Security Class: Top-Secret ( ) Secret ( ) Internal ( ) Public (√)

RKNN SDK Quick Start

(Technology Department, Graphic Computing Platform Center)

Mark: [ ] Editing [√] Released	Version	V1.7.5
	Author	Rao Hong
	Completed Date	2023-07-31
	Auditor	Vincent
	Reviewed Date	2023-07-31

Revision History

Version no.	Author	Revision Date	Revision description	Auditor
V0.9.9	Rao Hong	2019-03-25	Initial version release	Vincent
V1.0.0	Rao Hong	2019-05-08	Synchronize the modification contents of RKNN-Toolkit-V1.0.0	Vincent
V1.1.0	Rao Hong	2019-06-28	1. Synchronize the modification contents of RKNN-Toolkit-V1.1.0 2. Rename document, from <RKNN-Toolkit Quick Setup Guide> to <RKNN-Toolkit Quick Start> 3. Add quick start for Windows/Mac OS X/ARM64 platform.	Vincent
V1.2.0	Rao Hong	2019-08-21	Synchronize the modification contents of RKNN-Toolkit-V1.2.0	Vincent
V1.2.1	Rao Hong	2019-09-26	Synchronize the modification contents of RKNN-Toolkit-V1.2.1	Vincent
V1.3.0	Rao Hong	2019-12-23	Synchronize the modification contents of RKNN-Toolkit-V1.3.0	Vincent
V1.3.2	Rao Hong	2020-04-03	Synchronize the modification contents of RKNN-Toolkit-V1.3.2	Vincent
V1.4.0	Rao Hong	2020-08-13	Synchronize the modification contents of RKNN-Toolkit-V1.4.0	Vincent
V1.6.0	Rao Hong	2020-12-31	Synchronize the modification contents of RKNN-Toolkit-V1.6.0	Vincent
V1.6.1	Rao Hong	2021-05-21	Synchronize the modification contents of RKNN-Toolkit-V1.6.1	Vincent
v1.7.0	Xing Zheng	2021-08-10	Synchronize the modification contents of RKNN-Toolkit-V1.7.0	Vincent
v1.7.1	Rao Hong	2021-11-20	Synchronize the modification contents of RKNN-Toolkit-V1.7.1	Vincent
v1.7.3	Rao Hong	2022-08-07	1. Ubuntu platform update system version (18.04) and version of Python used (3.6). 2. Add development environment setup.	Vincent

Version no.	Author	Revision Date	Revision description	Auditor
V1.7.5	Rao Hong	2023-07-31	1. Update the description of requirements and example usage. 2. Add quick start guide for RKNN model deployment. 3. Replace main features introduction with overview. 4. Document renamed to RKNN SDK Quick Start.	Vincent

Content

1 Overview

This document is aimed at zero-based users, and introduces in detail how to quickly use RKNN Toolkit to complete model conversion, and deploy the model to Rockchip EVB development board for inference through the C API provided by RKNPU.

Applicable chip: RK1806 / RK1808 / RK3399Pro / RV1109 / RV1126.

RKNN Toolkit project download address: https://github.com/rockchip-linux/rknn-toolkit

RKNPU project download address: https://github.com/rockchip-linux/rknpu

2 Development environment setup

This chapter introduces the preparation of the development environment before using RKNN Toolkit. For common problems of development environment setup, please refer to chapter 3 of the document *Rockchip_Trouble_Shooting_RKNN_Toolkit_EN.pdf*.

2.1 Reference system configuration

The system configuration recommended for RKNN Toolkit is as follows:

Table 2-1 Recommend system configuration

Hardware/operating system	Recommend
CPU	Intel Core i3 or above or equivalent Xeon / AMD processor
RAM	16G or above
operating system	Linux: Ubuntu 16.04 64bit or Ubuntu 18.04 64bit MacOS: 10.13.5 Windows: 7 or 10

Note: It is recommended to use the PC with the recommended configuration to complete the model conversion stage. Model conversion can also be completed when using specified firmware on RK3399Pro, RK1808 computing card and other devices, but because of the limited CPU and memory resources of these devices, it is not recommended to use them for model conversion.

2.2 Requirements/Dependencies

It is recommended to meet the following requirements in the operating system environment:

Table 2-2 Run time environments

Operation System	Ubuntu18.04（x64）or later Windows 7（x64）or later Mac OS X 10.13.5（x64） or later Debian 9.8（aarch64）or later
Python	3.5 / 3.6 / 3.7 / 3.8
Python Requirements	'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'

Note:

1. Windows only support Python 3.6 currently; MacOS support Python 3.6 and Python 3.7; Arm64 platform support Python 3.5(Debian 9.8) and python 3.7(Debian 10).
2. Linux x86 platform removes support for python3.5 and adds support for Python3.8 since RKNN Toolkit Version 1.7.3.
3. Since some dependencies cannot be installed or used normally in the Python3.8 environment, in the Linux x86_64 / Python3.8 environment, upgrade the TensorFlow version to 2.2.0.
4. When using pip to install RKNN Toolkit, tensorflow/torch/mxnet will not be installed by default. Please

install the corresponding version according to actual needs. The version in the above table is the recommended version.

5. The application on ARM64 platform does not need to require flask, sklearn, requests.
6. Jinja2 is only used when creating custom op.

2.3 Connect to Rockchip NPU device

During model evaluation or deployment, Rockchip NPU equipment needs to be connected. This section will give the connection method of RK3399Pro, RK1808, RV1109, RV1126 development board, and how to confirm whether the device is successfully connected.

Note:

If you are using the Toybrick development board, please refer to the boot and serial port debugging content in the following link:

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 Connect to the device through the USB-OTG port

The connection between RKNN Toolkit and Rockchip NPU development board is created base on the USB-OTG interface. In the stage of model evaluation or model deployment, it is necessary to connect the PC and Rockchip NPU device for the related function.

The position of the RK3399Pro USB-OTG interface is shown in the red circle as shown in the figure below, and it is connected to the PC through the USB Type-C cable:

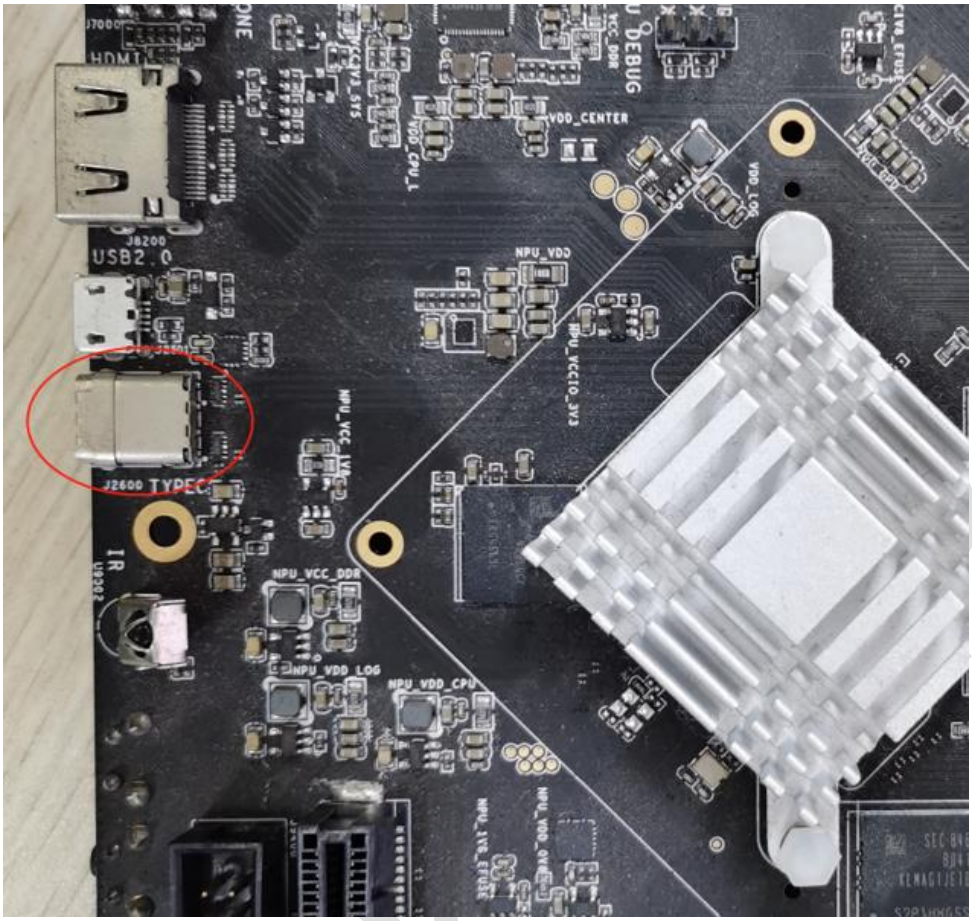

Figure 2-1 RK3399Pro USB-OTG port

The location of the RK1808 USB-OTG interface is shown in the figure below, and it is connected to the PC via a Micro-USB cable:

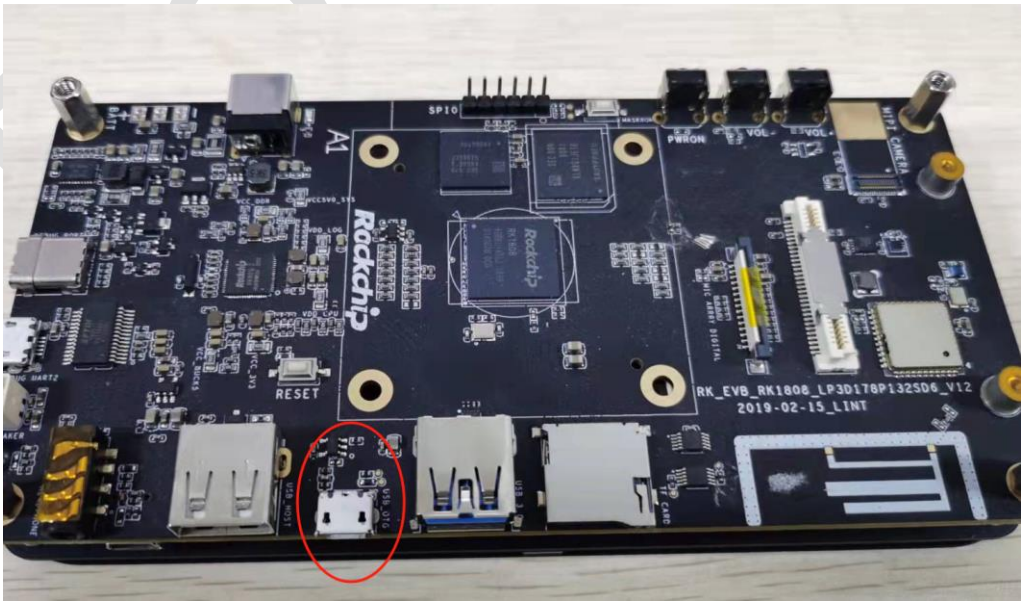

Figure 2-2 RK1808 USB-OTG port

The position of the USB-OTG interface of RV1109/RV1126 is shown in the red circle as shown in the figure below, and it is connected to the PC through the Micro-USB cable:

Figure 2-3 RV1109/RV1126 USB-OTG port

2.3.2 Connect the device through the DEBUG port

In the model evaluation or model deployment step, if you encounter an error on the development board, connection to the NPU device through the DEBUG port can help to grab the error log.

The location of the RK3399Pro NPU DEBUG interface is shown in the figure below. It is connected to the PC through the USB-serial conversion board. The PC needs to install Minicom or Putty etc. software:

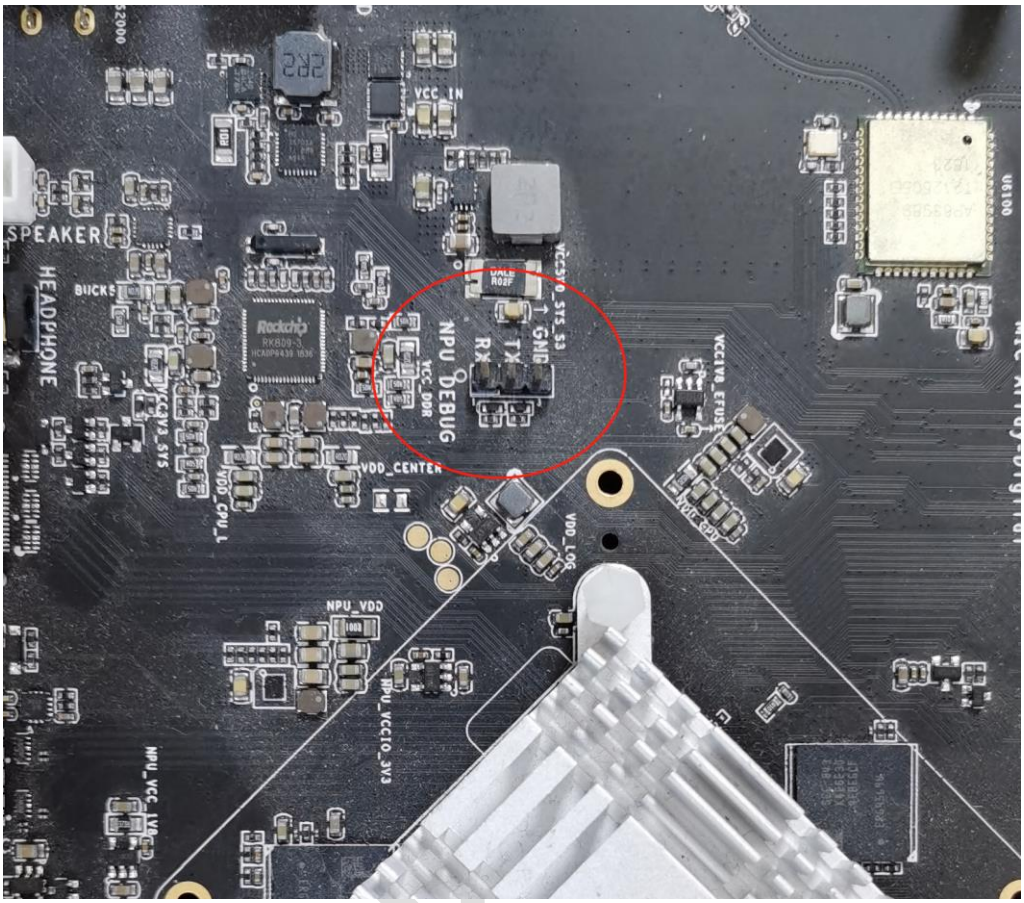

Figure 2-4 RK3399Pro DEBUG port

The position of the RK1808 DEBUG interface is shown in the red circle in the figure below, and it is connected to the PC through the Micro-USB cable:

Figure 2-5 RK1808 DEBUG port

The position of the RV1109/RV1126 DEBUG interface is shown in the red circle in the figure below, and it is connected to the PC through the Micro-USB cable:

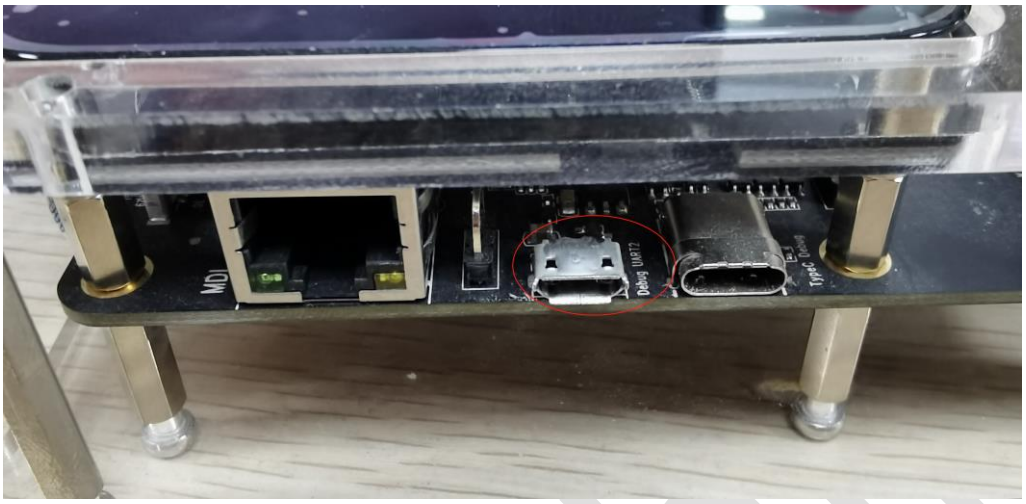

Figure 2-6 RV1109/RV1126 DEBUG port

2.3.3 Confirm that the device is already connected

If the device is connected correctly, execute the "python -m rknn.bin.list_devices" command on the PC's terminal to list the IDs of the connected devices. The reference output is as follows:

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

It's shown that the PC is currently connected to two development boards, the device id of the first development board is cf8f01ae745b49ce, and the device id of the second development board is 1126.

Note:

1. If you are using an x86_64 Linux system, you may need to update the USB device permissions, for the first time to connect to a Rockchip NPU devices, otherwise the system may not have read and write permissions on the devices. The update way is to execute the update_rk1808_usb_rule.sh script in the platform-tools/update_rk_usb_rule/linux/ directory, and restart the PC system on after execution.

2. If you are using Rockchip NPU device on Windows, you need to turn on the NTB communication

mode first (RK1808 and Toybrick development boards are turned on by default). For tuning on NTB mode, enter the development board system through adb and modify the file /etc/init.d/.usb_config, Add one line in this file: usb_ntb_en, and then restart the development board. The original line usb_adb_en should be retained, otherwise the development board system cannot be accessed through adb.

2.3.4 Connection FAQs

For common problems of device connection, please refer to chapter 1.5 of the document *Rockchip_Trouble_Shooting_RKNN_Toolkit_EN.pdf*.

3 Ubuntu platform Quick Start Guide

This chapter mainly describes how to quickly setup and use RKNN-Toolkit based on Ubuntu 18.04, Python3.6.

3.1 Environment Preparation

- One x86_64 bit computer with ubuntu18.04
- One RK1808 or RV1126 EVB board.
- Connect RK1808 (or RV1126) device to PC through OTG USB, use 'adb devices' command to check, and the result is as below:

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

The content marked in red is the device ID, the first is the RK1808 development board, and the second is the RV1126 development board.

3.2 Install RKNN-Toolkit

1. Install Python3.6

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

2. Install pip3

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

3. Obtain RKNN-Toolkit install package, and then execute below steps:

   a) Enter package directory:

```
cd package/
```

   b) Install Python dependency

```
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) Install RKNN-Toolkit

```
sudo pip3 install rknn_toolkit-1.7.x-cp35-cp35m-linux_x86_64.whl
```

d) Check if RKNN-Toolkit is installed successfully or not

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

The installation is successful if the import of RKNN module doesn't fail.

3.3 Execute the example attached in the install package

3.3.1 Example running on RV1126

Here take mobilenet_v1 as example. Please refer to below steps to run this demo:

1. Enter examples/lite/mobilenet_v1 directory

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

2. Execute test.py script, show top5 of classification result and performance data:

```
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
============================================================
```

The main process of this example include: create RKNN object, model configuration, load TensorFlow Lite model, structure RKNN model, export RKNN model, load pictures and infer to get TOP5 result, evaluate model performance, release RKNN object.

Other demos in the examples directory are executed the same way as mobilenet_v1. These models are mainly used for classification, target detection and image segmentation.

Note: If the PC is connected to multiple RKNPU devices, you can specify the device model and ID to distinguish them when executing the "test.py" script. The example is as follows:

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

3.3.2 Example running on RK1808

The mobilnet_v1 example in the SDK runs on the RV1126, if you want to run this example on RK1808 EVB board, you can refer to the following steps：

1. Enter examples/lite/mobilenet_v1 directory

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

2. Execute test script and specify target, such as:

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

3. Execute test.py script, show top5 of classification result and performance data:

```
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
============================================================
```

4 Windows platform Quick Start Guide

This chapter introduces how to use RKNN-Toolkit on Windows platforms with python 3.6.

4.1 Environmental preparations

- One pc with Windows 7 (64bit) or Windows 10 (64bit).
- One RK1808 or RV1126 EVB board.
- Connect RK1808 (or RV1126) EVB board to PC through USB(OTG). If this is first time to use RK1808 (or RV1126) Compute Stick, we need install driver first. Installation method is as follows:
  - Open SDK package, and enter directory: platform-tools/drivers_installer/windows-x86_64, run the zadig-2.4.exe program as an administrator to install the computing stick driver:

    1. Confirm the equipment and the driver to be installed:

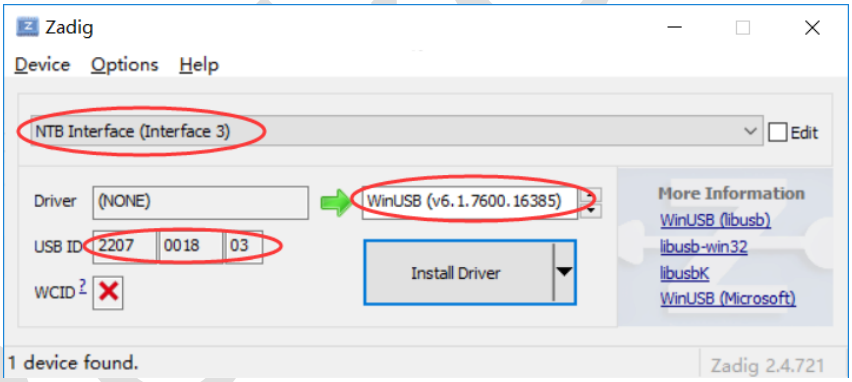

Figure 4-1 Driver installation interface

Note: The USB ID should start with **2207**; the driver choose default: WinUSB.

2. Click Install Driver.

3. If the installation is successful, the following interface will appear:

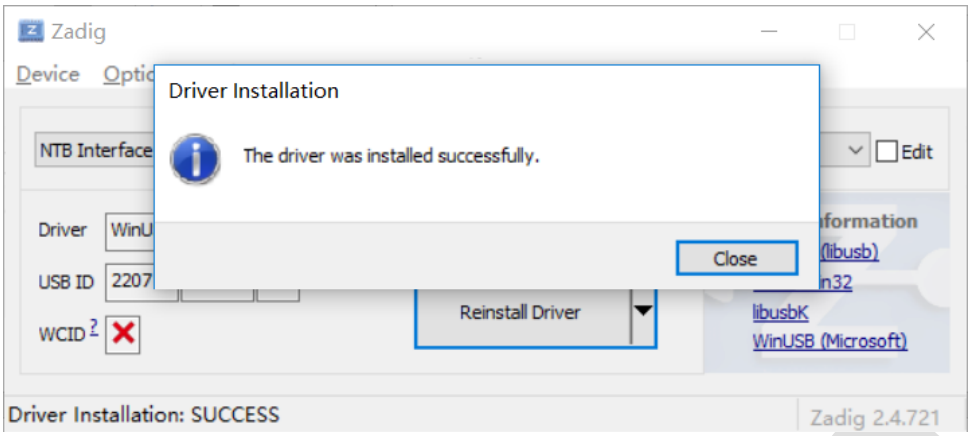

Figure 4-2 Example of successful driver installation

- After installation, if the NTB Interface in the Windows Device Manager does not have an exclamation point, and as shown below, the installation is successful.

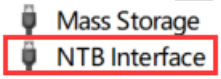

Figure 4-3 Device status when the driver is installed successfully

Note: Please reboot compute after installing driver.

4.2 Install RKNN-Toolkit

Before install RKNN-Toolkit, make sure python3.6 has been installed. This can be determined by executing python –version in cmd, as explained below. Python 3.6 is already installed on the system.

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

Get RKNN-Toolkit SDK package, then perform the following steps:

1. Enter directory: rknn-toolkit-v1.7.x/packages

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

2. Install Python dependency.

```
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
```

Note: demos in examples/mxnet require gluoncv.

3. Install RKNN-Toolkit.

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

4. Check if RKNN-Toolkit is installed successfully or not.

```
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
>>>
```

4.3 Example running on RV1126 or RK1808

When using RV1126 or RK1808 to run the example on the Windows platform, the running steps are the same as the Ubuntu platform, so it won't be repeated here.

5 Mac OS X platform Quick Start Guide

This chapter introduces how to use RKNN-Toolkit on Mac OS X platforms with python 3.6.

5.1 Environmental preparations

- One pc with MacOS 10.13.5 (or higher version).
- One RK1808 or RV1126 EVB board.
- Connect RK1808 (or RV1126) EVB board to PC through USB(OTG), execute program 'npu_transfer_proxy' in directory 'platform-tools/ntp/mac-osx-x86_64', check weather EVB board has connected. Result should looks like below:

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

Note: The red line is the RK1808 EVB board. Device id is "515e9b401c060c0b".

5.2 Install RKNN-Toolkit

Get RKNN-Toolkit SDK package, then perform the following steps:

1. Enter directory: rknn-toolkit-v1.7.x/packages

```
cd packages/
```

2. Install Python dependency.

```
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
```

Note: gluoncv are used in example.

3. Install RKNN-Toolkit.

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

4. Check if RKNN-Toolkit is installed successfully or not.

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

5.3 Running the sample attached in the installation package

When using RV1126 or RK1808 to run the example on the Mac OS platform, the running steps are the same as the Ubuntu platform, so it won't be repeated here.

6 ARM64 platform Quick Start Guide

This chapter introduces how to use RKNN-Toolkit on ARM64 platforms (Debian 9.8 systems) with python3.5.

Note: The model conversion process needs to use more CPU or memory resources. It`s recommended to complete this part of the operation on PC side, and then use the Python interface provided by RKNN Toolkit Lite or RKNPU C API to deploy the model on RK3399Pro board.

6.1 Environmental preparations

- An RK3399Pro with Debian 9.8 operating system. Make sure that the remaining space of the root partition is greater than 5GB.
- If can not find npu_transfer_proxy or npu_transfer_proxy.proxy in /usr/bin directory, we need copy the npu_transfer_proxy in rknn-toolkit-v1.7.x/platform-tools/ntp/linux_aarch64 directory to /usr/bin/ directory, and go to the directory and execute the following command (you have to start the program after each reboot, so please add it to boot script):

```
sudo ./npu_transfer_proxy &
```

6.2 Install RKNN-Toolkit

1. Execute the following command to update the system packages which will be used later when installing Python dependencies.

```
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. Execute the following command to update pip.

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

3. Install Python package tool.

```
pip3 install wheel setuptools
```

4. Install dependency package h5py.

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

5. Install TensorFlow. Since there is no corresponding source on the debian9 system, you need to search for the wheel package on the Internet and install it.
6. Install torch and torchvision. Since there is no corresponding source on the debian9 system, you need to search for the wheel package on the Internet and install it.
7. Install mxnet. Since there is no corresponding source on the debian9 system, you need to search for the wheel package on the Internet and install it.
8. Install opencv-python. Since there is no corresponding source on the debian9 system, you need to search for the wheel package on the Internet and install it.
9. Install RKNN-Toolkit and the corresponding wheel package is in the rknn-toolkit-v1.7.x/packages directory

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

Note: Since some libraries that RKNN-Toolkit relies on need compile and install on the ARM64 platform after downloading the source code, this step will take a long time.

6.3 Running the sample attached in the installation package

Take mobilenet_v1 as an example, which is a Tensorflow Lite model for image classification.

The running steps are as below:

1. Enter examples/lite/mobilenet_v1 directory

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

2. Run test.py script and specify target

```
linaro@linaro-alip: ~/mobilenet_v1$ python3 test.py rk3399pro
```

3. Show the TOP5 of classification and performance data after the script execution as below:

```
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
============================================
```

The main operations of this example include: create RKNN object, model configuration, load TensorFlow Lite model, structure RKNN model, export RKNN model, load pictures and infer to get TOP5 result, evaluate model performance, release RKNN object.

Other demos in the examples directory are executed the same way as mobilenet_v1. These models are mainly used for classification, target detection and image detection.

7 Model Deployment Quick Start Guide

This chapter takes rknn_mobilenet_demo running on the RV1126 Linux ARM32-bit platforms as an example to illustrate how to deploy the RKNN model to the Rockchip NPU platform through the C API provided by the RKNPU project.

For the RK3399Pro platform, please refer to the following project:

https://github.com/airockchip/RK3399Pro_npu

7.1 Download RKNPU project

RKNPU project link: https://github.com/rockchip-linux/rknpu

The download command reference is as follows:

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

7.2 Prepare the compilation tool

To compile executable programs for the ARM platform on the Linux x86 platform, you need to use the cross-compilation toolchain provided by ARM.

7.2.1 Download the cross-compiler tool

For the RV1109 / RV1126 platform, please download the cross compilation tool through the link below:

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

For the RK1808 platform, please download the cross-compilation tool through the link below:

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

For the RK1806 platform, please download the cross-compilation tool through the link below:

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 Set the compilation tool path

Decompress the compilation tool downloaded in the previous step, and record the absolute path of the folder.

Open the DEMO compilation script build.sh, and modify the GCC_COMPILER of the corresponding platform. For example, if the target platform is RV1126, modify the script as follows:

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

The `~/opt/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf` is the path decompressed in the previous step.

7.3 Update RKNN model

Refer to Section 3.3 to run the RKNN Toolkit example to generate the RKNN model. Copy the generated RKNN model to the rknpu/rknn/rknn_api/examples/rknn_mobilenet_demo/model directory, and rename the RKNN model according to the corresponding platform.

7.4 Build demo

1. Enter the rknn_mobilenet_demo folder in the terminal command line window:

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

2. Run the build.sh script compilation program

```
./build.sh
```

Note: If the compilation fails, it prompts that cmake is not installed, you can execute the following command to install cmake and then run the compilation script again.

```
sudo apt install cmake
```

7.5 Run demo on development board

Refer to Chapter 2.3.1 to connect the development board to the PC through the USB-OTG port.

1. Upload the compiled program (install/rknn_mobilenet_demo folder) to the /data/ directory of the board

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

2. Enter the development board system

```
adb shell
```

3. Enter the directory where the program is located

```
cd /data/rknn_mobilenet_demo
```

4. Run the program to identify the category of the test picture.

   Usage: ./rknn_mobilenet_demo <rknn_model> <image>

   - rknn_model: model path. Choose according to the specific platform.
   - image: the path of test picture.

   Reference command:

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

   Note: You can also directly execute the run_rk180x.sh or run_rv1109_rv1126.sh script provided by DEMO to run the example program.

5. The reference results are as follows:

```
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
```

Note:

- The index number with the highest score in the classification results is 156, and the labels corresponding to this index number are `Pekinese, Pekingese, Peke`.
- The printed Time is the time-consuming of the model inference interface “rknn_run”. The “FPS” is converted from the previous time-consuming.
- Using different platforms/versions of RKNN Toolkit and NPU drivers, there may be slight differences in inference results and time-consuming.

8 Appendix

8.1 Reference documents

OP support list:

RKNN_OP_Support.md

RKNN Toolkit manual:

Rockchip_User_Guide_RKNN_Toolkit_EN.pdf

Trouble shooting manual:

Rockchip_Trouble_Shooting_RKNN_Toolkit_EN.pdf

Custom OP define manual:

Rockchip_Developer_Guide_RKNN_Toolkit_Custom_OP_EN.pdf

Visualization function manual:

Rockchip_User_Guide_RKNN_Toolkit_Visualization_EN.pdf

RKNN Toolkit Lite manual:

Rockchip_User_Guide_RKNN_Toolkit_Lite_EN.pdf

The above documents can be found in the SDK/doc directory. You can also visit the following link to view: https://github.com/rockchip-linux/rknn-toolkit/tree/master/doc

8.2 Issue feedback

All the issue can be feedback via the follow ways:

- QQ group: 1025468710
- Github link: https://github.com/rockchip-linux/rknn-toolkit/issues
- Rockchip Redmine: https://redmine.rock-chips.com/

Note: Redmine account can only be registered by an authorized sales. If your development board is from the third-party manufacturer, please find the corresponding manufacturer to give

feedback first.