是 x86_64 Linux 系统，第一次使用 Rockchip NPU 设备时可能需要更新 USB 设备权限，否则系统对这些设备没有读写权限。更新方法是执行 platform-tools/update_rk_usb_rule/linux/目录下的 update_rk1808_usb_rule.sh 脚本，执行完后需要重启系统。
2. 如果是在 Windows 上使用 Rockchip NPU 设备，需要先开启 NTB 通信模式（RK1808 和 Toybrick 开发板默认已开启），开启方法是通过 adb 进入开发板系统，修改文件 /etc/init.d/.usb_config，在这个文件中增加一行：usb_ntb_en，然后重启开发板。原来的

usb_adb_en 要保留，否则无法通过 adb 进入开发板系统。

2.3.4 常见问题

设备连接常见问题请参考《Rockchip_Trouble_Shooting_RKNN_Toolkit_CN.pdf》文档 1.5 章节。

3 Ubuntu 平台快速上手

本章节以 Ubuntu 18.04、Python3.6 为例说明如何快速上手使用 RKNN-Toolkit。

3.1 环境准备

- 一台安装有 ubuntu18.04 操作系统的 x86_64 位计算机。
- RK1808 或 RV1126 EVB 板。
- 将 RK1808 或 RV1126 EVB 板通过 USB 连接到 PC 上，使用 adb devices 命令查看，结果如下：

```
rk@rk:~$ adb devices
List of devices attached
515e9b401c060c0b        device
c3d9b8674f4b94f6        device
```

其中红色字段是设备编号。

3.2 安装 RKNN-Toolkit

1. 安装 Python3.6

```
sudo apt-get install python3
```

2. 安装 pip3

```
sudo apt-get install python3-pip
```

3. 获取 RKNN-Toolkit 安装包，然后执行以下步骤：

   a) 进入 package 目录：

```
cd package/
```

   b) 安装 Python 依赖

```
pip3 install tensorflow==1.14.0
pip3 install mxnet==1.5.0
pip3 install torch==1.10.0+cpu -f https://download.pytorch.org/whl/torch_stable.html
pip3 install torchvision==0.11.0+cpu -f https://download.pytorch.org/whl/torch_stable.html
pip3 install gluoncv
```

c) 安装 RKNN-Toolkit

```
sudo pip3 install rknn_toolkit-1.7.x-cp36-cp36m-linux_x86_64.whl
```

d) 检查 RKNN-Toolkit 是否安装成功

```
rk@rk:~/rknn-toolkit-v1.7.x/package$ python3
>>> from rknn.api import RKNN
>>>
```

如果导入 RKNN 模块没有失败，说明安装成功。

3.3 运行安装包中附带的示例

3.3.1 在 RV1126 上运行示例

这里以 mobilenet_v1 为例，运行该示例的具体步骤如下：

1. 进入 examples/tflite/mobilenet_v1 目录

```
rk@rk:~/rknn-toolkit-v1.7.x/examples/tflite/mobilenet_v1$
```

2. 执行 test.py 脚本，输出分类结果 TOP5 和性能数据：

```
mobilenet_v1
-----TOP 5-----
[156]: 0.8603515625
[155]: 0.0833740234375
[205]: 0.0123443603515625
[284]: 0.00726318359375
[260]: 0.002262115478515625

============================================================
                        Performance
============================================================
Total Time(us): 4759
FPS: 210.13
============================================================
```

这个例子涉及到的主要流程包括：创建 RKNN 对象；模型配置；加载 TensorFlow Lite 模型；构建 RKNN 模型；导出 RKNN 模型；加载输入图形并推理，得到 TOP5 类别和对应的得分；评估模型性能；释放 RKNN 对象。

Examples 目录中的其他示例，执行方式与 mobilenet_v1 示例相同。这些示例主要面向分类，目标检测和图像分割等任务。

注：如果 PC 连接多个 RKNPU 设备，可以在执行”test.py”脚本时指定设备型号和 ID 进行区分，示例如下：

```
rk@rk:~/mobilenet_v1$ python test.py rv1126 c3d9b8674f4b94f6
```

3.3.2 在 RK1808 上运行示例

工具包中带的 mobilenet_v1 示例是在 RV1126 上运行的，如果要在 RK1808 EVB 板上运行这个示例，可以参考以下步骤：

1. 进入 examples/tflite/mobilenet_v1 目录

```
rk@rk:~/rknn-toolkit-v1.7.x/examples/tflite/mobilenet_v1$
```

2. 执行 test.py 脚本，指定目标设备平台，示例如下：

```
rk@rk:~/rknn-toolkit-v1.7.x/examples/tflite/mobilenet_v1$ python test.py rk1808
```

注：如果有多个设备，在执行命令最后加上设备 ID 进行区分。

3. 输出分类结果 TOP5 和性能数据：

```
mobilenet_v1
-----TOP 5-----
[156]: 0.85205078125
[155]: 0.09185791015625
[205]: 0.012237548828125
[284]: 0.006473541259765625
[194]: 0.0024929046630859375

When performing performance evaluation, inputs can be set to None to use fake inputs.
============================================================
                        Performance
============================================================
Total Time(us): 5499
FPS: 181.85
============================================================
```

4 Windows 平台快速上手指南

本章节说明如何在 Windows 系统、Python3.6 环境中使用 RKNN-Toolkit。

4.1 环境准备

- 一台安装有 Windows 10（或 Windows7）操作系统的 PC。
- 一个 RK1808 或 RV1126 开发板。
- 将 RK1808 或 RV1126 开发板通过 USB 连接到 PC 上。第一次使用开发板时需要安装相应的驱动，安装方式如下：
  - 进入 SDK 包 platform-tools/drivers_installer/windows-x86_64 目录，以管理员身份运行 zadig-2.4.exe 程序安装计算棒的驱动，如下图所示：

    1. 确认待安装的设备及需要安装的驱动，确认完后点 Install Driver 开始安装驱动。

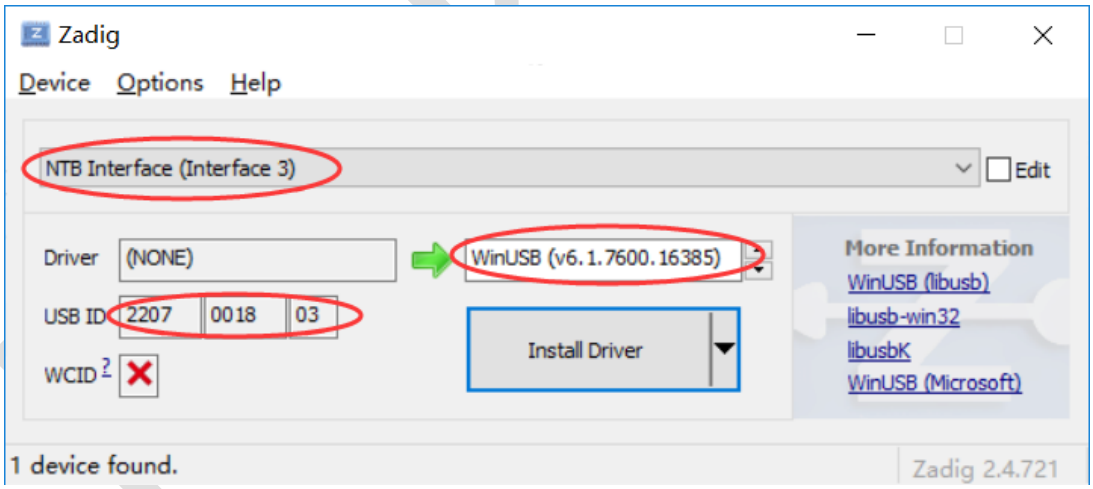

图 4-1 驱动安装界面

注：待安装的设备 USB ID 以 **2207** 开头；安装的驱动选择默认的 WinUSB。

2. 安装成功后会出现如下界面：

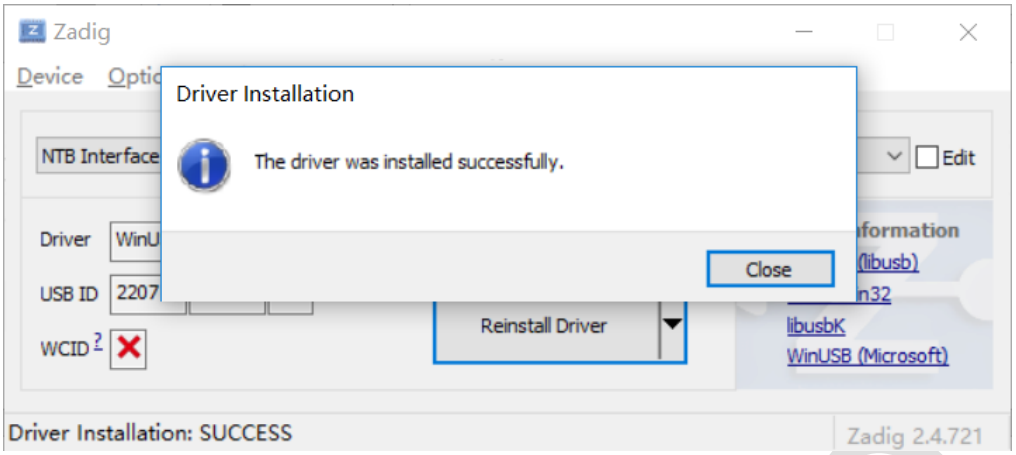

图 4-2 驱动安装成功示例

- 安装完后如果 windows 设备管理器的中的 NTB Interface 设备没有感叹号，且如下所示，说明安装成功：

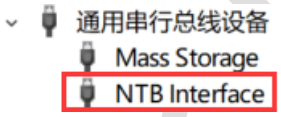

图 4-3 驱动安装成功时的设备状态

注：安装完驱动后需要重启计算机。

4.2 安装 RKNN-Toolkit

安装 RKNN-Toolkit 前需要确保系统里已经安装有 Python3.6。在 cmd 里执行 python –version 确定，执行结果如下说明 Python3.6 已经安装：

```
C:\Users\rk>python –version
Python 3.6.8
```

获取 RKNN-Toolkit SDK 包，然后执行以下步骤：

1. 在 sdk 根目录以管理员权限执行 cmd，然后进入 package 目录：

```
D:\workspace\rknn-toolkit-v1.7.x>cd packages
```

2. 安装 Python 依赖：

```
pip install tensorflow==1.14.0
pip install torch==1.10.0+cpu -f https://download.pytorch.org/whl/torch_stable.html
pip install torchvision==0.11.0+cpu -f https://download.pytorch.org/whl/torch_stable.html
pip install mxnet==1.5.0
pip install gluoncv
```

注：gluoncv 在运行 examples/mxnet 中的例子时会用到。

3. 安装 RKNN-Toolkit

```
pip install rknn_toolkit-1.7.x-cp36-cp36m-win_amd64.whl
```

4. 检查 RKNN-Toolkit 是否安装成功

```
D:\workspace\rknn-toolkit-v1.7.x\packages>python
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from rknn.api import RKNN
>>>
```

如果导入 RKNN 模块没有失败，说明安装成功。

4.3 在 RV1126 或 RK1808 上运行示例

Windows 平台上使用 RV1126 或 RK1808 运行示例的步骤与 Ubuntu 平台相同，这里不再赘述。

5 Mac OS X 平台快速上手指南

本章节说明如何在 Mac OS X 系统、Python3.6 环境中使用 RKNN-Toolkit。

5.1 环境准备

- 一台安装有 MacOS 10.13.5（或更高版本）操作系统的 Mac PC。
- 一个 RK1808 或 RV1126 开发板。
- 将 RK1808 或 RV1126 开发板通过 USB（OTG 口）连接到 PC 上。在 PC 上进入 SDK 包 platform-tools/ntp/mac-osx-x86_64 目录，运行 npu_transfer_proxy 程序查看是否存在可用的 Rockchip NPU 设备，命令如下：

```
macmini:ntp rk$ ./npu_transfer_proxy devices
List of ntb devices attached
515e9b401c060c0b        2bed0cc1        USB_DEVICE
```

其中红色字段是设备编号。

5.2 安装 RKNN-Toolkit

获取 RKNN-Toolkit SDK 包，执行以下步骤：

1. 进入 rknn-toolkit-v1.7.x/packages 目录：

```
cd packages/
```

2. 安装 Python 依赖

```
pip3 install tensorflow==1.14.0
pip3 install mxnet==1.5.0
pip3 install torch==1.10.0 torchvision==0.11.0
pip3 install gluoncv
```

注：examples/mxnet 中的例子依赖 gluoncv。

3. 安装 RKNN-Toolkit

```
pip3 install rknn_toolkit-1.7.x-cp36-cp36m-macosx_10_15_x86_64.whl
```

4. 检查 RKNN-Toolkit 是否安装成功

```
(rknn-venv)macmini:rknn-toolkit-v1.7.x rk$ python3
>>> from rknn.api import RKNN
>>>
```

如果导入 RKNN 模块没有失败，说明安装成功。

5.3 在 RV1126 或 RK1808 上运行示例

MacOS 平台上使用 RV1126 或 RK1808 运行示例的步骤与 Ubuntu 平台相同，这里不再赘述。

6 ARM64 平台快速上手指南

本章节说明如何在 ARM64 平台（Debian 9.8 系统）、Python3.5 环境中使用 RKNN-Toolkit。

注：模型转换过程需要使用较多的 CPU / 内存资源，建议这部分操作在 PC 端完成，之后再用 RKNN Toolkit Lite 提供的 Python 接口或者 RKNPU C API 接口在 RK3399Pro 板子上部署模型。

6.1 环境准备

- 一台安装有 Debian 9.8 操作系统的 RK3399Pro，并且确保 root 分区剩余空间大于 5GB。
- 如果/usr/bin 目录下没有 npu_transfer_proxy 或 npu_transfer_proxy.proxy 程序，则将 rknn-toolkit-v1.7.x/platform-tools/ntp/linux_aarch64 目录下的 npu_transfer_proxy 拷贝到/usr/bin/目录下，并进到该目录执行以下命令（每次重启后都要启动该程序，可以将它加到开机脚本中）：

```
sudo ./npu_transfer_proxy &
```

6.2 安装 RKNN-Toolkit

1. 执行以下命令更新系统包，这些包在后面安装 Python 依赖包时会用到。

```
sudo apt-get update
sudo apt-get install cmake gcc g++ libprotobuf-dev protobuf-compiler
sudo apt-get install liblapack-dev libjpeg-dev zlib1g-dev
sudo apt-get install python3-dev python3-pip python3-scipy
```

2. 执行以下命令更新 pip

```
pip3 install --upgrade pip
```

3. 安装 Python 打包工具

```
pip3 install wheel setuptools
```

4. 安装依赖包 h5py/gluoncv

```
sudo apt-get build-dep python3-h5py && \
pip3 install h5py
pip3 install gluoncv
```

5. 安装 TensorFlow，由于 debian9 系统上没有相应的源，需要在网上自行搜索 whl 包安装。
6. 安装 torch 和 torchvision，由于 debian9 系统上没有相应的源，需要在网上自行搜索 whl 包安装。
7. 安装 mxnet，由于 debian9 系统上没有相应的源，需要在网上自行搜索 whl 包安装。
8. 安装 opencv-python，由于 debian9 系统上没有相应的源，需要在网上自行搜索 whl 包安装。
9. 安装 RKNN-Toolkit，相应的 whl 包在 rknn-toolkit-v1.7.x\packages\目录下：

```
pip3 install rknn_toolkit-1.7.x-cp35-cp35m-linux_aarch64.whl --user
```

注：由于 RKNN-Toolkit 依赖的一些库在 ARM64 平台上需要下载源码后编译安装，所以这一步会耗费较长时间。

6.3 运行安装包中附带的示例

这里以 mobilenet_v1 为例。示例中的 mobilenet_v1 是一个 Tensorflow Lite 模型，用于图片分类。

运行该示例的步骤如下：

1. 进入 examples/tflite/mobilenet_v1 目录。

```
linaro@linaro-alip:~/rknn-toolkit-v1.7.x/ $ cd examples/tflite/mobilenet_v1
```

2. 执行 test.py 脚本，并指定目标平台为 RK3399Pro。

```
linaro@linaro-alip: ~/rknn-toolkit-v1.7.x/examples/tflite/mobilenet_v1$ python3 test.py rk3399pro
```

3. 脚本执行后得到如下结果：

```
mobilenet_v1
-----TOP 5-----
[156]: 0.85107421875
[155]: 0.09173583984375
[205]: 0.01358795166015625
[284]: 0.006465911865234375
[194]: 0.002239227294921875

============================================
                Performance
============================================
Total Time(us): 5761
FPS: 173.58
============================================
```

这个例子涉及到的主要操作有：创建 RKNN 对象；模型配置；加载 TensorFlow Lite 模型；构建 RKNN 模型；导出 RKNN 模型；加载图片并推理，得到 TOP5 结果；评估模型性能；释放 RKNN 对象。

examples 目录中的其他示例的执行方式与 mobilenet_v1 相同，这些模型主要用于图像分类、目标检测和图像分割。

注：RK3399Pro 上的内存资源有限，可能导致部分示例运行失败。

7 模型部署快速上手指南

此章节以 rknn_mobilenet_demo 在 RV1126 Linux ARM32 位平台上运行为例，说明如何通过 RKNPU 工程提供的 C API 将 RKNN 模型部署到瑞芯微 NPU 平台上。

对于 RK3399Pro 平台，请参考如下工程：https://github.com/airockchip/RK3399Pro_npu

7.1 下载 RKNPU 工程

RKNPU 工程链接：https://github.com/rockchip-linux/rknpu

下载命令参考如下：

```
rk@rk:~/npu/$ git clone https://github.com/rockchip-linux/rknpu.git
```

7.2 准备编译工具

在 Linux x86 平台上编译 ARM 平台的可执行程序，需要借助 ARM 提供的交叉编译工具链。

7.2.1 下载交叉编译工具

对于 RV1109 / RV1126 平台，请通过如下链接下载交叉编译工具：

https://developer.arm.com/-/media/Files/downloads/gnu-a/8.3-2019.03/binrel/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf.tar.xz?revision=e09a1c45-0ed3-4a8e-b06b-db3978fd8d56&rev=e09a1c450ed34a8eb06bdb3978fd8d56&hash=9C4F2E8255CB4D87EABF5769A2E65733

对于 RK1808 平台，请通过如下链接下载交叉编译工具：

https://releases.linaro.org/components/toolchain/binaries/6.3-2017.05/aarch64-linux-gnu/gcc-linaro-6.3.1-2017.05-x86_64_aarch64-linux-gnu.tar.xz

对于 RK1806 平台，请通过如下链接下载交叉编译工具：

https://releases.linaro.org/components/toolchain/binaries/6.3-2017.05/arm-linux-gnueabihf/gcc-linaro-6.3.1-2017.05-x86_64_arm-linux-gnueabihf.tar.xz

7.2.2 设置编译工具路径

将上一步下载的编译工具解压缩，记录文件夹的绝对路径。

打开 DEMO 编译脚本 build.sh，修改对应平台的 GCC_COMPILER，例如目标平台是 RV1126，则将脚本修改如下：

```
GCC_COMPILER=
~/opt/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf/bin/arm-linux-gnueabihf
```

其中~/opt/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf 为前一步解压缩后的路径。

7.3 更新 RKNN 模型

参考第 3.3 章节运行 RKNN Toolkit 示例，生成 RKNN 模型。将生成的 RKNN 模型复制到 rknpu/rknn/rknn_api/examples/rknn_mobilenet_demo/model 目录下，根据相应平台重命名 RKNN 模型。

7.4 编译示例程序

1. 在终端命令行窗口进入 rknn_mobilenet_demo 文件夹

```
cd rknpu/rknn/rknn_api/examples/rknn_mobilenet_demo/model
```

2. 运行 build.sh 脚本编译程序

```
./build.sh
```

注：如果在编译时提示 cmake 未安装，可以执行如下命令安装 cmake 后再运行编译脚本。

```
sudo apt install cmake
```

7.5 在开发板上运行示例

参考 2.3.1 章节通过 USB-OTG 口将设备连接到 PC 上。

1. 把编译好的程序（install/rknn_mobilenet_demo 文件夹）上传到开发板的/data/目录下。

```
adb push install/rknn_mobilenet_demo /data/
```

2. 进入开发板系统

```
adb shell
```

3. 进入程序所在目录

```
cd /data/rknn_mobilenet_demo
```

4. 运行程序识别测试图片的类别。

用法：./rknn_mobilenet_demo <rknn_model> <image>

- rknn_model: 模型路径。根据具体的平台选择。
- image: 测试图片路径。

参考命令：

```
./rknn_mobilenet_demo ./model/mobilenet_v1_rv1109_rv1126 model/dog_224x224.jpg
```

注：也可以直接执行 DEMO 提供的 run_rk180x.sh 或 run_rv1109_rv1126.sh 脚本运行实例程序。

5. 参考结果如下：

```
rknn_run
    0: Elapse Time = 3.85ms, FPS = 259.47
 --- Top5 ---
156: 0.865723
155: 0.083862
205: 0.012428
284: 0.005913
194: 0.001842
```

注：

- 分类结果中得分最高的索引号是 156，该索引号对应的标签是`Pekinese, Pekingese, Peke`。
- 打印的耗时是调用模型推理接口 rknn_run 的耗时，FPS 根据该耗时换算而来。

- 使用不同平台/不同版本的 RKNN Toolkit 和 NPU 驱动，推理结果和耗时可能存在细微差异。

8 附录

8.1 参考文档

OP 支持列表：《RKNN_OP_Support.md》

RKNN Toolkit 使用指南：《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》

问题排查手册：《Rockchip_Trouble_Shooting_RKNN_Toolkit_CN.pdf》

自定义 OP 使用指南：《Rockchip_Developer_Guide_RKNN_Toolkit_Custom_OP_CN.pdf》

可视化功能使用指南：《Rockchip_User_Guide_RKNN_Toolkit_Visualization_CN.pdf》

RKNN Toolkit Lite 使用指南：《Rockchip_User_Guide_RKNN_Toolkit_Lite_CN.pdf》

以上文档均存放在 SDK/doc 目录中，也可以访问以下链接查阅：

https://github.com/rockchip-linux/rknn-toolkit/tree/master/doc

8.2 问题反馈渠道

请通过 RKNN QQ 交流群，Github Issue 或瑞芯微 redmine 将问题反馈给 Rockchip NPU 团队。

RKNN QQ 交流群：1025468710

Github issue: https://github.com/rockchip-linux/rknn-toolkit/issues

Rockchip Redmine: https://redmine.rock-chips.com/

注：Redmine 账号需要通过销售或业务人员开通。如果是第三方开发板，请先找对应厂商反馈问题。