User's Manual

RENESAS

e-AI Translator V1.4.0

User's Manual

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.4.01 Mar 2020

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
5. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
6. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
11. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.4.0-1 November 2017)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Table of Contents

Page

Terminology

Some specific words used in this user's manual are defined below.

e-AI

An abbreviation of Embedded Artificial Intelligence.

e-AI Translator

A tool that converts and imports the inference processing of neural network models which have been trained in an open-source deep-learning framework, Caffe or TensorFlow or Keras, into files of code for the integrated development environment for MCUs and MPUs. The e-AI Translator operates as a plug-in for the e^2 studio integrated development environment for Renesas MCUs and MPUs.

Neural network

A model that has been made to provide computers with training abilities by representing the mechanisms of the neural circuits of a human brain.

Deep-learning framework

Software for defining and training neural network models.

Host machine

The personal computer on which the e-AI Translator runs.

User system

An application system in which an MCU is used, and that is run by the user.

User program

An application program that runs on the MCU.

1. Overview

The e-AI Translator is a tool that converts and imports the inference processing of neural network models which have been trained in an open-source deep-learning framework, Caffe or TensorFlow or Keras, into files of source code for the e^2 studio integrated development environment for MCUs and MPUs.

This document describes the methods of installation, usage, and points for caution of the e-AI Translator.

The e-AI Translator operates as a plug-in for the e^2 studio integrated development environment for Renesas MCUs and MPUs.

1.1. Operating Environment

Table 1-1 shows the operating environment of the e-AI Translator and required software to be installed.

Table 1-1 Operating Environment

Item	Name of Software, Supported Versions, etc.
Supported deep-learning frameworks	Caffe*[1], TensorFlow, Keras*[2]
Operating environment	The 64-bit edition of Windows 7, Windows 8.1, or Windows 10
Required software to be installed	Python 3.5.3
	The following packages from Python Package need to be installed. *[3] Tensorflow *[5] progressbar33 prettytable h5py pycryptodome configparser psutil keas (only when using standalone Keras)
	Microsoft Visual C++ 2015 Redistributable
	e^2 studio V6.3 or later version
Supported devices*[4]	RL78 family, RX family, RA family, RE family, RZ family, and Renesas Synergy^{TM} microcontrollers

Notes: 1. Caffe does not officially support the Windows environment.
For details on setting up the Ubuntu environment for it, refer to section 2.6, Installing Caffe (Only when Using Caffe).

2. The support version of Keras are both standalone Keras 2.2.4 and TensorFlow-backend Keras 2.1.6-tf.

3. For details on installing from Python Package, refer to section 2.4, Installing the Required Python Packages.

4. The usage of ROM and RAM increases with the size of the neural network. We recommend using a device with a large memory capacity.

5. The support version of TensorFlow is 1.12.0 or earlier.

1.2. Types of Convertible Neural Networks

Table 1-2 shows the types of neural networks that can be converted by the e-AI Translator in its current version.

Further types of neural network will be gradually added in future updates of the plug-in.

Note: Non-supported neural networks cannot be converted correctly. Be sure to confirm that the neural network to be converted only consists of supported types.

Table 1-2 Types of Convertible Neural Networks

Item	Types to be Supported
Algorithms*	Convolutional neural network (CNN), Auto encoder, etc.
Convolution layer	Convolution and Deconvolution (Transposed convolution)
Pooling layer	Max. pooling and Average pooling
Fully connected layer	Fully connected (Inner product)
Normalization layer	LRN (Local Response Normalization), Batch Normalization
Activation functions	Rectified linear unit (ReLU), Sigmoid, and Tanh
Classification function	Softmax

Note: The current version does not support recurrent neural networks (RNNs).

1.3. Supported API list

Table 1-3 shows the supported API of TensorFlow and supported Layer types of Caffe that can be converted by the e-AI Translator in its current version.

Further APIs and Layer types will be gradually added in future updates of the plug-in.

Table 1-3 List of supported API / Layer types (1/2)

TensorFlow API	Keras API	Caffe Layer types
tf.nn.conv2d tf.contrib.layers.conv2d tf.layers.conv2d	keras.layers.Conv2D tf.keras.layers.Conv2D	Convolution
tf.nn.conv2d_transpose tf.contrib.layers.conv2d_transpose tf.layers.conv2d_transpose	keras.layers.Conv2DTranspose tf.keras.layers.Conv2DTranspose	Deconvolution
tf.nn.max_pool tf.contrib.layers.max_pool2d tf.layers.max_pooling2d	keras.layers.MaxPooling2D tf.keras.layers.MaxPool2D	Pooling with pooling_param {pool: MAX}
tf.nn.avg_pool tf.contrib.layers.avg_pool2d tf.layers.average_pooling2d	keras.layers.AveragePooling2D tf.keras.layers.AveragePooling2D	Pooling with pooling_param {pool: AVE}
tf.nn.relu	keras.activations.relu tf.keras.activations.relu	ReLU
tf.nn.tanh tf.tanh	keras.activations.tanh tf.keras.activations.tanh	TanH
tf.sigmoid	keras.activations.sigmoid tf.keras.activations.sigmoid	Sigmoid
tf.nn.softmax tf.contrib.layers.softmax tf.nn.softmax_cross_entropy_with_logits tf.nn.softmax_cross_entropy_with_logits_v2	keras.activations.softmax tf.keras.activations.softmax keras.layers.Softmax tf.keras.layers.Softmax	Softmax
tf.matmul tf.contrib.layers.fully_connected tf.layers.dense	keras.layers.Dense tf.keras.layers.Dense	InnerProduct
-	-	Input
tf.add tf.nn.bias_add tf.contrib.layers.bias_add	-	-
tf.reshape tf.layers.flatten tf.contrib.layers.flatten	keras.layers.Reshape tf.keras.layers.Reshape keras.layers.Flatten tf.keras.layers.Flatten	-Note

Note: There is a caution about using "Flatten" and "Reshape" of Caffe, refer to "4. Points of Caution" for the detail.

Table 1-4 List of supported API / Layer types (2/2)

TensorFlow API	Keras API	Caffe Layer types
tf.nn.lrn tf.nn.local_response_normalization	-	LRN (Local Response Normalization)
tf.layers.batch_normalization[Note]	keras.layers.BatchNormalization tf.keras.layers.BatchNormalization	BatchNorm

Note: There is a caution about using this API, refer to "4. Points of Caution" for the detail.

Following APIs are only used in the training phase, so these APIs are ignored by translation of e-AI Translator.

- Loss functions
- Dropout functions
- Clip functions

1.4. Changes of e-AI Translator from V1.0.2 to V1.4.0

This chapter explains the changes of e-AI Translator from V1.0.2 to V1.4.0.

- Support of Keras framework both standalone Keras 2.2.4 and TensorFlow-backend Keras 2.1.6-tf. (TensorFlow-backend Keras 2.1.6-tf is included in TensorFlow 1.12.0)

- Support of TensorFlow version 1.12.0.

- Support of ”Batch Normalization” translation.

- Support of multiple neural network translation into single user program.
  Refer to “3.2. Translation of multiple neural networks” for the detail.

- Addition of specifying function for input/output node name when using TensorFlow trained model. Refer to “3.1. Basic procedure for using e-AI Translator” for the detail.

- Change of the format of license file.
  (Cannot use the e-AI Translator’s license file of older version. Be careful.)

2. Installing the e-AI Translator

This chapter explains how to install the e-AI Translator.

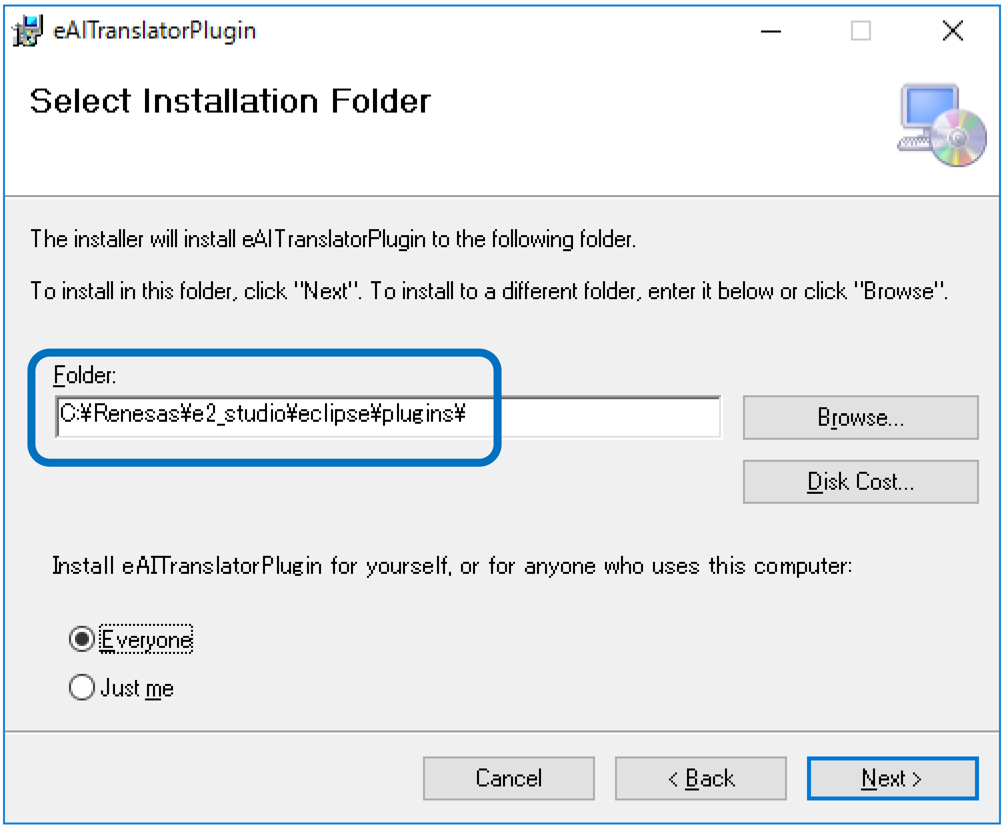
The e-AI Translator is a plug-in for the e^2 studio integrated development environment for Renesas MCUs and MPUs. After installing or if you have already installed the e^2 studio, install the e-AI Translator through the procedure below.

2.1. Installing the e-AI Translator

1. Extract zipped installer file which you have downloaded from our Web site.
2. Double-click on the extracted file "setup.exe".
3. Proceed with installation according to the instructions of the installer.

Note that the destination folder for installation must be the folder where the e^2 studio is installed.

Note: If the installation destination folder for the e^2 studio has not been changed, the folder setting for the e-AI Translator need not be changed from the default.



eAITranslatorPlugin

Select Installation Folder

The installer will install eAITranslatorPlugin to the following folder.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder: C:¥Renesas¥e2_studio¥eclipse¥plugins¥

Install eAITranslatorPlugin for yourself, or for anyone who uses this computer:

● Everyone

○ Just me

Figure 2.1 Specifying the Destination Folder for Installation

2.2. Copying the License File for the e-AI Translator

Copy the file "license.txt" that you have downloaded from the Web site to the following folder.

C:\Renesas\e2_studio\eclipse\plugins\com.renasas.eaitranslator_1.4.0

Note: If you have changed the destination folder for installation in section 2.1, change the destination folder for copying to match the path you have selected.

2.3. Installing Python 3.5.3

1. Download Python 3.5.3 from the following Web site.
   https://www.python.org/downloads/release/python-353/
   "Windows x86-64 executable installer"
2. Double-click on the installer to start installation.
   Click on "Install Now".
   Proceed with installation according to the instructions of the installer.

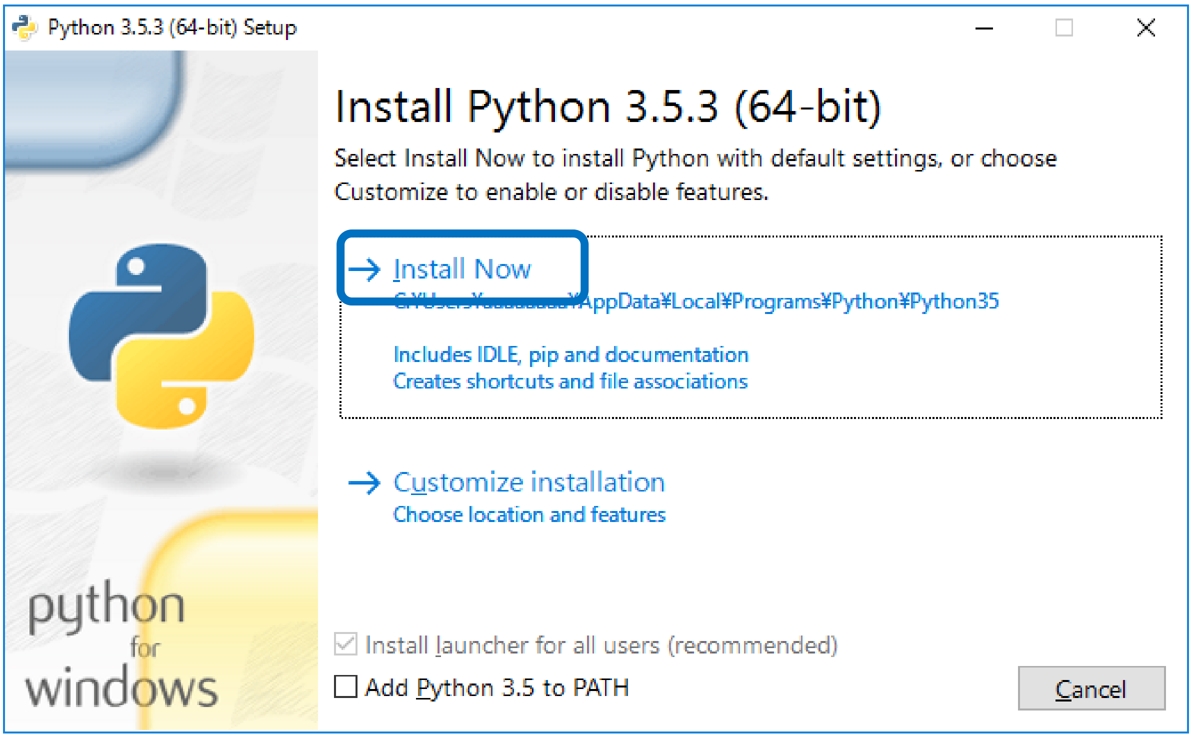
   Remark1: Both "checked" and "unchecked" settings are OK about "Add Python 3.5 to PATH".

   Remark2: For the license conditions of Python, read "License.txt" in its installation folder.

   [Installation folder for Python]

   C:\Users\*<windows-user-name>*\AppData\Local\Programs\Python\Python35\Scripts

   Note: "*<windows-user-name>*" is the user name that you use for logging on Windows.


Python 3.5.3 (64-bit) Setup

Install Python 3.5.3 (64-bit)

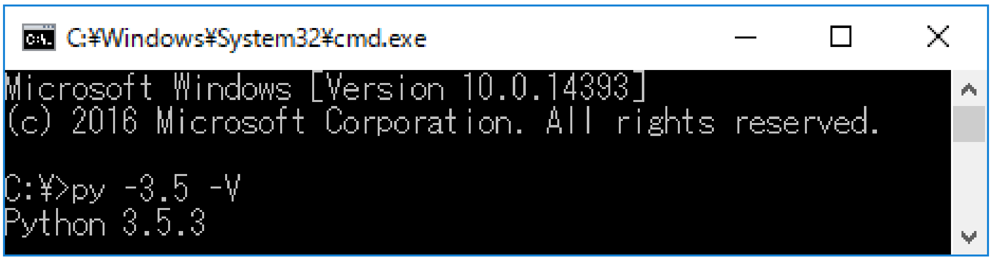
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

→ Install Now
C:¥Users¥xxxxxxxx¥AppData¥Local¥Programs¥Python¥Python35
Includes IDLE, pip and documentation
Creates shortcuts and file associations
→ Customize installation
Choose location and features
python for windows
Cancel

☑ Install launcher for all users (recommended)
□ Add Python 3.5 to PATH

Figure 2.2 Installing Python 3.5.3

3. Confirm the installation by the following steps.
   - Open command prompt of Windows
   - Execute the command "py -3.5 -V"
   - When the result is "Python 3.5.3", the installation is normally finished.
     (No need to do the procedure 4.)

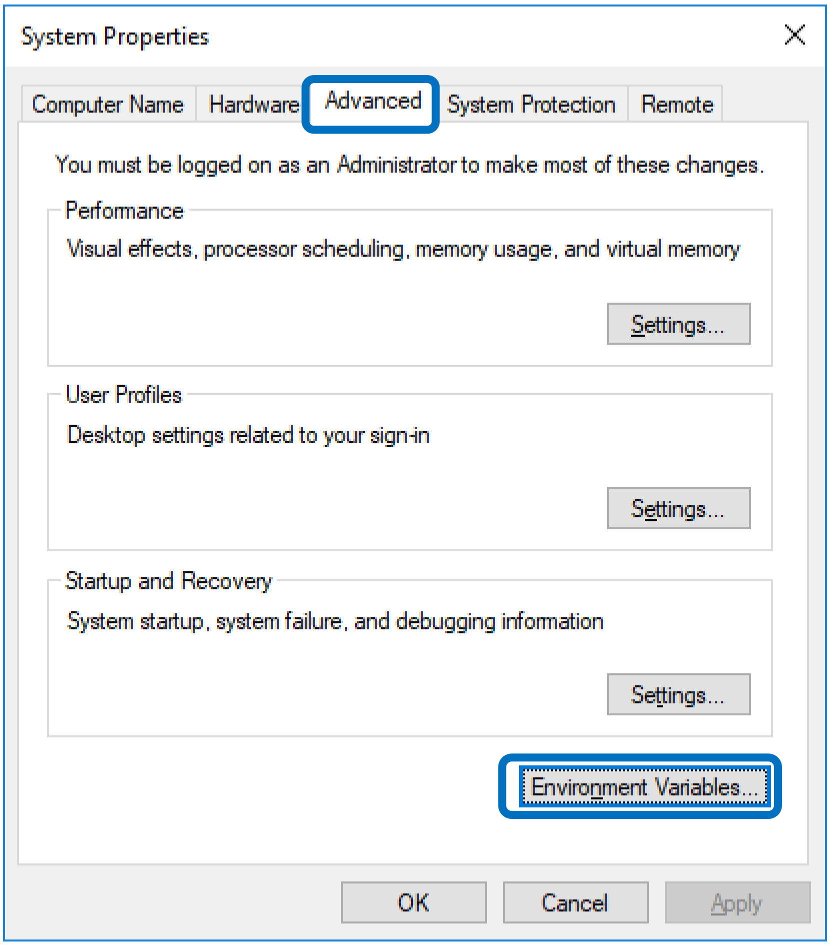


```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:¥>py -3.5 -V
Python 3.5.3
```

Figure 2.3 Confirmation of Installation Python 3.5.3

4. When the result is error in procedure 3, edit environment variables of Windows by the following steps. These window images are Windows 10 as an example.
   - Open "System Properties, then click "Environment Variables" button in the "Advanced" tab.



You must be logged on as an Administrator to make most of these changes.

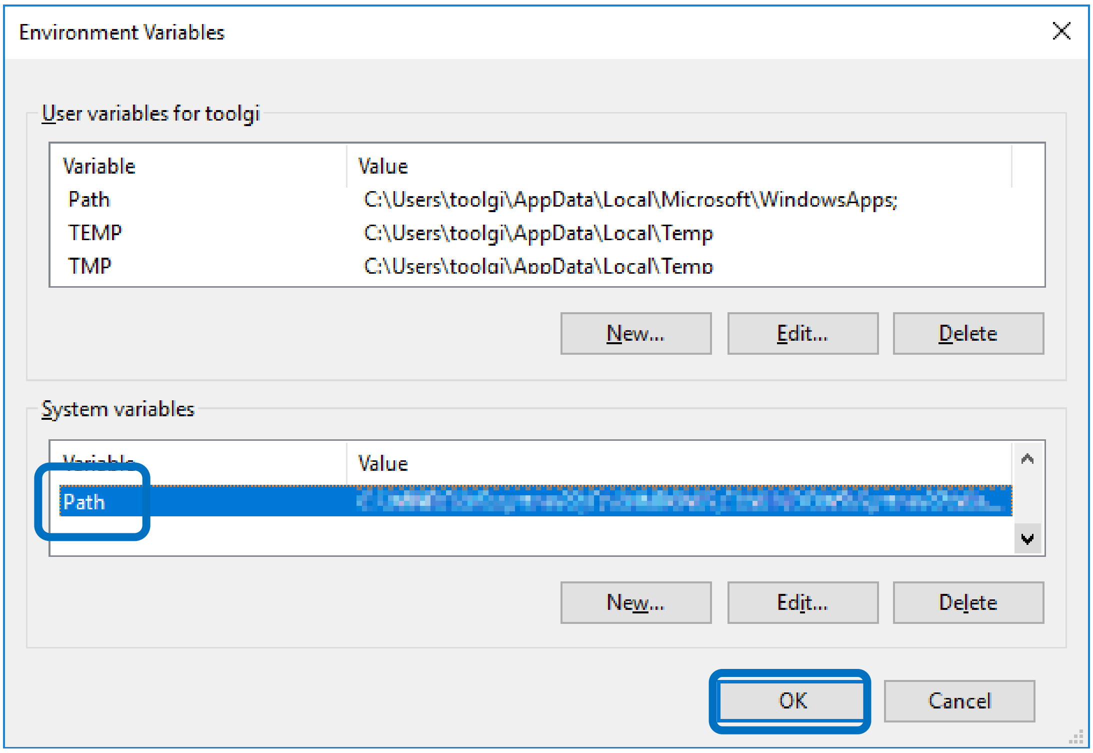
Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your sign-in

Startup and Recovery
System startup, system failure, and debugging information

Figure 2.4 System Properties

- Select "Path" from Open "System variables", then click "Edit" button.

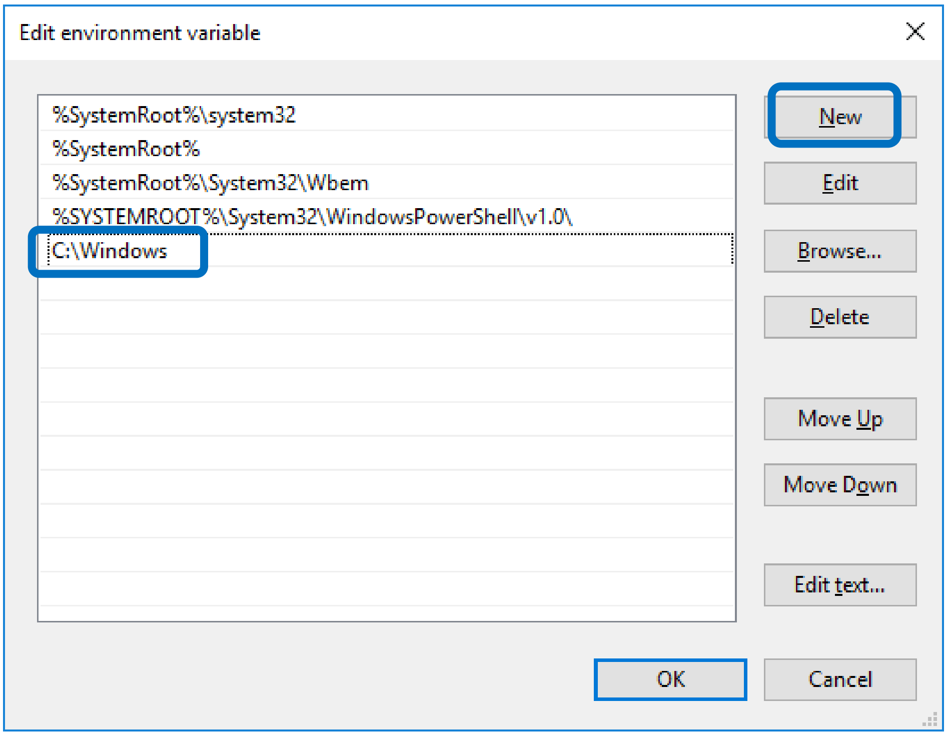


Variable	Value
Path	C:\Users\toolgi\AppData\Local\Microsoft\WindowsApps;
TEMP	C:\Users\toolgi\AppData\Local\Temp
TMP	C:\Users\toolgi\AppData\Local\Temp

Variable	Value
Path	[redacted]

Figure 2.5 Setting of Environment Variables 1

- Click "New" button, then add "C:\Windows". Click "OK" button to close each dialog, then confirm procedure 3 again.



```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
C:\Windows
```

Figure 2.6 Setting of Environment Variables 2

2.4. Installing the Required Python Packages

- Installing TensorFlow

1. Start the command prompt of Windows.
2. Run the following command.

   cd C:\Users\*<windows-user-name>*\AppData\Local\Programs\Python\Python35\Scripts

   Note: "*<windows-user-name>*" is the user name that you use for logging on Windows.

3. Run the following command.

   pip3 install --upgrade tensorflow==1.12.0

   Note: Up to 10 minutes will be required until the command ends.

- Installing ProgressBar

  After you have finished installing above, run the following command.

  pip3 install progressbar33

- Installing Prettytable

  After you have finished installing above, run the following command.

  pip3 install prettytable

- Installing h5py

  After you have finished installing above, run the following command.

  pip3 install h5py

- Installing pycryptodome

  After you have finished installing above, run the following command.

  pip3 install pycryptodome

- Installing configparser

  After you have finished installing above, run the following command.

  pip3 install configparser

- Installing psutil

  After you have finished installing above, run the following command.

  pip3 install psutil

- Installing Keras (only when using standalone Keras)

  After you have finished installing above, run the following command.

  pip3 install keras==2.2.4

2.5. Installing Microsoft Visual C++ 2015 Redistributable

1. Download 64bit version installer of Microsoft Visual C++ 2015 Redistributable "vc_redist.x64.exe" from the following Web site.
   https://www.microsoft.com/en-US/download/details.aspx?id=52685

2. Double-click on the installer to start installation.
   Proceed with installation according to the instructions of the installer.

2.6. Installing Caffe (Only when Using Caffe)

Since Caffe does not officially support operation in a Windows environment, use the following operating environment.

- OS: Ubuntu 16.04
- Versions of Python which are available for Ubuntu: 2.7.12 (Python 3 is not available.)

Install Caffe in Ubuntu by following the procedure below.

1. Install dependencies for Caffe.
   a. sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler
   b. sudo apt-get install --no-install-recommends libboost-all-dev
   c. sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
   d. sudo apt-get install libatlas-base-dev
   e. sudo apt-get install git

2. Download Caffe.
   a. cd ~
   b. git clone git://github.com/BVLC/caffe.git

3. Compile Caffe.
   a. cd ~/caffe
   b. cp Makefile.config.example Makefile.config
   c. Open the Makefile.config file in an editor and delete "#" at the beginning of the 8th line (including "CPU_ONLY:=1").
   d. make all
   e. make test
   f. make runtest

4. Compile the Python interface for Caffe.
   a. cd ~/caffe/python
   b. sudo apt-get install python-pip
   c. sudo apt-get install gfortran
   d. for req in $(cat requirements.txt); do sudo pip install $req; done
   e. sudo apt-get install python-dev python-numpy python-skimage
   f. cd ..
   g. make pycaffe

5. Confirm the operation after compilation.
   a. Add the path for Caffe.
      Example of commands:
      (bash) export PYTHONPATH=~/caffe/python/:$PYTHONPATH
      (tcsh) setenv PYTHONPATH ~/caffe/python/:$PYTHONPATH
   b. Run the "python -V" command to confirm that the version of Python is 2.7.12.
      ("x" is a numerical value.)
   c. Run the "python" command to enable the input of Python commands.
   d. Run the "import caffe" command to check that no errors have occurred.

3. Using the e-AI Translator

This chapter explains how to use the e-AI Translator.

3.1. Basic Procedure for Using the e-AI Translator

1. Start the e^2 studio.
   Create a new project as the project or open an existing project.

2. Start the e-AI Translator.
   Click on the [TR] button or select [Configure and Translate] from the [Translator] menu.

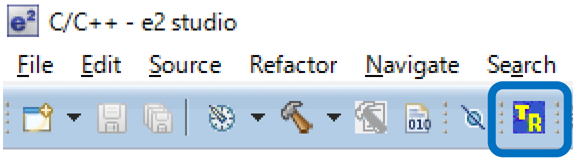


Figure 3.1 Start Button for the e-AI Translator

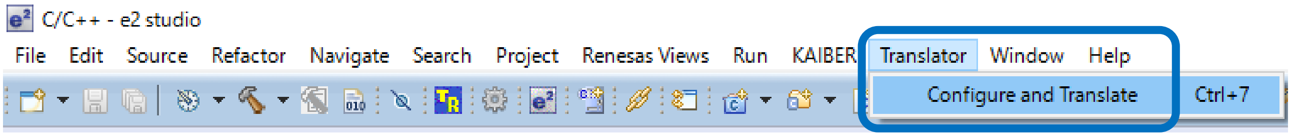
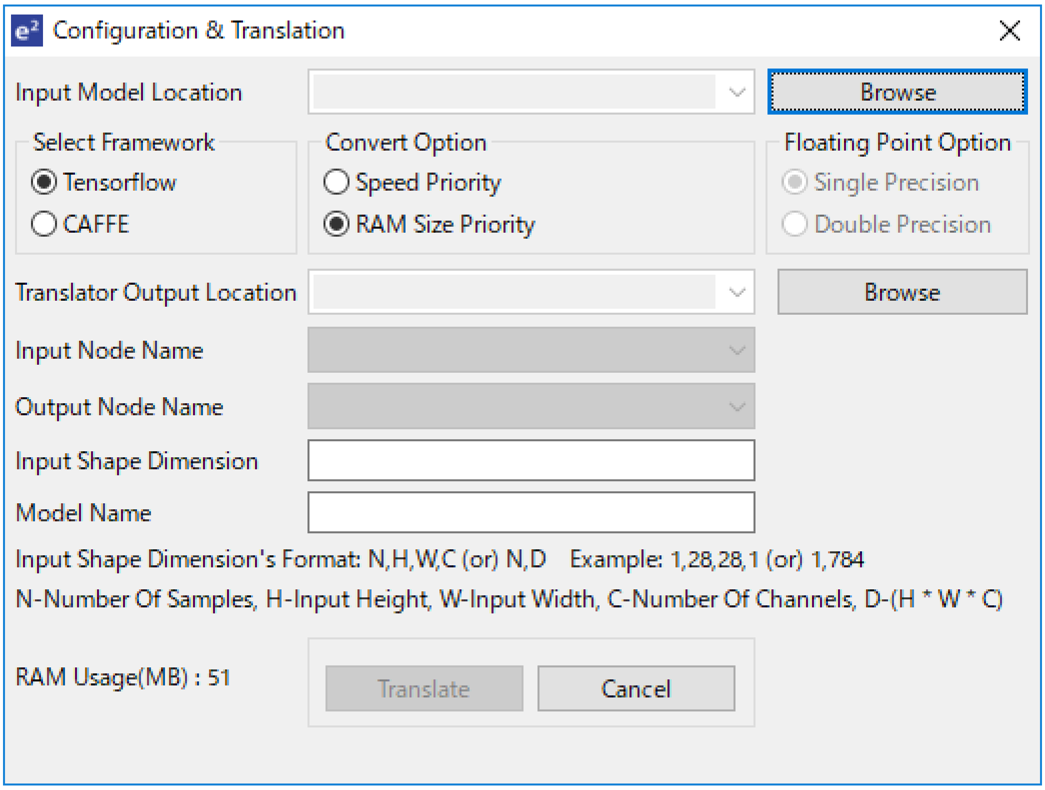


Figure 3.2 Start Menu of the e-AI Translator

3. The e-AI Translator is activated as shown below.



Configuration & Translation

Input Model Location

Select Framework
● Tensorflow
○ CAFFE

Convert Option
○ Speed Priority
● RAM Size Priority

Floating Point Option
● Single Precision
○ Double Precision

Translator Output Location

Input Node Name

Output Node Name

Input Shape Dimension

Model Name

Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784
N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 51

Figure 3.3 e-AI Translator

Table 3-1 Descriptions of Each Area of the e-AI Translator

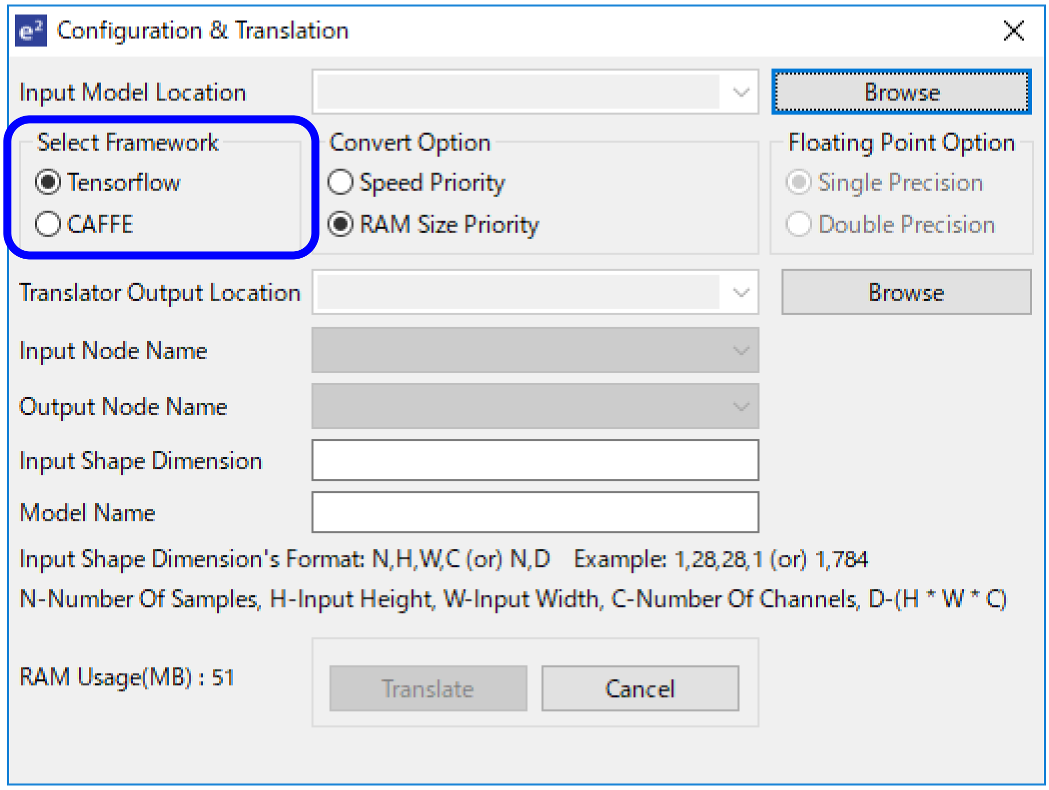
Name	Description
Input Model Location	Specifies the folder containing the trained model created by Caffe or TensorFlow or Keras.
Select Framework	Specifies the type of framework used for training by the AI. Select “tensorflow” when using Keras.
Convert option	Selects whether the speed of execution or reduction of the RAM size is given priority during conversion.
Floating Point Option	Specifies the floating-point precision (single or double) which is used for arithmetic processing by the neural network simulation. In the current version, this is fixed to single precision. Specify single precision when using the products of training by the framework. • Single precision: 32-bit floating-point data • Double precision: 64-bit floating-point data
Translator Output Location	Specifies the output destination folder for the source files and header files that are output as the results of conversion.
Input Node Name	Displays input node name which is extracted from the trained model when using TensorFlow.
Output Node Name	Displays output node name which is extracted from the trained model when using TensorFlow.
Input Shape Dimensions	Specifies the size of the input layer of the neural network. This is only required when the framework is TensorFlow.
Model Name	This area is used only when using multiple neural network. Refer to “3.2. Translation of multiple neural networks” for the detail.
RAM Usage (MB)	Displays PC RAM usage as a reference information when translating.
Translate button	Converts the trained model of the AI to source files and header files.

4. Set [Select Framework].

   Specify the framework used for training by the AI.

   - TensorFlow: Keras or TensorFlow was used for training by the AI.
   - Caffe: Caffe was used for training by the AI.

   Note: Specify [Select Framework] setting before specifying [Input Model Location].



Configuration & Translation

Input Model Location

Select Framework
● Tensorflow
○ CAFFE

Convert Option
○ Speed Priority
● RAM Size Priority

Floating Point Option
● Single Precision
○ Double Precision

Translator Output Location

Input Node Name

Output Node Name

Input Shape Dimension

Model Name

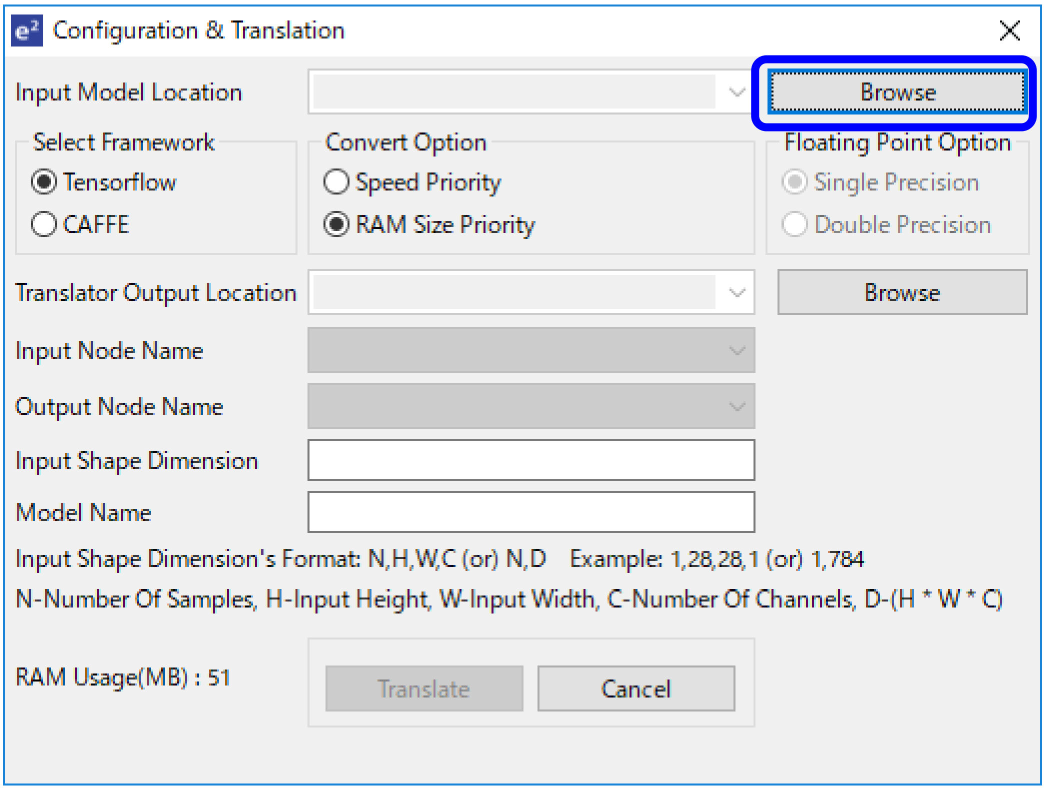
Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 51

Figure 3.4 Setting [Select Framework]

5. Click on the [Browse] button to set [Input Model Location].
   Specify the folder containing the trained model created by Caffe or TensorFlow or Keras.
   One-byte characters can be used for the folder name and path name. Do not use two-byte characters.
   The files listed following the screenshot must have been input in the cases of Caffe and TensorFlow.


Configuration & Translation

Input Model Location

Select Framework
● Tensorflow
○ CAFFE

Convert Option
○ Speed Priority
● RAM Size Priority

Floating Point Option
○ Single Precision
○ Double Precision

Translator Output Location

Input Node Name

Output Node Name

Input Shape Dimension

Model Name

Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784
N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 51

Figure 3.5 Setting [Input Model Location]

In the case of TensorFlow:

- The following four files are stored in the same folder. Specify this folder as [Input Model Location].
  - Checkpoint file
  - Trained model: A file having a name with an extension that includes "data".
  - Definition file 1 of the neural network structure: The name of this file has the extension ".meta".
  - Definition file 2 of the neural network structure: The name of this file has the extension ".index".

In the case of Keras:

- Specify the folder including the HDF5 format trained model (file extension: *.h5) as [Input Model Location].

In the case of Caffe:

- Prepare the following two files as the result of training by Caffe in the Ubuntu operating environment. Store the files in the same folder.
  - Definition file of the neural network structure for inference: The name of this file has the extension ".prototxt". Note
  - Trained model: The name of this file has the extension ".caffemodel".

  Note: Use ".prototxt" file for inference. The ".prototxt" file for learning cannot be used.

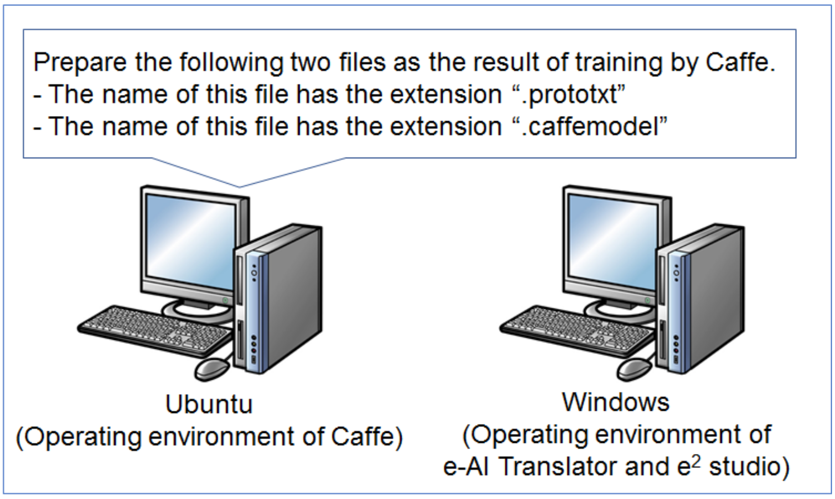

Prepare the following two files as the result of training by Caffe.
- The name of this file has the extension ".prototxt"
- The name of this file has the extension ".caffemodel"
Ubuntu
(Operating environment of Caffe)
Windows
(Operating environment of
e-AI Translator and e2 studio)

Figure 3.6 Preparing the Results of Training in the Ubuntu Environment

- Copy the following seven Python script files from the Windows environment to the Ubuntu environment.
  - os_invariant.pyc
  - os_invariant_caffemodel_generator.pyc
  - os_invariant_network_decoder_caffe.pyc
  - caffe_model_validation.pyc
  - error_code.pyc
  - os_invariant_caffemodel_generator.g
  - os_invariant_caffemodel_generator_h.g

  These script files are available in the following folder.
  C:\Renesas\e2_studio\eclipse\plugins\com.renasas.eaitranslator_1.4.0\Translator\os_invariant

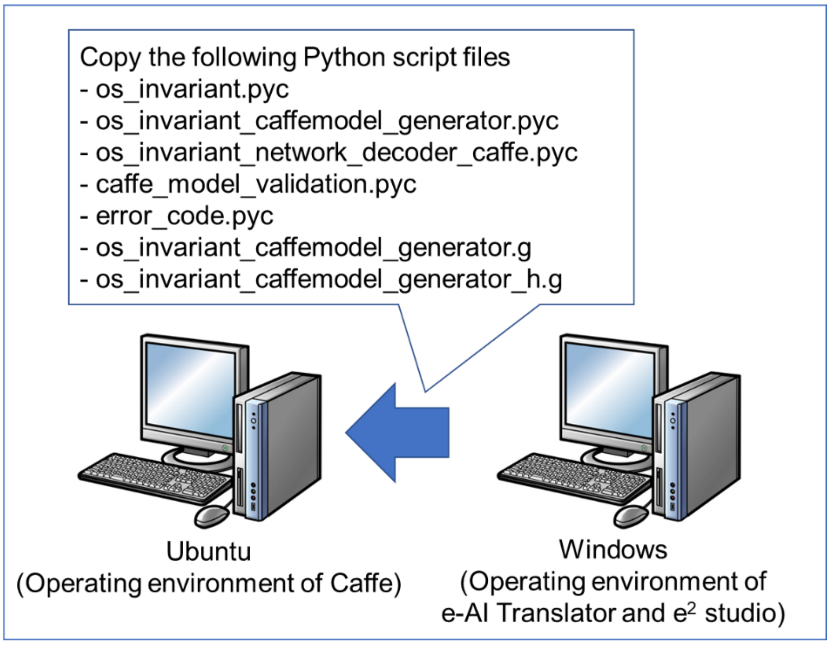

Copy the following Python script files
- os_invariant.pyc
- os_invariant_caffemodel_generator.pyc
- os_invariant_network_decoder_caffe.pyc
- caffe_model_validation.pyc
- error_code.pyc
- os_invariant_caffemodel_generator.g
- os_invariant_caffemodel_generator_h.g
Ubuntu
(Operating environment of Caffe)
Windows
(Operating environment of
e-AI Translator and e^2 studio)

Figure 3.7 Copying the Python Script Files

- With the Python script file "os_invariant.pyc" described above, convert the formats of "*.prototxt" and "*.caffemodel" files in Ubuntu. After this script file is executed, two files having the extension ".pickle" are generated.

  Example of usage: The "*.prototxt" and "*.caffemodel" are assumed to be stored in the "../model/caffe" folder in the Ubuntu environment.

  ```
  $ python os_invariant.pyc -mpath ../model/caffe
  ```

  Note: Execute these script files in the Python 2.7.6 environment.

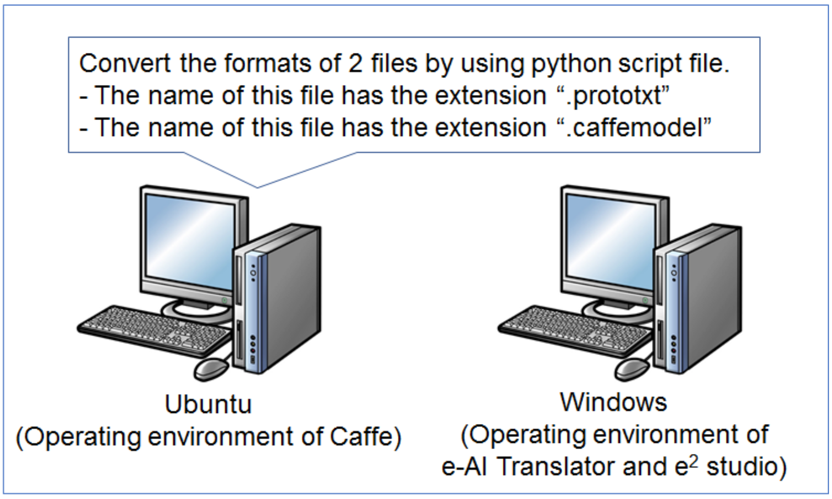

Convert the formats of 2 files by using python script file.
- The name of this file has the extension ".prototxt"
- The name of this file has the extension ".caffemodel"
Ubuntu
(Operating environment of Caffe)
Windows
(Operating environment of
e-AI Translator and e² studio)

Figure 3.8 Converting Files with the Python Script Files

- Copy the two generated files from the Ubuntu environment to the Windows environment.
- (One of these files has the extension ".pickle", another file has the extension "*.h5".)
  Store these two files in the same folder. This folder is specified as [Input Model Location].

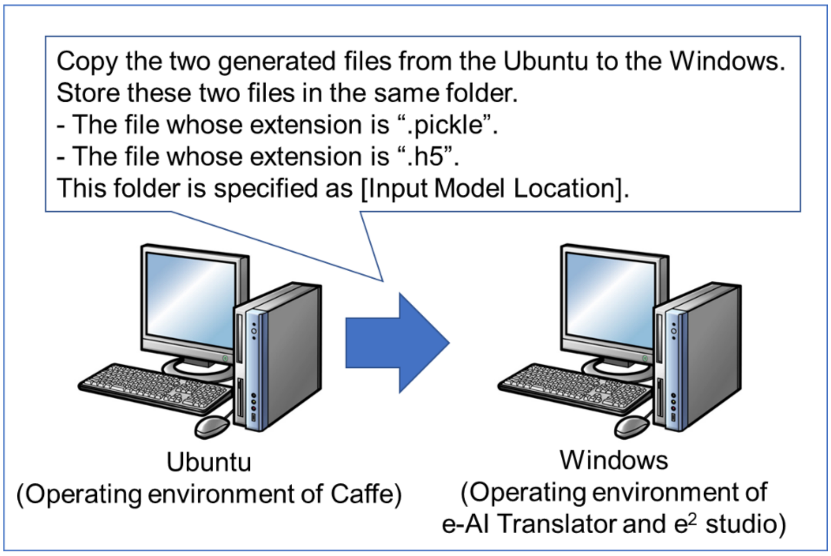
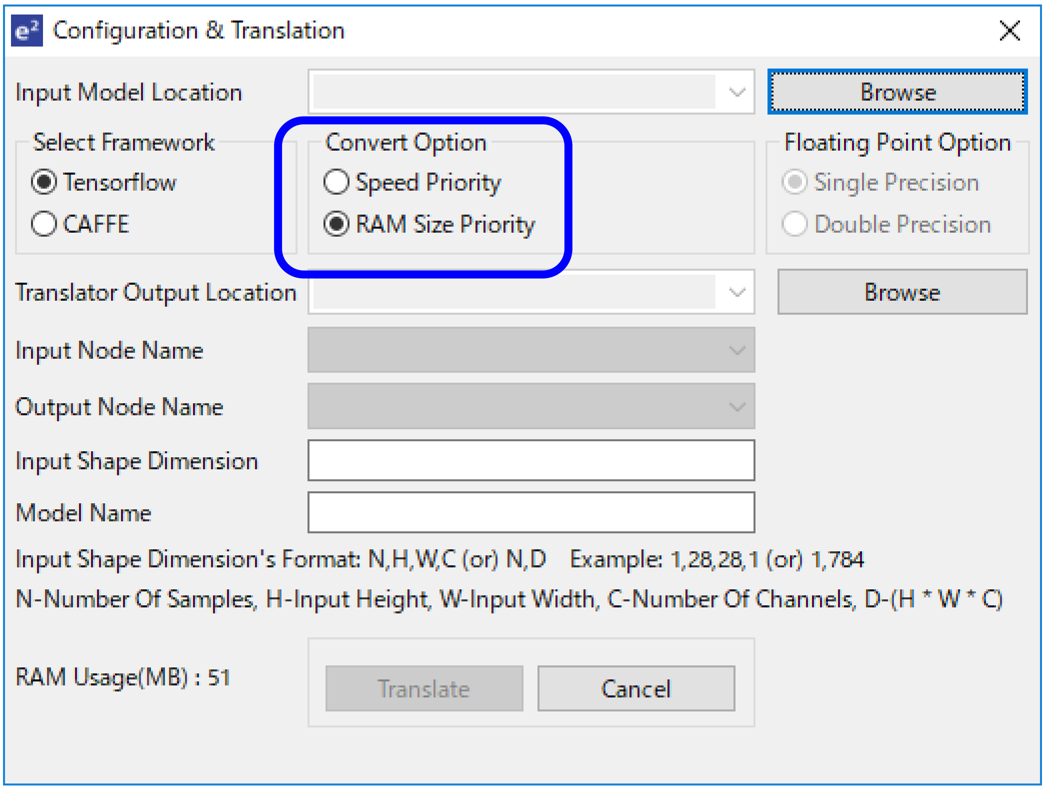

Copy the two generated files from the Ubuntu to the Windows.
Store these two files in the same folder.
- The file whose extension is ".pickle".
- The file whose extension is ".h5".
This folder is specified as [Input Model Location].
Ubuntu
(Operating environment of Caffe)
Windows
(Operating environment of
e-AI Translator and e² studio)

Figure 3.9 Copying Files and Specifying [Input Model Location]

6. Set [Convert option].
   Specify the quality to which priority is to be given when the trained model of the AI is converted.
   - Speed priority: The speed of execution is given priority in conversion to produce the program.
   - RAM size priority: Reduction of the RAM size is given priority in conversion to produce the program.

   Note that “Floating Point Option” is an option that specifies the floating-point precision (single or double) which is used for arithmetic processing by the neural network simulation.

   In the current version, this is fixed to single precision. Single precision is used in the converted product of training in the Deep-learning framework.



Configuration & Translation

Input Model Location

Select Framework
● Tensorflow
○ CAFFE

Convert Option
○ Speed Priority
● RAM Size Priority

Floating Point Option
● Single Precision
○ Double Precision

Translator Output Location

Input Node Name

Output Node Name

Input Shape Dimension

Model Name

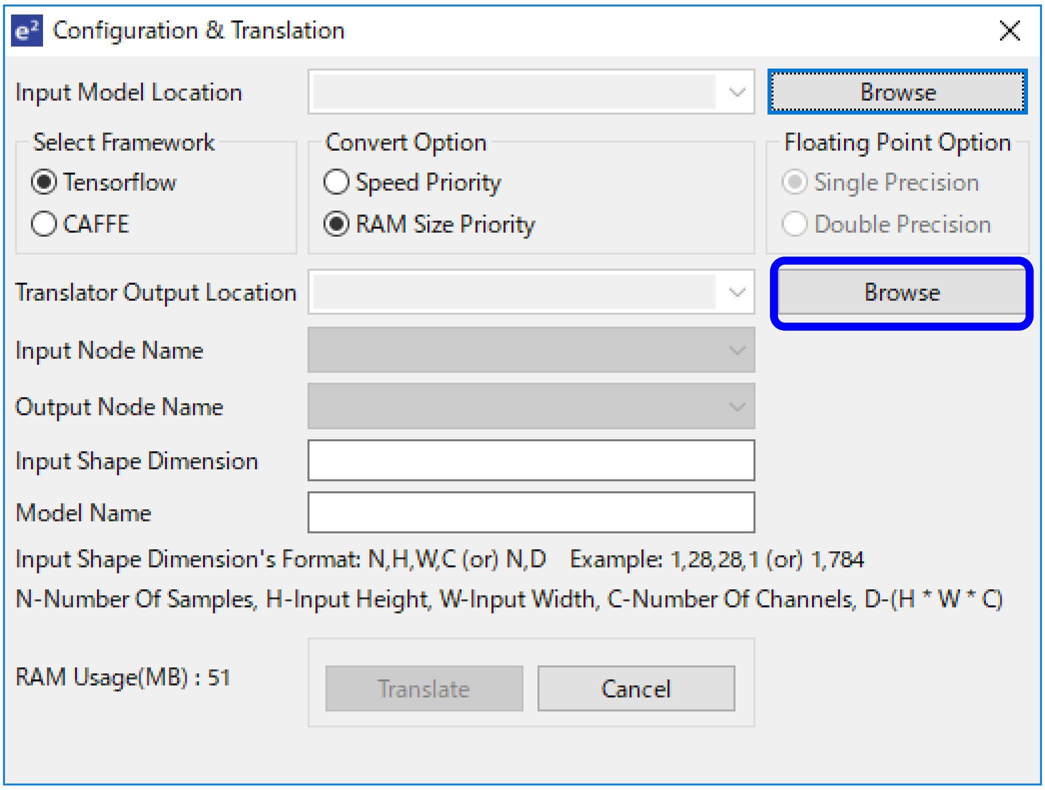
Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 51

Figure 3.10 Specifying [Convert option]

7. Click on the [Browse] button to set [Translator Output Location].
   Specify the destination folder for output of the source and header files from conversion.
   One-byte characters can be used for the folder name and path name. Do not use two-byte characters.
   Select the folder where the source files of the e^2 studio have been registered (e.g. the "src" folder in the project folder).



Configuration & Translation

Input Model Location

Select Framework

● Tensorflow

○ CAFFE

Convert Option

○ Speed Priority

● RAM Size Priority

Floating Point Option

○ Single Precision

○ Double Precision

Translator Output Location

Input Node Name

Output Node Name

Input Shape Dimension

Model Name

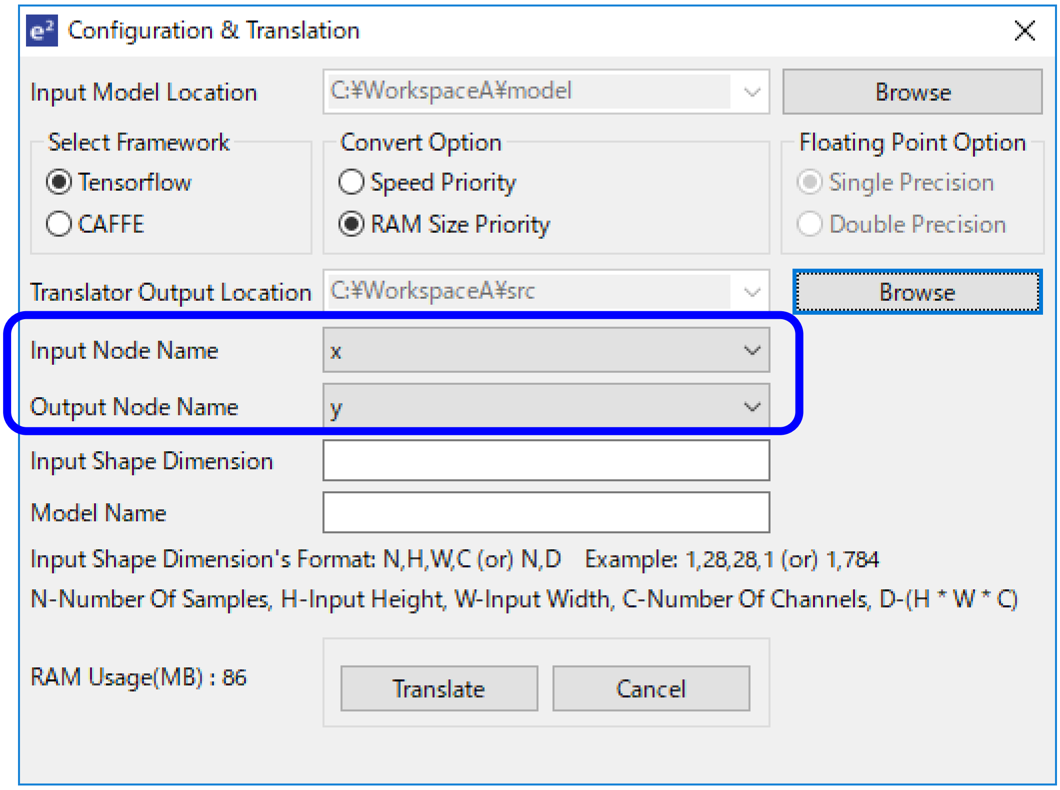
Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 51

Figure 3.11 Specifying [Translator Output Location]

8. Confirm [Input Node Name] and [Output Node Name]. (Only when using TensorFlow)
   Confirm whether [Input Node Name] and [Output Node Name] are correct or not.
   When they are not correct, specify the correct name from the pull-down list.



Configuration & Translation

Input Model Location: C:¥WorkspaceA¥model [Browse]

Select Framework: ● Tensorflow ○ CAFFE

Convert Option: ○ Speed Priority ● RAM Size Priority

Floating Point Option: ● Single Precision ○ Double Precision

Translator Output Location: C:¥WorkspaceA¥src [Browse]

Input Node Name: x

Output Node Name: y

Input Shape Dimension:

Model Name:

Input Shape Dimension's Format: N,H,W,C (or) N,D   Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 86

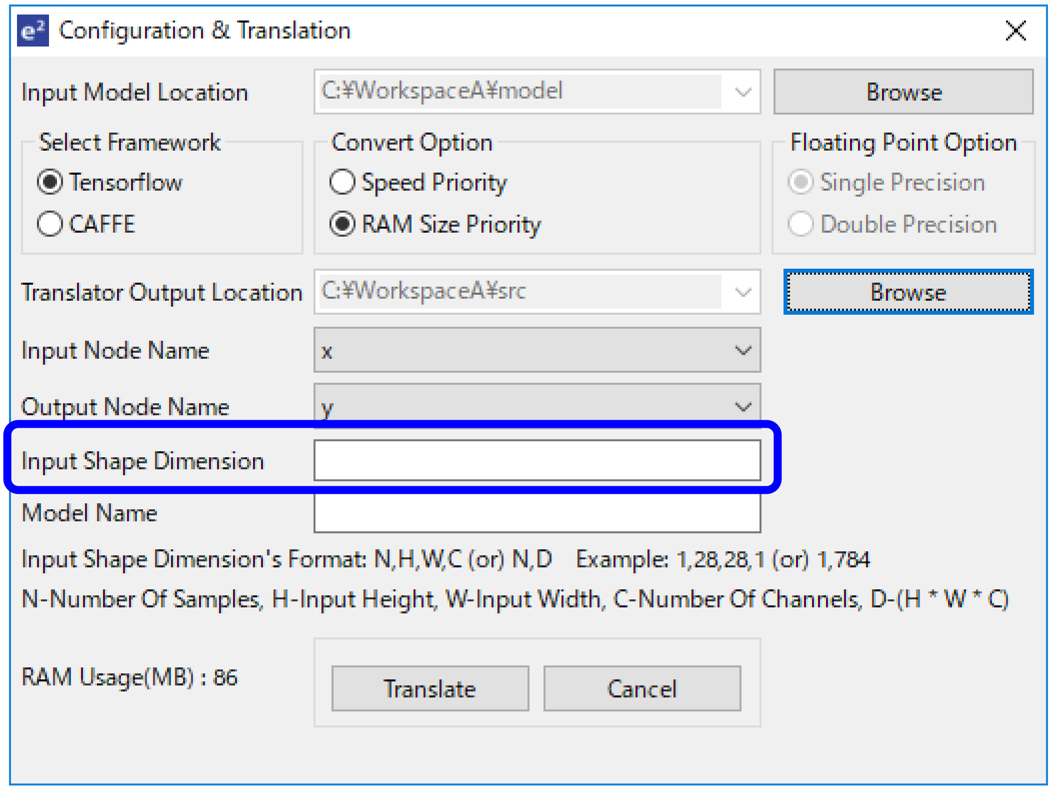
Figure 3.12 Confirm [Input Node Name] and [Output Node Name]

9. Set [Input Shape Dimensions] (only when using TensorFlow and Keras).

   Specify the size of the input layer of the neural network in the order of "N,H,W,C" or "N,D". This is only required when TensorFlow was used as the framework.

   Only numerical characters and comma (,) can be used for this input area, and input them as one-byte characters.

   N: Number of Samples; H: Input Height; W: Input Width; C: Number of Channels; D: H*W*C



Configuration & Translation

Input Model Location: C:¥WorkspaceA¥model

Select Framework: ● Tensorflow ○ CAFFE

Convert Option: ○ Speed Priority ● RAM Size Priority

Floating Point Option: ● Single Precision ○ Double Precision

Translator Output Location: C:¥WorkspaceA¥src

Input Node Name: x

Output Node Name: y

Input Shape Dimension:

Model Name:

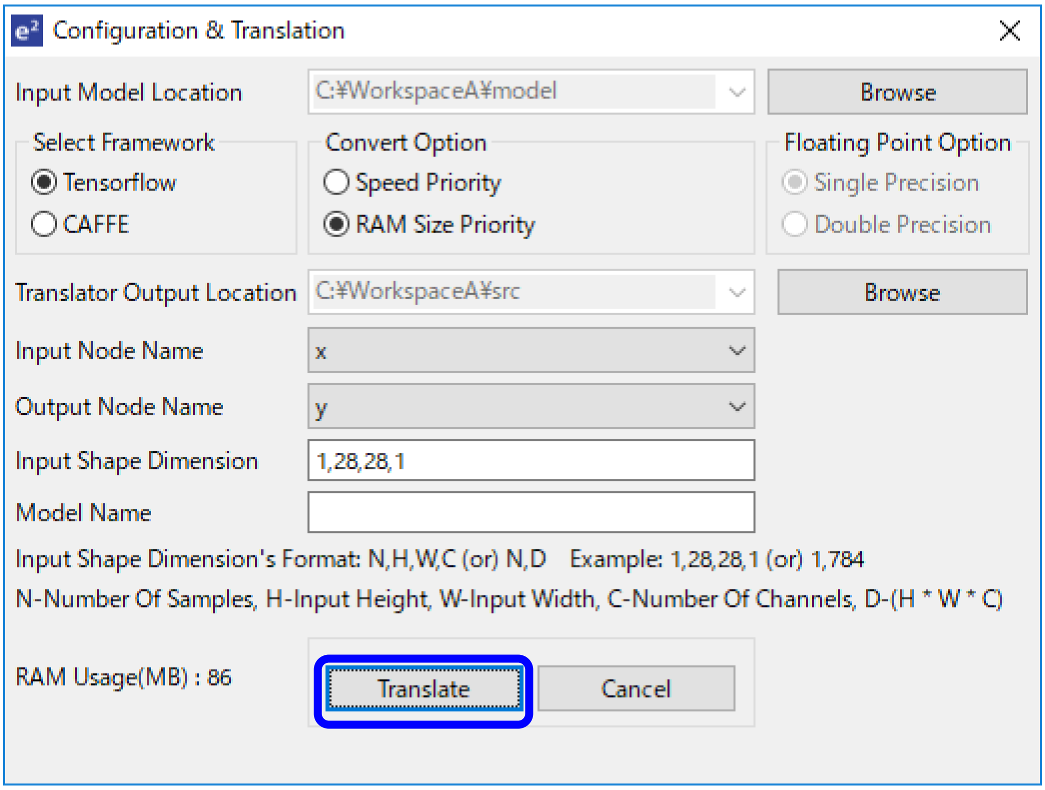
Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 86

Figure 3.13 Specifying [Input Shape Dimensions]

10. Click on the [Translate] button to convert the trained model into C-source and header files.
    Note: Need to set [Model Name] area only when using multiple neural networks.
    Be blank when using single neural network.



Configuration & Translation

Input Model Location: C:¥WorkspaceA¥model　Browse

Select Framework: ● Tensorflow ○ CAFFE

Convert Option: ○ Speed Priority ● RAM Size Priority

Floating Point Option: ● Single Precision ○ Double Precision

Translator Output Location: C:¥WorkspaceA¥src　Browse

Input Node Name: x

Output Node Name: y

Input Shape Dimension: 1,28,28,1

Model Name:

Input Shape Dimension's Format: N,H,W,C (or) N,D　Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 86

Figure 3.14　[Translate] Button

11. The converted C-source and header files are output to the "Translator" folder under the "src" folder. To build from the header files, add this folder to the include file path in the settings of the build tool. Call the "dnn_compute" function in the "dnn_compute.c" file to use inference processing of the AI.

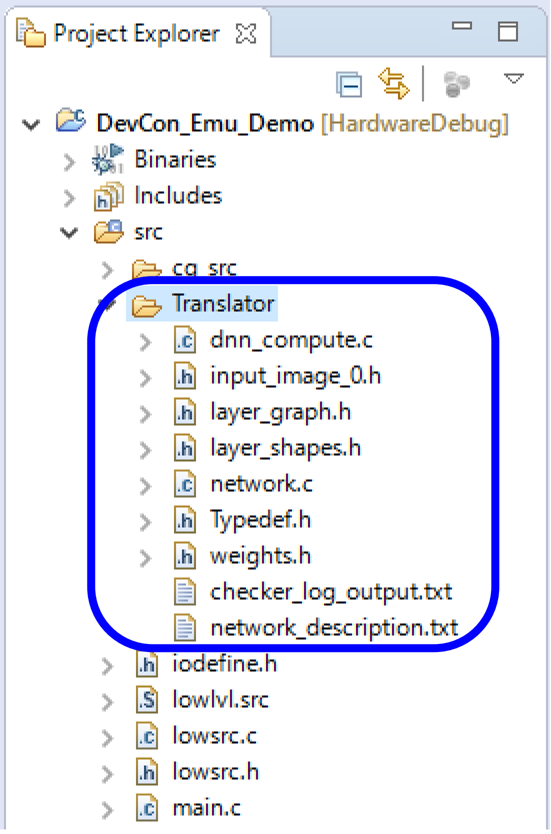


Figure 3.15 Output Folder and Files from Conversion

12. An outline of the results of conversion can be checked by reading the following two files in the "Translator" folder.
    - checker_log_output.txt: The ROM and RAM sizes to be used can be checked. note
      Also, the number of multiply-and-accumulate operations can be checked as information related to the speed of execution.
    - network_description.txt: Information on the converted neural network structure can be checked.

    Note: Estimation function of ROM size and RAM size are used by neural network.
    This file shows ROM size and RAM size when specifying "RAM size priority" as "convert option" in its current version. Even if "convert option" is changed, the estimation result is not changed.

13. Build the program which includes AI processing.
    Subsequently, the procedure for debugging the program is the same as that for programs in general.
    For the procedure for debugging the program through the e^2 studio, refer to "e^2 studio Integrated Development Environment User's Manual: Getting Started Guide" (document No.: R20UT4374EJ).

3.2. Translation of multiple neural networks

When using the multiple neural networks in single user program, the translation procedure is as follows.

In this chapte, these 2 neural networks are used as an example.

- Neural network for abnormal signal detection.
- Neural network for abnormal sound detection.

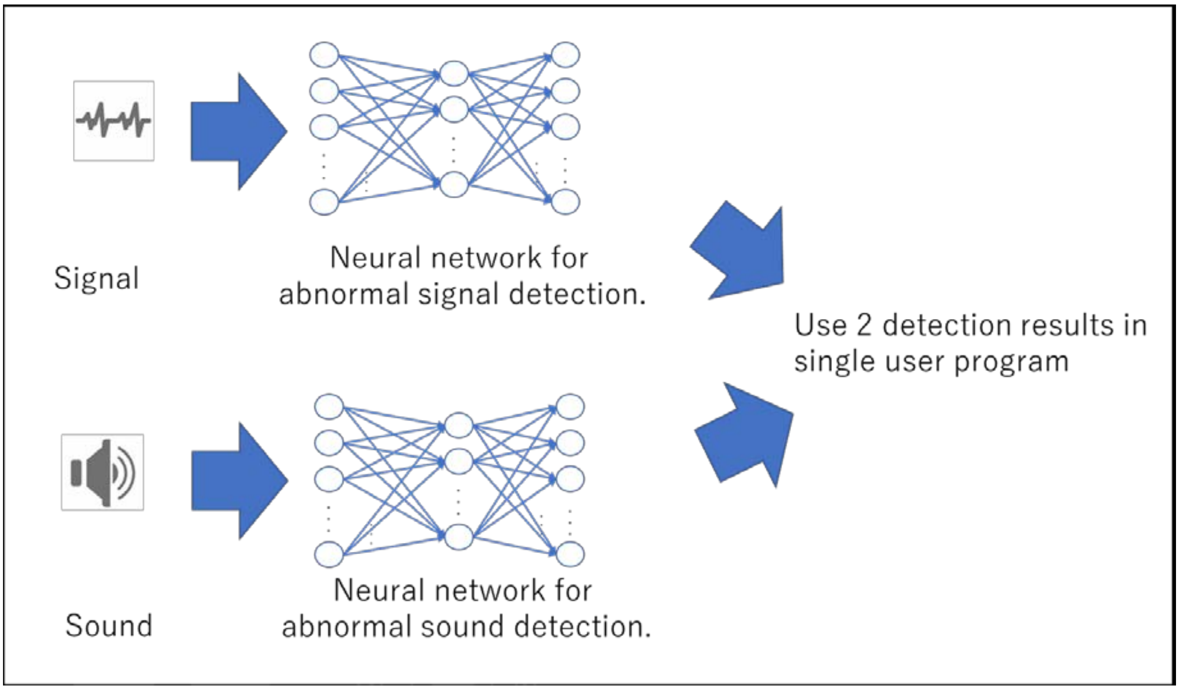
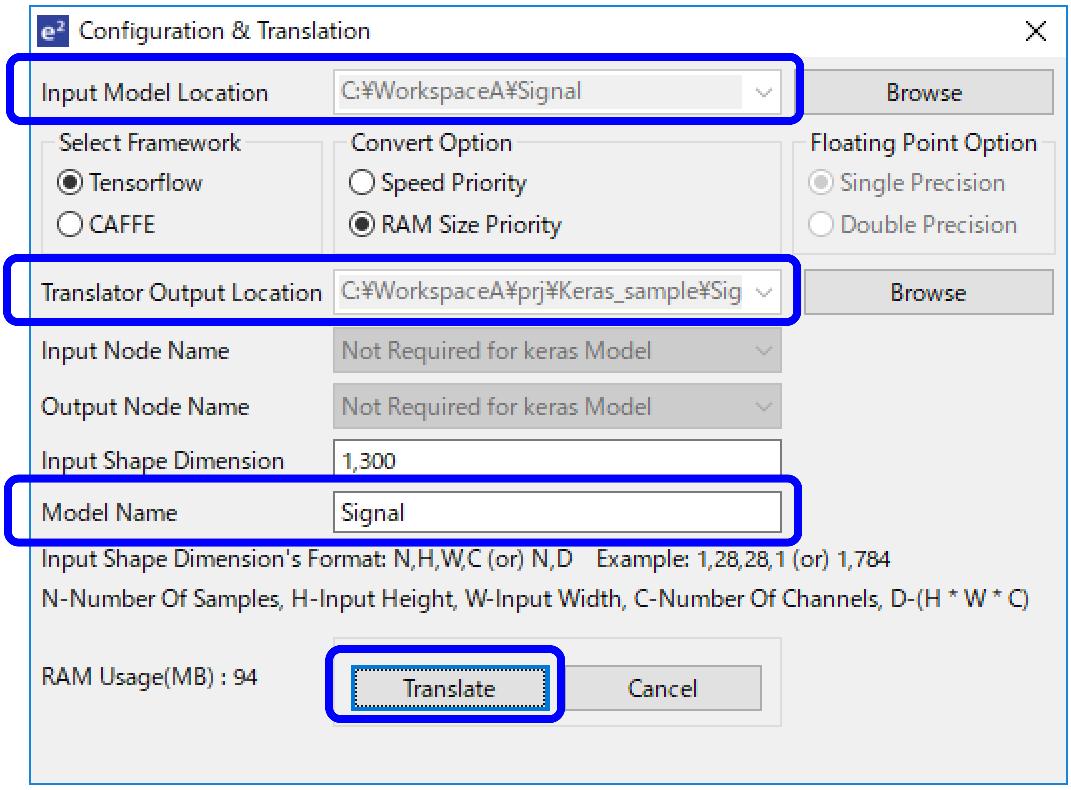

Signal
Neural network for abnormal signal detection.
Use 2 detection results in single user program
Sound
Neural network for abnormal sound detection.

Figure 3.16 Translation example of 2 neural networks

1. Translate the 1st neural network for abnormal signal detection.
   - Launch e-AI Translator, then specify the trained model of 1st neural network to [Input Model Location].
   - Specify different folder to [Translator Output Location] between 1st neural network and 2nd neural network.
   - Specify the 1st neural network name to [Model Name] for distinction of each neural network.
     In this case, "Signal" is used as the model name.
     Note: Only one-byte alphanumeric characters can be used, the maximum length is 8 characters.
   - After above settings, click [Translate] button for the translation.



Configuration & Translation

Input Model Location: C:¥WorkspaceA¥Signal

Select Framework: ● Tensorflow ○ CAFFE

Convert Option: ○ Speed Priority ● RAM Size Priority

Floating Point Option: ● Single Precision ○ Double Precision

Translator Output Location: C:¥WorkspaceA¥prj¥Keras_sample¥Sig

Input Node Name: Not Required for keras Model

Output Node Name: Not Required for keras Model

Input Shape Dimension: 1,300

Model Name: Signal

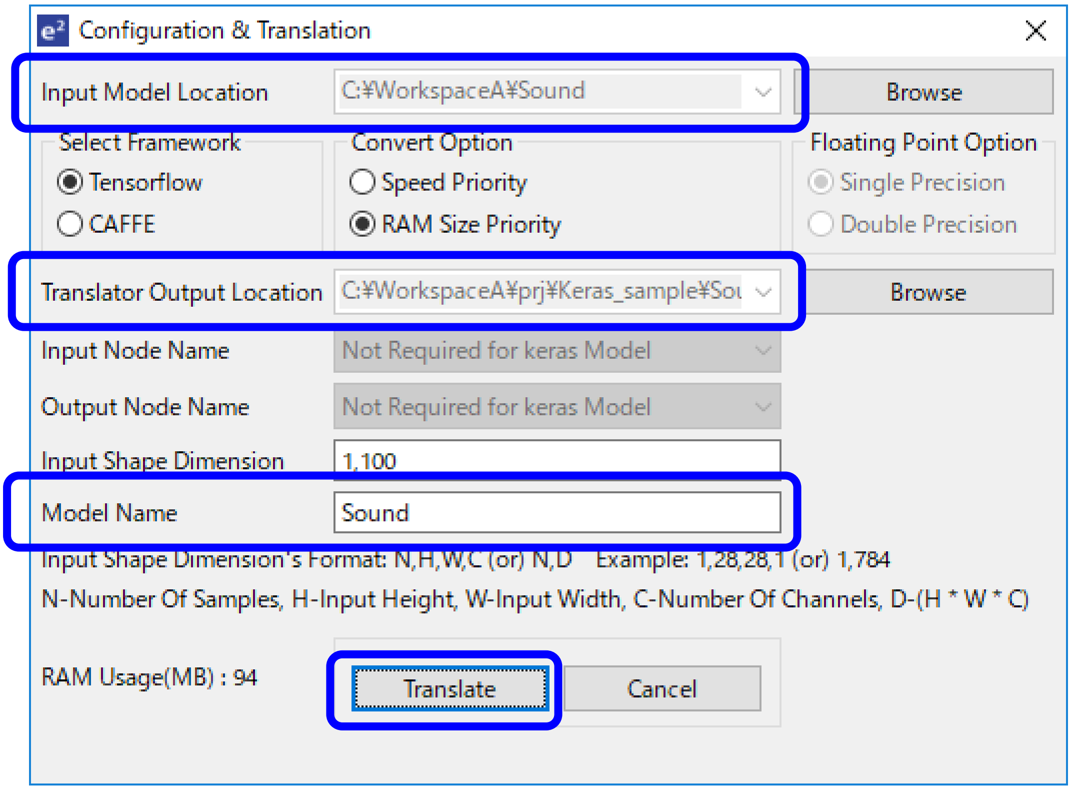
Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 94

Figure 3.17 Translation of 1st neural network for abnormal signal detection

2. Translate the 2nd neural network for abnormal sound detection.
   - Launch e-AI Translator, then specify the trained model of 2nd neural network to [Input Model Location].
   - Specify different folder to [Translator Output Location] between 1st neural network and 2nd neural network.
   - Specify the 2nd neural network name to [Model Name] for distinction of each neural network.
     In this case, "Sound" is used as the model name.
     Note: Only one-byte alphanumeric characters can be used, the maximum length is 8 characters.
   - After above settings, click [Translate] button for the translation.



Configuration & Translation

Input Model Location: C:¥WorkspaceA¥Sound

Select Framework: ● Tensorflow ○ CAFFE

Convert Option: ○ Speed Priority ● RAM Size Priority

Floating Point Option: ● Single Precision ○ Double Precision

Translator Output Location: C:¥WorkspaceA¥prj¥Keras_sample¥Sou

Input Node Name: Not Required for keras Model

Output Node Name: Not Required for keras Model

Input Shape Dimension: 1,100

Model Name: Sound

Input Shape Dimension's Format: N,H,W,C (or) N,D Example: 1,28,28,1 (or) 1,784

N-Number Of Samples, H-Input Height, W-Input Width, C-Number Of Channels, D-(H * W * C)

RAM Usage(MB) : 94

Figure 3.18 Translation of 2nd neural network for abnormal sound detection

3. Confirm the translated source files.

   Source files and header files are translated with the Model Name. However, "typedef.h" and "imput_image_0.h" don't have the Model Name, because they are same files.

   For using the inference function, call the following functions/

   - Inference function for signal: dnn_compute_Signal() function in "dnn_compute_Signal.c"
   - Inference function for sound: dnn_compute_Sound() function in "dnn_compute_Sound.c"

Table 3-2 Translated source files and header files

	1st neural network for abnormal signal detection	2nd neural network for abnormal sound detection
Translation Result	dnn_compute_Signal.c	dnn_compute_Sound.c
	layer_graph_Signal.h	layer_graph_Sound.h
	layer_shapes_Signal.h	layer_shapes_Sound.h
	network_Signal.c	network_Sound.c
	weights_Signal.h	weights_Sound.h
	Typedef.h	Typedef.h
	input_image_0.h	input_image_0.h

4. Marge "network_Signal.c" and "network_Sound.c".

   The functions and variables for each neural network layers are defined in these 2 files.

   When using the same layer in these 2 neural networks, build error occurs because of multiple definition of functions and variables.

   Therefore, these multiple definitions need to be commented out.

5. Build user program.

   Multiple neural networks can be used in single user program by this procedure.

3.3. Sample of a Trained Model and Sample Code of the Main Function

The installation folder of the e-AI Translator contains samples of a trained model and the main function. Use these samples as required.

[Stored directory]

```
C:\Renesas\e2_studio\eclipse\plugins\com.renasas.eaitranslator_1.4.0\Translator
   |
   |--- main.c: Sample code of a main function (calling of the functions for inference processing)
   |--- input: Samples of input characters 0 to 9; header file format
   |
   --- model
        |
        |--- caffe: Trained model of the AI using Caffe (before conversion of the file format)
        |--- caffe_windows: Trained model of the AI using Caffe (after conversion of the file format)
        ---- tensorflow: Trained model of the AI using TensorFlow
```

Handwriting recognition processing for Modified National Institute of Standards and Technology (MNIST) is provided as the sample program.

- A handwritten character from 0 to 9 is entered to the neural network simulation and the program from the e-AI Translator makes an inference as to which number has been entered.
- The size of the entered numerals is 28 × 28 pixels, and they are grayscale.
- The e-AI Translator outputs the probabilities of which number has been entered.

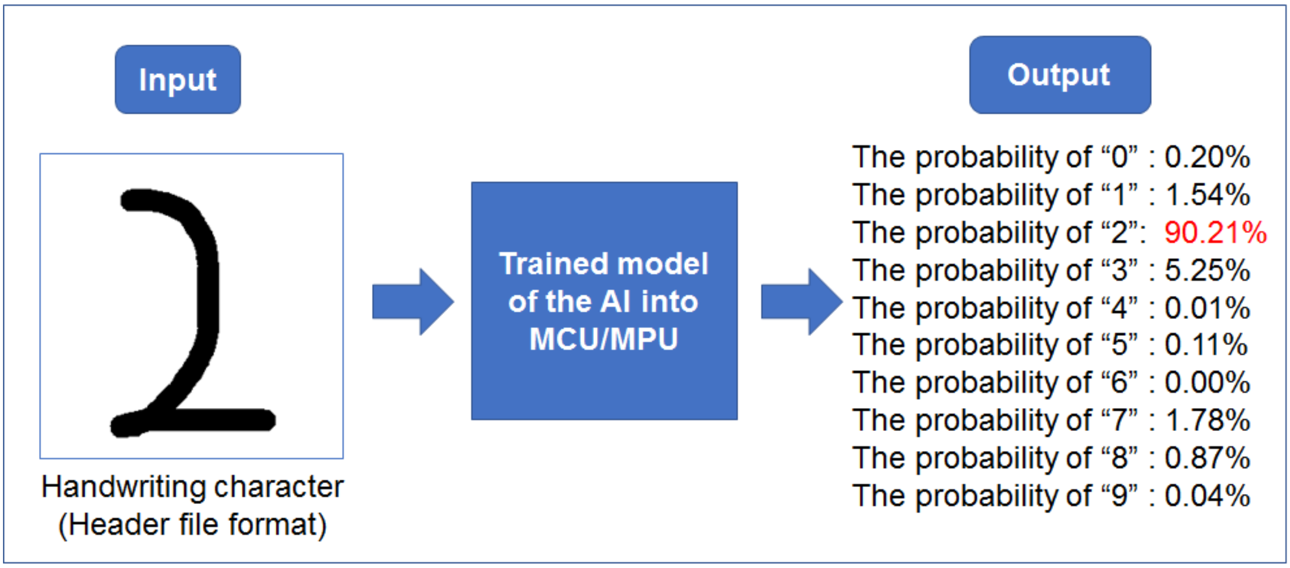

Input
Handwriting character
(Header file format)
Trained model of the AI into MCU/MPU
Output
The probability of "0" : 0.20%
The probability of "1" : 1.54%
The probability of "2": 90.21%
The probability of "3" : 5.25%
The probability of "4" : 0.01%
The probability of "5" : 0.11%
The probability of "6" : 0.00%
The probability of "7" : 1.78%
The probability of "8" : 0.87%
The probability of "9" : 0.04%

Figure 3.19 Image of Behavior of the Sample

4. Points for Caution

- The non-supported types of neural network cannot be correctly converted.

  For the supported types of neural network, refer to section 1.2, Types of Convertible Neural Networks, and section 1.3, Supported API list.

- When TensorFlow is used for the training, do not use "tf.name_scope".

  If "tf.name_scope" is used, the e-AI Translator may not correctly convert the neural network.

- When TensorFlow is used for the training, save trained model after finishing final epoch.

  (Final epoch: final loop of training)

- When using preprocessing such as Reshape and so on for the input layer of neural network, this preprocessing is removed in the inference function of e-AI Translator. Therefore, use the data shape after preprocessing for the input of the inference function.

  And, when using the 4D shape input data for the inference function, the input data order of the inference function needs to be [N, C, H, W] format. In the training phase of TensorFlow and Keras, the 4D shape input data order needs to be [N, H, W, C] format. Be careful the difference of 4D shape input data order.

  N: Number of Samples, H: Input Height, W: Input Width, C: Number of Channels

-

- The neural networks including these configurations cannot be translated.
  - The input layer is activation function (relu, tanh sigmoid and so on).
  - The input layer is normalization layer (Batch Normalization, LRN).
  - The input layer is pooling layer (Max Pooling, Average pooling).

- When Caffe is used for the training, do not use "Reshape" and "Flatten" layer. The translation error occurs.

  When connecting from 4D shape output layer (Convolution, Pooling, etc) to 2D shape input layer (Fully connected, etc), connect these layers without "Reshape" and "Flatten". The shape of data is converted automatically by Caffe.

- When using Batch Normalization layer, standalone Keras or TensorFlow-backend Keras or Caffe is recommended as a framework for training. When using TensorFlow, the following procedure is required to create the trained model for inference operation, the procedure is complex.

  [Procedure to create the trained model for inference operation]

  1.Set the attribute “training” of API “tf.layers.batch_normalization” to “true”, then build computation graph.

  2.When starting the training session, specify attribute "extra_update_ops" as following example.

  Note: Other options for training session are omitted in the example.

  Example:

  ```
  extra_update_ops = tf.get_collection(tf.GraphKeys.UPDATE_OPS)
  sess.run(extra_update_ops)
  ```

  3.Save the trained model after the training session.

  4.Set the attribute “training” of API “tf.layers.batch_normalization” to “false”, then build computation graph which is same as procedure1 again.

  5.Restore the trained model created by procedure2 to the computation graph built by procedure4.

  6.Save the model. Don’t train the model in this procedure.

  7.Launch e-AI Translator, then translate the model saved by procedure6.

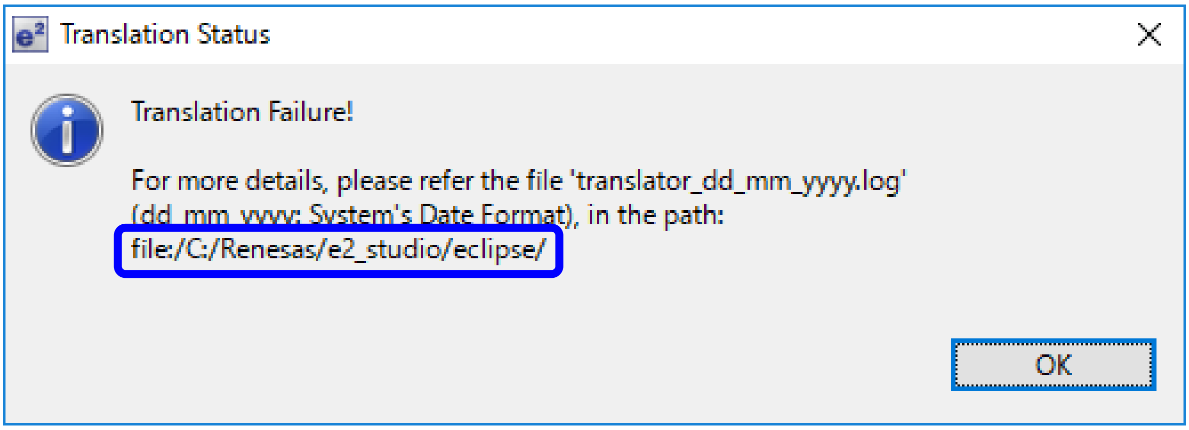
5. Error message

This chapter explains error message of e-AI Translator and how to solve each error.

When e-AI Translator generates the error, following dialog is displayed.

The log file shows error number "eAI-xxx", and is generated in the folder which is displayed in error dialog.

When the error occurs, confirm the solution from this error number.

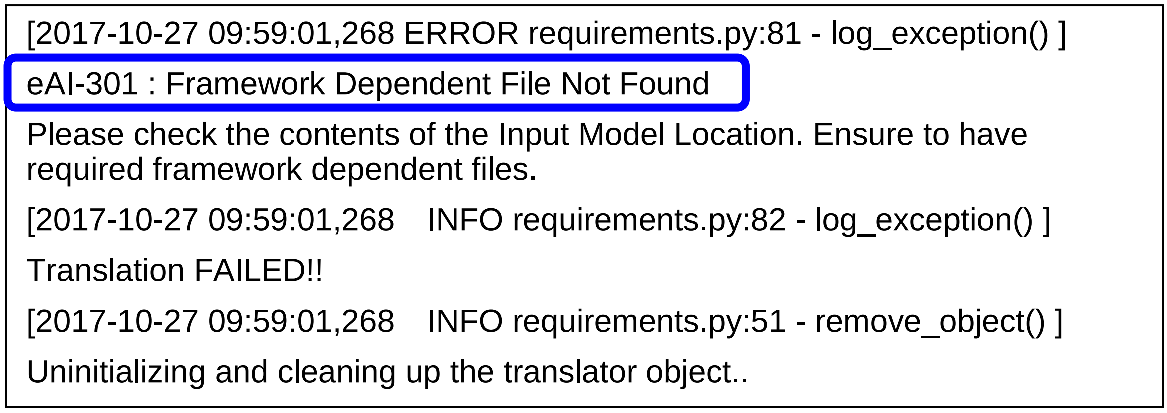


Translation Status

Translation Failure!

For more details, please refer the file 'translator_dd_mm_yyyy.log' (dd_mm_yyyy: System's Date Format), in the path: file:/C:/Renesas/e2_studio/eclipse/

Figure 5.1 Folder of log file in error dialog



```
[2017-10-27 09:59:01,268 ERROR requirements.py:81 - log_exception() ]
eAI-301 : Framework Dependent File Not Found
Please check the contents of the Input Model Location. Ensure to have
required framework dependent files.
[2017-10-27 09:59:01,268    INFO requirements.py:82 - log_exception() ]
Translation FAILED!!
[2017-10-27 09:59:01,268    INFO requirements.py:51 - remove_object() ]
Uninitializing and cleaning up the translator object..
```

Figure 5.2 Error number in the log file

Table 5-1 List of error numbers and solutions (1/6)

Error category	Error number	Error message and the solution
About setup	eAI-101	Python not installed. The error when Python 3.5.3 is not installed. Install Python 3.5.3 by referring to “2.3. Installing Python3.5.3”.
	eAI-102	64bits Python version is not installed. The error when Python 3.5.3 is not 64bit version but 32bit version. Install 64bit version of Python 3.5.3, and confirm whether Windows on PC is 64bit or not.
	eAI-103	“tensorflow” package is not installed. The error when tensorflow package is not installed. Install tensorflow package by referring to “2.4. Installing the required Python Packages”.
	eAI-104	“progressbar” package is not installed. The error when progressbar package is not installed. Install progressbar package by referring to “2.4. Installing the required Python Packages”.
	eAI-105	“prettytable” package is not installed. The error when prettytable package is not installed. Install prettytable package by referring to “2.4. Installing the required Python Packages”.

Table 5-2 List of error numbers and solutions (2/6)

Error category	Error number	Error message and the solution
About setup	eAI-107	“h5py” package is not installed The error when h5py package is not installed. Install h5py package by referring to “2.4. Installing the required Python Packages”.
	eAI-108	“pycrypto” package is not installed The error when pycryptodome package is not installed. Install pycryptdome package by referring to “2.4. Installing the required Python Packages”.
	eAI-110	“configparser” package is not installed The error when configparser package is not installed. Install configparser package by referring to “2.4. Installing the required Python Packages”.
	eAI-111	“psutil” package is not installed The error when psutil package is not installed. Install psutil package by referring to “2.4. Installing the required Python Packages”.
	eAI-112	"Keras" package is not installed The error when Keras package is not installed. Install Keras package by referring to “2.4. Installing the required Python Packages”. However, when using TensorFlow-backend Keras, confirm the installation of TensorFlow correctly.

Table 5-3 List of error numbers and solutions (3/6)

Error category	Error number	Error message and the solution
About setting value	eAI-206	Invalid Input Data-file Name. The error when files in the folder as "Input Model Location" have illegal file extension. Confirm whether specified folder is correct or not.
	eAI-208	Invalid Input Shape Dimensions. The error when the specified characters to "Input Shape Dimensions" cannot be used. Only numerical characters and comma (,) can be used for this input area, and input them as one-byte characters.
	eAI-209	Input Shape Dimensions Not Found. The error when "Input Shape Dimensions" is not specified. Specify the input layer size of neural network to "Input Shape Dimensions".

	eAI-210	Incorrect Input Shape Dimensions.
		The error when the setting value of "Input Shape Dimensions" is not correct. Specify the size of the input layer of the neural network in the order of "N,H,W,C" or "N,D".
		N: Number of Samples; H: Input Height; W: Input Width;
		C: Number of Channels; D: H*W*C;

	eAI-211	Incorrect Input/output node name The error when using TensorFlow, input node name or output node name is not correct. Confirm the node name in "Input Node Name" area and "Output Node Name" area.
	eAI-212	Model name error The error when specifying incorrect "Model Name". Confirm to fullfill these 2 conditions for the model name. - Specify only one-byte alphanumeric characters. - Specify within 8 charactors.

Table 5-4 List of error numbers and solutions (4/6)

Error category	Error number	Error message and the solution
About framework	eAI-301	Framework Dependent File Not Found. The error when the files in the folder specified by “Input Model Location” are not found. Confirm whether the specified folder is correct or not, and whether all required files are existing or not by referring to procedure 4 of “3.1. Basic Procedure for Using the e-AI Translator”.
	eAI-302	Caffe Network Configuration Mismatch. The error when file extensions are not ”*.caffemodel” and ”*.prototxt” as trained model of Caffe. Confirm whether the specified folder is correct or not, and whether the file extension is correct or not. This error is not output from e-AI Translator, but is output from Python script files used to convert file format from Ubuntu environment to Windows environment.
	eAI-303	Memory Error in Tensorflow. The error when the memory size of PC is not enough. Close PC application software, then try to use e-AI Translator again.
	eAI-304	Error in importing frozen graph. The error when e-AI Translator cannot get required parameters from trained model. If this error occurs, contact Renesas.
	eAI-305	Unsupported network. The error when using unsupported neural network. Confirm whether unsupported neural network is used or not by referring to “1.2. Types of Convertible Neural Networks” and ”1.3. Supported API / unsupported API list”.

Table 5-5 List of error numbers and solutions (5/6)

Error category	Error number	Error message and the solution
About framework	eAI-306	Unsupported Algorithm/Model The error when using unsupported model. Confirm whether unsupported layers are included or not in the trained model.
	eAI-307	Tensorflow Version mismatch The error when using the trained model which is created by unsupported version of TensorFlow. The support version of TensorFlow is 1.12.0 or earlier.
	eAI-308	Keras model file versions mismatch The error when using the trained model which is created by unsupported version of Keras. Confirm the version as follows. - Standalone Keras: 2.2.4 - TensorFlow-backend Keras: 2.1.6-tf (Included in TensorFlow 1.12.0)
	eAI-309	Multiple Bias Error The error when using multiple bias addition in the trained model. Confirm whether multiple bias addition is existed or not in the training script. Example: The case to add bias to the output of "tf.layers.conv2d" API. ("tf.layers.conv2d" API has bias addition operation, therefore bias addition after the API is not required.)

Table 5-6 List of error numbers and solutions (6/6)

Error category	Error number	Error message and the solution
About accessing files	eAI-401	File Open Error. The error when the files in the folder specified by "Input Model Location" cannot be opened. Confirm whether the folder access setting is inhibited or not, and whether other PC application software is using these files or not.
	eAI-402	File Creation Error. The error when the file creation is failed to the folder specified by "Translator Output Location". Confirm whether the folder write access setting is inhibited or not.
	eAI-403	File Overwrite Error. The error when overwriting the files in the folder specified by "Translator Output Location" is failed. Confirm whether the folder write access setting is inhibited or not, and whether other PC application software is using these files or not.
	eAI-404	File Not Found. The error when the files are not found in the folder specified by "Input Model Location". Confirm whether specified folder is correct or not.
Others	eAI-501	Uncaught Exception. The error which cannot be classified. If this error occurs, contact Renesas.

Revision History

Table 6-1 Revision History (1/2)

Rev.	Date	Description	
		Page	Summary
1.01	Sep 8, 2017	—	First edition issued
2.00	Nov 6, 2017	Overall	Change of the version of e-AI Translator from V1.0.0 to V1.0.1. Change of the GUI images and the descriptions of installation folder by this version up.
		P5	Addition of “Microsoft Visual C++ 2015 Redistributable” as required software to be installed.
		P7, P8	Addition of “1.3. Supported API / unsupported API list”.
		P9	Addition of “1.4. Changes of e-AI Translator from V1.0.0 to V1.0.1”
		P12, P13	Addition of the confirmation after installing Python.
		P15, P16, P21	Modification of Python version from 2.7 to 2.7.6 for installing Ubuntu environment.
		P19	Addition of character information which can be used for the folder name and path name as “Input Model Location”.
		P20, P21	Change of the procedure to convert the trained model from Ubuntu environment to Windows environment.
		P23	Addition of character information which can be used for the folder name and path name as “Translator Output Location”.
		P24	Addition of character information which can be used for “Input Shape Dimensions”. Addition of input format expansion for “Input Shape Dimensions”.
		P27	Addition of 1 caution.
		P28 - P31	Addition of “5. Error message”.
2.01	May 15, 2018	P5, P14	Addition of description about TensorFlow version to install.
		P20	Addition of description about “.prototxt” file when using Caffe.
3.00	July 24, 2018	Overall	Change of the version of e-AI Translator from V1.0.1 to V1.0.2. Change of the GUI images and the descriptions of installation folder by this version up.
		P5, P9, P14	Modification about supporting of TensorFlow version 1.8.0.
		P6, P7, P8	Modification about supporting of LRN (Local Response Normalization).
		P9	Addition about changes of e-AI Translator V1.0.2.
		P18	Addition of “RAM Usage(MB)” explanation.
		P21, P22	Modification about specification change of Python script files which are used when converting file format from Ubuntu environment to Windows environment.

Table 6-2 Revision History (2/2)

Rev.	Date	Description	
		Page	Summary
4.00	Feb 3, 2020	Overall	Change of the version of e-AI Translator from V1.0.2 to V1.4.0. Change of the GUI images and the descriptions of installation folder by this version up.
		P5, P7, P8, P21	Addition about supporting Keras framework.
		P6, P8	Addition about supporting of Batch Normalization.
		P14	Addition of required software, modification about the version of software.
		P9	Addition about changes of e-AI Translator V1.4.0.
		P15-P16	Modification of the explanation about Caffe installation, because Ubuntu14.04 is end of life.
		P31-P34	Addition about the procedure to use multiple neural networks.
		P36-P37	Addition / remove of cautions.
		P38-P44	Addition of error numbers and messages.
4.01	Mar 10, 2020	P5, P14	Correction of errors

e-AI Translator V1.4.0 User’s Manual

Publication Date: Rev.1.01 Sep. 08, 17
Rev.4.01 Mar. 10, 20

Published by: Renesas Electronics Corporation

e-AI Translator V1.4.0

Renesas Electronics Corporation