Customization Guide

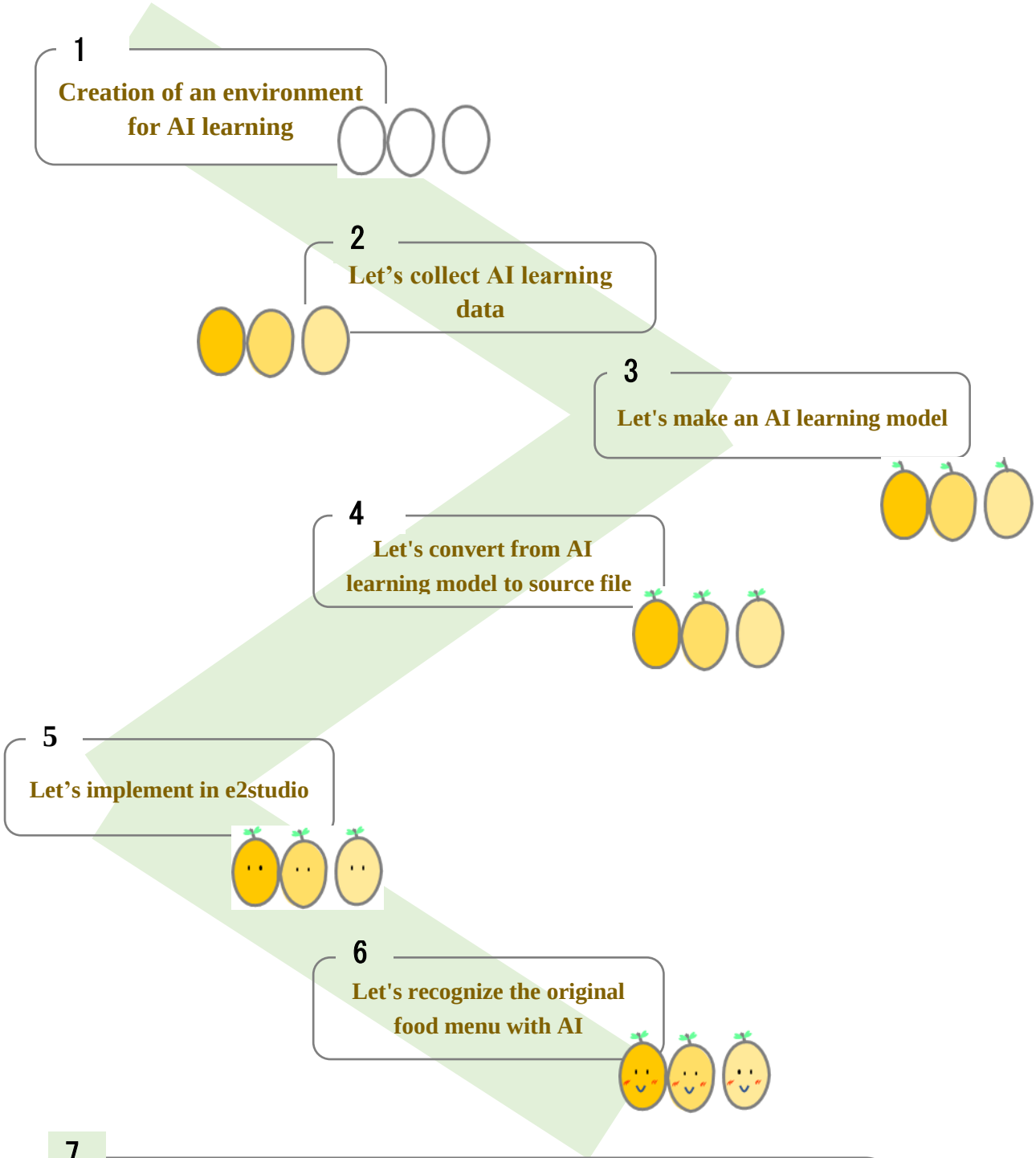
GR-MANGO
AI Customization Guide

EPMP-IMB-20-0265-01 2020.12

目次

1. Introduction

This document describes the procedure for **changing to your favorite food menu** on GR-MANGO and recognizing the menu with AI.



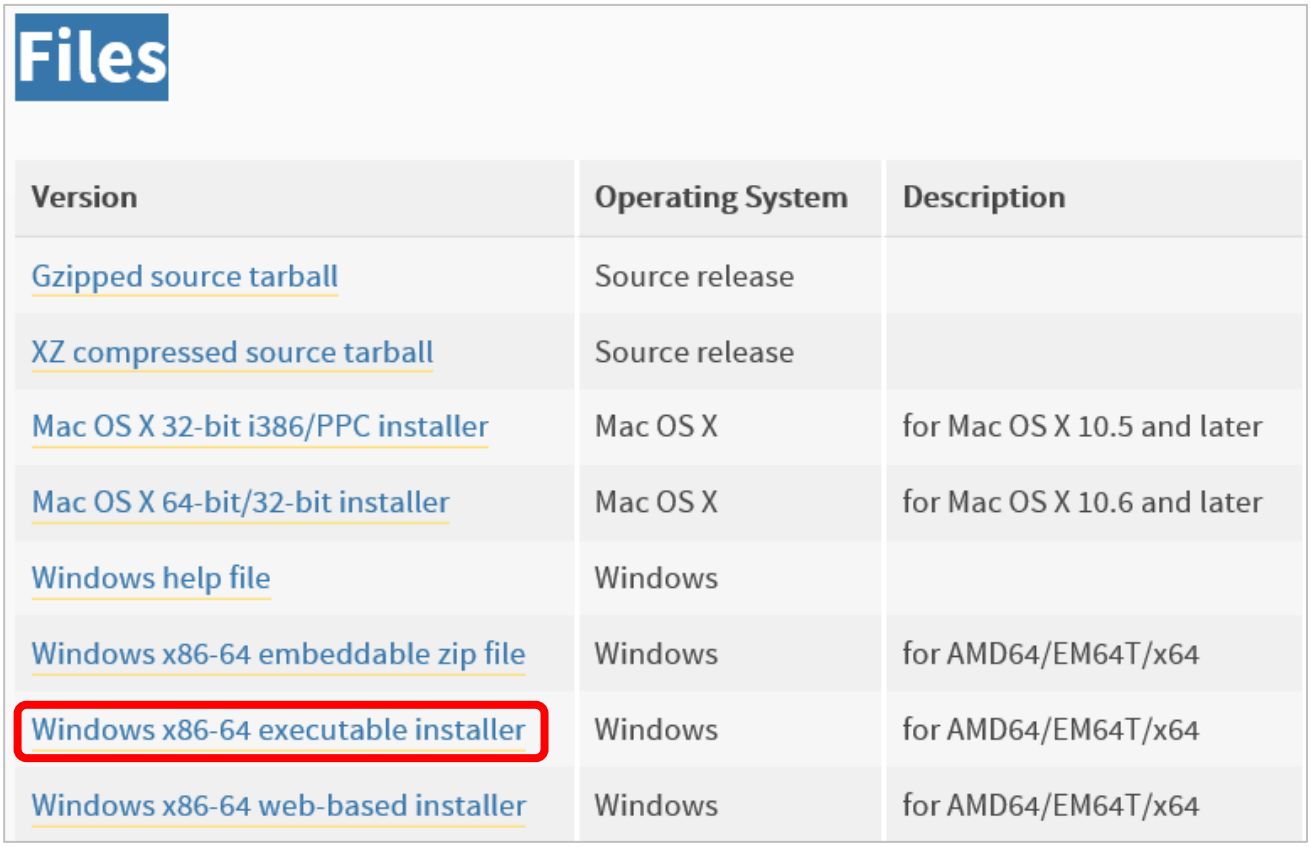
2. Creation of an environment for AI learning

2.1 Installing an environment for AI learning

*If you have done "e-AI starting package", please read only "2.1.2(1)Installing tensorflow" and confirm that only "2.1.2(7)Version confirmation".

2.1.1 Installing Python

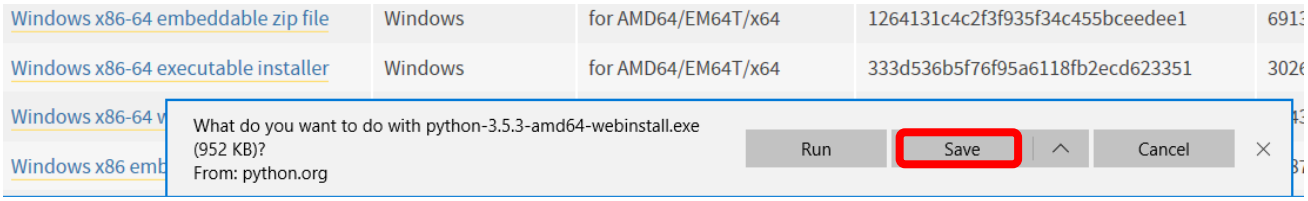
Download Python3.5.3 from the following website.
https://www.python.org/downloads/release/python-353/
Scroll down the screen and click "Files" in "Windows x86-64 executable installer".



Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
Mac OS X 32-bit i386/PPC installer	Mac OS X	for Mac OS X 10.5 and later
Mac OS X 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64

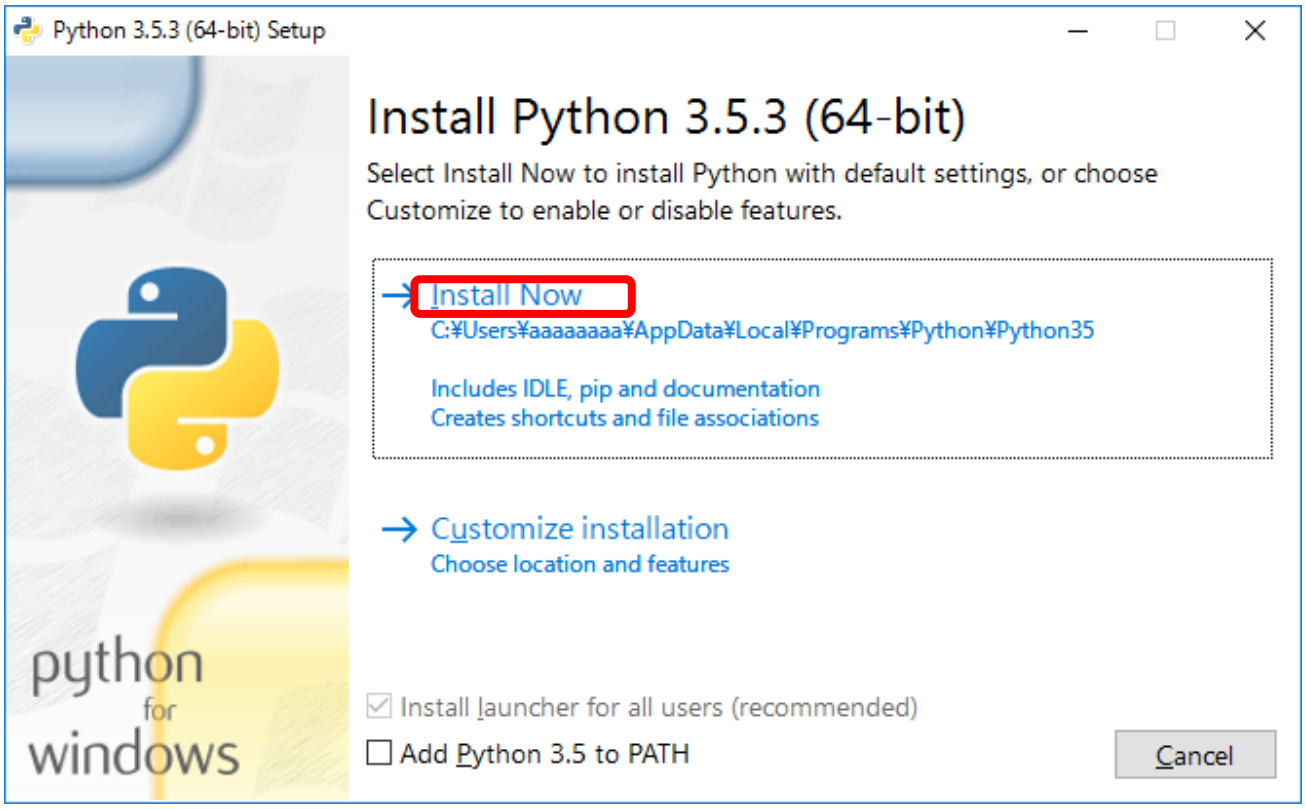
Click save button.



Double-click the installer, start the installation.
Double-click "Install Now". After that, proceed with the installation according to the instructions of the installer.

Note 1: Execute the setting of "Add Python 3.5 to PATH" without checking, because the version is setting when

executing the command.
Note 2: Read "Lisence.txt" in the installation folder for the license of Python.
[Install folder of Python]
C:/Users/<windows-user-name>/AppData/Local/Programs/Python/Python35/Scripts
* "<windows-user-name>" : set the user name for logging on to Windows.



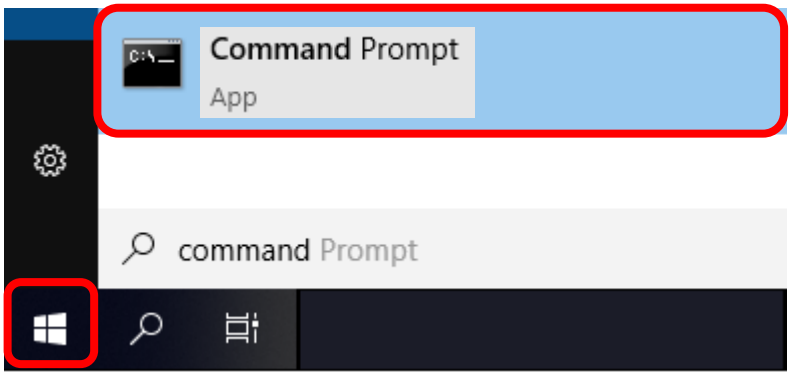
After installation, confirm the installation was successful following the step.
Excute the command prompt.
Excute the following command to confirm the version.
The following command can be used by copy and paste.

```
py -3.5 -V
```

If the result of the command is "Python3.5.3", the installation was successful.

2.1.2 Installing Python packages

(1) Start "Command Prompt" from the Windows menu.



(2) Move the folder to the location where you installed it in "2.1.1Installing Python".
Execute the following command.

```
cd C:¥Users¥renesas¥AppData¥Local¥Programs¥Python¥Python35¥Scripts
```

*The following is an example of creating a windows user name "renesas".

(1) Installing tensorflow

Execute the following command.

```
pip3 install --upgrade tensorflow==2.0.1
```

(2) Installing ProgressBar

Execute the following command.

```
pip3 install progressbar33==2.4
```

(3) Installing Prettytable

Execute the following command.

```
pip3 install prettytable==0.7.2
```

(4) Installing imageio

Execute the following command.

```
pip3 install imageio==2.6.1
```

(5) Installing keras

Execute the following command.

```
pip3 install keras==2.2.4
```

(6) Installing matplotlib

Execute the following command.

```
pip3 install matplotlib==3.0.3
```

(7) Version confirmation

Execute the following command.

```
pip3 list
```

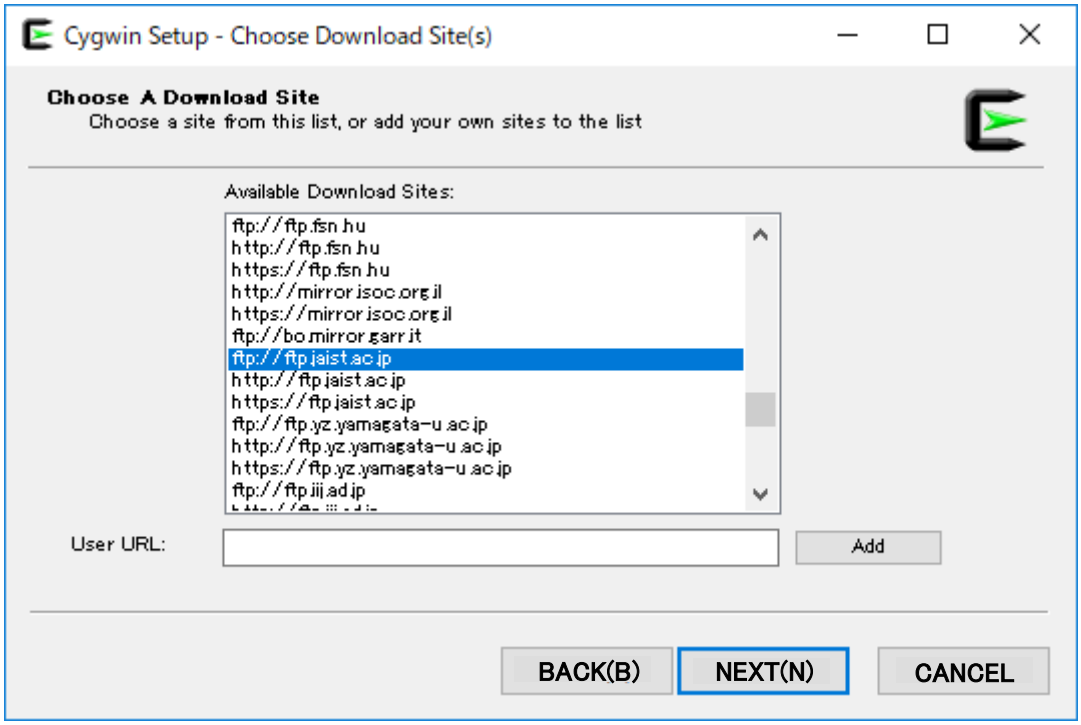
Please confirm that the installed version is correct.

tensorflow	2.0.1
ProgressBar	2.4
Prettytable	0.7.2
imageio	2.6.1
keras	2.2.4
matplotlib	3.0.3

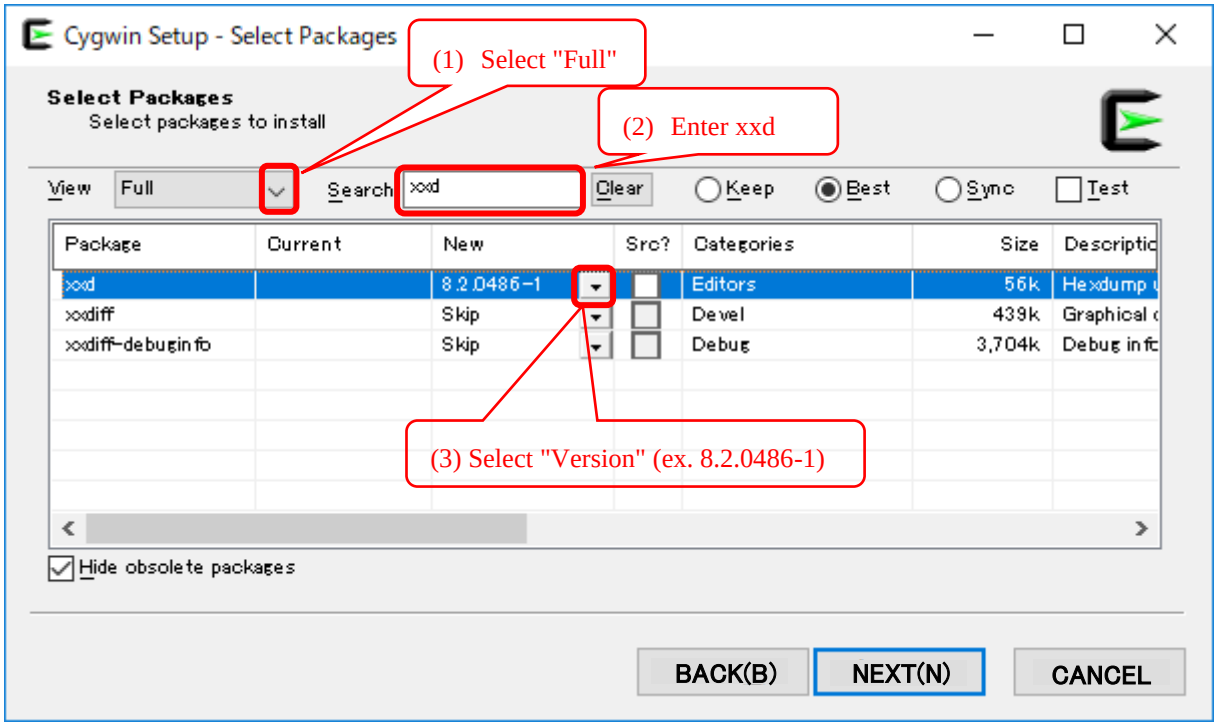
2.2 Installing Cygwin

*** Cygwin is installed to use the Linux xxd command. If you have an environment that can execute Linux commands such as Ubuntu, you do not need to install it.**

(1) Download Cygwin by clicking **setup-x86_64.exe** on the https://www.cygwin.com/.
(2) Double-click setup-x86_64.exe. We will proceed to the next without changing.
(3) Sellect a server in your area.



(4) Select the package by referring to the figure below.



(5) Press the "Next" and "Next" buttons.

3. Let's collect AI learning data

We will collect data used in AI learning. we will write two methods.

3.1 Take many pictures

We will take pictures to be used in learning AI.

If you pay attention to the following points, the recognition accuracy will increase..

POINT!

To raise the recognition rate of AI

1 Use a camera that can take pictures with good image quality.
2 Shoot so that unnecessary things are not reflected.
3 Shoot at the place where AI is actually performed.
4 Take a lot of pictures.

FAQ

How many photos should we take?

F : How many phots will we take?
A : More than 1,000 photos for each type, 1,000 photos / type in this guide, totaling 15,000 photos I'm using.
F : Wow! That is too bad.
A : It's very difficult. But imagine a baby or child recognizing something for the first time. You have to look at the mango over and over again to know "This is a mango!". But if that's difficult or you don't have the time, there's an easier way.
F : There is also an easy way.
A : Yes. There are some points to note, so I will introduce them. Let's move on to the next chapter.
F : Yes, please.

Next, proceed to "**3.3Save** the collected AI learning data in a folder"

3.2 Use the dataset

There are datasets on the WEB. You can use them freely. However, some have poor image quality, and some are not commercially available. Please use with caution.

URL

Pictures of foods*

https://data.vision.ee.ethz.ch/cvl/datasets_extra/food-101/

http://data.vision.ee.ethz.ch/cvl/food-101.tar.gz

Others

https://www.tensorflow.org/datasets/catalog/overview?hl=ja

Note: For commercial use, you need to check with the owner of each image.

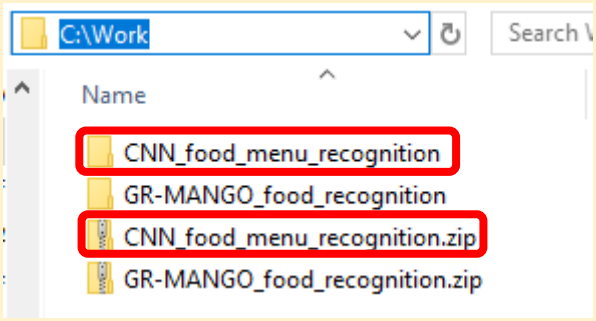
3.3 Save the collected AI learning data in a folder

Save the learning data collected in the previous chapter.

1. **Unzip the zip of the AI learning environment set for food menu recognition included in this package.**

   Unzip CNN_food_menu_recognization.zip.

   In the example, a folder of Work is created on the C drive.



2. Store photos after renaming the folder that contains the image data

Renames the folder for each category collected in the specified folder. Change the folder name of the category you want to change. The example changes caesar_salad to beer. Make the same changes for both train in the blue frame and valid in the green frame. Store the photos in the created folder.

COLUMN
Why do we create two folders with same name? What are train and valid?

When creating an AI learning model, we first train AI learning.
At the same time, we also perform verification to confirm that learning is progressing properly.

Image data (train) used for training and image data (valid) used for verification are generally separated. This time, the AI model used in the beginner's guide is divided into a ratio of 75% for training and 25% for verification (12,000 training and 3,000 verification). Please include a photo to decide the ratio based on the number of sheets you have prepared.

4. Let's make an AI learning model

4.1 Change the Python file

Chage the code of food_menu_recognition.py

4.1.1 Change the categories

Before

```
Label_list = ['caesar_salad','club_sandwich','donuts',
  'edamame','french_fries','fried_rice','hamburger',
  'hot_dog','oysters',
  'pancakes','pizza','ramen',
  'spaghetti_carbonara','steak','sushi']
```

After

```
Label_list = ['beer','club_sandwich','donuts',
  'edamame','french_toast','fried_rice','hamburger',
  'hot_dog','oysters',
  'pancakes','pizza','ramen',
  'spaghetti_meatsauce','steak','sushi']
```

4.1.2 Change the number of AI learning data

Before

```
train_data_num = 12000
valid_data_num =  3000
test  data  num   =  1000
```

After

```
train_data_num =  1200
valid_data_num =   300
test_data_num   =   100
```

4.1.3 Change structure of CNN

The following is an example when you want to change the layer of Conv2D from 11 to 13.

Before

```
model = Sequential()

model.add(Conv2D(8,    (3, 3), padding = 'same', activation
                = 'relu', input_shape = (data_height, data_width, data_channel)))
model.add(Conv2D(8,    (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(AveragePooling2D(pool_size = (2, 2)))
model.add(Dropout(0.2))
#model.add(BatchNormalization())

model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(32, (3, 3), padding = 'same', activation = 'relu'))
```

After

```
model = Sequential()

model.add(Conv2D(8,    (3, 3), padding = 'same', activation
                = 'relu', input_shape = (data_height, data_width, data_channel)))
model.add(Conv2D(8,    (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(8,    (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(AveragePooling2D(pool_size = (2, 2)))
model.add(Dropout(0.2))
#model.add(BatchNormalization())

model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(16, (3, 3), padding = 'same', activation = 'relu'))
model.add(Conv2D(32, (3, 3), padding = 'same', activation = 'relu'))
model.add(AveragePooling2D(pool_size = (2, 2)))
```

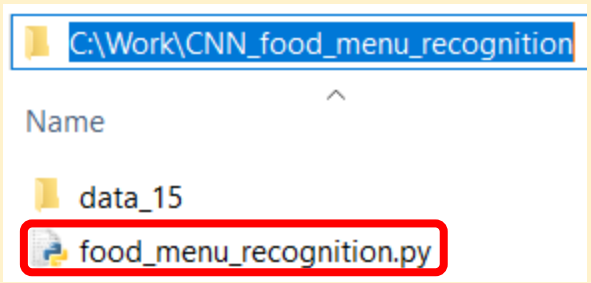
You can change the code in other parts as well.

Please refer to the contents of "2. AI learning guide of e-AI starting package" or "Keras official website".

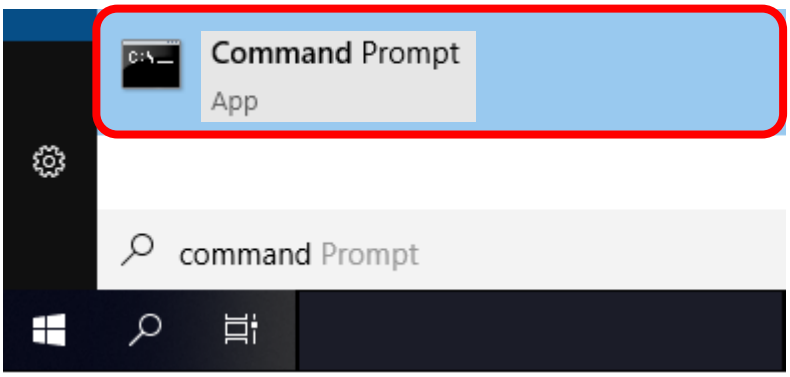
URL Keras Document https://keras.io/ja/

4.2 Create an AI learning model

Execute the following food_menu_recognition.py file.



(1) Start "Command Prompt" from start menu.



(2) Chnage the directory. Execute the following command.
(The example work directory is C: ¥ Work)

```
cd C:¥Work¥CNN_food_menu_recognition
```

(3) Execute the following command.

```
py -3.5 food_menu_recognition.py
```

The following log will appear. The execution time was about 24 hours using a Windows PC when generating the AI model of the beginner guide.

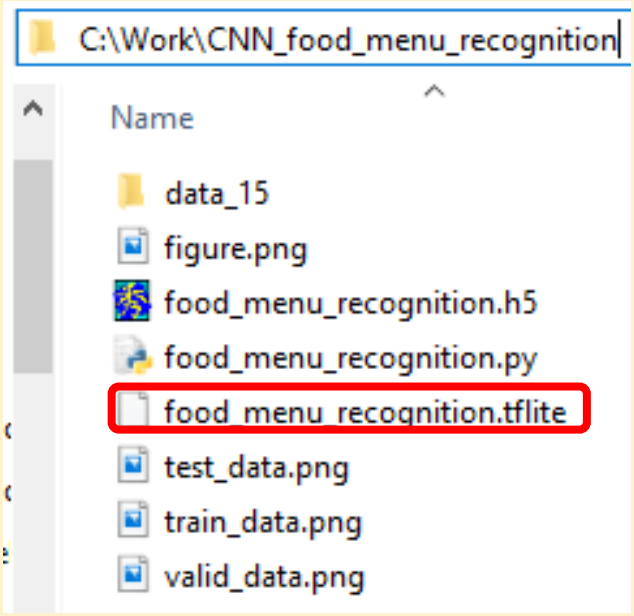
```
C:¥Users¥        >cd C:¥CNN_food_menu_recognition

C:¥CNN_food_menu_recognition>py -3.5 food_menu_recognition.py
15
Found 12000 images belonging to 15 classes.
Found 3000 images belonging to 15 classes.
Found 3000 images belonging to 15 classes.
2020-12-07 16:24:55.473630: I tensorflow/core/platform/cpu_feature
ctions that this TensorFlow binary was not compiled to use: AVX2
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 128, 128, 8)       224
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 128, 128, 8)       584
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 128, 128, 16)      1168
_________________________________________________________________
average_pooling2d (AveragePo (None, 64, 64, 16)        0
_________________________________________________________________
dropout (Dropout)            (None, 64, 64, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 64, 64, 16)        2320
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 64, 16)        2320
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 64, 32)        4640
_________________________________________________________________
average_pooling2d_1 (Average (None, 32, 32, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization (BatchNo (None, 32, 32, 32)        128
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 32, 32, 32)        9248
_________________________________________________________________
average_pooling2d_2 (Average (None, 16, 16, 32)        0
```

(2)

(3)

(4) AI model (Tensorflow lite micro format) completed

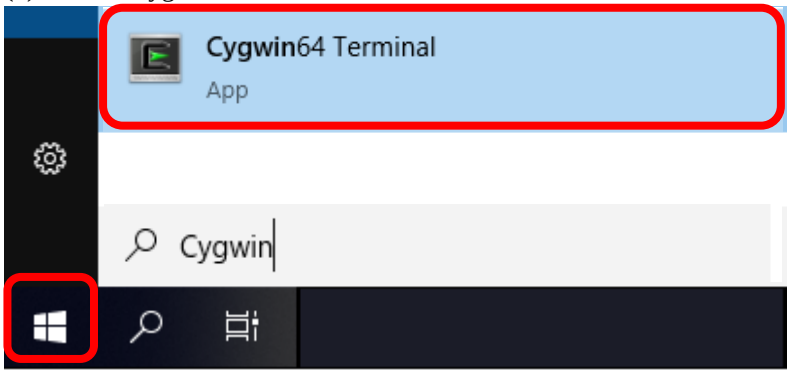
If the AI learning execution is successful, food_menu_recognition.tflite will be generated. figure.png contains a graph showing the transition of AI learning, and test_data.png, train_data.png, and valid_data.png contain some images of the data used in the test, training, and evaluation.



5. Let's convert from AI learning model to source file

We will change the "food_menu_recognition.tflite" created in Chapter 5 into the source code.
Run the Linux command xxd on Windows to convert the tflite format file to the source file.

(1) Start "Cygwin" from the Windows menu.



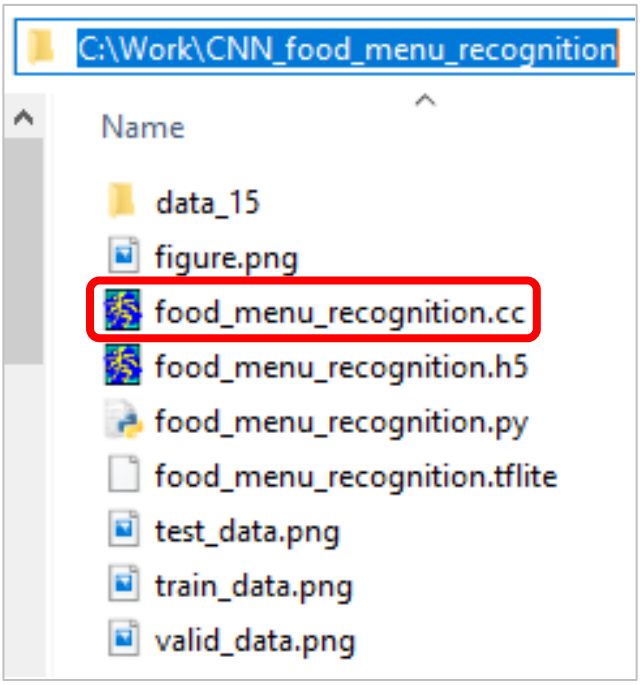
(2) Move to the folder located the food_menu_recognition.tflite. Execute the following command.

```
cd c:/Work¥CNN_food_menu_recognition
```

(3) Converts ".tflm" format to ".cc" files. Execute the following command.

```
xxd -I food_menu_recognition.tflite > food_menu_recognition.cc
```

(4) A food_menu_recognition.cc file will be created



Please open the file. It contains an array of weight data called food_menu_recognition_tflite and data called food_menu_recognition_tflite_len that represents the length of the array. Add const to make no changes.

Before (1)

```
unsigned char food_menu_recognition_tflite[] = {
  0x20, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x12, 0x00, 0x1c, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x10, 0x00, 0x14, 0x00, 0x00, 0x00, 0x18, 0x00, 0x12, 0x00, 0x00, 0x00,
  0x03, 0x00, 0x00, 0x00, 0xc8, 0x4d, 0x09, 0x00, 0xf8, 0x14, 0x09, 0x00,
  0xe0, 0x14, 0x09, 0x00, 0x3c, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x04, 0x00, 0x08, 0x00, 0x08, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00,
  0x32, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00, 0x6d, 0x69, 0x6e, 0x5f,
  0x72, 0x75, 0x6e, 0x74, 0x69, 0x6d, 0x65, 0x5f, 0x76, 0x65, 0x72, 0x73,
  0x69, 0x6f, 0x6e, 0x00, 0x33, 0x00, 0x00, 0x00, 0x98, 0x14, 0x09, 0x00,
  0x88, 0x02, 0x09, 0x00, 0x80, 0x02, 0x09, 0x00, 0xf0, 0xfd, 0x08, 0x00,
                                    :
  0xf2, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0a, 0x00, 0x0e, 0x00, 0x07, 0x00, 0x00, 0x00, 0x08, 0x00,
  0x0a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x02, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0a, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00,
  0x0a, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00
};
unsigned int food_menu_recognition_tflite_len = 609944;
[EOF]
```

(2)

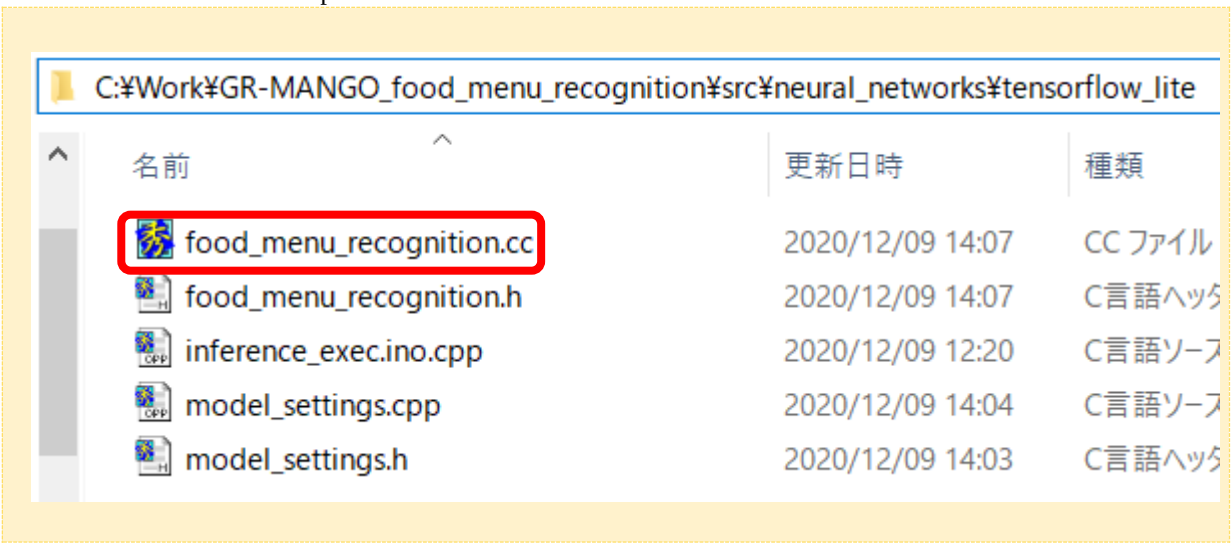
After

```
const unsigned char food_menu_recognition[] = {
  0x20, 0x00, 0x00, 0x00, 0x54, 0x46, 0x4c, 0x33, 0x00, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x12, 0x00, 0x1c, 0x00, 0x04, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x10, 0x00, 0x14, 0x00, 0x00, 0x00, 0x18, 0x00, 0x12, 0x00, 0x00, 0x00,
  0x03, 0x00, 0x00, 0x00, 0xc8, 0x4d, 0x09, 0x00, 0xf8, 0x14, 0x09, 0x00,
  0xe0, 0x14, 0x09, 0x00, 0x3c, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
  0x01, 0x00, 0x00, 0x00, 0x0c, 0x00, 0x00, 0x00, 0x08, 0x00, 0x0c, 0x00,
  0x04, 0x00, 0x08, 0x00, 0x08, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00,
  0x32, 0x00, 0x00, 0x00, 0x13, 0x00, 0x00, 0x00, 0x6d, 0x69, 0x6e, 0x5f,
  0x72, 0x75, 0x6e, 0x74, 0x69, 0x6d, 0x65, 0x5f, 0x76, 0x65, 0x72, 0x73,
  0x69, 0x6f, 0x6e, 0x00, 0x33, 0x00, 0x00, 0x00, 0x98, 0x14, 0x09, 0x00,
  0x88, 0x02, 0x09, 0x00, 0x80, 0x02, 0x09, 0x00, 0xf0, 0xfd, 0x08, 0x00,
  0xe0, 0xd9, 0x08, 0x00, 0xd8, 0xd9, 0x08, 0x00, 0xd0, 0xd9, 0x08, 0x00,
                                    :
  0xf2, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x03, 0x03, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0a, 0x00, 0x0e, 0x00, 0x07, 0x00, 0x00, 0x00, 0x08, 0x00,
  0x0a, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x02, 0x00, 0x00, 0x00,
  0x00, 0x00, 0x0a, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00,
  0x0a, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x00
};
const unsigned int food_menu_recognition_len = 609944;
[EOF]
```

6. Let's implement in e²studio

6.1 Change AI model

Overwrite the file created in Chapter 5 with the file in the red frame below.



6.2 Change the categories

Change the following files.

6.2.1 inference_exec.h

Before

```
typedef struct
{
    uint32_t us_total;
    float    score;
    char     *category;
    float    caesar_salad_score;
    float    club_sandwich_score;
    float    donuts_score;
    float    edamame_score;
    float    french_fries_score;
    float    fried_rice_score;
    float    hamburger_score;
    float    hot_dog_score;
    float    oysters_score;
    float    pancakes_score;
    float    pizza_score;
    float    ramen_score;
    float    spaghetti_carbonara_score;
    float    steak_score;
    float    sushi_score;
    char     *price;
}
inference_result_t;
```

After

```
typedef struct
{
    uint32_t us_total;
    float    score;
    char     *category;
    float    beer_score;
    float    club_sandwich_score;
    float    donuts_score;
    float    edamame_score;
    float    french_fries_score;
    float    french_toast_score;
    float    hamburger_score;
    float    hot_dog_score;
    float    oysters_score;
    float    pancakes_score;
    float    pizza_score;
    float    ramen_score;
    float    spaghetti_meatsauce_score;
    float    steak_score;
    float    sushi_score;
    char     *price;
}
inference_result_t;
```

6.2.2 inference_exec.ino.cpp

Before

```
/* Process the inference results. */
inference_result.caesar_salad_score          = output->data.f[k_caesar_salad];
inference_result.club_sandwich_score         = output->data.f[k_club_sandwich];
inference_result.donuts_score                = output->data.f[k_donuts];
inference_result.edamame_score               = output->data.f[k_edamame];
inference_result.french_fries_score          = output->data.f[k_french_fries];
inference_result.fried_rice_score            = output->data.f[k_fried_rice];
inference_result.hamburger_score             = output->data.f[k_hamburger];
inference_result.hot_dog_score               = output->data.f[k_hot_dog];
inference_result.oysters_score               = output->data.f[k_oysters];
inference_result.pancakes_score              = output->data.f[k_pancakes];
inference_result.pizza_score                 = output->data.f[k_pizza];
inference_result.ramen_score                 = output->data.f[k_ramen];
inference_result.spaghetti_carbonara_score   = output->data.f[k_spaghetti_carbonara];
inference_result.steak_score                 = output->data.f[k_steak];
inference_result.sushi_score                 = output->data.f[k_sushi];
```

After

```
/* Process the inference results. */
inference_result.beer_score                  = output->data.f[k_beer];
inference_result.club_sandwich_score         = output->data.f[k_club_sandwich];
inference_result.donuts_score                = output->data.f[k_donuts];
inference_result.edamame_score               = output->data.f[k_edamame];
inference_result.french_fries_score          = output->data.f[k_french_fries];
inference_result.french_toast_score          = output->data.f[k_french_toast];
inference_result.hamburger_score             = output->data.f[k_hamburger];
inference_result.hot_dog_score               = output->data.f[k_hot_dog];
inference_result.oysters_score               = output->data.f[k_oysters];
inference_result.pancakes_score              = output->data.f[k_pancakes];
inference_result.pizza_score                 = output->data.f[k_pizza];
inference_result.ramen_score                 = output->data.f[k_ramen];
inference_result.spaghetti_meatsauce_score   = output->data.f[k_spaghetti_meatsauce];
inference_result.steak_score                 = output->data.f[k_steak];
inference_result.sushi_score                 = output->data.f[k_sushi];
```

6.2.3 model_settings.cpp

Before

```
/* Process the inference results. */
inference_result.caesar_salad_score          = output->data.f[k_caesar_salad];
inference_result.club_sandwich_score         = output->data.f[k_club_sandwich];
inference_result.donuts_score                = output->data.f[k_donuts];
inference_result.edamame_score               = output->data.f[k_edamame];
inference_result.french_fries_score          = output->data.f[k_french_fries];
inference_result.fried_rice_score            = output->data.f[k_fried_rice];
inference_result.hamburger_score             = output->data.f[k_hamburger];
inference_result.hot_dog_score               = output->data.f[k_hot_dog];
inference_result.oysters_score               = output->data.f[k_oysters];
inference_result.pancakes_score              = output->data.f[k_pancakes];
inference_result.pizza_score                 = output->data.f[k_pizza];
inference_result.ramen_score                  = output->data.f[k_ramen];
inference_result.spaghetti_carbonara_score   = output->data.f[k_spaghetti_carbonara];
inference_result.steak_score                 = output->data.f[k_steak];
inference_result.sushi_score                 = output->data.f[k_sushi];
```

After

```
/* Process the inference results. */
inference_result.beer_score                  = output->data.f[k_beer];
inference_result.club_sandwich_score         = output->data.f[k_club_sandwich];
inference_result.donuts_score                = output->data.f[k_donuts];
inference_result.edamame_score               = output->data.f[k_edamame];
inference_result.french_fries_score          = output->data.f[k_french_fries];
inference_result.french_toast_score           = output->data.f[k_french_toast];
inference_result.hamburger_score             = output->data.f[k_hamburger];
inference_result.hot_dog_score               = output->data.f[k_hot_dog];
inference_result.oysters_score               = output->data.f[k_oysters];
inference_result.pancakes_score              = output->data.f[k_pancakes];
inference_result.pizza_score                 = output->data.f[k_pizza];
inference_result.ramen_score                  = output->data.f[k_ramen];
inference_result.spaghetti_meatsauce_score   = output->data.f[k_spaghetti_meatsauce];
inference_result.steak_score                 = output->data.f[k_steak];
inference_result.sushi_score                 = output->data.f[k_sushi];
```

6.2.4 model_settings.h

Before

```
constexpr int kCategoryCount = 15;

// Index=15

constexpr int k_caesar_salad          = 0;
constexpr int k_club_sandwich         = 1;
constexpr int k_donuts                = 2;
constexpr int k_edamame               = 3;
constexpr int k_french_fries          = 4;
constexpr int k_fried_rice            = 5;
constexpr int k_hamburger             = 6;
constexpr int k_hot_dog               = 7;
constexpr int k_oysters               = 8;
constexpr int k_pancakes              = 9;
constexpr int k_pizza                 = 10;
constexpr int k_ramen                 = 11;
constexpr int k_spaghetti_carbonara   = 12;
constexpr int k_steak                 = 13;
constexpr int k_sushi                 = 14;
```

After

```
constexpr int kCategoryCount = 15;

// Index=15

constexpr int k_beer                  = 0;
constexpr int k_club_sandwich         = 1;
constexpr int k_donuts                = 2;
constexpr int k_edamame               = 3;
constexpr int k_french_fries          = 4;
constexpr int k_french_toast          = 5;
constexpr int k_hamburger             = 6;
constexpr int k_hot_dog               = 7;
constexpr int k_oysters               = 8;
constexpr int k_pancakes              = 9;
constexpr int k_pizza                 = 10;
constexpr int k_ramen                 = 11;
constexpr int k_spaghetti_meatsauce   = 12;
constexpr int k_steak                 = 13;
constexpr int k_sushi                 = 14;
```

6.3 Change structure of CNN

6.3.1 inference_exec.ino.cpp

I will write an example when I want to change the layer of Conv2D from 11 to 13.
However, when using conv2D with conditions not described in Chapter **8.2**, set bit to 0.

Before

```
void inference_exec_init(void) {

    static tflite::MicroErrorReporter micro_error_reporter;
    error_reporter = &micro_error_reporter;
    model = tflite::GetModel(food_menu_recognition_tflite);
    drp_en_conv_layer = 0x07ff;       /* DRP Layer enable bit */
    drp_conv_total_num = 0;
```

After

```
void inference_exec_init(void) {

    static tflite::MicroErrorReporter micro_error_reporter;
    error_reporter = &micro_error_reporter;
    model = tflite::GetModel(food_menu_recognition_tflite);
    drp_en_conv_layer = 0x1fff;       /* DRP Layer enable bit */
    drp_conv_total_num = 0;
```

Case of the 11 layers

0x07fff → 0b0000 0111 1111 1111

11bits are "1".

Case of the 13 layers

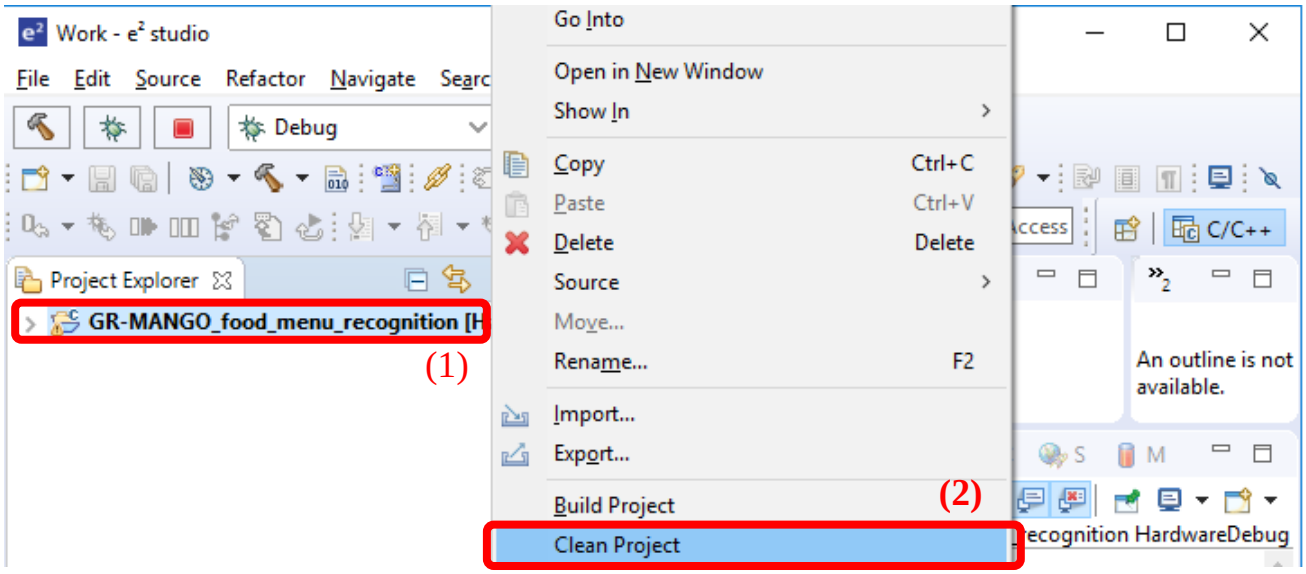
0x1ffff → 0b0001 1111 1111 1111

13bits are "1".

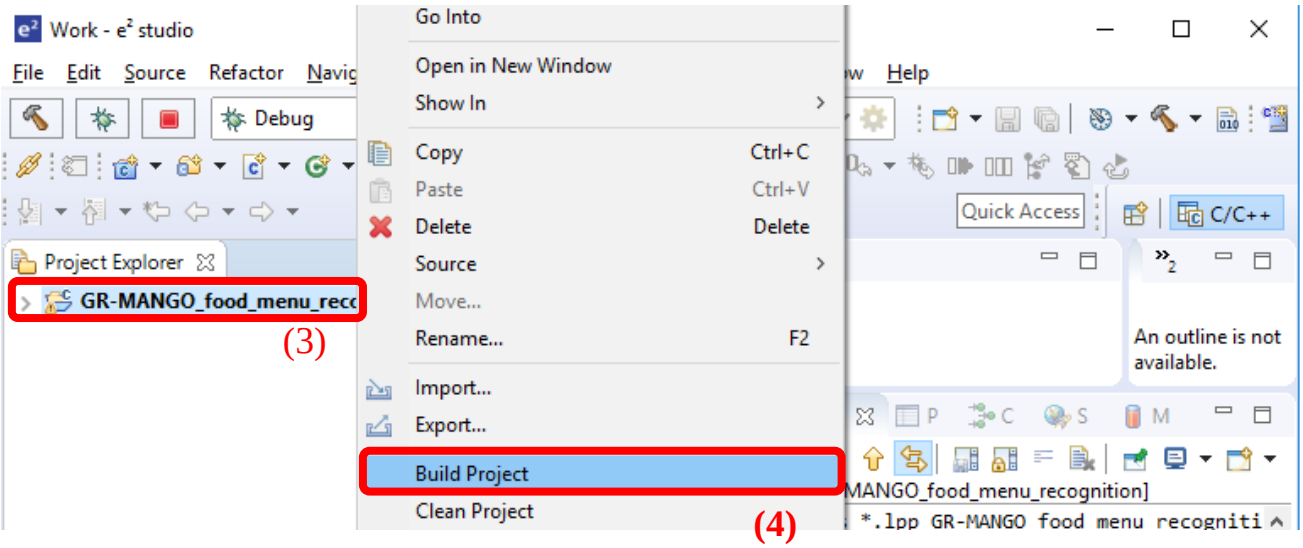
However, when using conv2D with conditions not described in Chapter8.28.2, set bit to 0. For example, if the 13th layer does not meet the conditions, set it to 0x0fff.

7. Let's recognize the original food menu with AI

1. Start e2studio by referring to "When using a debugger" in "GR-MANGO_AI_beginner's guide".
2. Right-click (1) on the red frame. Click Clean Project.



3. Right-click (3) on the red frame and click (4) Build Project.



4. Load and execute by referring to GR-MANGO AI_Beginner 's guide.

5. A demo will be displayed.

8. Appendix

8.1 Why can AI processing of color images be realized with less memory usage?

Using TFLM

TFLM is an abbreviation for TensorFlow Lite for Microcontrollers.

TFLM is open source provided by Google.

A port of TensorFlow Lite designed to run machine learning models on devices such as memory-limited microcontrollers.

TFLM has a function that can support quantization, and we can expect a memory reduction effect of about 1/4. There are several types, this time with a slight decrease in model accuracy, but we are using post-training integer quantized, which can reduce the size of the model.

The source code included in this package may be updated. Please check from the URL information below.

Words

TensorFlow : Open source that platform developed by Google for machine learning.

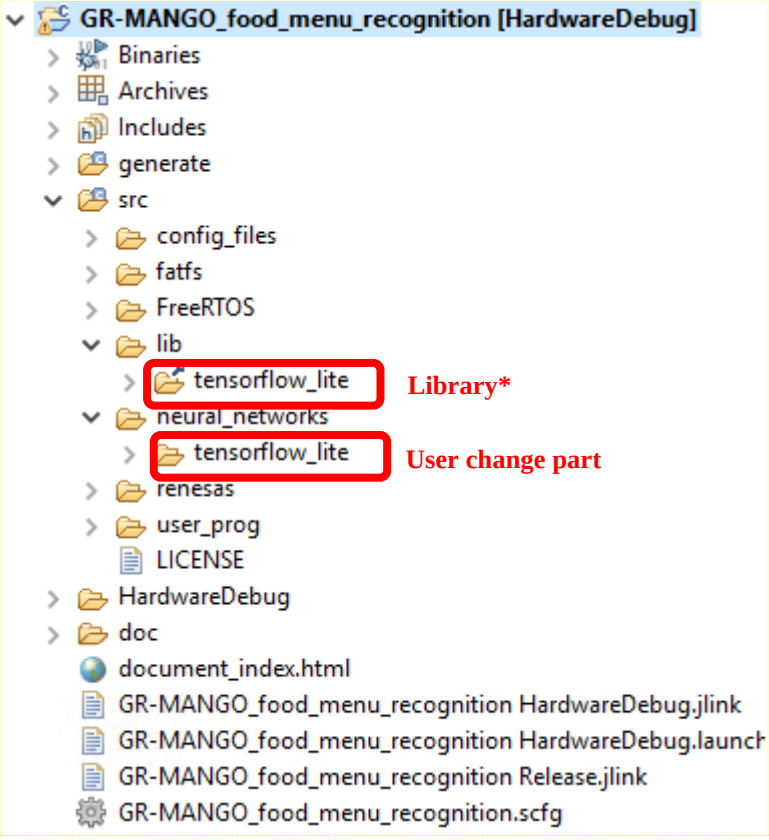
TensorFlow Lite : Open source that enables inference on the device

URL

TFLM : https://www.tensorflow.org/lite/microcontrollers

TensorFlow : https://www.tensorflow.org/?hl=ja

Souce Code



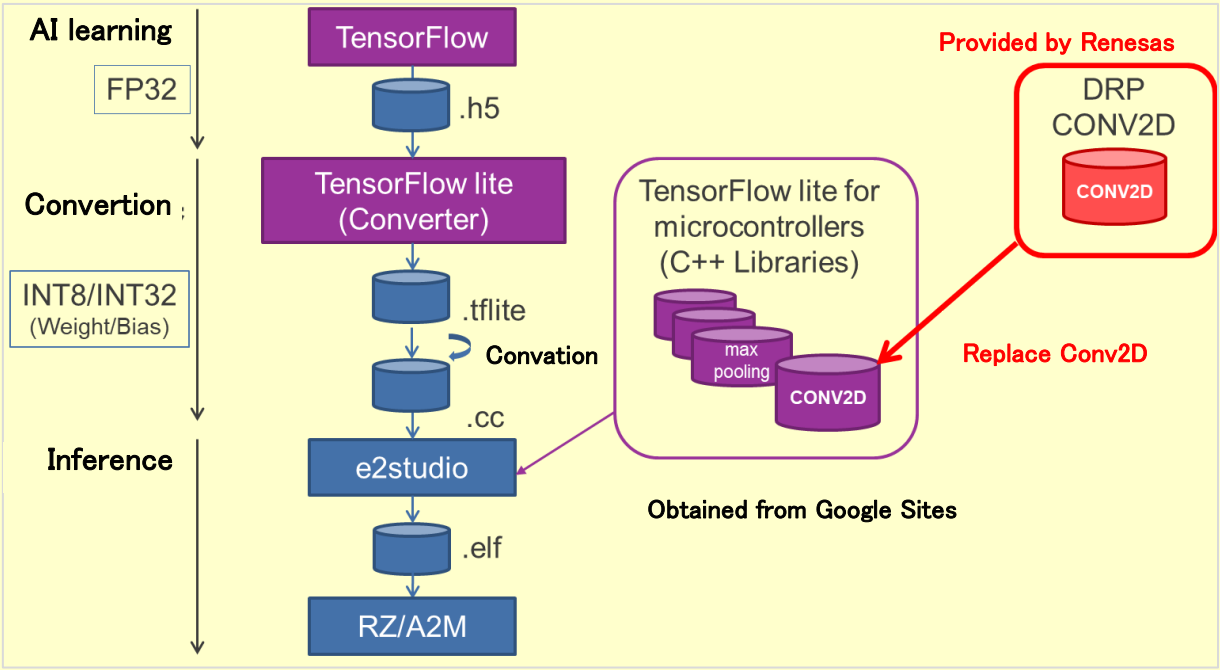
Note: Some changes have been made at Renesas Electronics. We added that in the comments.

8.2 Why can AI processing of color images be performed in less than 400 msec?

Speeding up using DRP library for CNN processing

By replacing the convolution layer of CNN with the DRP library, it is possible to perform AI processing about 5 - 7 times faster than the CPU.

Flow



DRP library : supported parameter of CONV2D

feature map X size, Y size	**4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256**
ich (input channel)	**3, 4, 8, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224**
och (output channel)	**4, 8, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224**
Feature map X size, Y size **Input / output channel constraints**	**X size x Ysize x och ≦ 1M(1,048,576)** **ich ≦ och**

Ex.) Parameter combination

X x Y x och = 256 x 256 x 16 ≦ 1M, ich = 8 ≦ 16 OK
X x Y x och = 16 x 16 x 224 ≦ 1M, ich = 112 ≦ 224 OK
X x Y x och = 160 x 120 x 8 ≦ 1M, ich = 3 ≦ 8 OK

DRP library : filter specification of CONV2D

- **Size fixed 3x3, stride X direction fixed 1, Y direction fixed 1**
- **Padding fixed 0, feature map input/output size, fixed SAME**

8.3 For commercial use

We are using the library of Apache License2.0

The TFLM described in chapter 8.1 is Apache License 2.0.
Please check the required contents before commercial use.

URL

Apache License 2.0 : **https://www.apache.org/licenses/LICENSE-2.0**

Note : This package contains some modifications by Renesas Electronics, which is clearly stated in the source code comments.

8.4 Why did we decide to make this package?

When we decided to make this package, a big illness was prevalent all over the world. We thought that if automation such as self-checkout progressed, many people would be able to avoid unnecessary and unurgent going out.

When I was shopping and saw an apple without a barcode, if we could recognize AI with a barcode reader at the same time, I could save the efforts of pasting the barcode, the cash register would not be crowded, and various people I thought it would be useful.

(There may be regions and countries that have already been realized and are widespread.)

Getting started with AI seems very difficult.AI. To lower that hurdle, we've created this guide for easy changes.

We sincerely hope that you will create an AI solution that will be useful to many people.

2020.12.24