密级状态：绝密(　　)　　秘密(　　)　　内部(　)　　公开(√)

RKNN-Toolkit 可视化使用指南

（技术部，图形显示平台中心）

文件状态： [　] 正在修改 [√] 正式发布	当前版本：	V0.1.0
	作　　者：	陈浩
	完成日期：	2019-11-22
	审　　核：	卓鸿添
	完成日期：	2019-11-22

福州瑞芯微电子股份有限公司

Fuzhou　Rockchips　Semiconductor　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	卓鸿添

目 录

1 主要功能说明

该功能以图形界面的形式呈现 RKNN-Toolkit 的各项功能，简化用户操作步骤。用户可以通过填写表单、点击功能按钮的形式完成模型的转换和推理等功能，而不需要再去手动编写脚本。当前支持以下功能：

1） 模型转换：支持 TensorFlow、TensorFlow Lite、MXNet、ONNX、Darknet、Pytorch、Caffe、Keras 模型转成 RKNN 模型（Keras 暂不支持），支持 RKNN 模型导入导出，后续能够在硬件平台上加载使用。当前暂不支持多输入模型。

2） 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、dynamic_fixed_point-16）以及混合量化。

3） 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、TB-RK1808 AI 计算棒上运行模型并获取推理结果。

4） 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、TB-RK1808 AI 计算棒上运行模型，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。

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获取模型在指定硬件平台 RK3399Pro、RK1808、TB-RK1808 AI 计算棒或 RK3399Pro Linux 开发板上运行时的内存使用情况。

6） 模型预编译：通过预编译技术，可以减少模型加载的时间，对于部分模型，还可以减少模型尺寸。但是预编译后的 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

本工具仅支持运行于 Ubuntu、Windows、MacOS 操作系统。系统依赖环境请参考《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》的 **2 系统依赖说明**章节。

2.1 安装

安装方法请参考《Rockchip_User_Guide_RKNN_Toolkit_CN.pdf》的 **3.1 安装**章节。

2.2 启动方法

1) 在安装该 whl 包的环境下输入以下命令，即可启动一个窗口。

   python -m rknn.bin.visualization

2) 若要再打开一个新窗口，打开一个新终端，在新终端下输入 python -m rknn.bin.visualization (需等到第 1 个窗口初始化完成后才能打开第 2 个，第 3 个以上的窗口)。

2.3 使用方法

2.3.1 初始界面

启动可视化后，初始界面如图 1。具体功能如下：

TensorFlow、TensorFlow Lite、MXNet、ONNX、Darknet、Pytorch、Caffe、Keras 图标为将原始模型转换为 RKNN 模型，RKNN 模型后续能够在硬件平台上加载使用。（Keras 目前暂不支持）

RKNN 图标为 RKNN 模型评估，支持的功能包括模型可视化、模型推理、性能评估、内存使用评估。

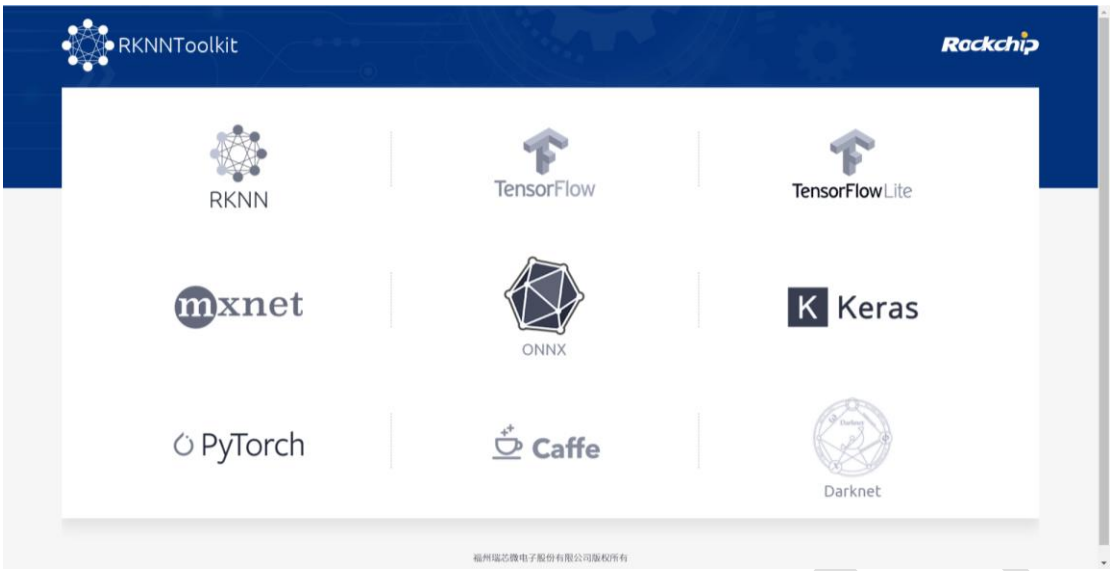

图 1 可视化初始界面

2.3.2 RKNN

RKNN 功能主要是对转换好的 RKNN 模型进行评估，支持的功能包括模型可视化、模型推理、性能评估、内存使用评估。

首先选择想要评估的 RKNN 模型，点击下一步进入 RKNN 模型可视化页面。

图 2 RKNN 模型选择

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。

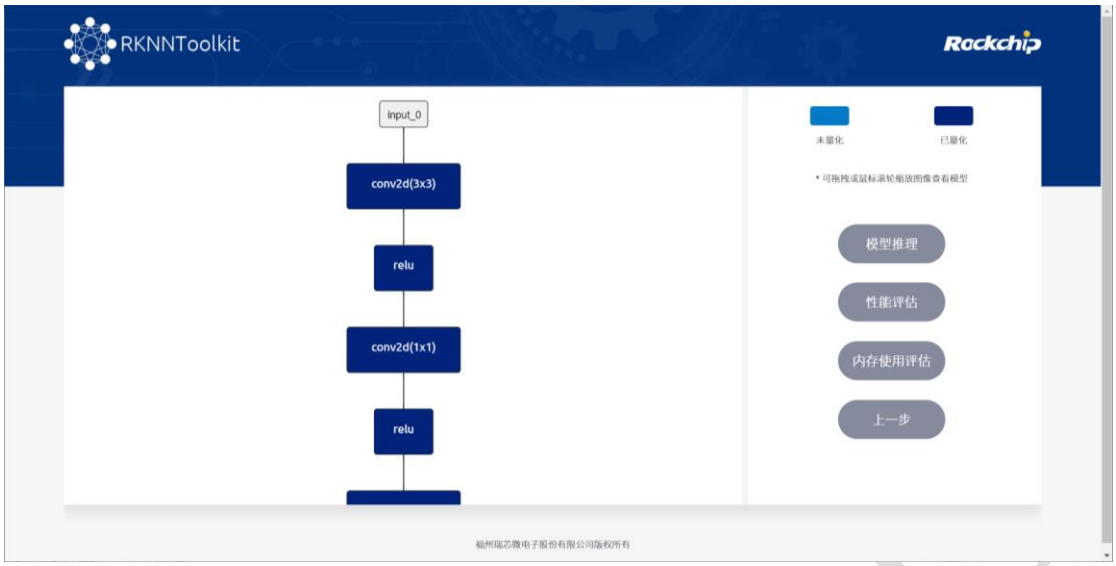

图 3 RKNN 模型可视化

该页面为模型推理、性能评估、内存使用评估功能，每一个选项的功能如下：

- **目标平台：**

  要执行评估功能的平台，支持的平台包括模拟器，RK1808 和 RK3399PRO。

- **设备 ID：**

  RK1808 和 RK3399PRO 的设备 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，则打印每一层的性能信息，否则只获取模型总的运行时间。如果目标平台是模拟器，该选项不生效。

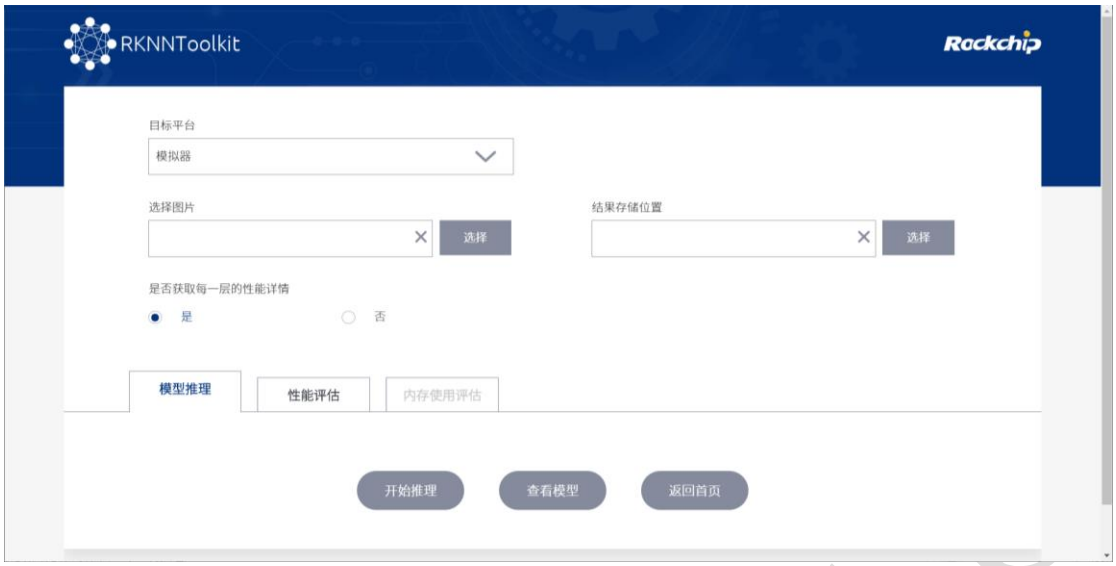

图 4 RKNN 模型评估页面

选择模型推理功能，模型会预测该图片，并将预测结果保存成 npy 文件。

图 5 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 6 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 7 RKNN 内存使用评估

2.3.3 TensorFlow

点击 TensorFlow 图标进入 TensorFlow 功能页面。在转成 RKNN 模型前首先需要进行参数配置，下面将介绍参数配置页面每个选项的具体功能。

参数配置 1 页面每个选项功能如下：

- **输入的通道均值和缩放系数：**

  包括四个值(M0 M1 M2 S0)，前三个值为均值(mean)，后面一个值为缩放系数(scale)。假

设输入为(Cin0, Cin1, Cin2)，输出为(Cout0,Cout1, Cout2)，则计算公式为：Cout0 = (Cin0 - M0)/S0，Cout1 = (Cin1 - M1)/S0，Cout2 = (Cin2 - M2)/S0。

- **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。

- **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，使用浮点数运算，并且不能使用混合量化功能。

- **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 8 TensorFlow 参数配置 1

填写完参数配置 1 后，点击下一步进入参数配置 2 页面。参数配置 2 页面每个选项功能如下：

- **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：**

  模型输入节点。

- **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。

- **均值：**

  输入的均值。只有当导入的模型是已量化过的模型时才需要设置该参数，且模型输入的三个通道均值都相同。建议上一步的量化类型设置为 none，否则模型会被重新量化。

- **缩放系数：**

  输入的缩放系数。只有当导入的模型是已量化过的模型时才需要设置该参数。建议上一步的量化类型设置为 none，否则模型会被重新量化。

- **输出节点：**

  模型的输出节点，支持多个输出节点。

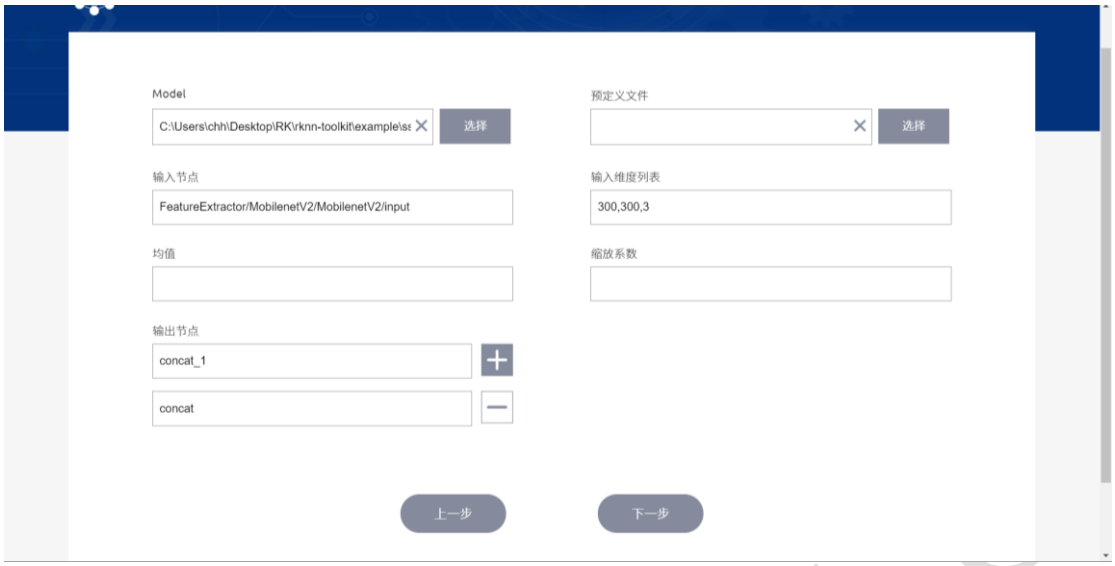

图 9 TensorFlow 参数配置 2

所有参数都配置好后，点击下一步，开始加载模型，量化模型。

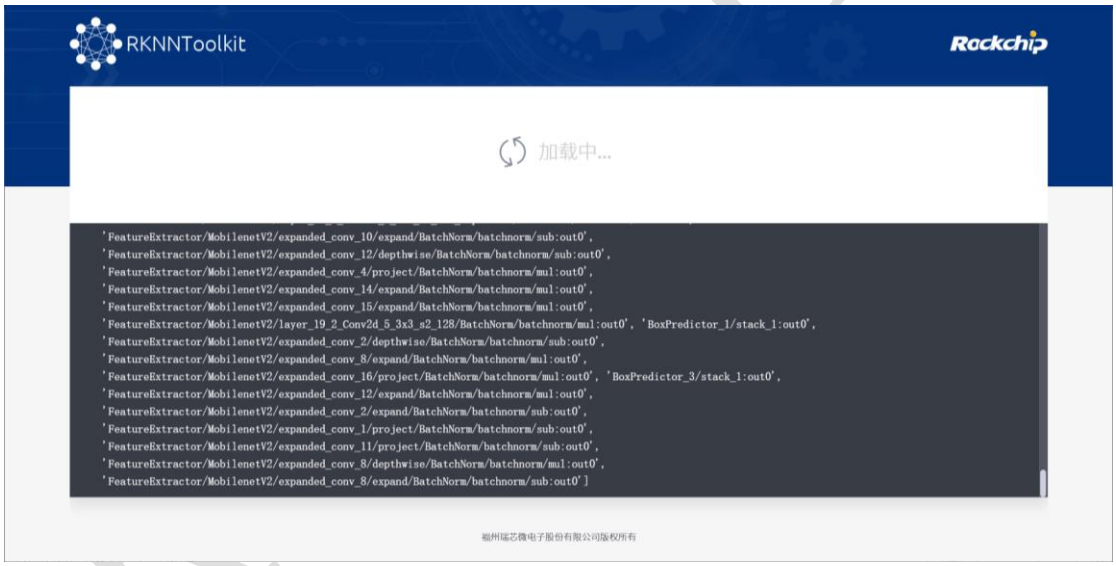

图 10 模型加载

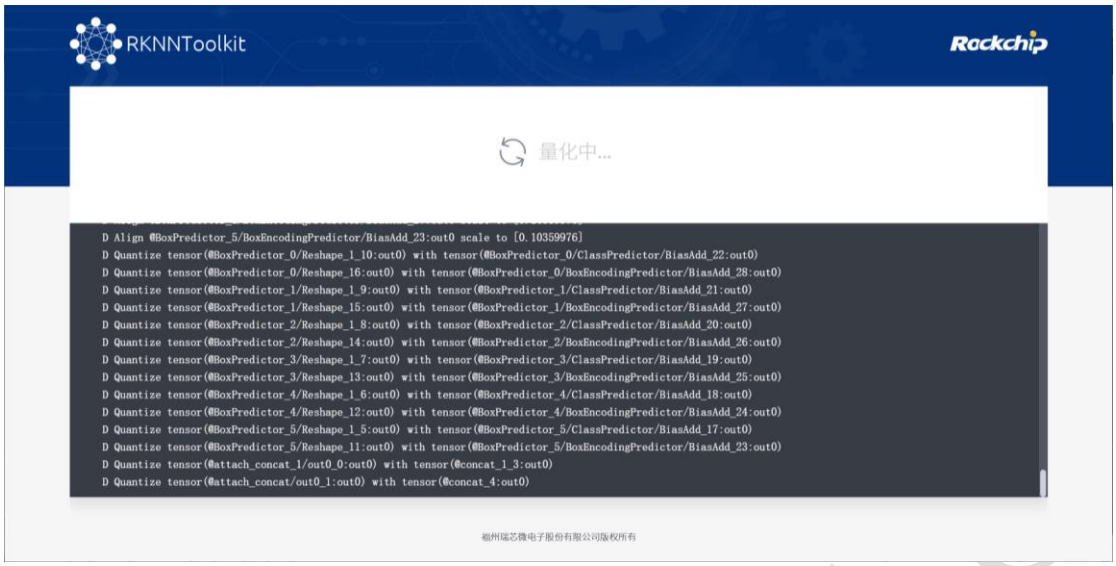

图 11 模型量化

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。

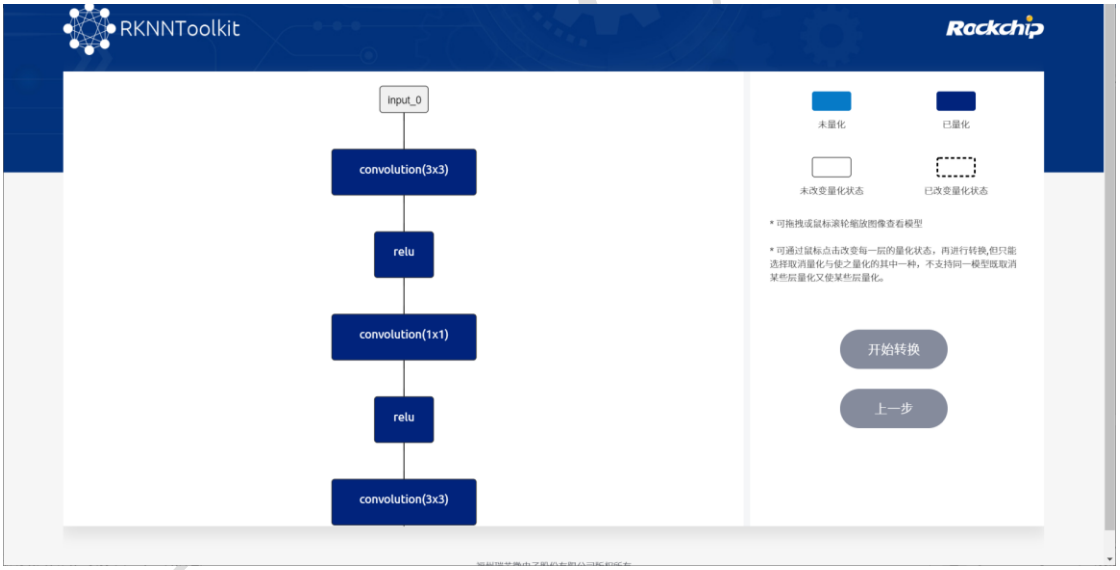

图 12 TensorFlow 模型可视化

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。

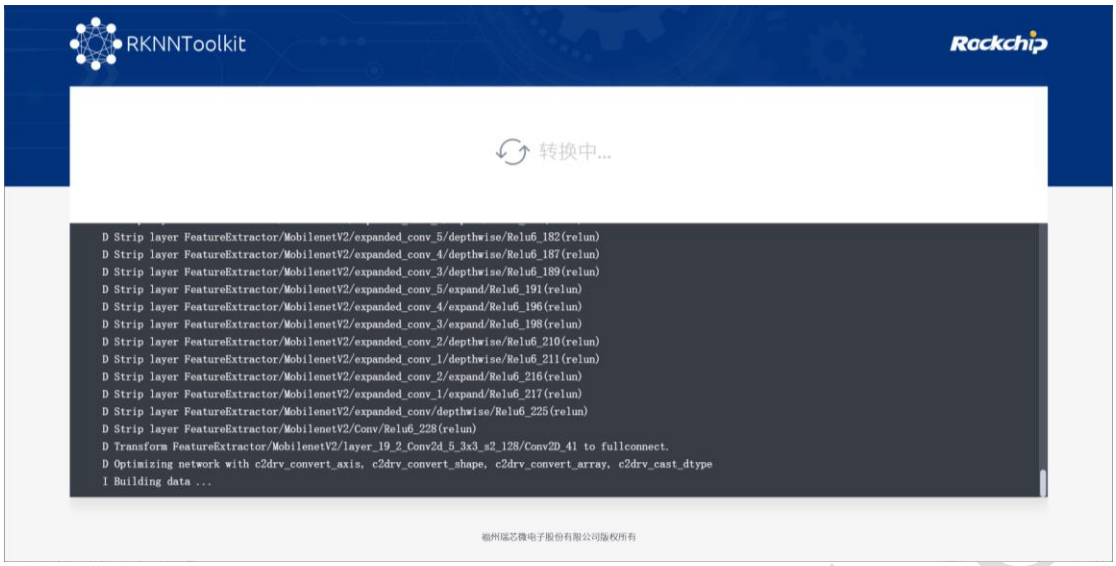

图 13 导出 RKNN 模型

图 14 混合量化

2.3.4 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 如下：

- **Model：**

TensorFlow Lite 模型文件（.tflite 后缀）所在路径。

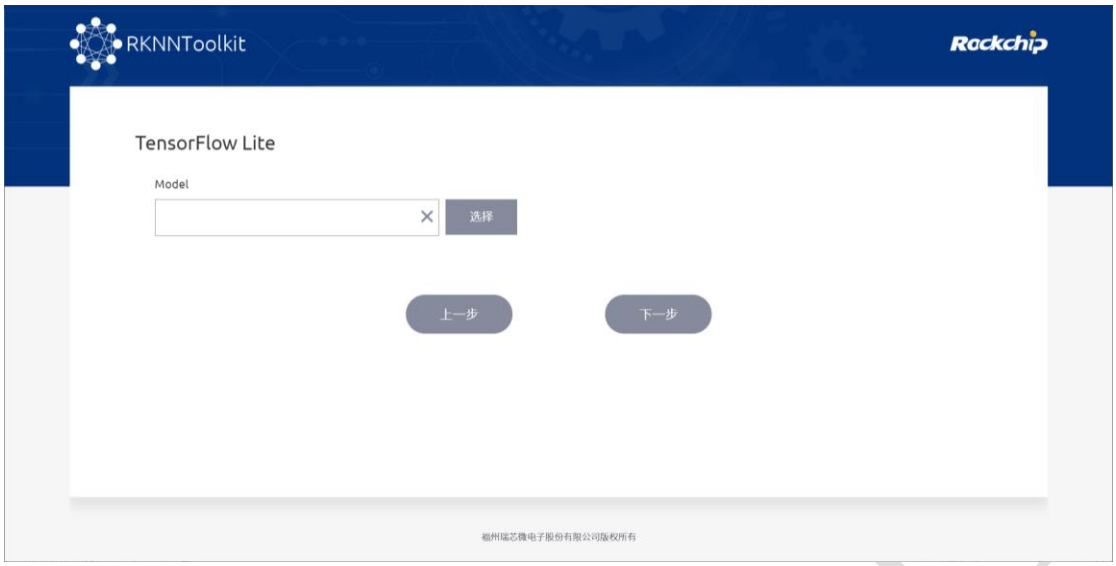

图 15 TensorFlow Lite 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。

2.3.5 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 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。

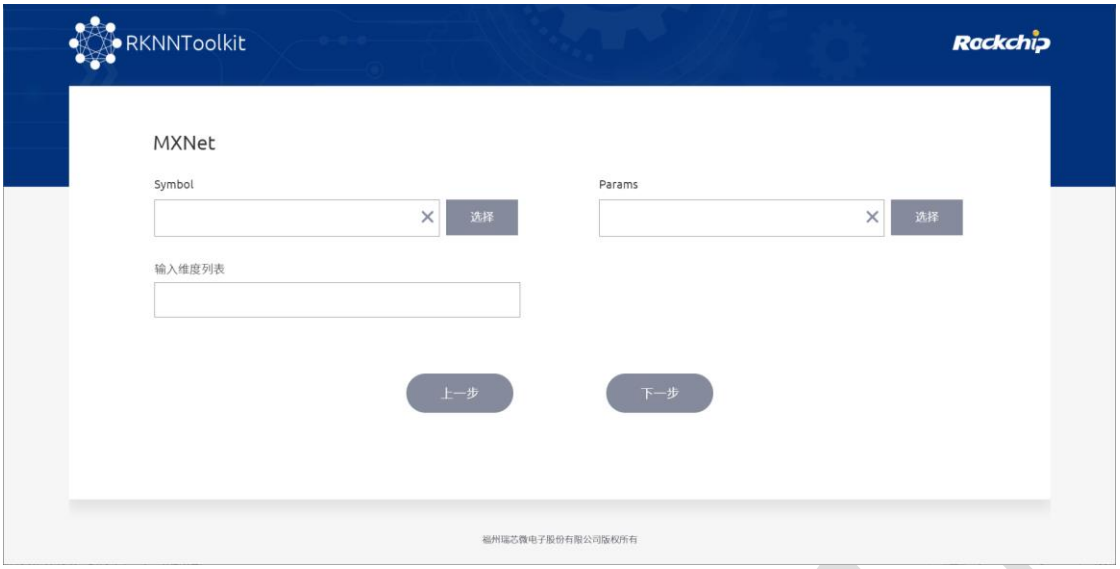

图 16 MXNet 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。

2.3.6 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 如下：

- **Model:**

  ONNX 模型文件（.onnx 后缀）所在路径。

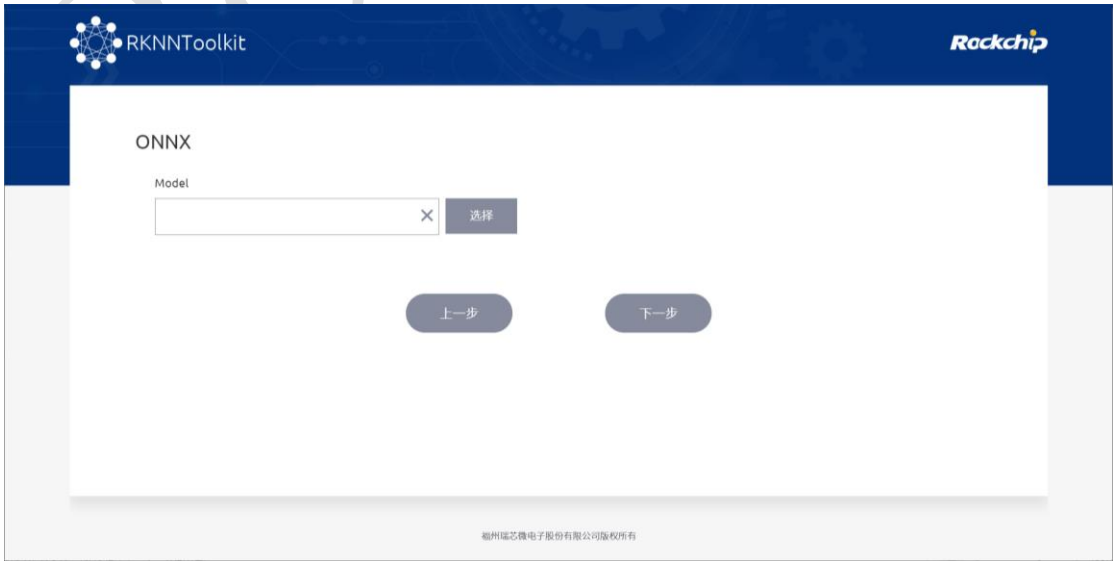

图 17 onnx 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。

2.3.7 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 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。

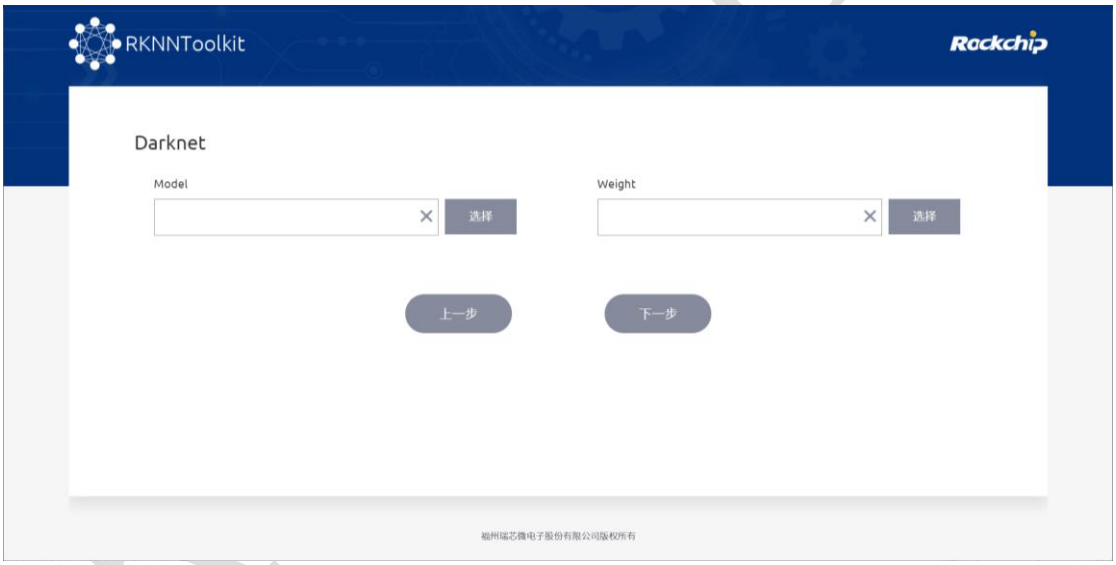

图 18 Darknet 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。

2.3.8 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 如下：

- **Model:**

  PyTorch 模型文件（.pt 后缀）所在路径。Pth 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。

- **输入节点:**

  选填，默认为模型输入节点。

- **输出节点:**

  选填，默认为模型输出节点。

图 19 PyTorch 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。

2.3.9 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。

参数配置 1 请参考 **2.3.3 TensorFlow 章节**的参数配置 1。

参数配置 2 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。

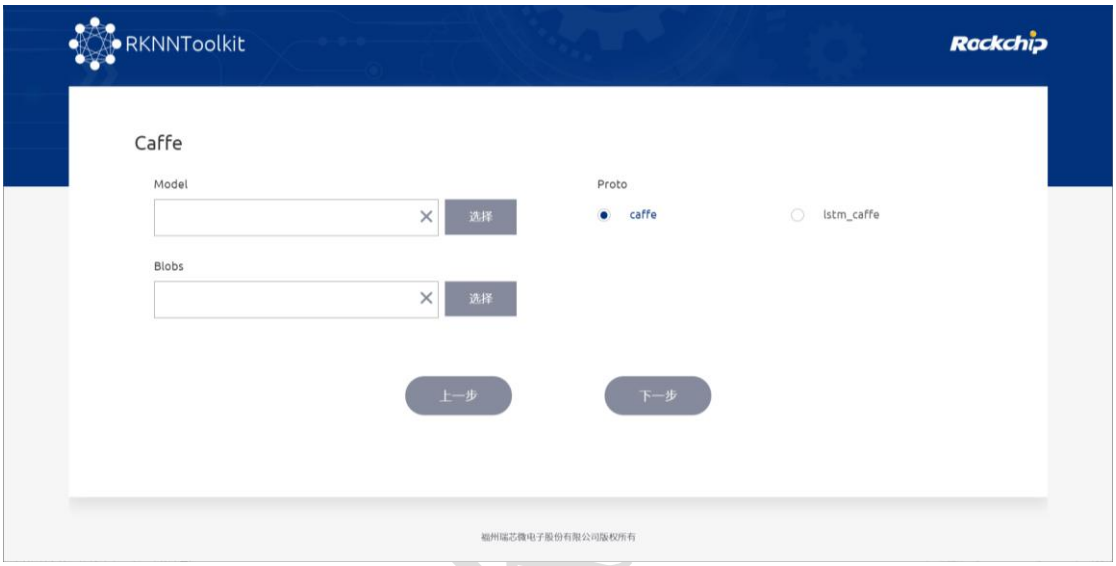

图 20 caffe 参数配置 2

模型加载、模型量化、混合量化、模型转换请参考 **2.3.3 TensorFlow 章节**。