

密级状态：绝密( ) 秘密( ) 内部( ) 公开(√)

## RKNN-Toolkit ARM 平台快速上手指南

(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：	V1.0.3b1
	作 者：	饶洪
	完成日期：	2019-05-27
	审 核：	卓鸿添
	完成日期：	2019-05-27

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
V1.0.3b1	饶洪	2019-05-27	初始版本	卓鸿添

## 目 录

1	主要功能说明 .....	1
2	系统依赖说明 .....	2
3	ARM64 平台（PYTHON3.5）快速上手指南 .....	3
3.1	环境准备 .....	3
3.2	安装 RKNN-TOOLKIT.....	3
3.3	运行安装包中附带的示例.....	4

## 1 主要功能说明

RKNN-Toolkit 是为用户提供在 PC、RK3399Pro、RK1808 或 RK3399Pro Linux 开发板上进行模型转换、推理和性能评估的开发套件，用户通过提供的 python 接口可以便捷地完成以下功能：

1) 模型转换：支持 Caffe、TensorFlow、TensorFlow Lite、ONNX、Darknet 模型转成 RKNN 模型，支持 RKNN 模型导入导出，后续能够在硬件平台上加载使用。

2) 模型推理：能够在 PC 上模拟运行模型并获取推理结果；也可以在指定硬件平台 RK3399Pro（或 RK3399Pro Linux 开发板）、RK1808 上运行模型并获取推理结果。

3) 性能评估：能够在 PC 上模拟运行并获取模型总耗时及每一层的耗时信息；也可以通过联机调试的方式在指定硬件平台 RK3399Pro、RK1808 上运行模型，或者直接在 RK3399Pro Linux 开发板上运行，以获取模型在硬件上完整运行一次所需的总时间和每一层的耗时情况。

4) 获取模型运行时的内存使用情况：通过联机调试的方式获取模型在指定硬件平台 RK3399Pro、RK1808 或 RK3399Pro Linux 开发板上运行时的内存使用情况。

5) 量化功能：支持将浮点模型转成量化模型，目前支持的量化方法有非对称量化（asymmetric\_quantized-u8），动态定点量化（dynamic\_fixed\_point-8 和 dynamic\_fixed\_point-16）。

## 2 系统依赖说明

本开发套件支持运行于 Ubuntu / Debian 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	Ubuntu16.04 (x64) 以上或 Debian9.8 以上
Python 版本	3.5/3.6
Python 库依赖	'numpy >= 1.16.1' 'scipy >= 1.1.0' 'Pillow >= 3.1.2' 'h5py >= 2.7.1' 'lmbd >= 0.92' 'networkx == 1.11' 'flatbuffers == 1.9', 'protobuf >= 3.5.2' 'onnx >= 1.3.0' 'flask >= 1.0.2' 'tensorflow >= 1.11.0' 'dill==0.2.8.2' 'opencv-python>=3.4.3.18' 'ruamel.yaml==0.15.82'

### 3 ARM64 平台（Python3.5）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。

#### 3.1 环境准备

- 一台安装有 Debian 9.8 操作系统的 RK3399Pro。
- 编辑 RK3399Pro 固件里的 parameter.txt 文件，修改 root 分区的大小，将 0x00700000@0x0007a000(rootfs),-@0x0077a000(userdata:grow) 改成 0x01300000@0x0007a000(rootfs),-@0x0137a000(userdata:grow)。之所以做这一步是因为现有 root 分区空间太小，后面安装依赖包会失败。
- 烧写固件到 RK3399Pro 上，参考《RK3399PRO\_LINUX\_SDK\_BETA\_V0.02\_20190321 发布说明.pdf》。
- 替换 NPU 固件，将/usr/share/npu\_fw 里的 boot.img 用 rknn-toolkit-1.0.3b1\RK3399Pro\firmware 目录下的 boot.img 替换掉。
- 将 rknn-toolkit-1.0.3b1\RK3399Pro\3rdparty 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 &
```

#### 3.2 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
```

更新完还需要修改/usr/bin/pip3，否则安装其他依赖包时会报 pip3 的错误。修改/usr/bin/pip3，如下：

```
from pip import main    -->    from pip install __main__
...
    sys.exit(main())    -->    sys.exit(__main__.__main__())
```

### 3. 安装 Python 打包工具

```
pip3 install wheel setuptools
```

### 4. 安装依赖包 h5py

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

### 5. 安装 requires.txt 文件中的依赖包，放在 rknn-toolkit-1.0.3b1/packages/requirement-packages 目录下：

```
pip3 install -r requires.txt
```

### 6. 安装 TensorFlow，相应的 whl 包在 rknn-toolkit-1.0.3b1/packages/requirement-packages 目录下

```
pip3 install tensorflow-1.11.0-cp35-none-linux_aarch64.whl
```

### 7. 安装 opencv-python，相应的 whl 包在 rknn-toolkit-1.0.3b1/packages/requirement-packages 目录下

```
pip3 install \
opencv_python_headless-4.0.1.23-cp35-cp35m-linux_aarch64.whl
```

### 8. 安装 RKNN-Toolkit，相应的 whl 包在 rknn-toolkit-1.0.3b1/packages/目录下

```
pip3 install rknn_toolkit-1.0.3b1-cp35-cp35m-linux_aarch64.whl --user
```

## 3.3 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 example/mobilenet\_v1 目录

```
linaro@linaro-alip:~/rknn-toolkit-1.0.3b1/ $ cd example/mobilenet_v1
```

2. 执行 test.py 脚本

```
linaro@linaro-alip: ~/rknn-toolkit-1.0.3b1/example/mobilenet_v1$ python3 test.py
```

3. 脚本执行完后得到如下结果：

```
--> config model
done
--> Loading model
done
--> Building model
done
--> Export RKNN model
done
--> Init runtime environment
done
--> Running model
mobilenet_v1
-----TOP 5-----
[156]: 0.8837890625
[155]: 0.0677490234375
[188 205]: 0.00867462158203125
[188 205]: 0.00867462158203125
[263]: 0.0057525634765625

done
--> Begin evaluate model performance
=====
                                Performance
=====
Total Time(us): 5790
FPS: 172.71
=====

done
```



example 目录中的 mobilenet\_v2 和 mobilenet-ssd 示例的执行方式与 mobilenet\_v1 相同，只是 mobilenet-ssd 示例中要执行的脚本是 `ssd.py`，且该示例执行完后会输出一张 `out.jpg` 图像，它会把检测到的对象在图像中用方框框出来。

注：有关 RKNN-Toolkit 更详细的用法和接口说明，请参考《RKNN-Toolkit 使用指南\_V1.0.0.pdf》手册。