密级状态：绝密(　　)　　秘密(　　)　　内部(　　)　　公开(√)

# RKNN Toolkit 可视化使用指南

（技术部，图形计算平台中心）

| | | |
|---|---|---|
| 文件状态：<br>[　] 正在修改<br>[√] 正式发布 | 当前版本： | V1.7.3 |
| | 作　　者： | 陈浩 |
| | 完成日期： | 2022-08-07 |
| | 审　　核： | 熊伟 |
| | 完成日期： | 2022-08-07 |



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 |
|---|---|---|---|---|
| v0.1.0 | 陈浩 | 2019-11-22 | 初始版本 | 熊伟 |
| v1.3.2 | 饶洪 | 2020-04-15 | 更新版本号 | 熊伟 |
| V1.4.0 | 饶洪 | 2020-08-13 | 调整文档结构 | 熊伟 |
| V1.6.0 | 饶洪 | 2020-12-31 | 增加 Keras 模型转换功能的使用方法 | 熊伟 |
| V1.6.1 | 饶洪 | 2021-05-21 | 更新版本号 | 熊伟 |
| V1.7.0 | 饶洪 | 2021-08-08 | 增加多通道 std 参数的使用说明 | 熊伟 |
| V1.7.1 | 饶洪 | 2021-11-17 | 更新参考资料章节 | 熊伟 |
| V1.7.3 | 饶洪 | 2022-08-07 | 更新版本号 | 熊伟 |

# 目 录

# 1 主要功能说明

可视化功能为 RKNN Toolkit 提供了图形化的人机交互界面。通过可视化功能，用户只需要填写表单并点击相应的功能按钮就可以完成模型转、模型推理、模型评估等功能，无需再编写脚本。这极大地简化了用户的使用流程，降低了用户的使用难度。当前可视化界面提供以下功能：

1） 模型转换：支持将 TensorFlow、TensorFlow Lite、MXNet、ONNX、Darknet、Pytorch、Caffe、Keras 等框架的模型转成 RKNN 模型。

2） 量化功能：在模型转换时，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、dynamic_fixed_point-16）以及混合量化。

3） 模型推理：能够在 PC（Linux x86_64）上模拟 NPU 运行模型并获取推理结果；也可以在指定目标设备（RK1806、RK1808、RK3399Pro、RV1109、RV1126）上运行模型并获取推理结果。

4） 性能评估：能够在 PC（Linux x86_64）上模拟 NPU 运行模型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目标设备上运行模型，并获取在设备上运行时的性能数据（总耗时及每一层的耗时）。

5） 内存评估：获取模型运行时的内存使用情况。通过联机调试的方式，在指定目标设备上运行 RKNN 模型，并获取模型在目标设备上的内存使用信息。

6） 模型预编译：在模型转换阶段，通过预编译技术，可以使生成的 RKNN 模型在模型加载时减少耗时，部分模型还可以压缩 RKNN 模型的尺寸。通过预编译技术生成的 RKNN 模型只能在带有 NPU 的硬件平台上运行，且该功能目前只有 x86_64 Ubuntu 平台支持。

# 2 系统依赖说明

目前只有 Ubuntu、Windows、MacOS 平台上的 RKNN Toolkit 提供可视化界面。相应的系统依赖环境请参考《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》的**系统依赖说明**章节。

# 3 可视化程序的安装和启动

## 3.1 安装

Ubuntu、Windows、MacOS 平台上的 RKNN Toolkit 自带可视化程序，只需要安装 RKNN Toolkit 即可。RKNN Toolkit 安装方法请参考《Rockchip_Quick_Start_RKNN_Toolkit_CN.pdf》。

## 3.2 启动方法

RKNN Toolkit 安装完成后，在终端输入以下命令打开可视化界面：

```
python3 -m rknn.bin.visualization
```

如果要开启更多可视化窗口，只需在打开一个窗口后，在新的终端上执行上述命令即可。

# 4 使用方法

## 4.1 可视化程序首页

打开可视化程序后，首页如下图所示：

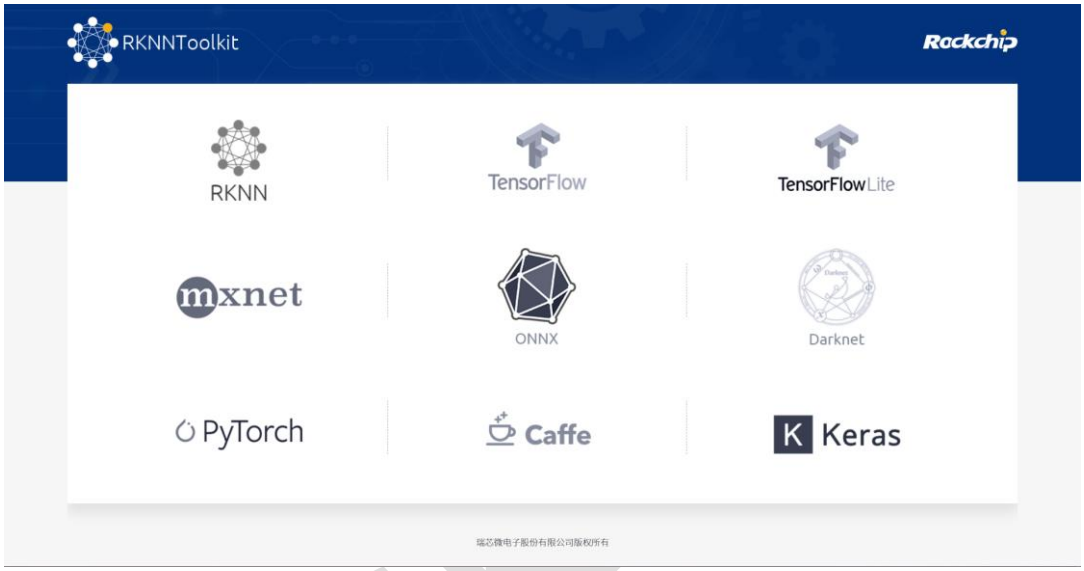


图 4-1 RKNN Toolkit 可视化界面首页

在首页排列着不同的深度学习框架和 RKNN 模型的图标。点击各深度学习框架的图标，将进入相应的模型转换界面。点击 RKNN 模型图标，将进入模型推理、性能评估、内存评估的界面。

## 4.2 模型转换

RKNN Toolkit 支持将 TensorFlow、TensorFlow Lite、MXNet、ONNX、Darknet、PyTorch、Caffe 和 Keras 等深度学习框架的模型转换成 RKNN 模型。点击首页中相应框架的图标，就可以进入该框架模型的转换界面。下面各小节将通过图文的方式展示如何通过可视化程序完成各框架模型的转换。

### 4.2.1 TensorFlow

点击 TensorFlow 图标进入 TensorFlow 模型转换页面。在转成 RKNN 模型前首先需要进行参数

配置，下面将给出参数配置页面每个选项的详细说明。

参数配置分两个页面，第一个页面是模型的通用配置，各个框架的模型在转换时都需要填写这部分内容；第二个页面则是每个框架的自有配置。

模型通用配置页面的配置项和说明如下：

- **目标芯片平台：**

  指定模型可以运行的芯片平台，[RK1806, RK1808, RK3399PRO]或[RV1109, RV1126]。

- **输入的通道均值：**

  通道均值和缩放系数在对模型输入进行预处理时使用。假设一个有三通道的输入，它的均值和缩放系数为(M0 M1 M2 S0 S1 S2)，各通道值用（IN0，IN1，IN2）表示，则预处理时将按以下公式进行计算 OUT0 =（IN0 – M0）/ S0; OUT1 = (IN1 – M1) / S1; OUT2 = (IN2 – M2) / S2。（OUT0, OUT1, OUT2）为预处理后的输出。

  该参数指定模型输入预处理时每个通道的均值。各通道之间用空格隔开，如果有多个输入，则各输入之间用“#”分隔。例如：127.5 127.5 127.5#127.5，表示该模型有两个输入，第一个输入有三个通道，均值均为 127.5；第二个输入只有一个通道，均值为 127.5。

- **输入缩放系数：**

  该参数指定模型输入预处理时每个输入的缩放系数。如果有多个输入，则每个输入的缩放系数用“#”分隔。例如：127.5 127.5 127.5#127.5，表该模型有两个输入，第一个输入在预处理时三个通道缩放系数均为 127.5；第二个输入在预处理时的缩放系数为 127.5，它有一个通道。

- **输入的通道顺序调整:**

  表示是否需要对图像通道顺序进行调整，只对 3 通道有效。’0 1 2’表示按照输入的通道顺序来推理，比如输入时是 RGB，那么推理的时候就按照 RGB 顺序；’2 1 0’表示会对输入做通道转换，比如输入时是 RGB，推理时会将其转成 BGR，反之亦然。如果有多个输入，用“#”分隔。

- **数据集:**

  量化时校正数据的数据集。目前支持文本文件格式，用户可以把用于校正的图片（jpg 或

png 格式）或 npy 文件路径放到一个.txt 文件中。文本文件里每一行表示一条路径信息。

- **Batch Size:**

  量化时每一批数据的大小。

- **Epochs:**

  量化时的迭代次数。每迭代一次，就选择 Batch Size 指定数量的图片进行量化校正。若为 -1 则会根据数据集总数和 Batch Size 自动计算。

- **量化类型:**

  如果量化类型选择 None，则不量化，使用浮点运算，并且不能使用混合量化功能。其他量化类型的详细介绍请参考《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》。

- **是否是 inception 系列模型:**

  如果模型是 inception v1/v3/v4，开启该选项可以提高性能。

- **是否打开预编译:**

  如果打开预编译，可以减少模型在硬件设备上的首次加载时间。但是打开这个开关后，转换出来的模型只能在硬件平台上使用。

- **RKNN 模型保存路径:**

  转换好的 RKNN 模型的存放位置。

- **RKNN 模型文件名:**

  转换得到的 RKNN 模型以该名字保存，文件后缀为 rknn。

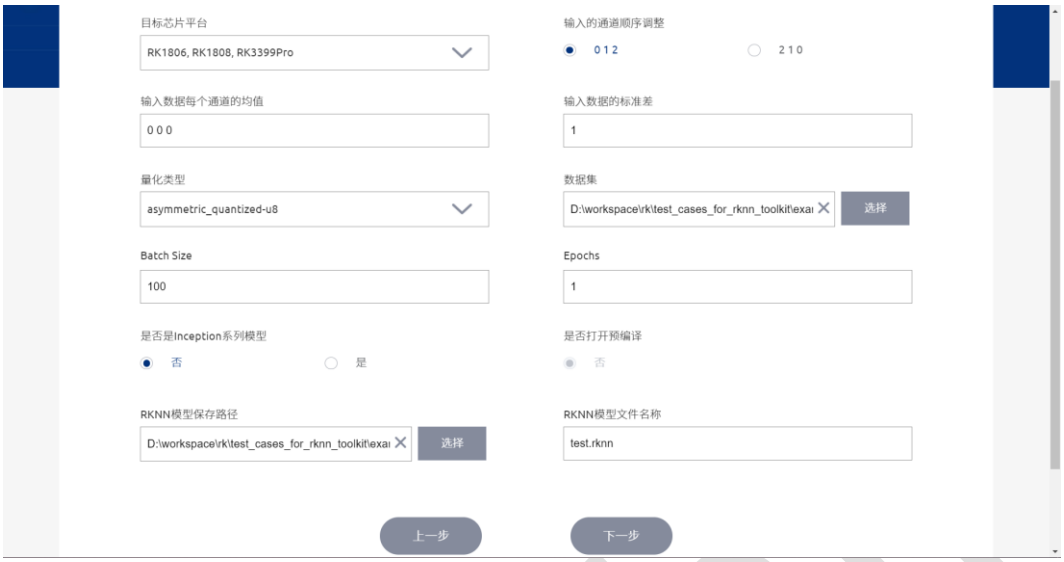


图 4-2 模型通用配置

填写完通用配置后，点击下一步进入 TensorFlow 模型配置页面。点击上一步返回首页。TensorFlow 配置页面每个选项的详细说明如下：

- **Model：**

  Pb 模型所在路径。

- **预定义文件：**

  为了支持一些控制逻辑，需要提供一个 npz 格式的预定义文件。可为空。

- **输入节点：**

  指定模型的输入节点。如果有多个输入，点击输入框右侧的加号按钮，添加新的输入节点。

- **输入维度列表：**

  每个输入节点对应的图片的尺寸和通道数，用逗号隔开。例如 224, 224, 3。如果有多个输入，点击输入框右侧的加号按钮，添加新的输入节点的尺寸和通道数。该参数需要和输入节点一一对应。

- **输出节点：**

  模型的输出节点，支持多个输出节点。


图 4-3 TensorFlow 参数配置

所有参数都配置好后，点击下一步，开始加载模型，量化模型。

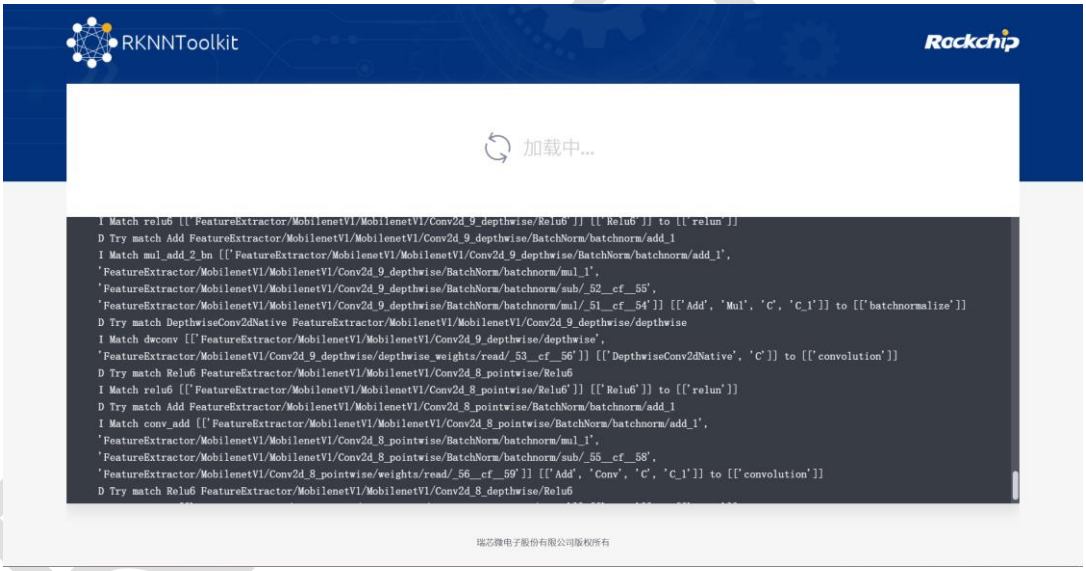


图 4-4 模型加载

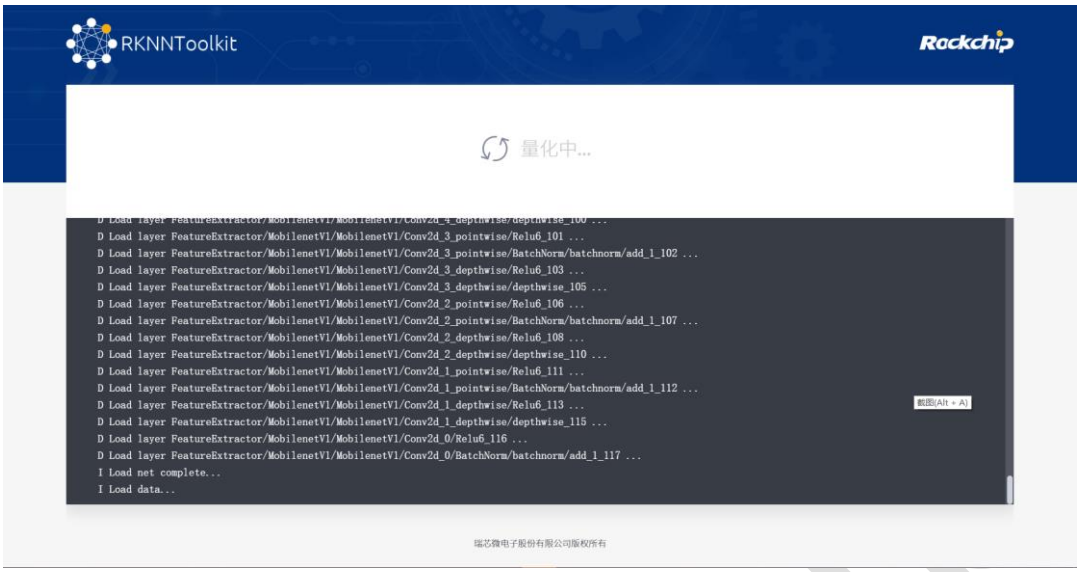


图 4-5 模型量化

加载模型，量化模型结束后，进入模型可视化界面。可视化页面展示了 TensorFlow 模型每一层的详细信息（包括层名和参数）。若当前窗口只显示模型部分信息，可拖拽或鼠标滚轮缩放图像来查看模型的其余部分。深蓝色为已量化的层，浅蓝色为未量化的层。

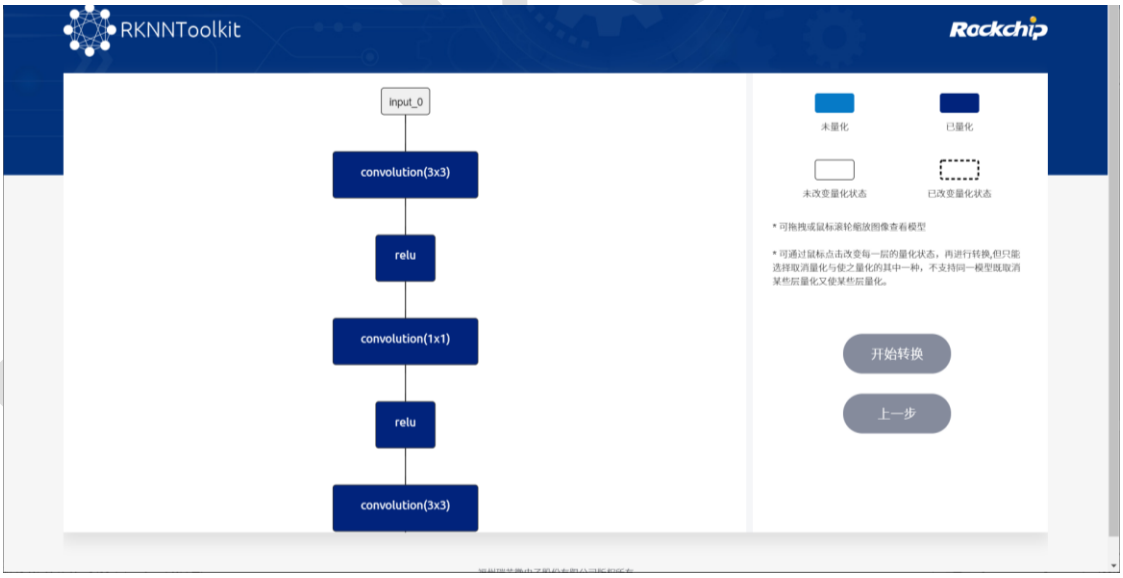


图 4-6 TensorFlow 模型可视化

可通过鼠标点击改变每一层的量化状态，比如将已量化的层改成未量化的层，或者将未量化的层改成已量化的层。若未改变原始的量化状态，点击开始转换，则直接导出 RKNN 模型；若改变了原始的量化状态，点击开始转换，则会先进行混合量化，再导出 RKNN 模型。

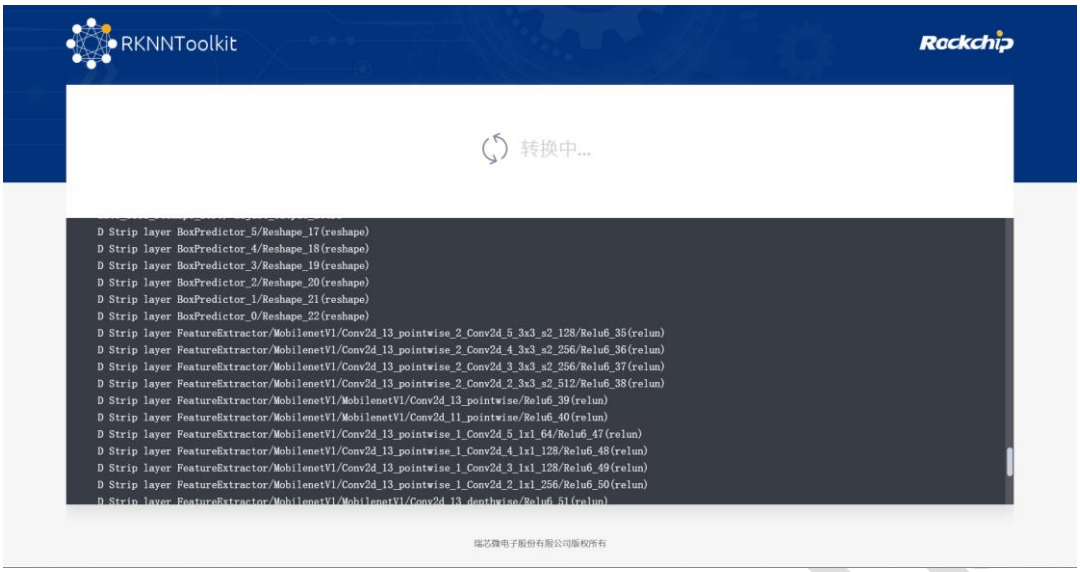


图 4-7 导出 RKNN 模型

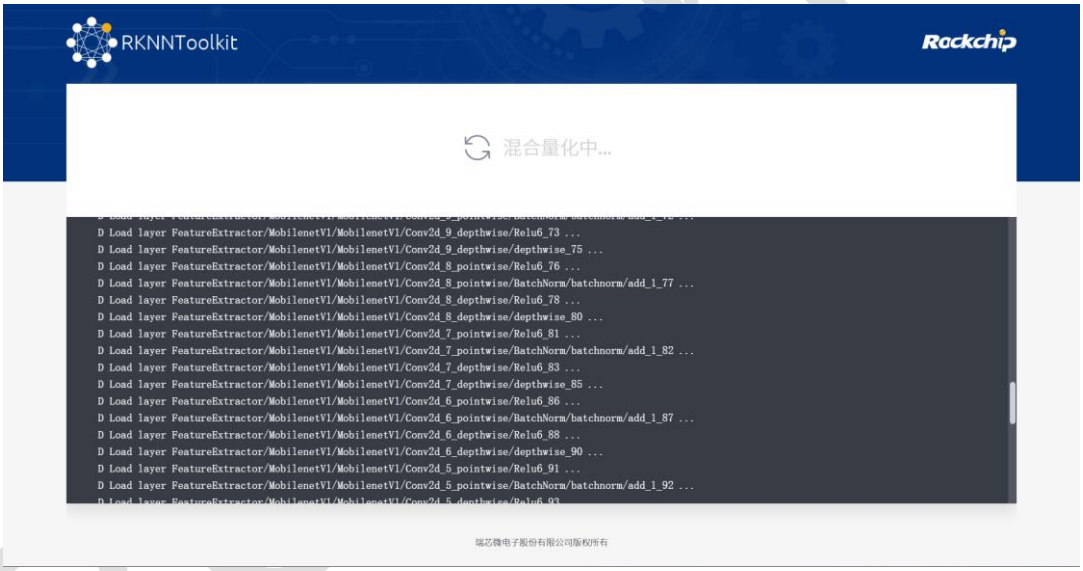


图 4-8 混合量化

### 4.2.2 TensorFlow Lite

点击 TensorFlow Lite 图标进入 TensorFlow Lite 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

TensorFlow Lite 框架自有配置页面的参数和详细说明如下：

- **Model：**

TensorFlow Lite 模型文件（.tflite 后缀）所在路径。

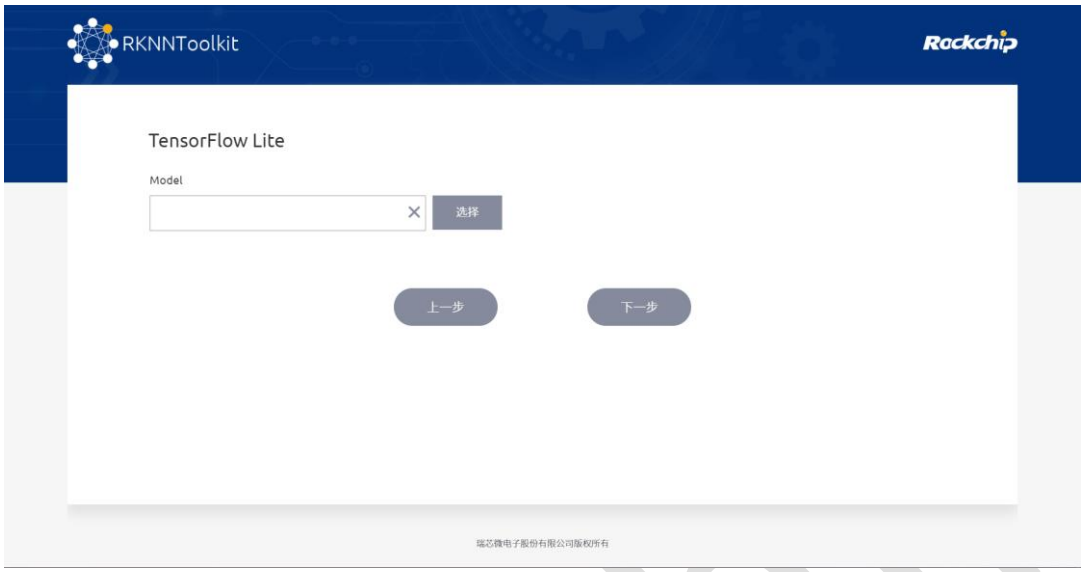


图 4-9 TensorFlow Lite 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.3 MXNet

点击 MXNet 图标进入 MXNet 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

MXNet 框架自有配置页面的参数和详细说明如下：

- **Symbol:**

  MXNet 模型文件（.json 后缀）所在路径。

- **Params:**

  MXNet 权重文件（. params 后缀）所在路径。

- **输入维度列表：**

  每个输入节点对应的图片的尺寸和通道数，用逗号隔开。例如 3, 224, 224。

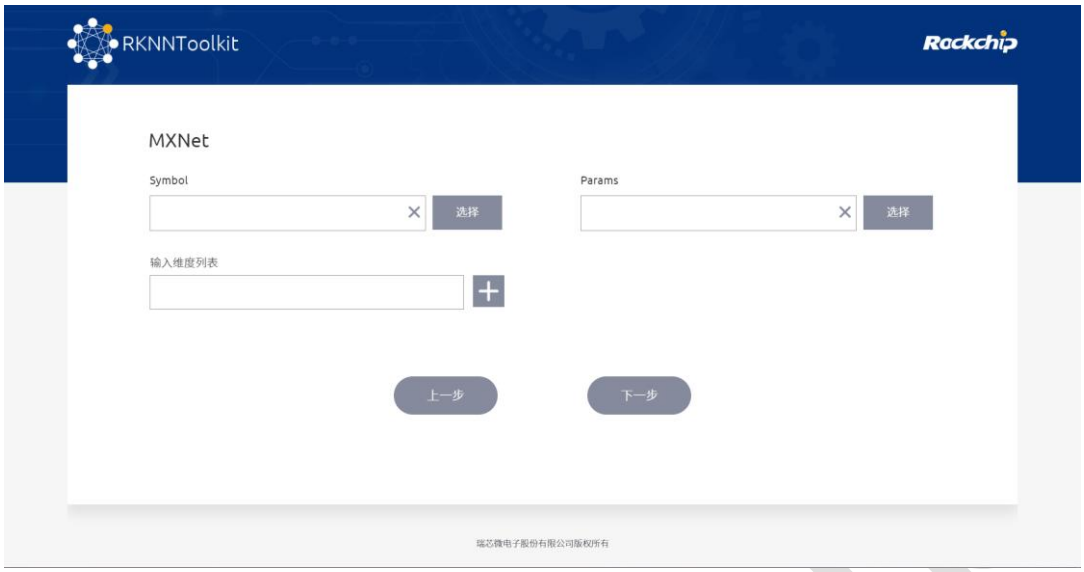


图 4-10 MXNet 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.4 ONNX

点击 ONNX 图标进入 ONNX 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

ONNX 框架自有配置页面的参数和详细说明如下：

- **Model:**

  ONNX 模型文件（.onnx 后缀）所在路径。

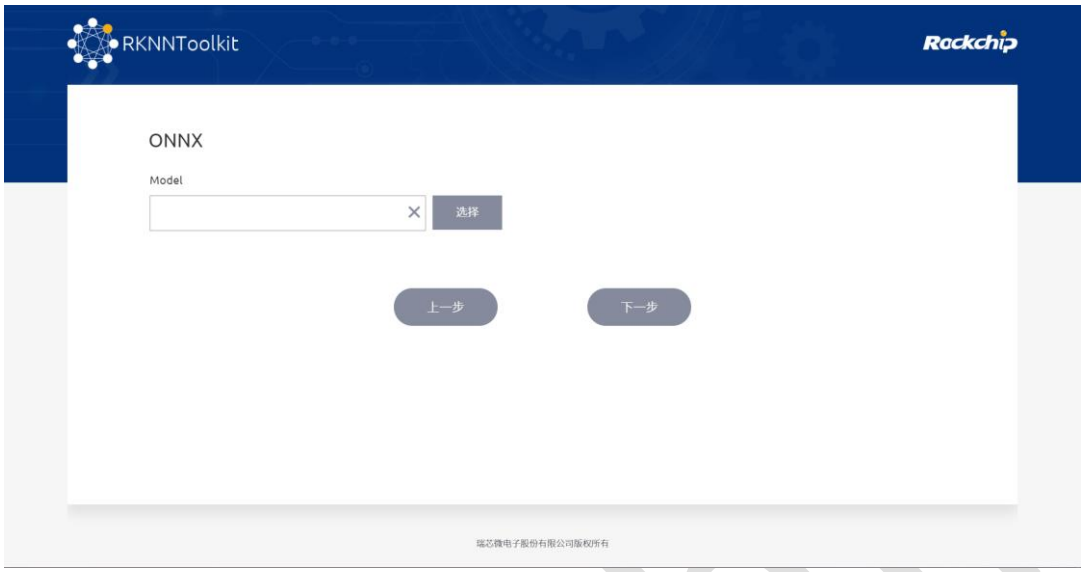


图 4-11 ONNX 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.5 Darknet

点击 Darknet 图标进入 Darknet 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

Darnet 框架自有配置页面的参数和详细说明如下：

- **Model:**

  Darknet 模型文件（.cfg 后缀）所在路径。

- **Weight:**

  权重文件（.weights 后缀）所在路径。

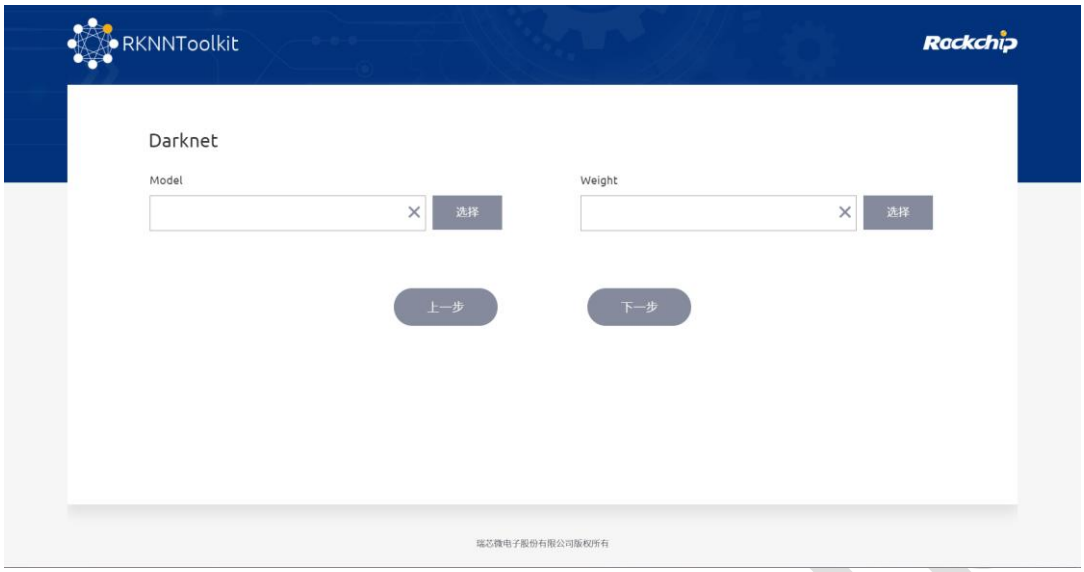


图 4-12 Darknet 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.6 PyTorch

点击 PyTorch 图标进入 PyTorch 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

PyTorch 框架自有配置页面的参数和详细说明如下：

- **Model:**

  PyTorch 模型文件（.pt 后缀）所在路径。以 pth 作为后缀的模型通常只包含权重，没有网络结构，转换前需调用相应函数（例如 torch.jit.trace），将 pth 模型转换成既有权重又有网络结构的 torchscript（.pt 后缀）模型。

- **输入维度列表:**

  每个输入节点对应的图片的尺寸和通道数，用逗号隔开。例如 3,224, 224。

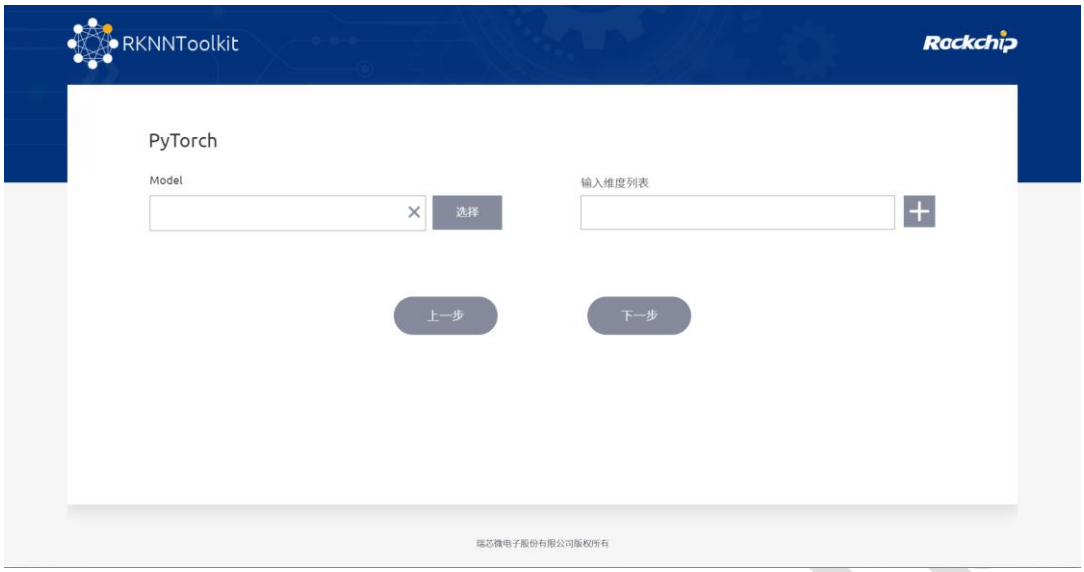


图 4-13 PyTorch 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.7 Caffe

点击 caffe 图标进入 caffe 功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

Caffe 框架自有配置页面的参数和详细说明如下：

- **Model:**

  caffe 模型文件（.prototxt 后缀文件）所在路径。

- **Proto:**

  caffe 模型的格式。

- **Blobs:**

  caffe 模型的二进制数据文件（.caffemodel 后缀文件）所在路径。

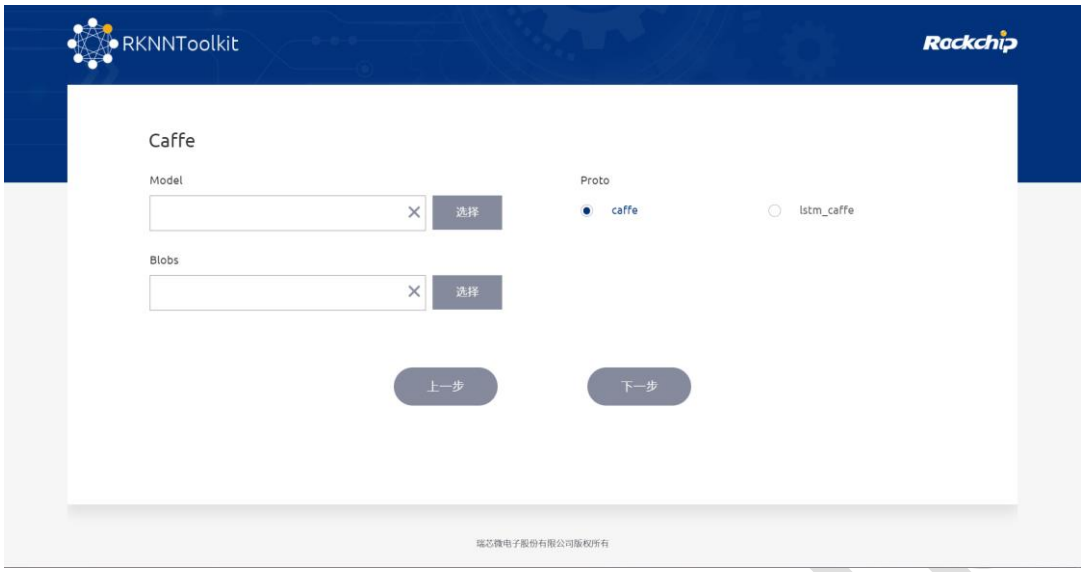


图 4-14 Caffe 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

### 4.2.8 Keras

点击 Keras 图标进入 Keras 模型转换功能页面，在转成 RKNN 模型前同样需要先进行参数配置。

模型通用配置请参考 **4.2.1 TensorFlow** 章节的详细说明。

Keras 框架自有配置页面的参数和详细说明如下：

- **Model:**

  Keras 模型文件（.h5 后缀）所在路径。

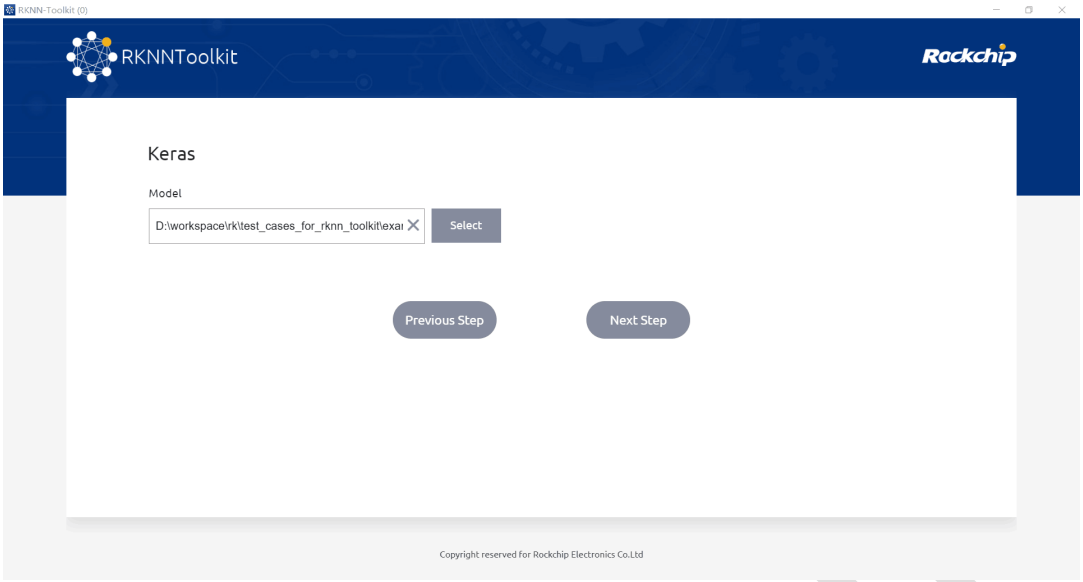


图 4-15 Keras 参数配置

模型加载、模型量化、混合量化、模型转换请参考 **4.2.1 TensorFlow** 章节的相关说明。

## 4.3 RKNN 模型使用

点击首页里的 RKNN 模型图标，将进入模型使用界面。该界面提供模型可视化、模型推理、性能评估、内存使用评估等功能。以下各小节通过图文形式介绍这些功能的使用方法。

### 4.3.1 模型可视化

在首页点击 RKNN 模型图标后，将出现一个 RKNN 模型选择的页面。在该页面选择我们接下来要使用的模型。模型选择页面如下图所示：

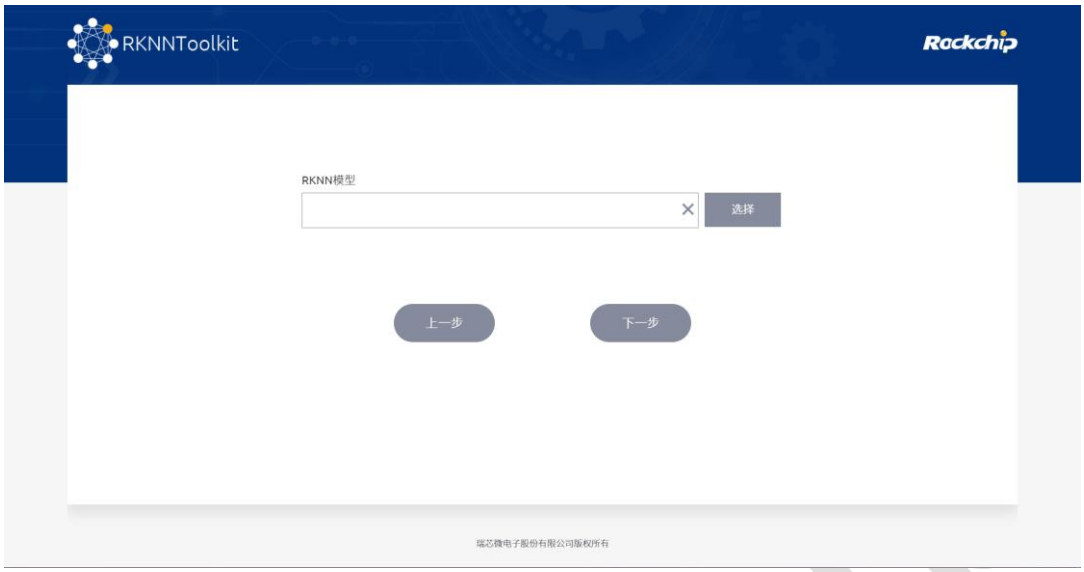


图 4-16 RKNN 模型选择

点击选择按钮进入 RKNN 模型选择界面，选择完以后，点击下一步按钮，进入模型可视化界面。该页面使用图形的形式展示模型。如果点击上一步按钮，则将返回首页。

可视化页面展示了 RKNN 模型每一层的详细信息（包括层名和参数）。若当前窗口只显示模型部分信息，可拖拽或鼠标滚轮缩放图像来查看模型的其余部分。深蓝色为已量化的层，浅蓝色为未量化的层。查看模型完毕后，可选择模型推理，性能评估或内存使用评估功能进入下一个页面。

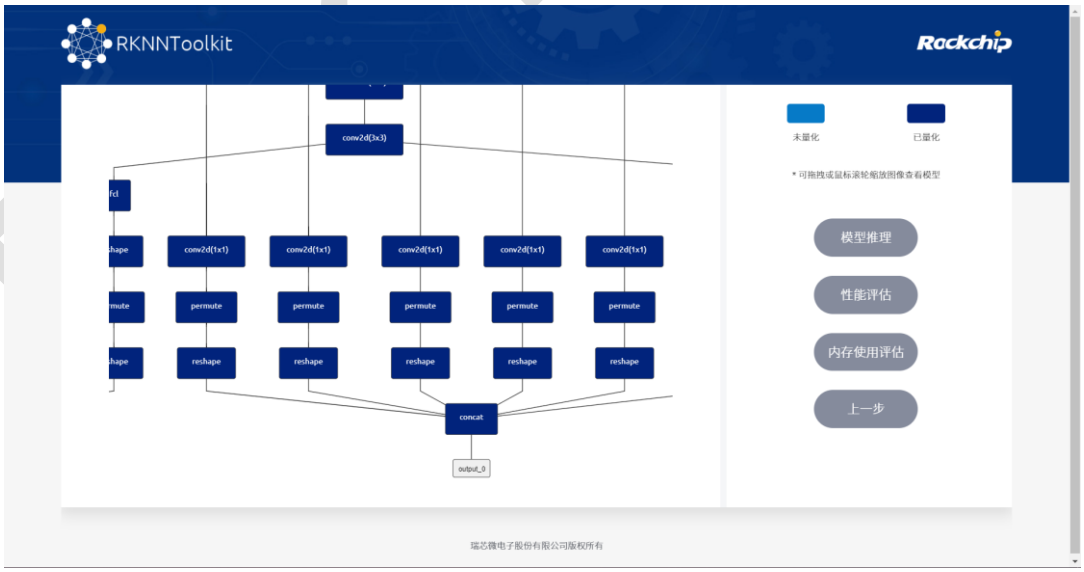


图 4-17 RKNN 模型可视化

### 4.3.2 模型使用

该页面提供模型推理、性能评估、内存使用评估功能，每一个参数的说明如下：

- **目标平台：**

  要运行 RKNN 模型的平台，当前支持的平台包括模拟器，RK1806、RK1808、RK3399PRO、RV1109、RV1126。其中模拟器只有 Ubuntu 平台上的工具支持。

- **设备 ID：**

  目标硬件设备的 ID 号，若查不到设备则为 none。当目标平台为模拟器时，该选项自动隐藏。

- **选择图片：**

  选择要评估的图片。若选择的图片尺寸小于模型输入尺寸，则会报错；若选择的图片尺寸大于模型输入尺寸，则会从图片左上方开始按照模型输入尺寸进行裁剪，再进行评估。

- **结果存储位置：**

  模型推理、性能评估、内存使用评估结果将会保存在该目录。模型推理结果会保存成 npy 文件，性能评估、内存使用评估结果会保存成 txt 文件。

- **是否获取每一层的性能详情：**

  如果设为是，则显示每一层的性能信息，否则只显示模型总的运行时间。如果目标平台是模拟器，该选项不生效。

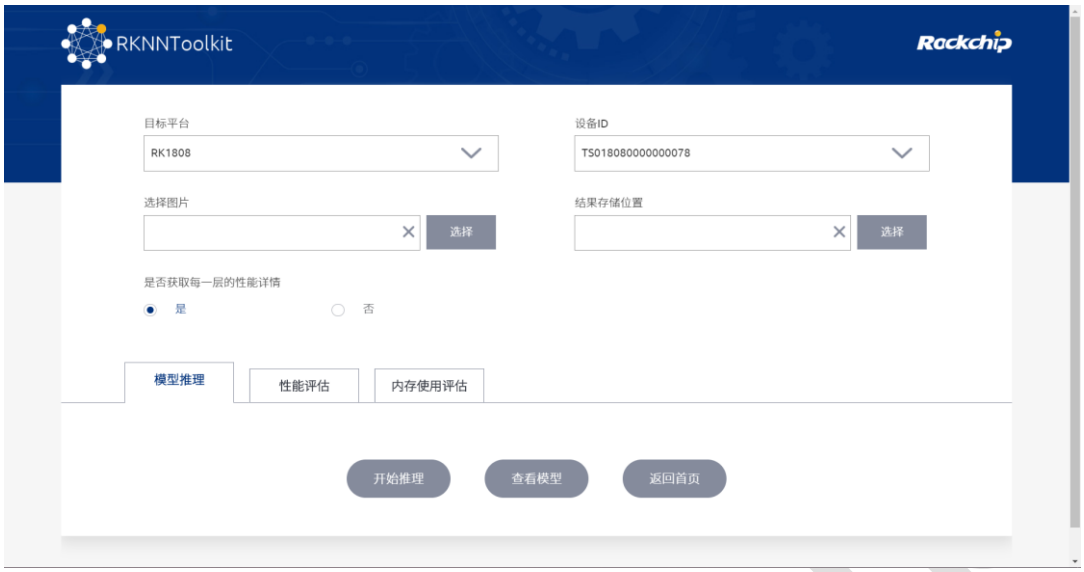


图 4-18 RKNN 模型评估页面

选择模型推理功能，模型会以该图片作为输入进行推理，并将结果保存成 npy 文件。


图 4-19 RKNN 模型推理

选择性能评估功能，将获取该模型的性能数据，并将性能评估结果保存成 txt 文件。


图 4-20 RKNN 性能评估

选择内存使用评估功能，将获取该模型的内存使用情况，并将结果保存成 txt 文件。


图 4-21 RKNN 内存使用评估

# 5 附录

## 5.1 参考文档

OP 支持列表：《RKNN_OP_Support.md》

快速上手指南：《Rockchip_Quick_Start_RKNN_Toolkit_CN.pdf》

RKNN Toolkit 使用指南：《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》

RKNN Toolkit Lite 使用指南：《Rockchip_User_Guide_RKNN_Toolkit_Lite_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》

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

## 5.2 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，请先找第三方厂方反馈问题。**