

密级状态：绝密() 秘密() 内部() 公开(√)

RKNN Toolkit Lite2 用户使用指南

(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： | V1.5.2 |
| | 作 者： | 饶洪 |
| | 完成日期： | 2023-08-21 |
| | 审 核： | 熊伟 |
| | 完成日期： | 2023-08-21 |

瑞芯微电子股份有限公司

Rockchip Electronics 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 |
|--------|-----|------------|--|-----|
| V1.2.0 | 饶洪 | 2022-01-13 | 初始版本。 | 熊伟 |
| V1.3.0 | 饶洪 | 2022-04-27 | 更新版本号。 | 熊伟 |
| V1.4.0 | 饶洪 | 2022-08-30 | 更新 init_runtime 接口 core_mask 参数的说明。 | 熊伟 |
| V1.5.0 | 饶洪 | 2023-05-18 | 1. 更新适配芯片，新增 RK3562； 2. 更新适配的 Python 版本，增加 ARM64 平台 Python3.8 和 Python3.10 的支持。 | 熊伟 |
| V1.5.2 | 饶洪 | 2023-08-21 | 更新版本号 | 熊伟 |

目 录

| | | |
|----------|-------------------|-----------|
| 1 | 主要功能说明 | 4 |
| 1.1 | 适用芯片 | 4 |
| 1.2 | 适用系统 | 4 |
| 2 | 开发环境部署 | 5 |
| 2.1 | 系统依赖说明 | 5 |
| 2.2 | 工具安装 | 5 |
| 3 | 使用说明 | 7 |
| 3.1 | 基本使用流程 | 7 |
| 3.2 | 示例 | 8 |
| 4 | API 详细说明 | 9 |
| 4.1 | RKNNLITE 初始化及对象释放 | 9 |
| 4.2 | 加载 RKNN 模型 | 9 |
| 4.3 | 初始化运行时环境 | 10 |
| 4.4 | 模型推理 | 11 |
| 4.5 | 查询 SDK 版本 | 12 |
| 4.6 | 查询模型可运行平台 | 13 |
| 5 | 附录 | 14 |
| 5.1 | 参考文档 | 14 |
| 5.2 | 问题反馈渠道 | 14 |

1 主要功能说明

RKNN Toolkit Lite2 为 Rockchip NPU 平台提供 Python 编程接口，帮助用户部署 RKNN 模型，加速 AI 应用的落地。

1.1 适用芯片

- RK3562
- RK3566
- RK3568
- RK3588 / RK3588S

注：文中统一用 RK3588 指代 RK3588 和 RK3588S。

1.2 适用系统

- Debian: 10 (aarch64)
- Debian: 11 (aarch64)

2 开发环境部署

2.1 系统依赖说明

本开发套件支持运行于 Debian 10 / 11 (aarch64) 操作系统。需要满足以下运行环境要求：

表 1 运行环境

| | |
|------------|---------------------------------------|
| 操作系统版本 | Debian 10 / 11 (aarch64) |
| Python 版本 | 3.7 / 3.8 / 3.9 / 3.10 |
| Python 库依赖 | 'numpy' 'ruamel.yaml' 'psutils' |

2.2 工具安装

目前可以通过 `pip3 install` 命令安装 RKNN Toolkit Lite2。

1. 如果系统中没有安装 `python3/pip3` 等程序，请先通过 `apt-get` 方式安装，命令如下：

```
sudo apt-get update
sudo apt-get install -y python3 python3-dev python3-pip gcc
```

注：安装部分依赖模块时，需要编译源码，所以将 `python3-dev` 和 `gcc` 这两个包也一并安装，避免后面安装依赖模块时编译失败。

2. 安装依赖模块: `opencv-python` 和 `numpy`

```
sudo apt-get install -y python3-opencv
sudo apt-get install -y python3-numpy
```

注：

- 1) RKNN Toolkit Lite2 本身并不依赖 `opencv-python`，但是在示例中需要使用该模块对图像进行处理。
 - 2) 在 Debian10 固件上直接通过 `pip3` 安装 `numpy` 可能失败，建议用上述方法安装。
3. 安装 RKNN Toolkit Lite2

各平台的安装包都放在 SDK 的 `packages` 文件夹下。进入 `packages` 文件夹，执行以下命令

安装 RKNN Toolkit Lite2:

```
# Python 3.7
pip3 install rknn_toolkit_lite2-1.x.0-cp37-cp37m-linux_aarch64.whl
# Python 3.8
pip3 install rknn_toolkit_lite2-1.x.0-cp38-cp38-linux_aarch64.whl
# Python 3.9
pip3 install rknn_toolkit_lite2-1.x.0-cp39-cp39-linux_aarch64.whl
# Python 3.10
pip3 install rknn_toolkit_lite2-1.x.0-cp310-cp310-linux_aarch64.whl
```

3 使用说明

RKNN Toolkit Lite2 主要用于 RKNN 模型在 Rockchip NPU 上的部署。

在使用 RKNN Toolkit Lite2 之前，用户需要先通过 RKNN Toolkit2 将各深度学习框架导出的模型转成 RKNN 模型。

RKNN Toolkit2 完整的安装包和使用文档可以从以下链接获取：

<https://github.com/rockchip-linux/rknn-toolkit2>

3.1 基本使用流程

使用 RKNN Toolkit Lite2 部署 RKNN 模型的基本流程如下图所示：

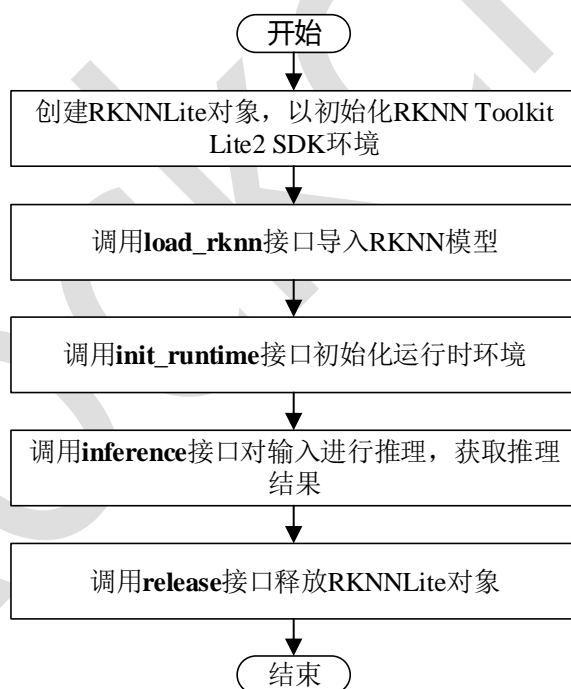


图 3-1-1 RKNN Toolkit Lite2 的基本使用流程

注：

1. 在调用 inference 接口进行推理之前，需要获取输入数据，并做相应的预处理，然后根据输入信息设置 inference 接口的中 data_format 等参数；
2. 在调用 inference 接口后，通常需要对推理结果进行相应的处理，以完成上层应用相关功

能。

3.2 示例

在 SDK/examples 目录提供了一个使用 RKNN Toolkit Lite2 进行模型推理的示例 inference_with_lite。执行该示例将加载 Resnet18 模型并进行推理，得到测试图片的 top5 分类结果。

运行该示例的方法：

1. 准备一块安装有 RKNN Toolkit Lite2 的开发板；
2. 将该示例推到开发板上；
3. 在开发板上进入 examples/inference_with_lite 目录，执行如下命令运行示例：

```
python3 test.py
```

参考运行结果如下所示：

```
-----TOP 5-----  
[812]: 0.9996383190155029  
[404]: 0.00028062614728696644  
[657]: 1.6321087969117798e-05  
[833 895]: 1.015903580992017e-05  
[833 895]: 1.015903580992017e-05
```


4 API 详细说明

4.1 RKNNLite 初始化及对象释放

在使用 RKNN Toolkit Lite2 时，需要先调用 RKNNLite()方法初始化一个 RKNNLite 对象，并在用完后调用该对象的 release()方法将资源释放掉。

初始化 RKNNLite 对象时，可以设置 *verbose* 和 *verbose_file* 参数，以打印详细的日志信息。其中 *verbose* 参数指定是否要在屏幕上打印详细日志信息；如果设置了 *verbose_file* 参数，且 *verbose* 参数值为 True，日志信息还将写到这个参数指定的文件中。

举例如下：

```
# 将详细的日志信息输出到屏幕，并写到 inference.log 文件中
rknn_lite = RKNNLite(verbose=True, verbose_file='./inference.log')
# 只在屏幕打印详细的日志信息
rknn_lite = RKNNLite(verbose=True)
...
rknn_lite.release()
```

4.2 加载 RKNN 模型

| API | load_rknn |
|-----|--------------------|
| 描述 | 加载 RKNN 模型。 |
| 参数 | path: RKNN 模型文件路径。 |
| 返回值 | 0: 加载成功 |
| | -1: 加载失败 |

举例如下：

```
# 从当前目录加载 resnet_18.rknn 模型
ret = rknn_lite.load_rknn('./resnet_18.rknn')
```

4.3 初始化运行时环境

在模型推理之前，必须先初始化运行时环境。

| | |
|-----|---|
| API | init_runtime |
| 描述 | 初始化运行时环境。 |
| 参数 | <p>core_mask: NPU 工作核心配置模式。</p> <p>RKNNLite.NPU_CORE_AUTO: 自动调度模式，自动运行在当前空闲的 NPU 核上。</p> <p>RKNNLite.NPU_CORE_0: 运行在 NPU0 核心上。</p> <p>RKNNLite.NPU_CORE_1: 运行在 NPU1 核心上。</p> <p>RKNNLite.NPU_CORE_2: 运行在 NPU2 核心上。</p> <p>RKNNLite.NPU_CORE_0_1: 表示同时运行在 NPU0、NPU1 核心上。</p> <p>RKNNLite.NPU_CORE_0_1_2: 表示同时运行在 NPU0、NPU1、NPU2 核心上。</p> <p>默认值为 NPU_CORE_AUTO。</p> <p>注：该参数只对 RK3588 有效。</p> |
| 返回值 | 0: 初始化运行时环境成功。 |
| | -1: 初始化运行时环境失败。 |

举例如下：

```
# 初始化运行时环境
ret = rknn_lite.init_runtime(core_mask=RKNNLite.NPU_CORE_AUTO)
if ret != 0:
    print('Init runtime environment failed')
    exit(ret)
```

4.4 模型推理

| | |
|-----|--|
| API | inference |
| 描述 | 对指定输入进行推理，返回推理结果。 |
| 参数 | inputs : 待推理的输入，如 OpenCV 读取的图片。类型是 list，列表成员是 ndarray。 |
| | data_format : 数据排列方式，可选值"nhwc"，只对四维输入有效。默认值为 None，表示所有输入的 layout 都是 NHWC。 |
| 返回值 | results : 推理结果，类型是 list，列表成员是 ndarray。 |

举例如下：

以分类模型为例，如 resnet18，代码如下（完整代码参考 `examples/inference_with_lite`）：

```
# 使用模型对图片进行推理，得到 TOP5 结果
.....
outputs = rknn_lite.inference(inputs=[img])
show_top5(outputs)
.....
```

输出的 TOP5 结果如下：

```
-----TOP 5-----
[812]: 0.999442994594574
[404]: 0.0004096269840374589
[657]: 3.284541890025139e-05
[833]: 2.6112385967280716e-05
[895]: 1.8509887013351545e-05
```

4.5 查询 SDK 版本

| | |
|-----|--|
| API | get_sdk_version |
| 描述 | 获取 Runtime, 驱动和 RKNN 模型版本信息。 注：使用该接口前必须完成模型加载和初始化运行环境。 |
| 参数 | 无 |
| 返回值 | sdk_version: runtime, 驱动和 RKNN 模型版本信息。类型为字符串。 |

举例如下：

```
# 获取 SDK 版本信息
.....
sdk_version = rknn_lite.get_sdk_version()
.....
```

返回的 SDK 信息参考如下：

```
I RKNN: [10:47:23.097] RKNN Runtime Information: librknrt version: 1.2.0b1
(a47985372@2021-12-06T10:20:14)
I RKNN: [10:47:23.097] RKNN Driver Information: version: 0.6.2
I RKNN: [10:47:23.098] RKNN Model Information: version: 1, toolkit version: 1.1.2(compiler
version: 1.1.2b13 (1e5726f57@2021-11-29T14:13:55)), target: RKNPU v2, target platform: rk3588,
framework name: PyTorch, framework layout: NCHW
```

4.6 查询模型可运行平台

| | |
|-----|--|
| API | list_support_target_platform |
| 描述 | 查询给定 RKNN 模型可运行的芯片平台。 |
| 参数 | rknn_model: RKNN 模型路径。如果不指定模型路径，则按类别打印 RKNN Toolkit Lite2 当前支持的芯片平台。 |
| 返回值 | support_target_platform: 返回模型可运行的芯片平台。如果 RKNN 模型路径为空或不存在，返回 None。 |

参考代码如下所示：

```
rknn_lite.list_support_target_platform(rknn_model='mobilenet_v1.rknn')
```

参考结果如下：

```
*****
Target platforms filled in RKNN model:    ['RK3588']
Target platforms supported by this RKNN model: ['RK3588']
*****
```

5 附录

5.1 参考文档

RKNN Toolkit2 模型转换相关文档请参考以下链接：

<https://github.com/rockchip-linux/rknn-toolkit2/tree/master/doc>

5.2 问题反馈渠道

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

- RKNN QQ 交流群：1025468710
- Github issue: <https://github.com/rockchip-linux/rknn-toolkit2/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，请先找该厂商反馈问题。