AN14241

How to Integrate Customer ML Model to NPU on MCX N94x

Rev. 1 — 1 March 2024 Application note

Document information

Information	Content
Keywords	MCX N94x, NPU, machine learning.
Abstract	This document describes how to integrate the customer ML model to NPU on the FRDM-MCXN947 board.

1 Introduction

This document describes how to integrate the customer ML model to NPU on the FRDM-MCXN947 board. MCX N94x and MCX N54x are based on dual high-performance Arm Cortex-M33 cores running at up to 150 MHz, which has 2 MB of on-chip Flash with optional full ECC RAM, and an integrated proprietary Neural Processing Unit (NPU). The integrated NPU delivers up to 40x faster machine learning (ML) throughput compared to a CPU core alone, enabling it to spend less time awake and reducing overall power consumption.

Table 1. Requirements

Software requirements	Hardware requirements
MCUXpresso IDE eIQ Toolkit MCUXpresso SDK for FRDM-MCXN947	FRDM-MCXN947 board Debug prob: USB-C

2 NPU overview

The eIQ Neutron Neural Processing Unit (NPU) is a highly scalable accelerator core architecture providing machine learning (ML) acceleration. The architecture provides power and performance-optimized NPUs integrated with NXP's wide portfolio of microcontrollers and application processors.

The eIQ Neutron NPUs offer support for various neural network types such as CNN, RNN, TCN, Transformer networks, and more. ML application development with the eIQ Neutron NPU is fully supported by the eIQ machine learning software development environment. The NPU used in MCX N94 is Neutron N1-16. Its block diagram is shown in Figure 1

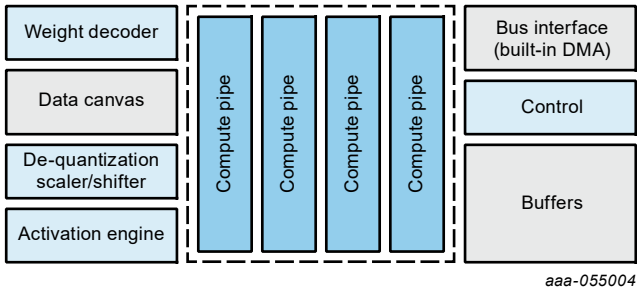

Figure 1. Neutron N1-16 block diagram

The eIQ Neutron N1-16 NPU found inside the MCX N94 has 4 compute pipes and each compute pipe contains 4 INT8 MAC (Multiply Accumulate) blocks for a total of 16 MAC blocks. This means that MCX N94 could execute 4.8G (150MHz * 4 * 4 * 2) INT8 operations per second.

The MCUXpresso Software Development Kit (MCUXpresso SDK) provides a comprehensive software package with a preintegrated TensorFlow Lite for Microcontrollers (TFLM).

The Neutron library is integrated into TFLM as well. Table 2 shows the operators supported by the NPU.

Table 2. The operators supported by the NPU

Operator	Operator input type	MCXN947/MCXN548 NPU
ADD	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes

Table 2. The operators supported by the NPU*...continued*

Operator	Operator input type	MCXN947/MCXN548 NPU
AVERAGE_POOL_2D	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
CONV_2D	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
DEPTHWISE_CONV_2D	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
FULLY_CONNECTED	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
UNIDIRECTIONAL_SEQUENCE_ LSTM	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	No
LOGISTIC (Sigmoid)	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
MAX_POOL_2D	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	Yes
MUL	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	No
SOFTMAX	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	No
SVDF	Float	No
	Uint8(PTQ)	No
	Int8(PCQ)	No

Note:

- *PTQ — Per-tensor quantized (asymmetric 8-bit quantization).*
- *PCQ — Per-channel quantized (symmetric 8-bit quantization).*

For more information, refer to each TensorFlow Lite User's Guide in `middleware/eiq/doc` *of the SDK.*

3 Software environment setup

The setup of the software environment is based on Windows.

1. Download MCUXpresso IDE, and select the Windows version to download, as shown in Figure 2

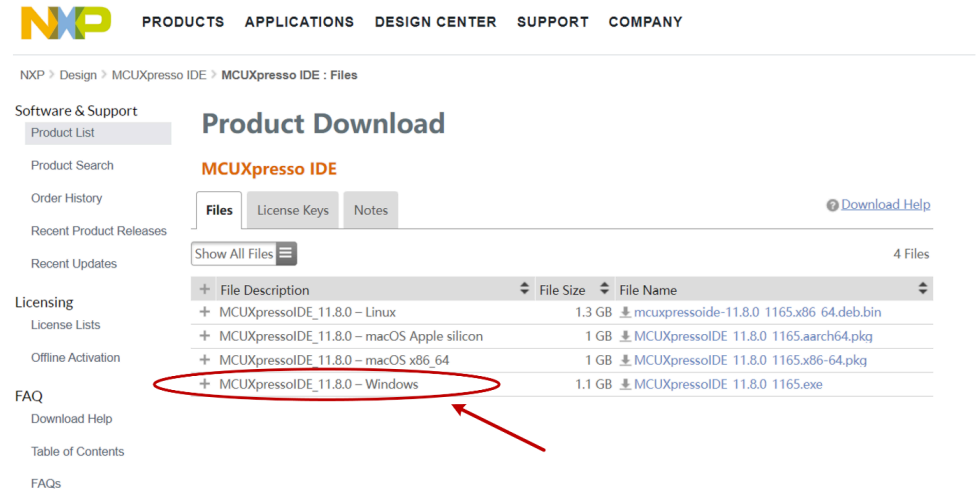

Figure 2. MCUXpresso IDE download

2. After downloading, double-click the downloaded installation package, select the installation location, and perform the installation until it is completed.

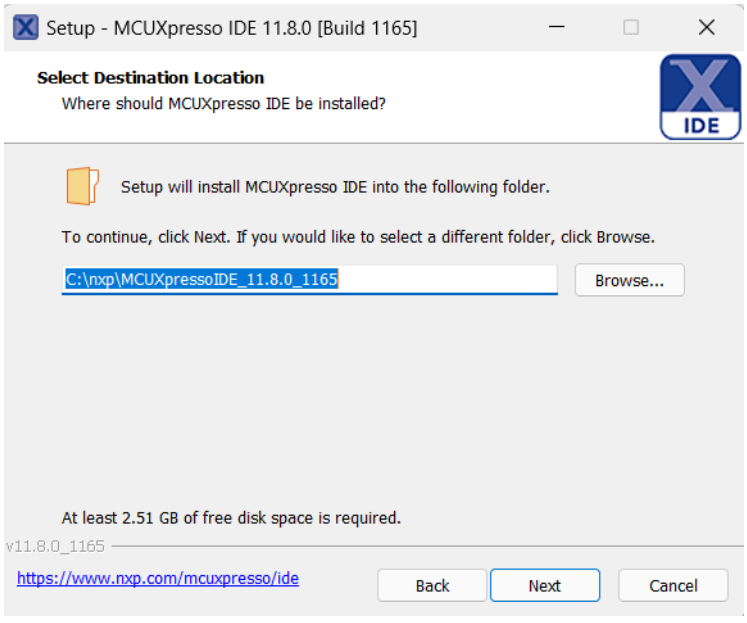

Figure 3. MCUXpresso IDE installation

Note: *Use a USB-C cable for both powering and debugging of the FRDM-MCXN947 board as shown in Figure 4. The other side of the USB-C cable must be plugged into your development computer.*

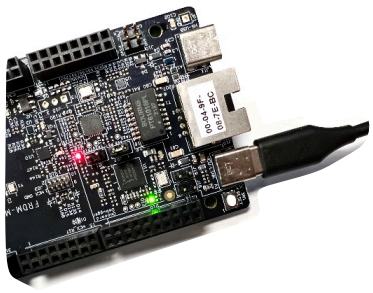

Figure 4. USB-C cable and FRDM-MCXN947 board

3. To install the required SDK, select FRDM-MCXN947 in MCUxpresso SDK Builder as shown in Figure 5.

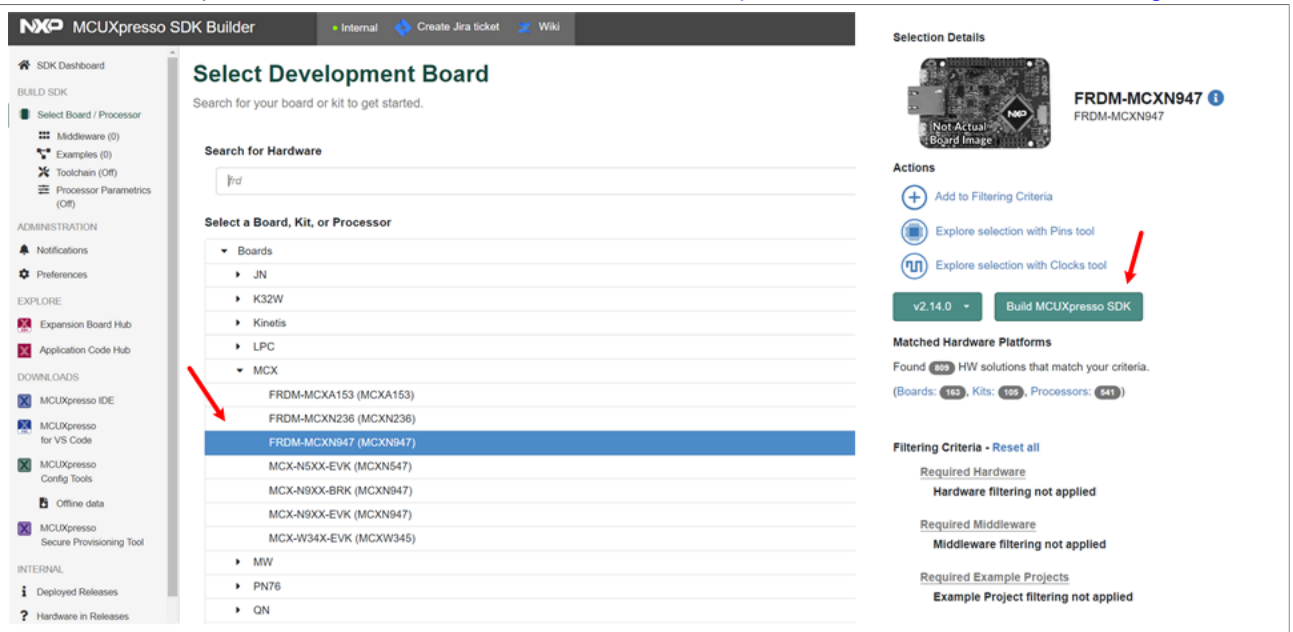

Figure 5. Selecting Development board

4. Select eIQ middleware, click **DOWNLOAD SDK** as shown in Figure 6.

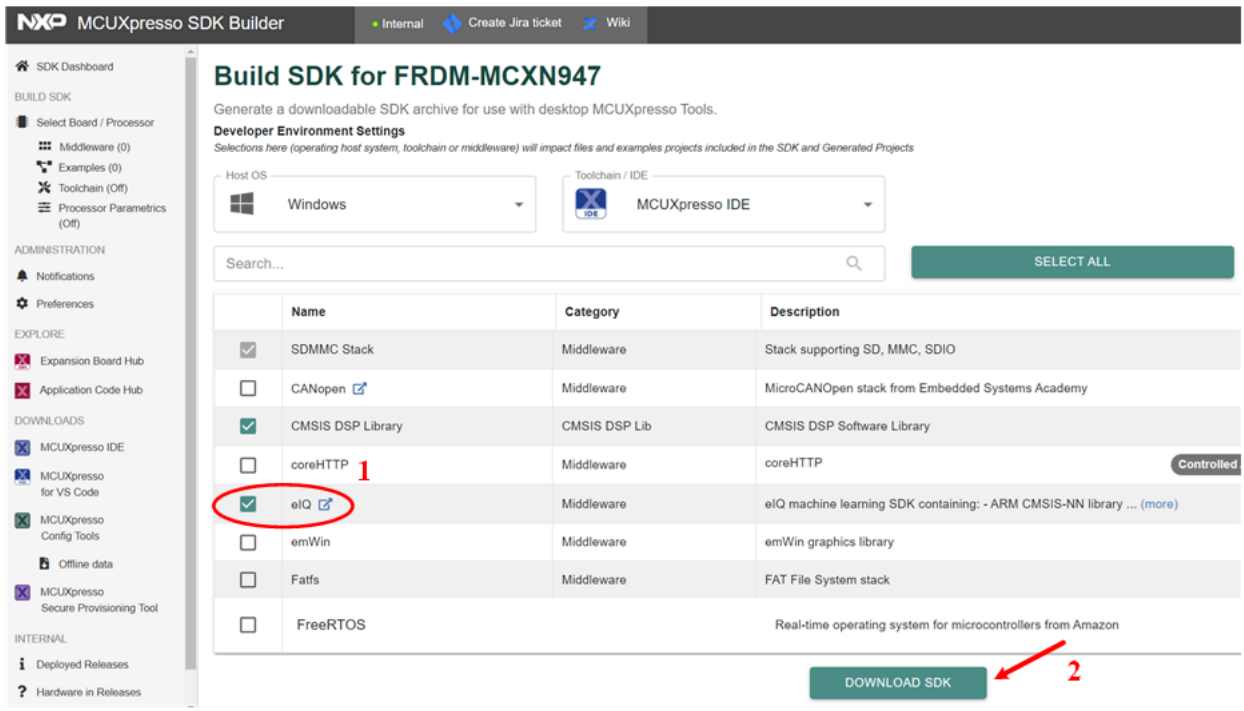

Figure 6. SDK download

5. After finishing building the SDK, download the zip SDK file as shown in Figure 7

Figure 7. zip SDK file download

6. Open MCUXpresso IDE and install the SDK by dragging and dropping the file into the SDK installation window of the MCUXpresso IDE as shown in Figure 8

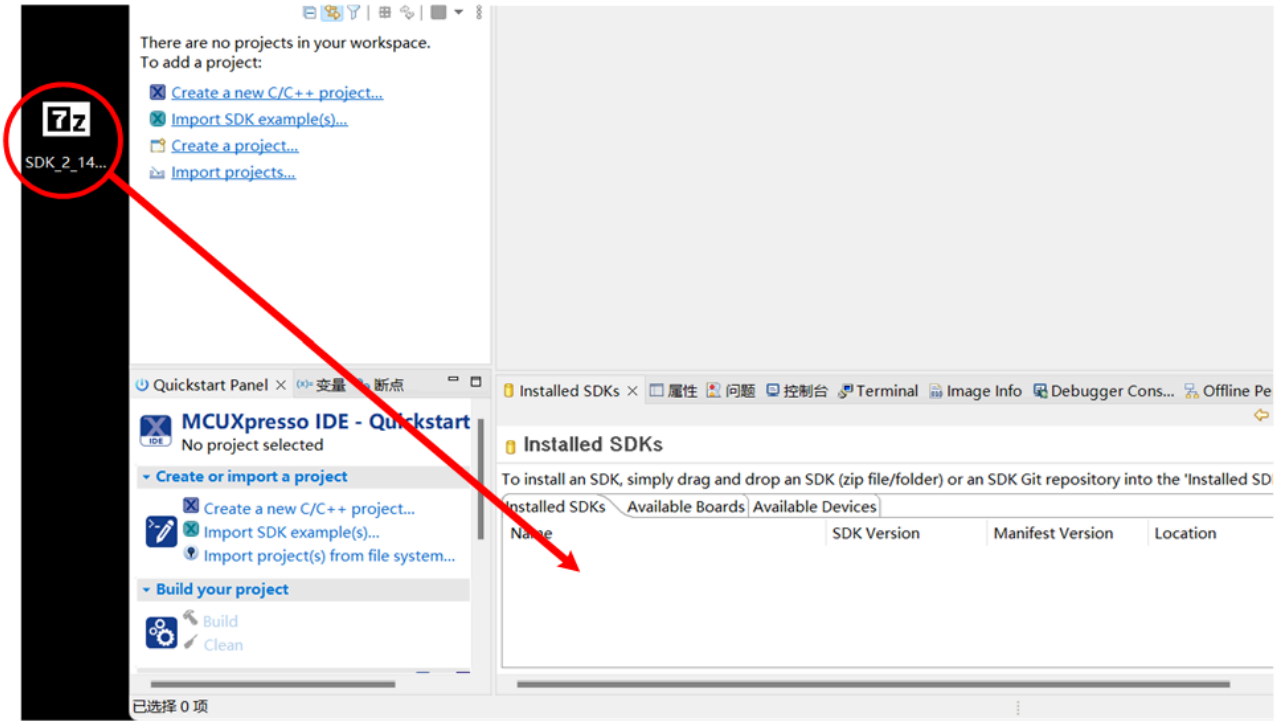

Figure 8. SDK installation

4 Converting customer model

To accelerate the model with the NPU, the model must first be converted by the Neutron Converter tool that is included as part of the eIQ Toolkit.

Download the eIQ Toolkit installation package in EIQ-TOOLKIT, double-click the downloaded installation package, select the installation location, and perform the installation until it is completed as shown in Figure 9

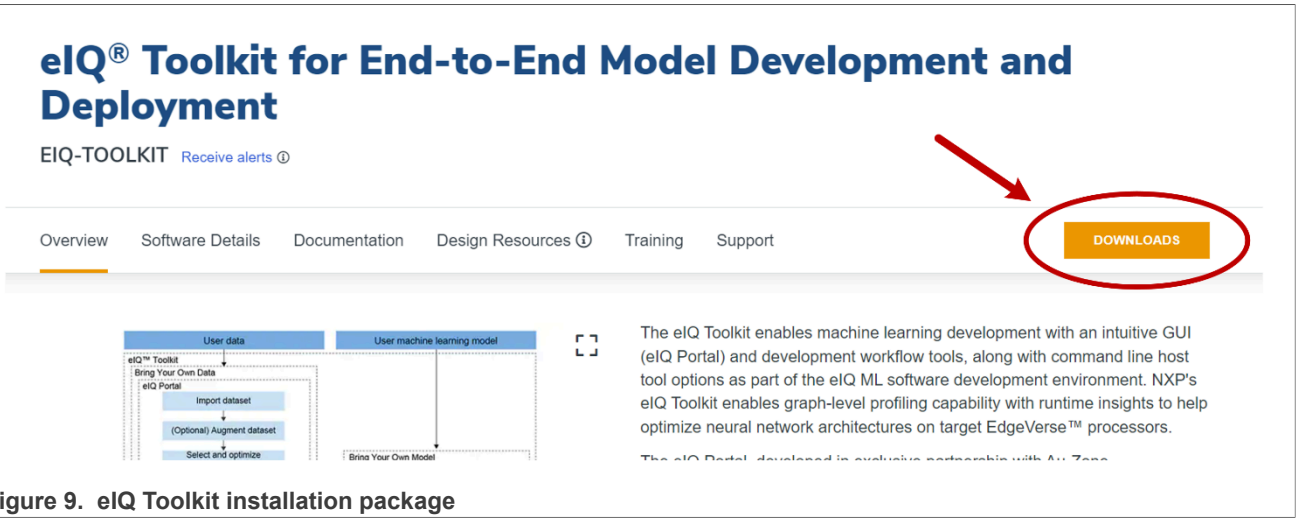

Figure 9. eIQ Toolkit installation package

There are two methods of converting custom models:

- using the command-line tool: neutron-converter
- using the eIQ portal application

4.1 Command-line tool: neutron-converter

1. Open the eIQ toolkit folder `eIQ_Toolkit_v2.1.0\bin\neutron-converter`. Two versions of the converter are shown in the folder. Choose the matching converter based on the SDK version. The corresponding relationship is shown in Table 3.

Table 3. SDK and converters

SDK	Converter
SDK-2.13.1	v1.0.0
SDK-2.14.0	v1.2.0

2. Using the command line, enter the folder of the converter, copy the custom model into the folder, and execute the command to convert the model as shown in Figure 10

Figure 10. Conversion command execution

4.2 eIQ portal application.

1. Open the eIQ portal application and click the **PLUG-INS** button on the top as shown in Figure 11

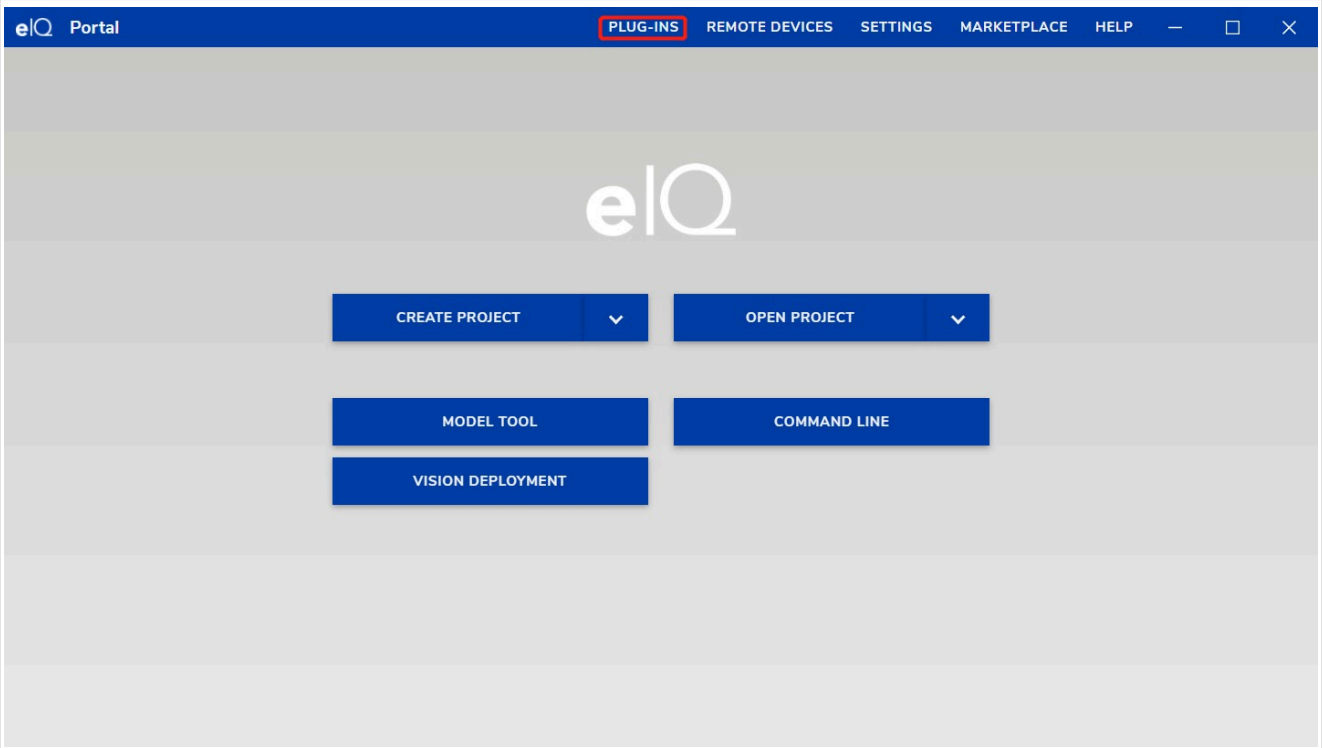

Figure 11. eIQ portal application

2. After the converter server initialization successfully, close the window and open the customer model in the MODEL TOOL. The neural network graph is shown in the window as in Figure 12.

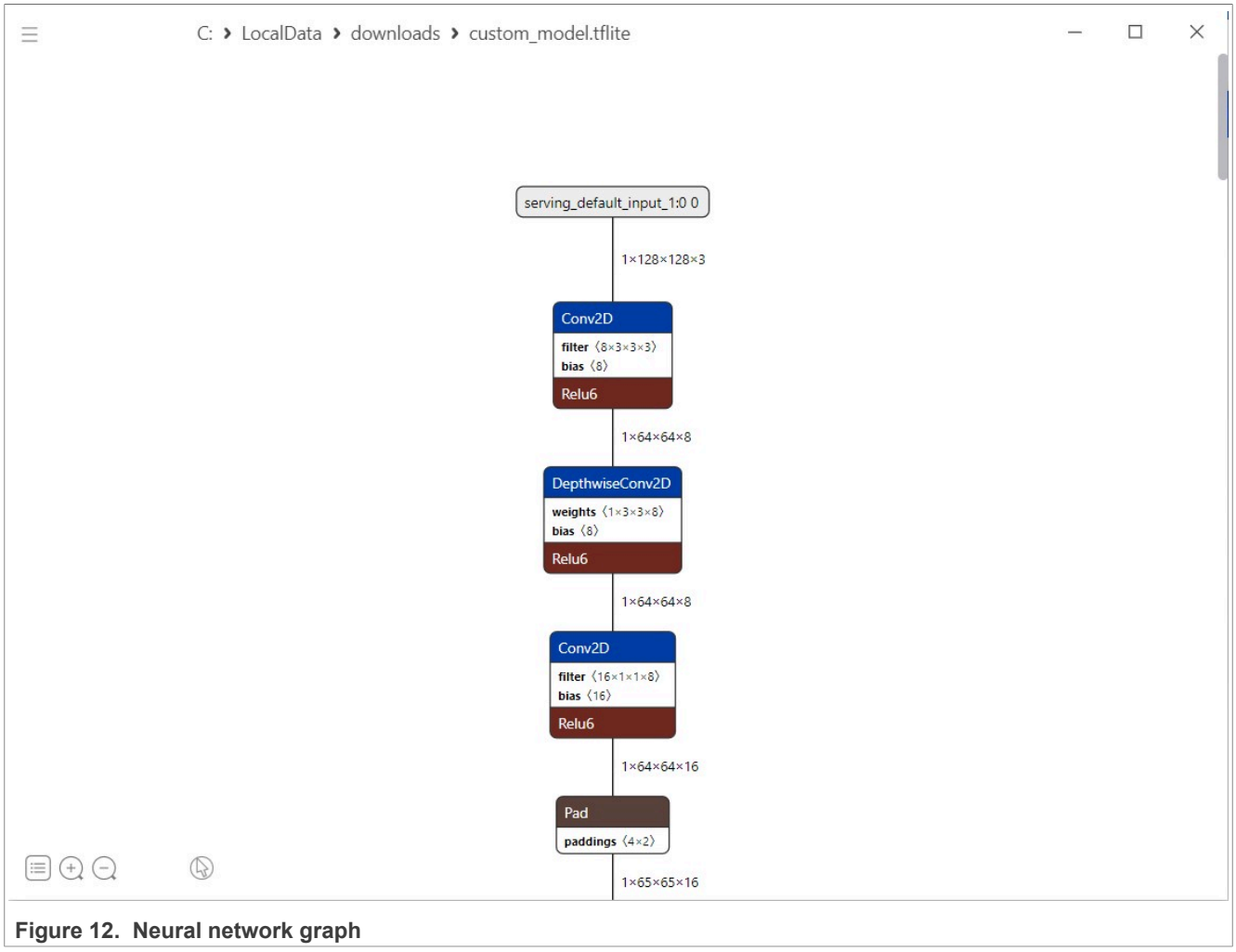

Figure 12. Neural network graph

3. Click the menu on the left top, and select TensorFlow Lite For Neutron in the Convert menu as shown in Figure 13.

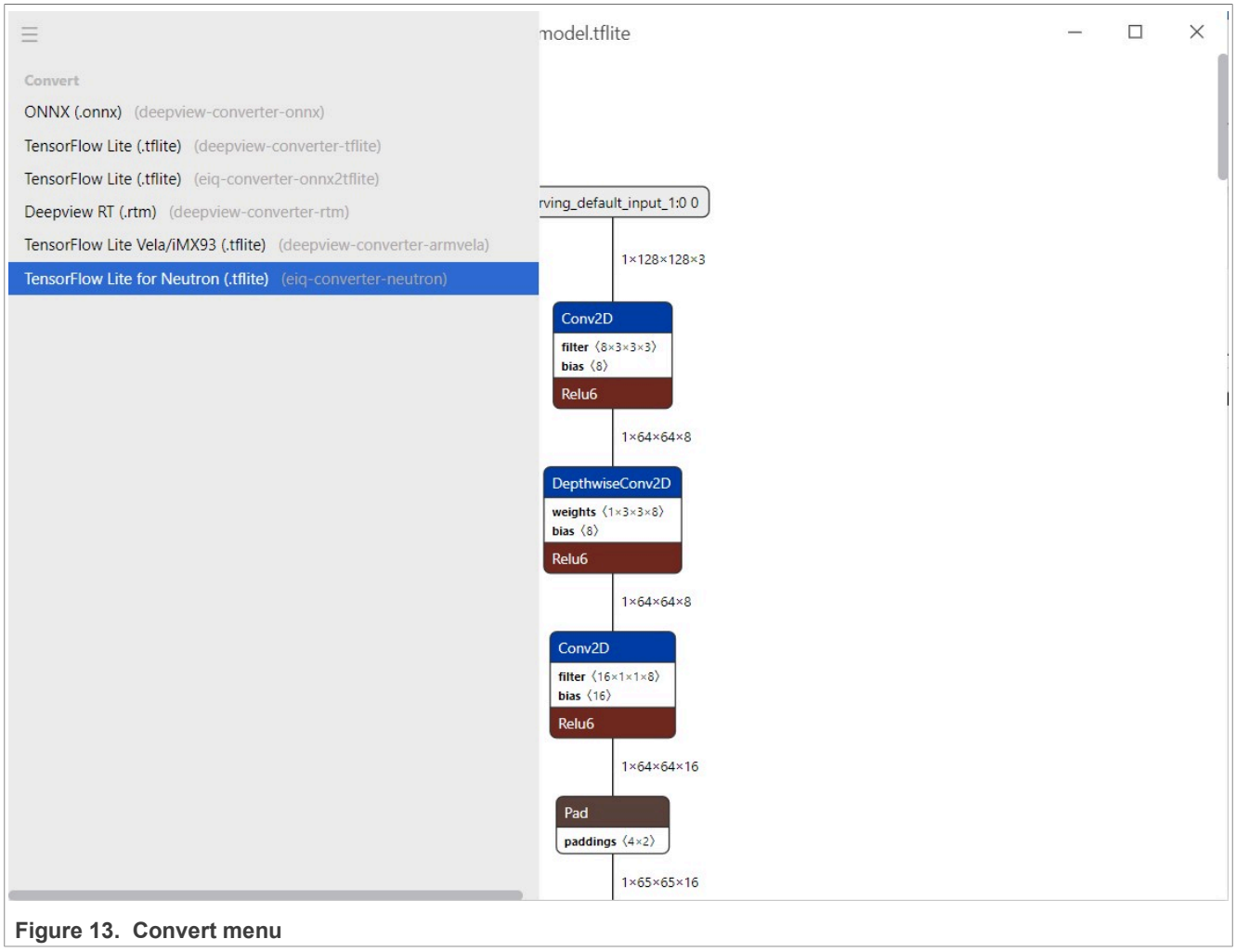

Figure 13. Convert menu

4. Click the **Convert** button, and select the save path for the converted model. After converting successfully, the converted graph is shown in the window as in Figure 14

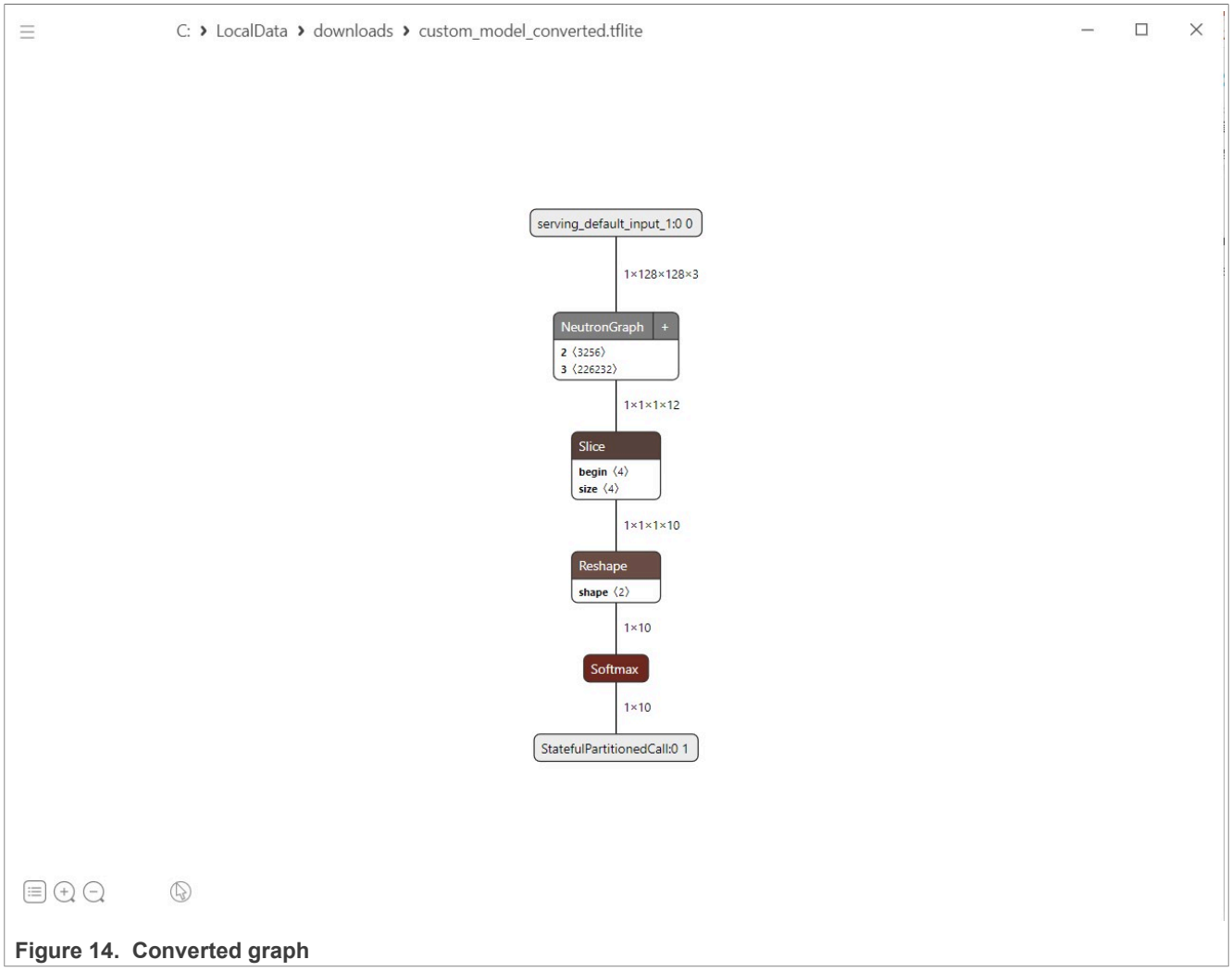

Figure 14. Converted graph

5. The Neutron Converter tool changes the graph of the model to convert operators that can be run on the NPU into custom NeutronGraph nodes as can be seen in Figure 14. The converted model is named `custom_model_converted.tflite`.
6. After converting, there are fewer operators and quantization factors are expressed more efficiently. The comparison between the model after and before being converted is shown in Table 4.

Table 4. Model after and before being converted

	Original model	After converting
File Size	299 KB	227 KB
Tensor Arena	157 KB	144 KB

5 Integrate the model into an example project

1. Import the `tflm_ciar10` project. To import the `tflm_ciar10` project, follow the steps in Figure 15 and Figure 16.

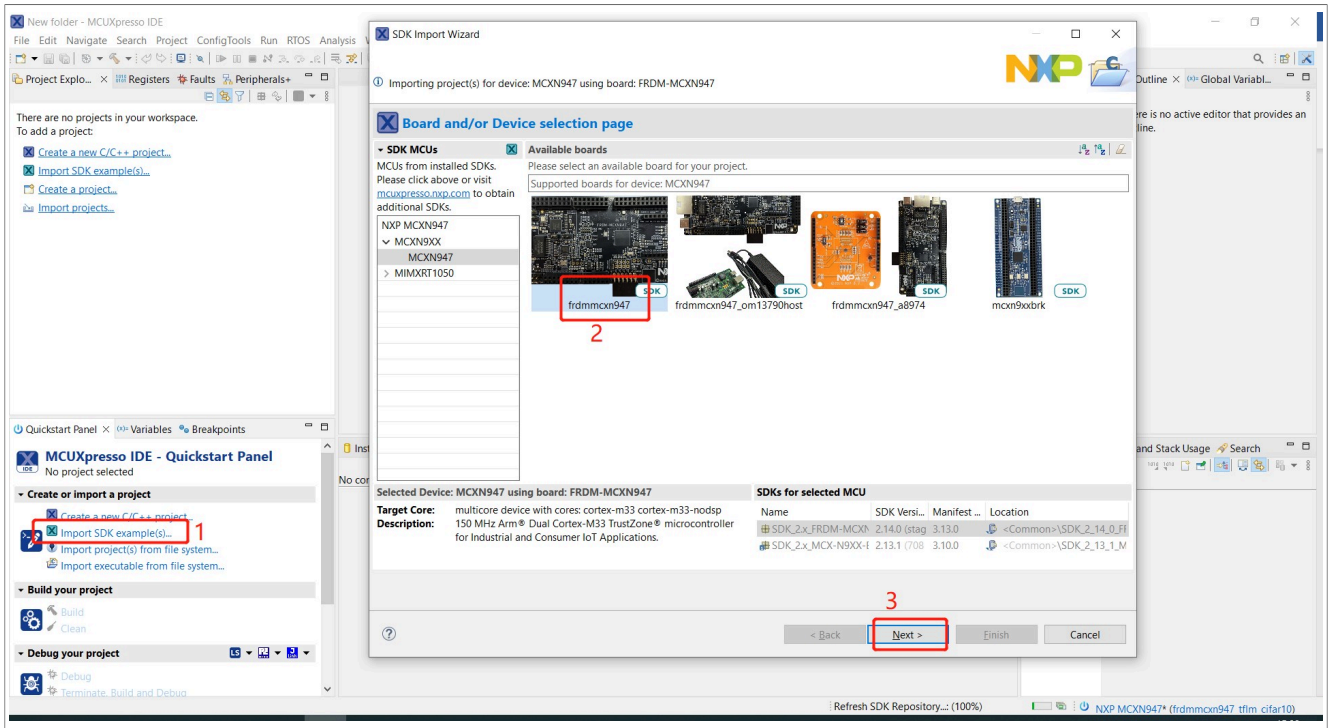

Figure 15. Importing `tflm_ciar10` project

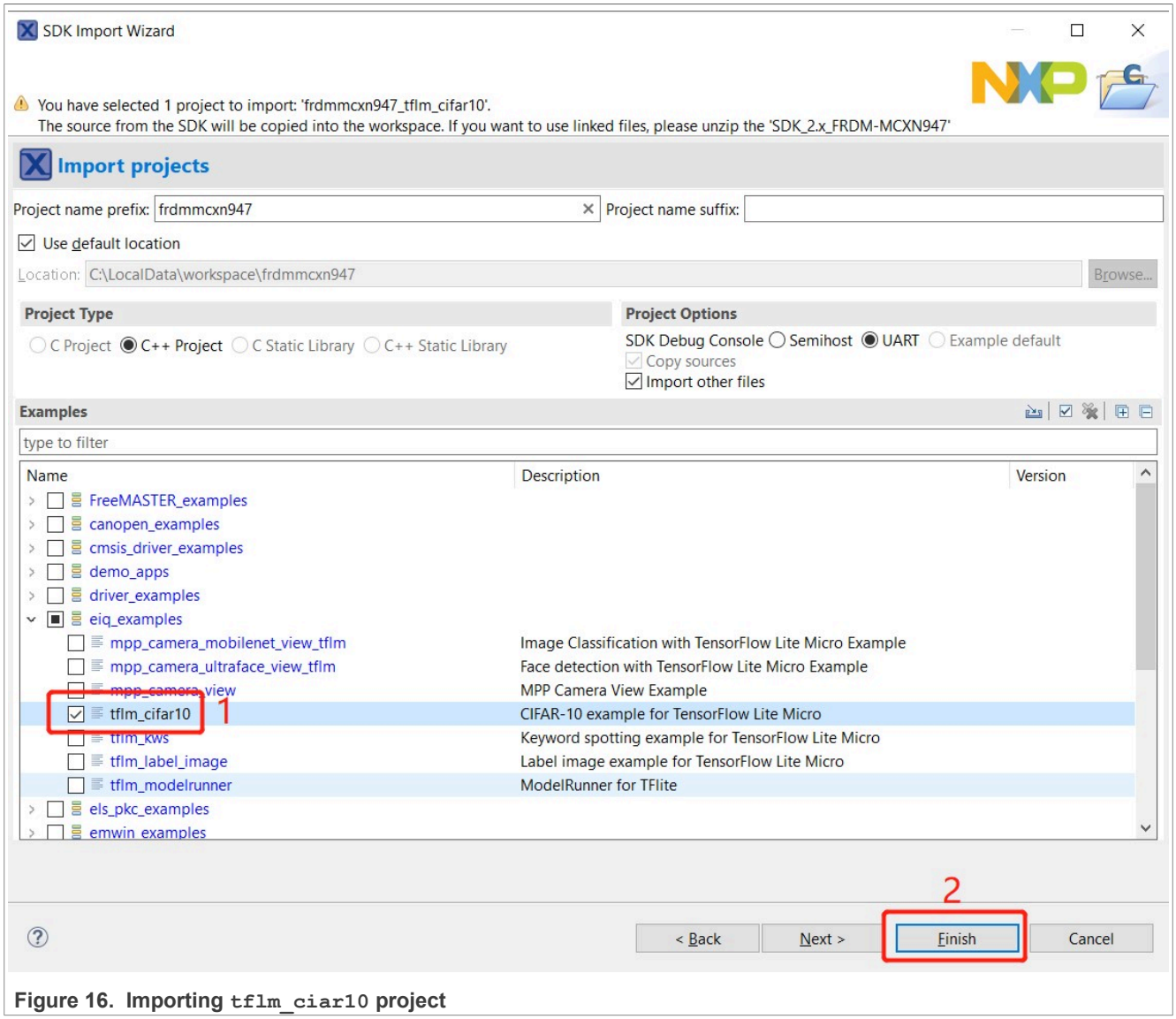

Figure 16. Importing `tflm_ciar10` project

2. Copy the converted model file into the folder of your project, create an assembly file named `model_data.s`, and include the model file into the data section `custom_model_data`. Export directives in lines 32 to 35 as shown in Figure 17.

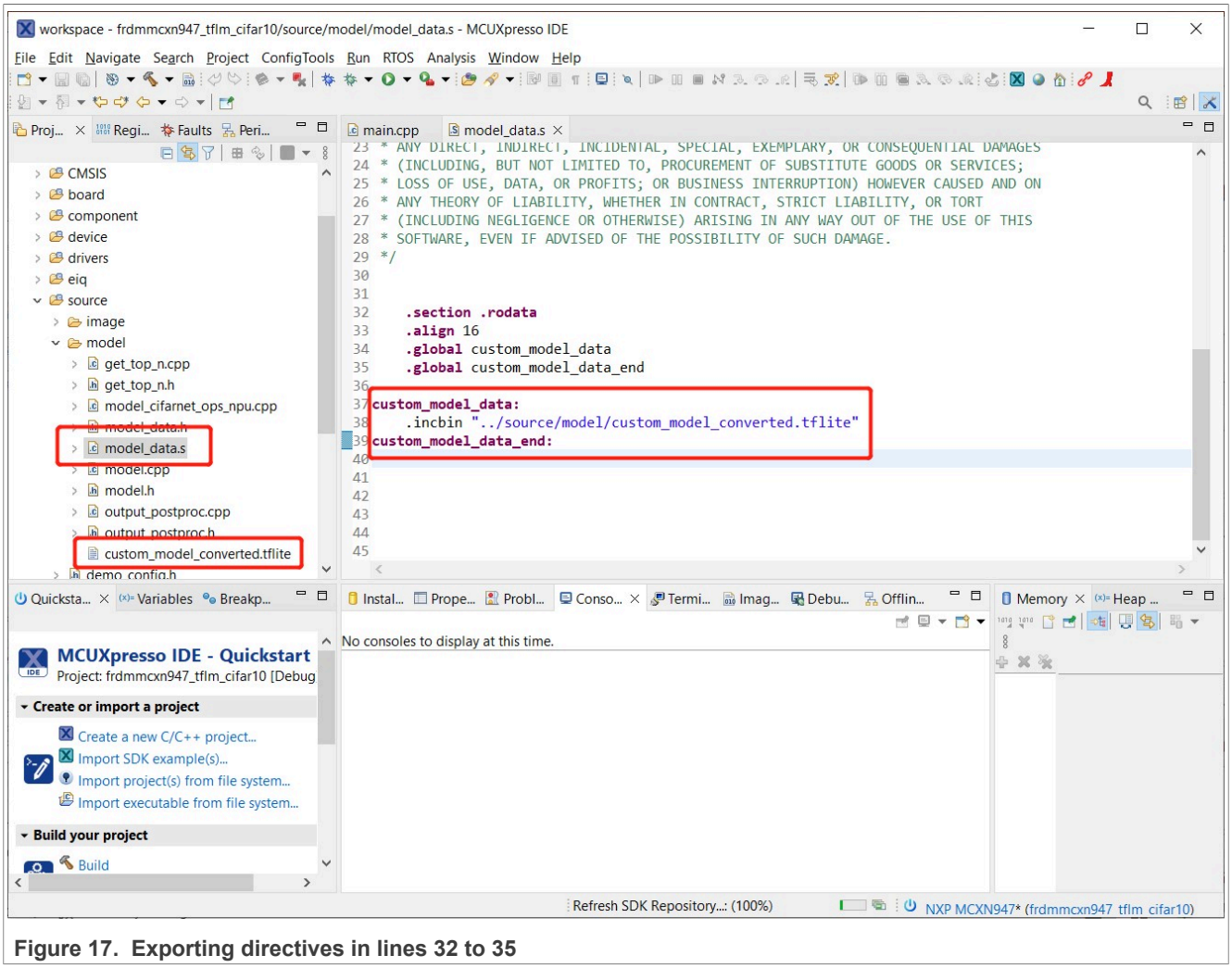

Figure 17. Exporting directives in lines 32 to 35

3. Modify `model.cpp`, change to use the custom model data as shown in Figure 18

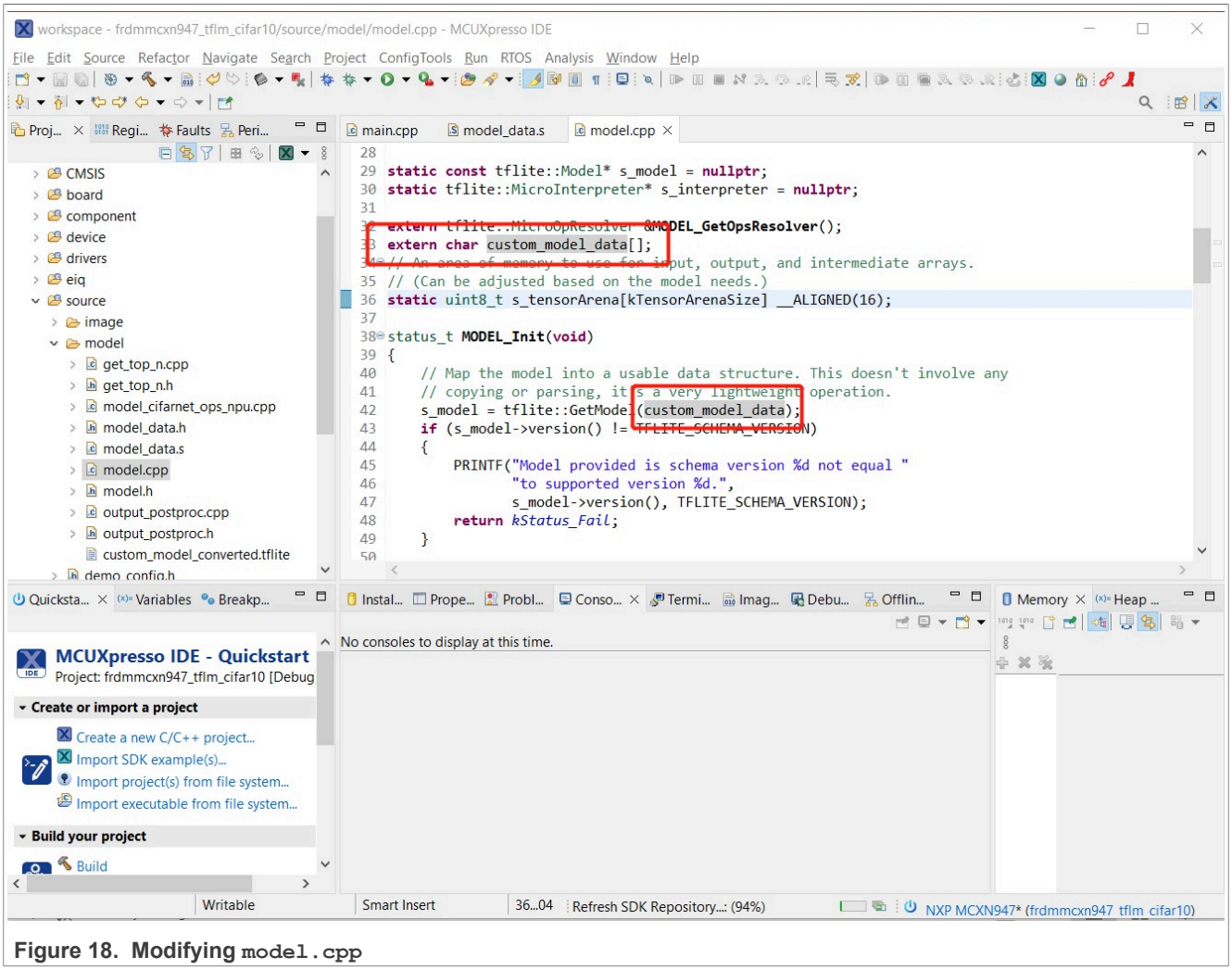

Figure 18. Modifying `model.cpp`

4. Modify `model_cifarnet_ops_npu.cpp`, and make sure all the operators used by the custom model are added in `s_microOpResolver` as shown in Figure 19.

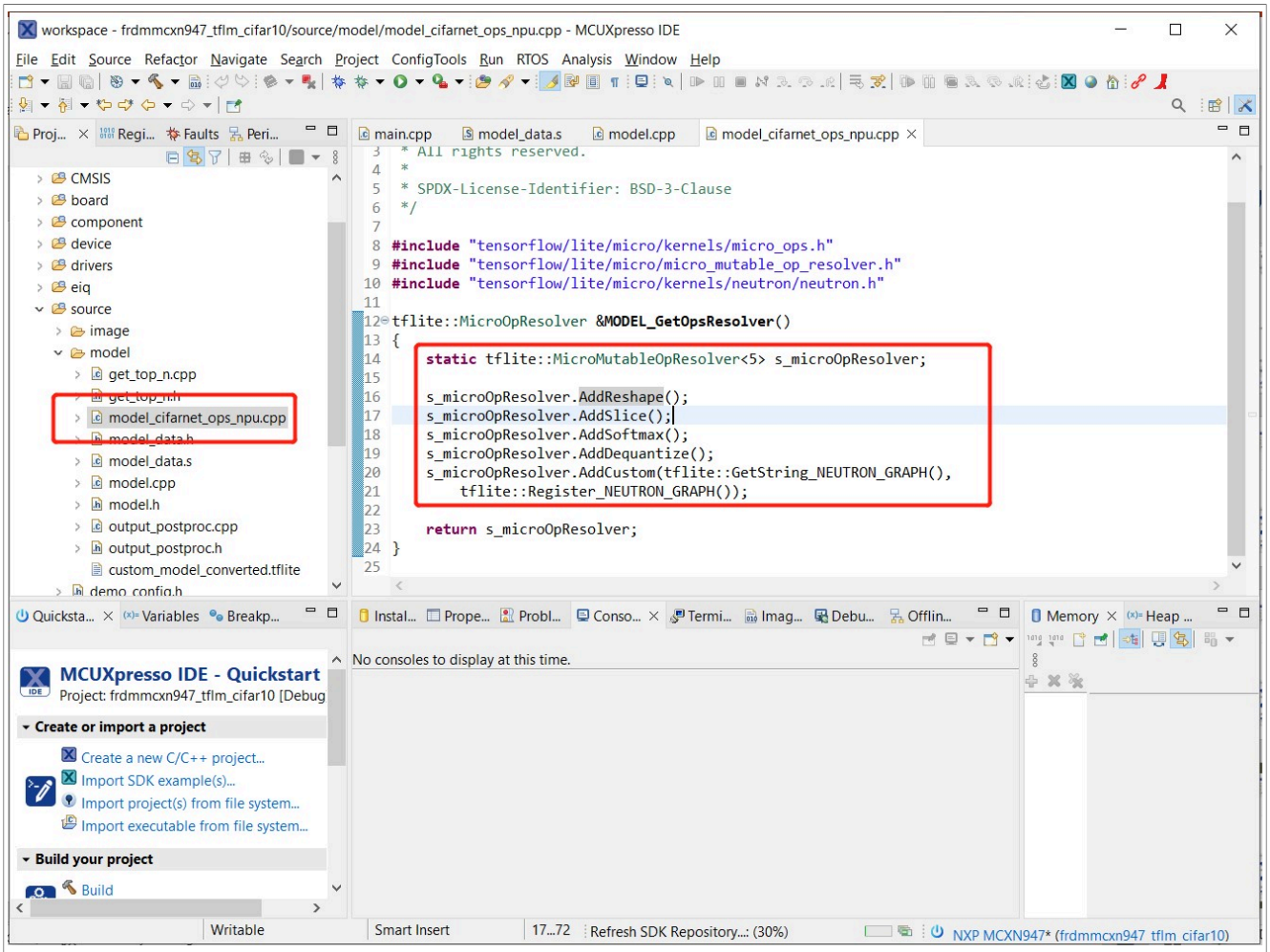

Figure 19. Modifying `model_cifarnet_ops_npu.cpp`

5. Prepare some test data, use the following Python script program to load the selected JPG file (included in the CIFAR10 test dataset), uncompress it, and export it to a C array, together with the array definition. Save it to `bird.h` within the same folder of the following Python script file. Ensure that the path to the image is correct, relative to the location from where the script is executed as shown in Figure 20.

```
import cv2
import numpy as np
img = cv2.imread('test/bird/batch_5_num_113.jpg')
img = cv2.resize(img, (128, 128))
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
with open('bird.h', 'w') as fout:
    print('#define STATIC_IMAGE_NAME "bird"', file=fout)
    print('static const uint8_t bird [] = {', file=fout)
    img.tofile(fout, ', ', '0x%02X')
    print('};\n', file=fout)
```

Figure 20. Script execution

6. Copy the header file (`bird.h`) into the image folder of the project, and modify `image_loade.c`. Switch to use the bird image data as shown in Figure 21.

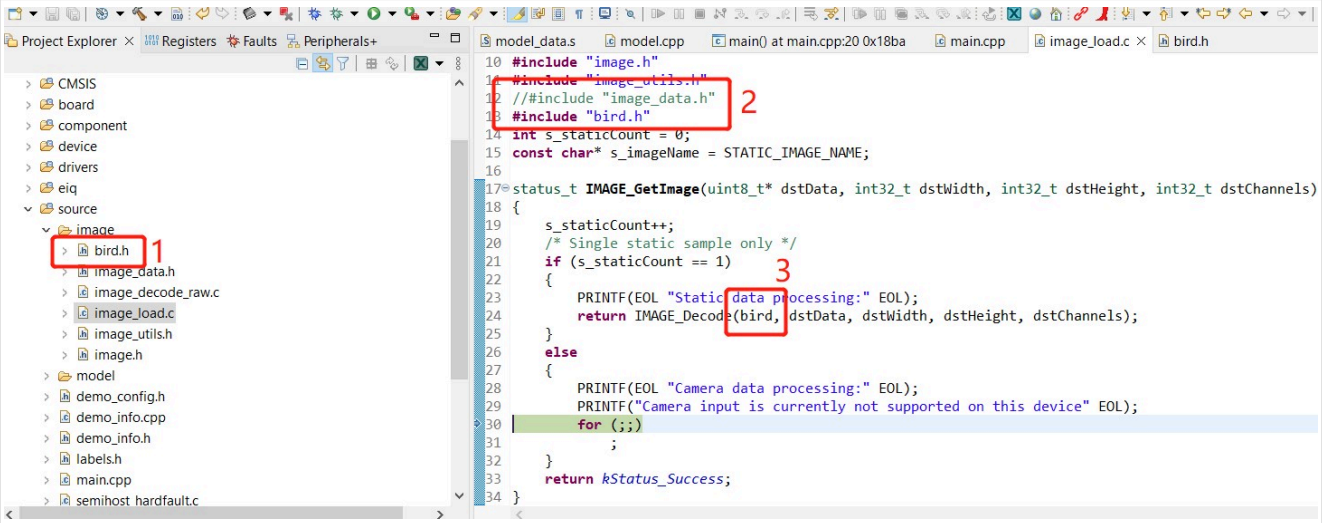

Figure 21. Using bird image data

7. Compile the project and download it to the FRDM-MCXN947 board.

6 Result

The result of running the program on the board is shown in Figure 22, the inference time is about 9 ms, and the score (confidence) of the result is 98%.

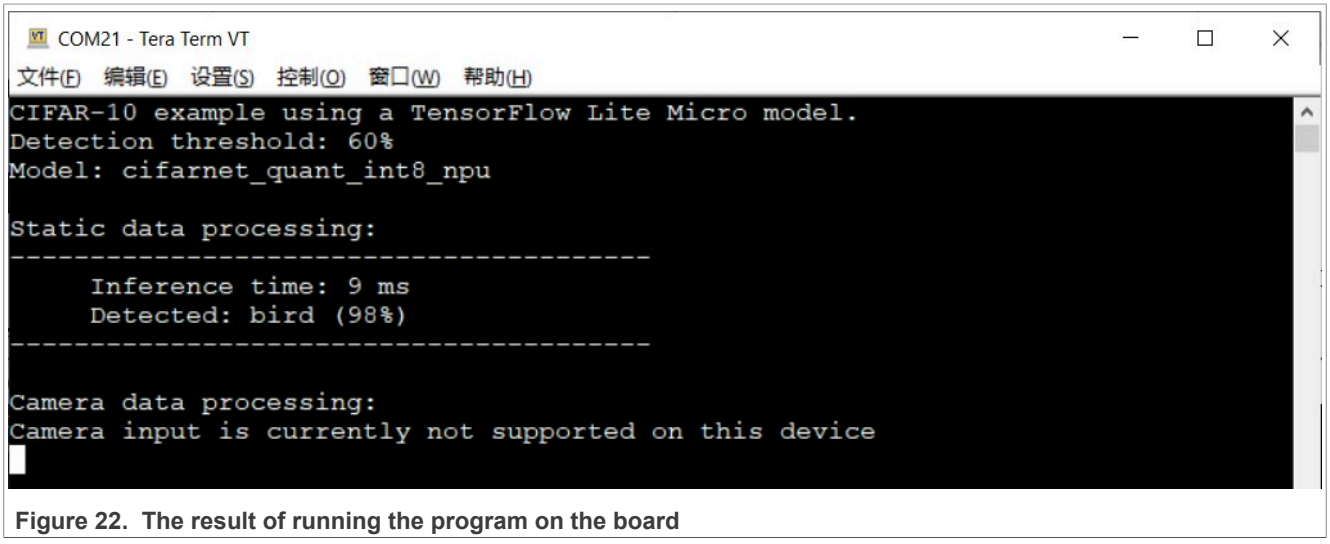

Figure 22. The result of running the program on the board

7 Revision history

Table 5. Revision history

Document ID	Release date	Description
AN14241 v.1	01 March 2024	Initial version

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

For more information, please visit: https://www.nxp.com

Date of release: 1 March 2024
Document identifier: AN14241