<Infrared Sensors>

Manual of MTT-V002 for training AI model

CTSA-231012-02

Table of Contents

Manual of MTT-V002 for training AI model

CTSA-231012-02

1. Introduction

1.1 Overview of this manual

An algorithm for using AI to detect heat sources from thermal images taken with the thermal diode infrared sensor MelDIR is provided as a reference code. An example of human detection results using this algorithm is shown in Figure 1.1.

If the AI model provided by our company does not provide sufficient detection results, the Model training tool for MelDIR (MTT-V002) can be used as a tool to create an AI model suitable for the user's environment.

This manual describes the sequence of steps required to create a model: extraction of a grayscale image from a captured thermal image, annotation, model creation using this tool, and writing to an MCU.

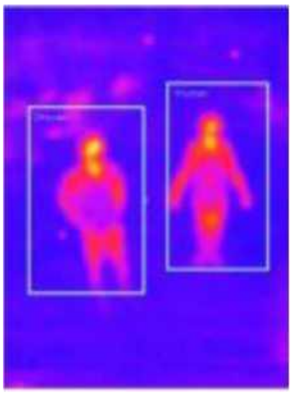


Figure 1.1 Example of person detection

1.2 Features and specifications of heat source detection algorithm

By utilizing the heat source detection algorithm and Model training tool for MelDIR, we can reduce the resources required to develop systems and services that require heat source detection incorporating MelDIR. The features of this algorithm are as follows.

・Developed as an edge AI targeting a general-purpose MCU for embedding. No servers or cloud required.

・Works only with general-purpose MCUs. No external memory, GPU, accelerator, or other external components.

・MelDIR-optimized algorithms and neural networks for high-performance, real-time detection.

・Everything from the algorithm to the development environment is based on OSS, making it easy to use for development.

・Model training tools can be used to create AI models for different applications and installation environments.

Table 1.1 Specifications of heat source detection algorithms and target device

Item	Specifications
MelDIR Model Number	MIR8060B1-01
Demo kit	EVB-8060B1-MN****
Reference code	SAPM-MD**V*
Target MCU	STM32H723_G(Cortex-M7 1177DMIPS@530MHz)
Deep learning algorithms	MobileNetV2-YOLOv3-Nano
Rom size	<1Mbyte
Ram size	<512Kbyte
Frame rate	7fps(=142ms/frame) Measured values under our demo kit

1.3 Overview of AI model creation

Figure 1.2 shows the flow for creating an AI model from thermal images. It consists of five processes: (1) Imaging of taken data (2) Extraction of images for supervisor data (3) annotation (4) model training, and (5) model conversion and writing, which are explained in this manual. For taking thermal images, we provide a demo kit, so please refer to the included manual.

The Model training tool for MelDIR (MTT-V002) provided by our company is a software used for (1) Imaging of taken data and (4) model training, and the other process uses software provided by companies other than our company. Please note that we are unable to support software manufactured or provided by companies other than our company beyond the scope of this manual. In addition, prior to product development, please check the terms and conditions of use of each software.

The Model training tool MTT-V002 for MelDIR runs on a PC with Microsoft Windows 10 installed. We recommend using a PC connected to the Internet. The time required for training in Chapter 5 Model training varies depending on CPU performance. We have confirmed that the training unit works in the following environments.

・Processor : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz (8 CPUs)

・Memory: 8GB RAM

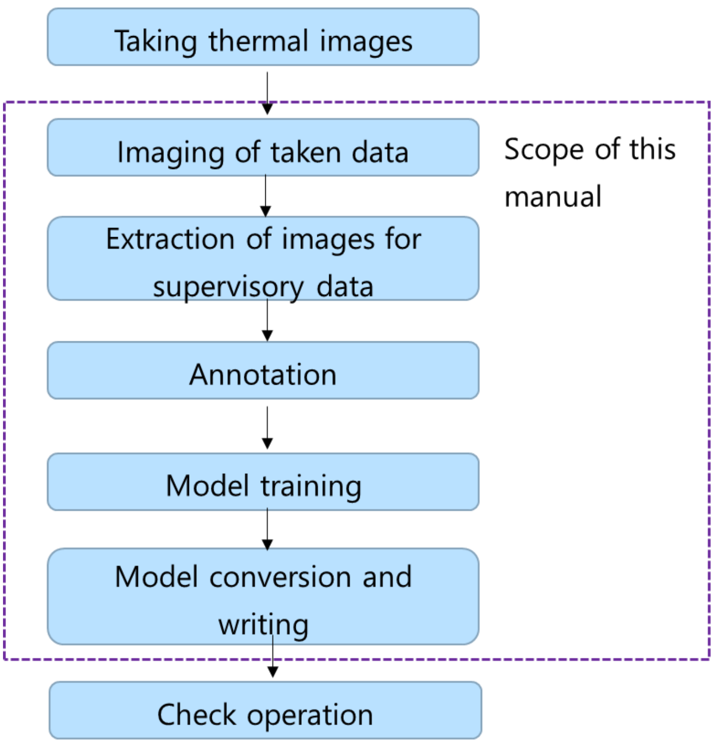


Fig. 1.2 Flow of AI model creation

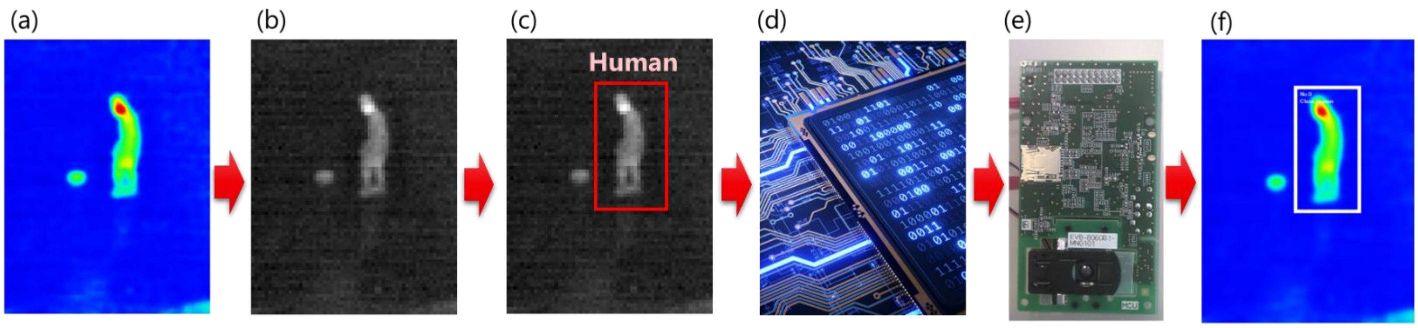


Fig. 1.3 Overview of AI model creation process (a) Taking thermal images (b) Making grayscale image (c) Annotation (d) Model training (e) Model conversion and writing (f) Behavior checking

Figure 1.3 outlines the processing in each step of AI model creation.

・Imaging and image extraction of captured data

In Figure 1.3(b), a grayscale bmp image used for model training is generated from the captured thermal image. From the generated image, the image for the supervisor data to be used for training is extracted.

Tools :MTT-V002

・Annotations

In Fig. 1.3 (c), the grayscale image is tagged to create supervisor data for machine learning.

Tools : Visual Object Tagging Tool (VoTT)

・Model training

In Fig. 1.3 (d), the AI model is trained and created by inputting the annotated thermal image into the Model training tool.

Tools : Model training tool for MelDIR (MTT-V002)

・Model conversion and writing

In Fig. 1.3 (e), the AI model is converted into a format that can be written to a microcomputer, and the writing is performed in a format that overwrites the reference code.

Tools : STM32 CubeIDE

Operation check

When all the steps are completed, it enables to detect the object on the thermal image as shown in Fig. 1.3 (f).

Tools : MIRVIEW-V40* or MIRVIEW-V50*

2. Installation and configuration of MTT-V002

2.1 Installation of MTT-V002

Follow the steps below to install MTT-V002.

(1) Double-click MTT-V002.exe.

(2) Click "Yes" when the user account control is displayed.

(3) Confirm the contents of the license agreement, select "I agree" and click "Next" (Fig. 2.1(a)).

(4) Select the installation directory and click "Next" (Fig. 2.1(b)).

(5) Specify the Start Menu folder and click "Next" (Fig. 2.1(c)).

(6) Click "Install" to execute the installation (Fig. 2.1(d)).

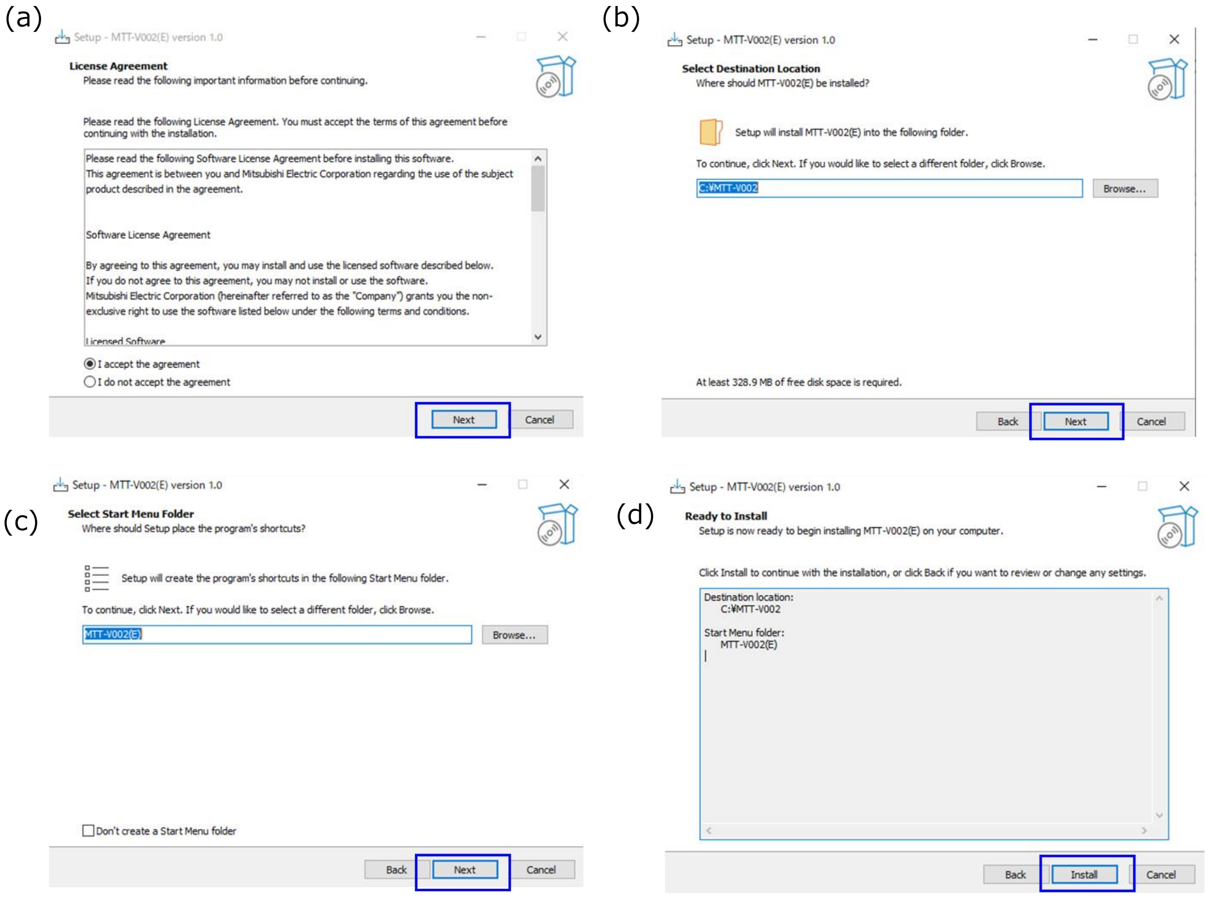


Fig. 2.1 Installation procedure for MTT-V002

2.2 Uninstallation of MTT-V002

(1) Double-click unins000.exe in the MTT-V002 folder.

(2) When the User Account Control dialog box appears, click "Yes.

(3) Click "Yes" when the confirmation screen for uninstallation is displayed.

(4) Click "OK" when the uninstallation completion screen is displayed.

2.3 Folder structure of MTT-V002

Figure 2.2 shows the folder structure of MTT-V002, and Table 2.1 outlines each folder and file.

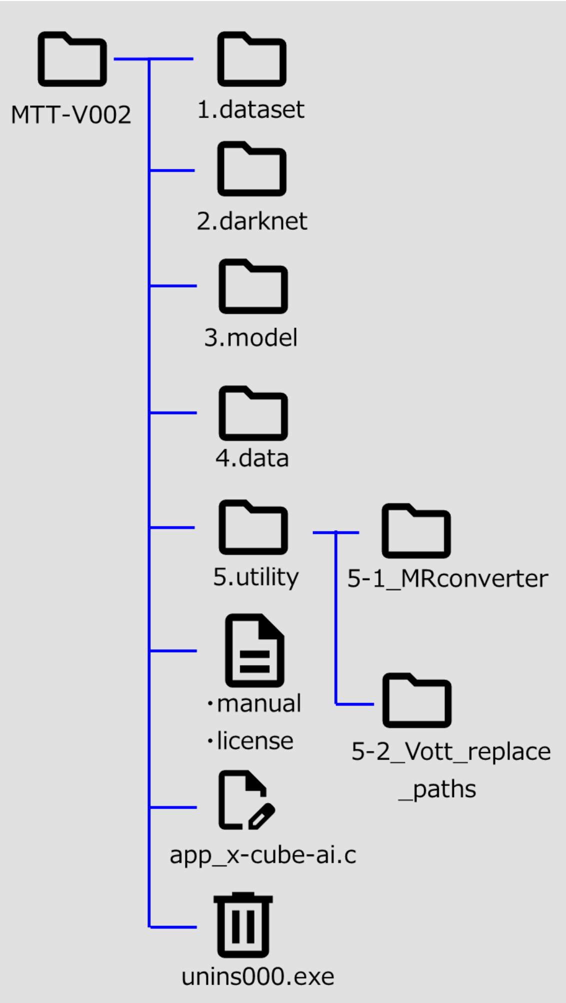


Fig. 2.2 Folder structure of MTT-V002

Table 2.1 Folder description for MTT-V002

Folder/file name	Process	Details
1.dataset	Chapter 5 Model training	Convert data saved in the data folder to be used by MTT-V002
2.darknet	Chapter 5 Model training	Train and generate AI models.
3.model	Chapter 5 Model training	Quantize AI model (weight file) and convert to tflite format.
4.data	Chapter 5 Model training	Folder to store the annotated Pascal VOC format data created in Chapter 4.
5.utility – 5-1_MRconverter	Chapter 3 Generation of bmp images	Conversion from captured data to grayscale bmp images.
5.utility – 5-2_Vott_replace_paths	Chapter 4 Annotations	Used to move the folder where the annotation data before exporting was saved
・Manual ・License	-	・This material ・Terms and conditions
app_x-cube-ai.c	Chapter 6 Writing to MCU	Used in the process of writing to an MCU
unisn000.exe	-	Used to uninstall MTT-V002

3. Generation and extraction of images for supervisor data

3.1 Cautions for taking of thermal images

An AI model will not be able to detect correctly if the orientation of the input image is upside down. For this reason, the orientation of MelDIR when capturing training data and the orientation of MelDIR when detecting with the AI model should be the same.

The "Other settings" tab of the viewer software has a function to rotate the image, but this function rotates the image on the PC. Since the AI detection is done inside the MCU, this function cannot change the orientation of the thermal image to be input to the AI. As shown in Figure 3.1, the three buttons under “Image direction” must be set to the same settings when thermal images are taken for training and when AI detects objects from thermal images.

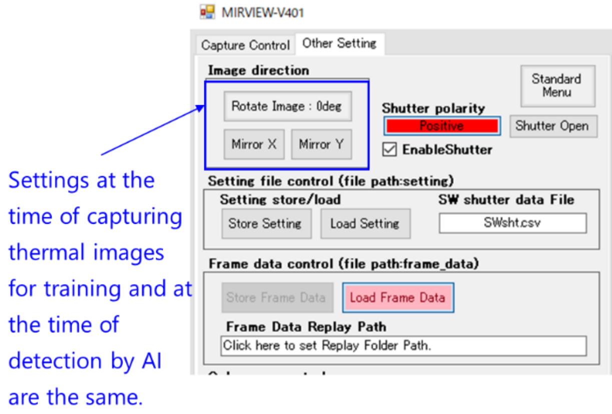


Fig. 3.1 Notes on viewer settings

3.2 Generation of grayscale bmp image from thermal image data

Use 5-1_MRconverter in the folder 5.utility to convert thermal images to bmp images. The folder structure of MRconverter is shown in Figure 3.2.

The recorded data to be converted can be saved in any location.

Copy the exe file in the MRconverter folder into the folder where you saved the data to perform the conversion.

The tool to be used depends on the version of the viewer software being used.

If thermal image data is saved in csv format using MIRVIEW-V401 or V402, please refer to 3.2.1 csv to bmp. If thermal image data is saved in recorded data file format (.MR1 or .MR2) using MIRVIEW-V5**, please refer to 3.2.2 MR to bmp. There are two versions of MR to bmp: a Windows version that runs on the command prompt and a Cygwin version. Please use whichever is more convenient for your environment.

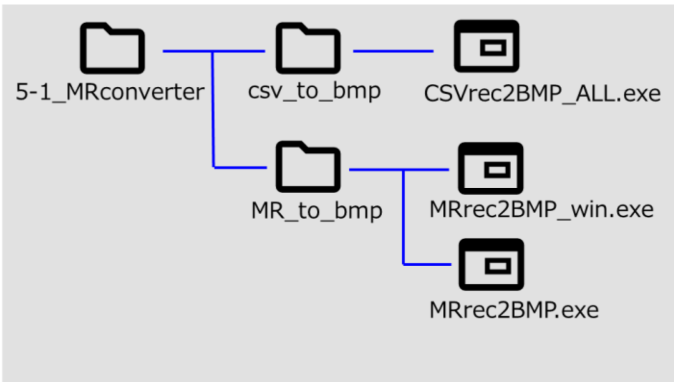


Fig 3.2 Structure of MRconverter

3.2.1 csv to bmp

1. First, install Cygwin. Go to https://www.cygwin.com/ and download the installer from Install Cygwin by running setup-x86_64.exe.

2. Double-click setup-x86_64.exe.

3. Follow the steps shown in Fig. 3.3 (a) - (i) to install the software.

(e) If you need to configure proxy settings, please contact your network administrator.

(f) It does not matter where you choose.

(g) You do not need to change any settings, just click “Next”.

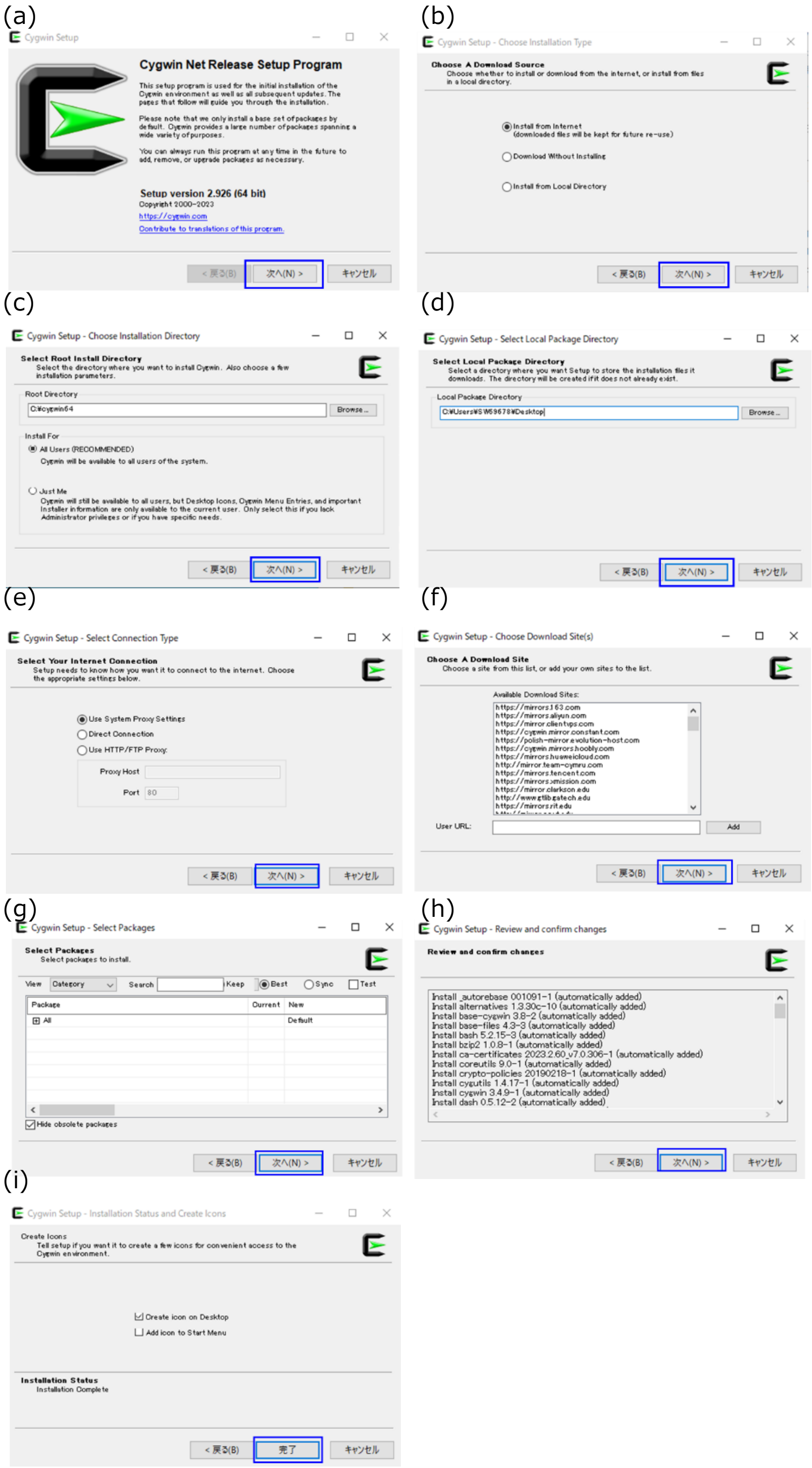


Fig. 3.3 Installation procedure of Cygwin

Manual of MTT-V002 for training AI model

・File conversion

Convertible csv files are 1D data files (file name "frame_Cont_1D_*.csv" or "frame_ContDrc_1D_*.csv"). Conversion is possible even if the files are not sequentially numbered.

1. Copy and paste CSVrec2BMP_ALL.exe into the folder where the csv file is saved.
   Example of exe file path: C:¥csv_data¥CSVrec2BMP_ALL.exe
2. Click Cygwin64 Terminal to start the terminal.
3. Navigate to the directory where you saved the exe file with the following command.
   Usage: cd <directory>.
   Command example: cd /cygdrive/c/csv_data
4. type the following two commands and press "Enter" key to convert all csv files in the folder to bmp files.
   Usage: . /CSVrec2BMP_ALL.exe -o TL <directory where exe file is saved>.
   Command example: . /CSVrec2BMP_ALL.exe -o TL /cygdrive/c/csv_data
   Figure 3.4 shows the screen during conversion.

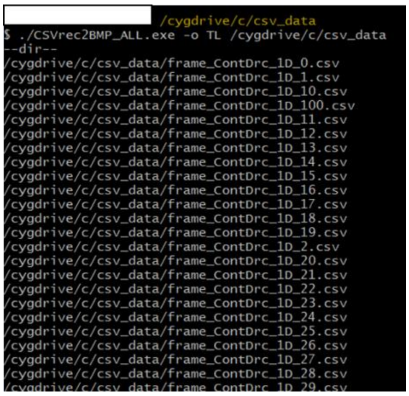


Fig. 3.4 Screen during conversion

3.2.2 MR to bmp

・Windows version

1. Copy and paste MRrec2BMP_win.exe into the folder where MR0, MR1, and MR2 files are saved. Both MR1 and MR2 can be converted simultaneously. The MR0 file is required to obtain image orientation information.
   Example of exe file path: C:¥MR_data¥MRrec2BMP_win.exe
2. Open the folder where the MR file is saved as shown in Figure 3.5, type "cmd" in the address bar and press "Enter" key.
3. Type the following command in the Command Prompt window.
   Example of bmp file destination: C:¥MR_data¥bmp
   If the folder specified in the output directory does not exist, a new folder will be created.
   Usage: MRrec2BMP_win <recording data(MR0)> <output directory>.
   Command example: MRrec2BMP_win MELD0001.MR0. ¥bmp
   Figure 3.6 shows the screen after execution.

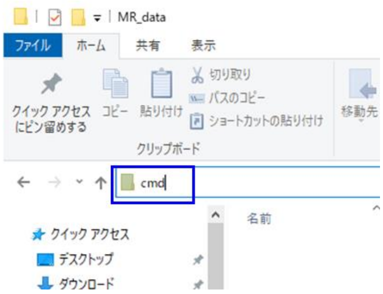


Fig. 3.5 Launching the command prompt

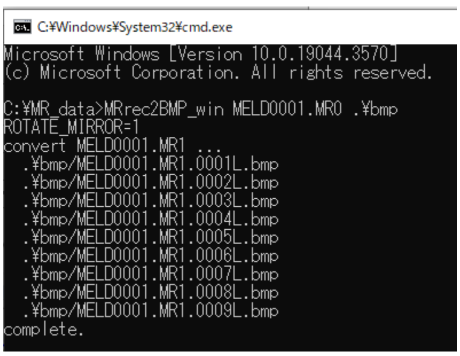


Fig. 3.6 Screen after conversion

・Cygwin version

1. Install Cygwin according to the procedure in section 3.2.1.
2. Copy and paste MRrec2BMP.exe into the folder where MR0, MR1, and MR2 files are saved. The MR0 file is required to obtain image orientation information.
   Example of exe file path: C:¥MR_data¥MRrec2BMP.exe
3. Click Cygwin64 Terminal to launch the terminal.
4. Go to the directory where the exe file is saved by the following command.
   Usage: cd <directory>.
   Command example: cd /cygdrive/c/MR_data
5. Enter the following command on the command prompt screen.
   Example of bmp file destination: C:¥MR_data¥bmp
   If the folder specified in the output directory does not exist, a new one will be created.
   Usage: MRrec2BMP <recording data (MR0)> <output directory>.
   Command example: . /MRrec2BMP . /MELD0001.MR0 . /bmp
   Figure 3.7 shows the screen after conversion.

Fig. 3.7 Screen after conversion

・Extraction of images for supervised training. (Tips)

It is not necessary to annotate all of the output grayscale images. Select the thermal images needed for training. This process is important for creating models with high detection rates, and here are some tips:

-Be careful not to end up with only similar images with no movement. It is preferable to extract images showing the heat source (person) to be detected in various positions, numbers, and positions. It can also reduce false positives if it includes only a background image without a heat source.

-The number of thermal images to be annotated should be about 200 or more for each class. Untagged heat sources are treated as noise (not detected). If multiple classes exist in the same image, it counts by each class.

- Avoid large differences in the number of annotations per class.

4. Annotation

4.1 What is annotation?

Machine learning using AI is broadly divided into supervised learning and unsupervised learning. In the heat source detection algorithm, models are created in a "supervised learning" method, and the process of creating teacher data by tagging images is annotation. For example, in a thermal image with people in it, information about the position, size and number of people in it is added to the thermal image.

4.2 Annotation work

The storage location of thermal images and annotation data is arbitrary, but for the sake of explanation, we will assume below that the folder structure is as shown in Figure 4.1.

Create a "Working folder" on the desktop, and create "Image" and "Annotation" folders in it. Save the prepared grayscale thermal images in bmp or png format in the "Image" folder.

Note that the annotation results include information on the file paths of the images, and if the folders are moved after annotation, .json and .votto files will need to be modified. If you wish to move files or merge annotation results by multiple people, please use the tools for path modification described in section 4.3.

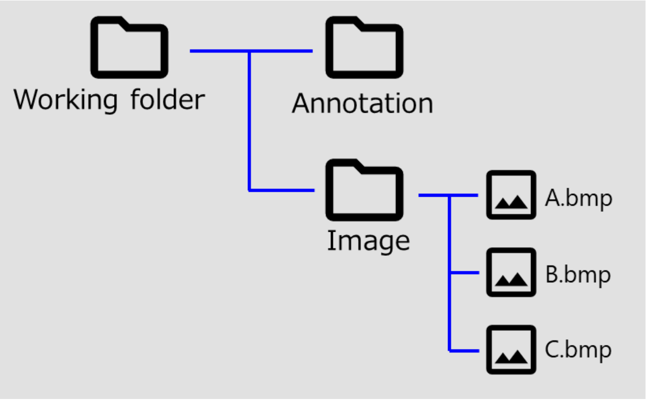


Fig.4.1 Structure of working folder

·Installing VoTT

We use Microsoft's annotation tool Visual Object Tagging Tool (VoTT) v 2.2.0 for annotation work. For an explanation of VoTT, please see reference material [1].

VoTT is available from the following HP:

https://github.com/Microsoft/VoTT/releases

The file to download is vott-220-win32.exe in Assets.

After downloading, double-click the exe file to install and install it.

•Creating a project

(a) When VoTT is activated, the screen shown in Fig. 4.2 appears. Click New Project.

(b)The project settings screen shown in Figure 4.3 appears. ① Please enter a project name that does not include Japanese characters in the display name.
In (c) - (m) below, the path of the folder in which the thermal images prepared in Fig. 4.1 are stored and the path of the folder in which the annotation results are stored are set.

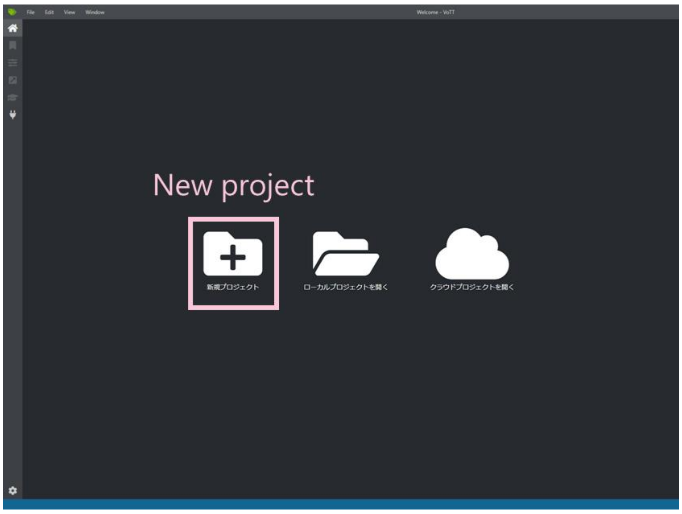


Fig.4.2 VoTT startup screen

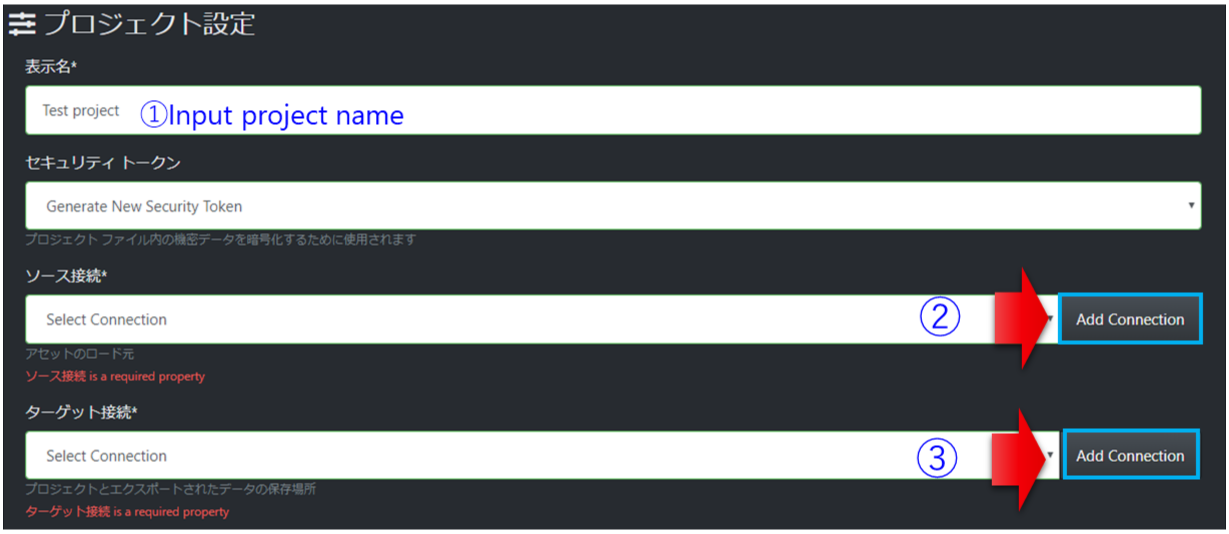


Fig.4.3 Project settings

(c) Click Add Connection on the right side of Figure 4.3 ② Source Connection.

(d) You should see the connection settings shown in Figure 4.4. For the display name, enter a name that identifies the folder where the thermal images are stored, such as Image folder.

(e) Select "Local File System" for "Provider" and set the "Image" folder in Figure 4.1.

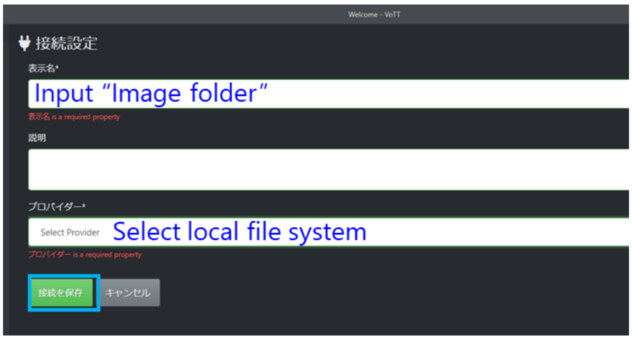


Fig.4.4 Source connection settings

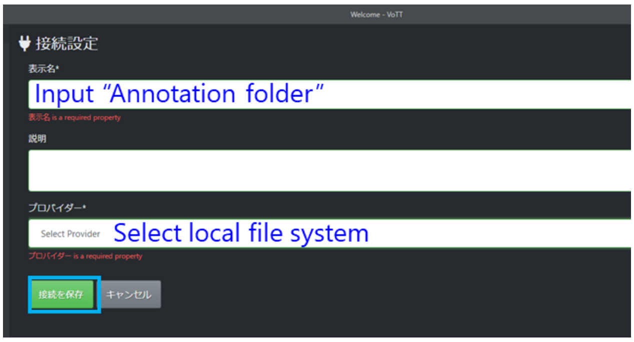


Fig.4.5 Target connection settings

(f) Click "Save Settings" at the bottom left.

(g) To return to the project setting screen shown in Fig. 4.3, click Add Connection on the right side of ③"Target Connection."

(h) When the connection settings shown in Figure 4.5 are displayed, enter a display name that identifies the folder in which annotation results will be stored (such as Annotation folder).

(i) Select "Local file system" for "Provider" and set the "Annotation folder" in Figure 4.1.

(j) Click "Save Settings" at the bottom left.

(k) When you return to the project setting screen, in the column under ④ "Source connection" in Fig. 4.6, set the folder to the Image folder set in Fig. 4.4.

(l) In the column under ⑤ “Target connection” in Fig. 4.6, set it to the Annotation folder set in Fig. 4.5.

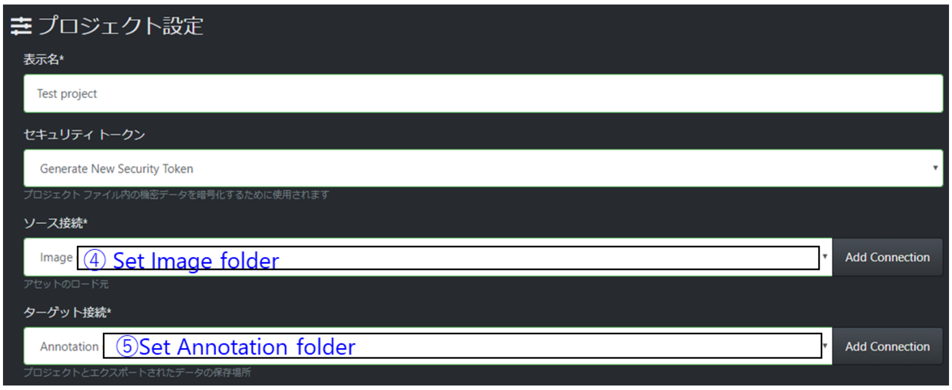


Fig.4.6 Source and target connection settings in a project setup

(m) Enter the class you want to detect (Human) in the additional blank at the bottom of the project settings shown in Figure 4.7 and press Enter.

Note that video settings are not used in this manual.

(n) Click "Save Project" at the bottom of the screen.

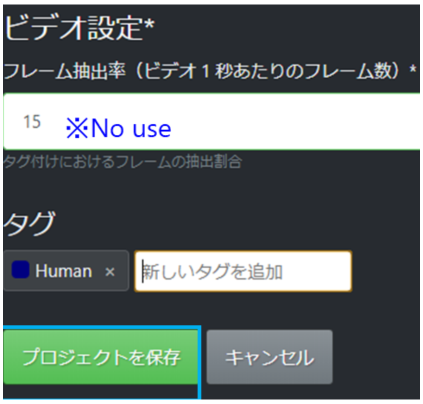


Fig.4.7 Tag settings

・Annotation working

Figure 4.8 illustrates an example of annotation work.

(a) Select the image you want to annotate from the thumbnail image on the left.

(b) Select the square from the toolbar at the top of the screen. Or press the "R" key on your keyboard.

(c) Circle the area of the photo with your mouth and click on the corresponding tag (human here) in the tag area on the right. Or key in the tag number (here "1").

(d) When you have finished tagging an image, go back to (a) to select the next image and repeat tagging.
You can save your work by pressing the Save button at the top of the screen.

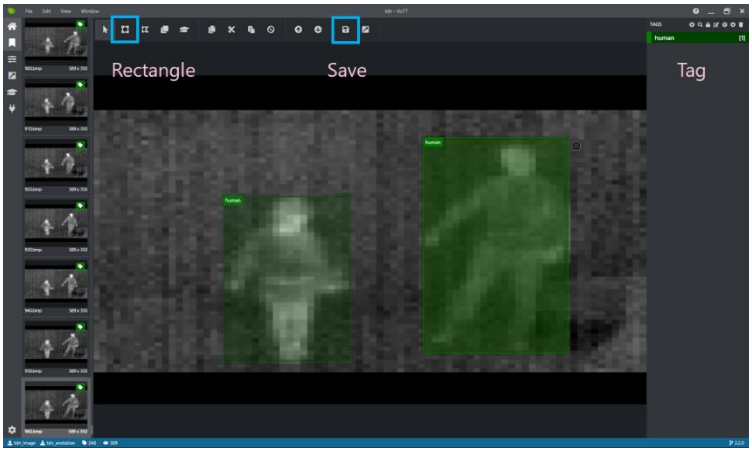


Fig.4.8 Annotation working screen

< Annotation working TIPS >

・The “s” key takes you to the next image.

・As shown in Fig. 4.9, a selected tag can be automatically assigned to a frame placed on an image by clicking a tag and clicking a key mark. You can also release it using the same procedure.

・As shown in Fig. 4.10, the color of the tag can be changed arbitrarily by clicking the part of the color bar to the left of the tag name and clicking the pen mark at the top. Tag color does not affect detection results.

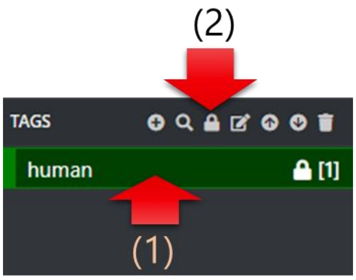


Fig.4.9 How to fix a tag

Fig.4.10 Changing the color of tags

・Export annotation results

(a) When the annotation work is finished, click the export button on the left of the screen to display the export settings screen shown in Fig. 4.11. Set Provider to Pascal VOC and Asset Status to All Assets *. (* You can change the asset status to another setting if necessary.) The standard test/training split should be 80% and adjust according to the number of thermal images and the convergence status of model training.

(b) Click "Save export settings" at the bottom of the screen.

(c) As shown in Figure 4.12, by pressing the export button at the top of the annotation work screen, the annotation results are exported in a format that can be input to the Model training tool. The data is saved under the folder name "display-name-export" in the folder of the target connection set up in Figure 4.6.

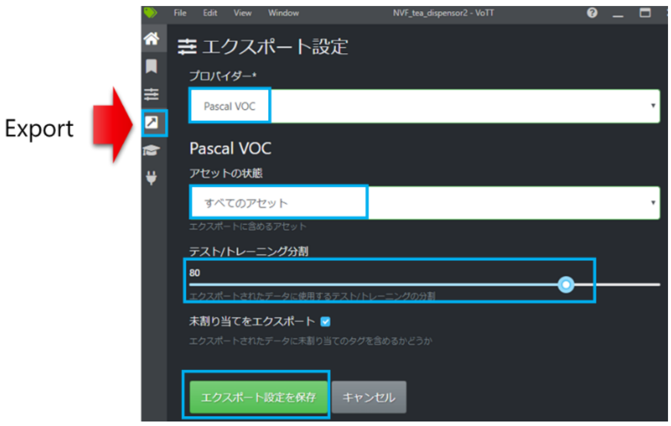


Fig.4.11 Export settings

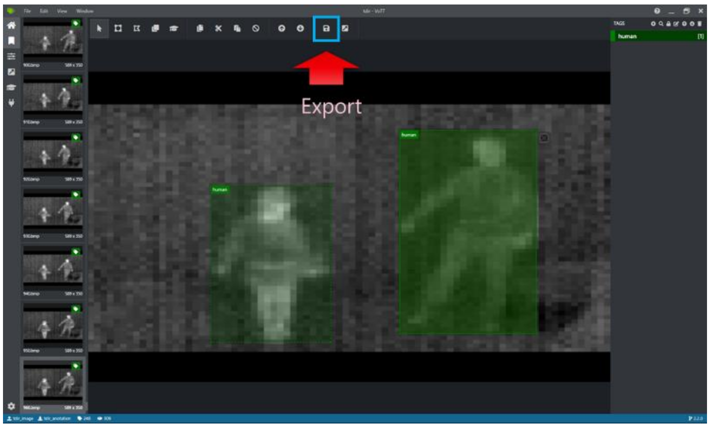


Fig.4.12 Running export

4.3 Path replacing tool for annotation data

Annotation data (.vott and .json) before exporting cannot be moved and used as is because each file contains the absolute path of the destination. For this reason, we provide an annotation data path replacing tool that can be used to change the destination of annotation data or merge the results of work by multiple users. Note that this tool has only been tested on annotation results of Visual Object Tagging Tool (VoTT) v2.2.0.

Figure 4.13 shows the folder structure of this tool. Table 4.1 also lists the input files.

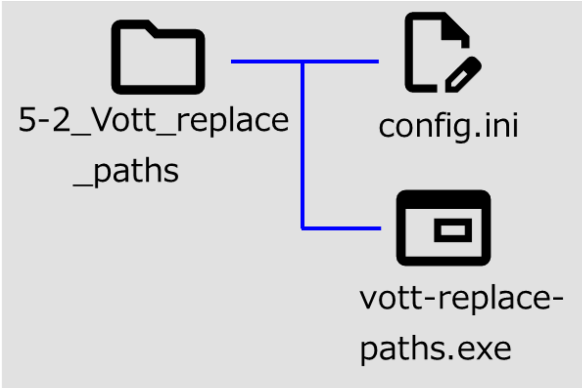


Fig.4.13 Folder structure of path replacing tool

Although the data to be converted to paths can be saved in any location, we assume here that the data is saved in C:¥Working_folder, and the folder structure below this is the same as in Figure 4.1.

Table 4.1 List of input files

Name	Description	Remarks
config.ini (setting file)	An INI format file describing various settings of vott-replace-paths	Enter your conditions
xxxx-asset.json (asset file)	A file containing the annotation results for each image	Stored in Annotation folder
Vott file	Configuration file for annotating by vott	Stored in Annotation folder
Image file	Image files to be annotated	Stored in Image folder

(a) First, obtain the security token key. Click on the Application Settings button in the lower left corner of Votto's work screen as shown in Figure 4.14.

(b) As shown in Figure 4.15, click the Copy button to the right of the target project name to copy the security token key. If you want to change the value of the security token, set the value to 32 or 44 characters.

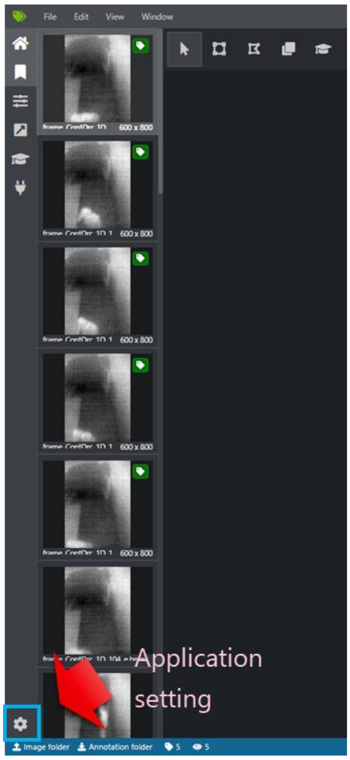


Fig. 4.14 VoTT working screen

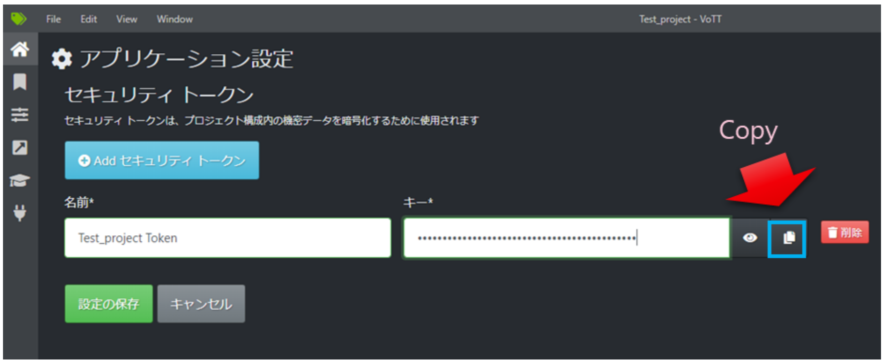


Fig. 4.15 Application settings

(c) Open config.ini in the folder “5-2_Vott_replace_paths” with Notepad, etc., set each item shown in Table 4.2, and save the file.

Table 4.2 List of setting item in config.ini

Name	Description	Example
security_token	Paste the security token key copied in Figure 4.14. 32 or 44 characters can be set.	00001111222233334444555566667777
source_connection	Set the Image folder path that contains the annotated images for path conversion.	C:¥Working_folder¥Image
target_connection	Set the Annotation folder path containing the asset and vott files for replacing path.	C:¥Working_folder¥Annotation

(d) Open the folder where vott_replace_paths.exe is saved as shown in Figure 4.16, type "cmd" in the address bar and press "Enter".

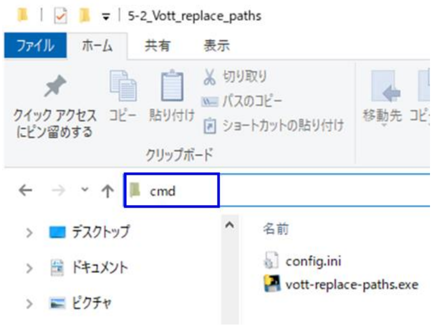


Fig. 4.16 Application setting

(e) Type the following command in the Command Prompt window and press "Enter" key.

Usage: vott-replace-paths.exe config.ini

Figure 4.16 shows the changes in the folder structure before and after the tool is executed. When the tool is executed, the file after path conversion is generated in the Annotation folder and the file before conversion is generated in the newly created backup folder, as shown in Figure 4.16 (b).

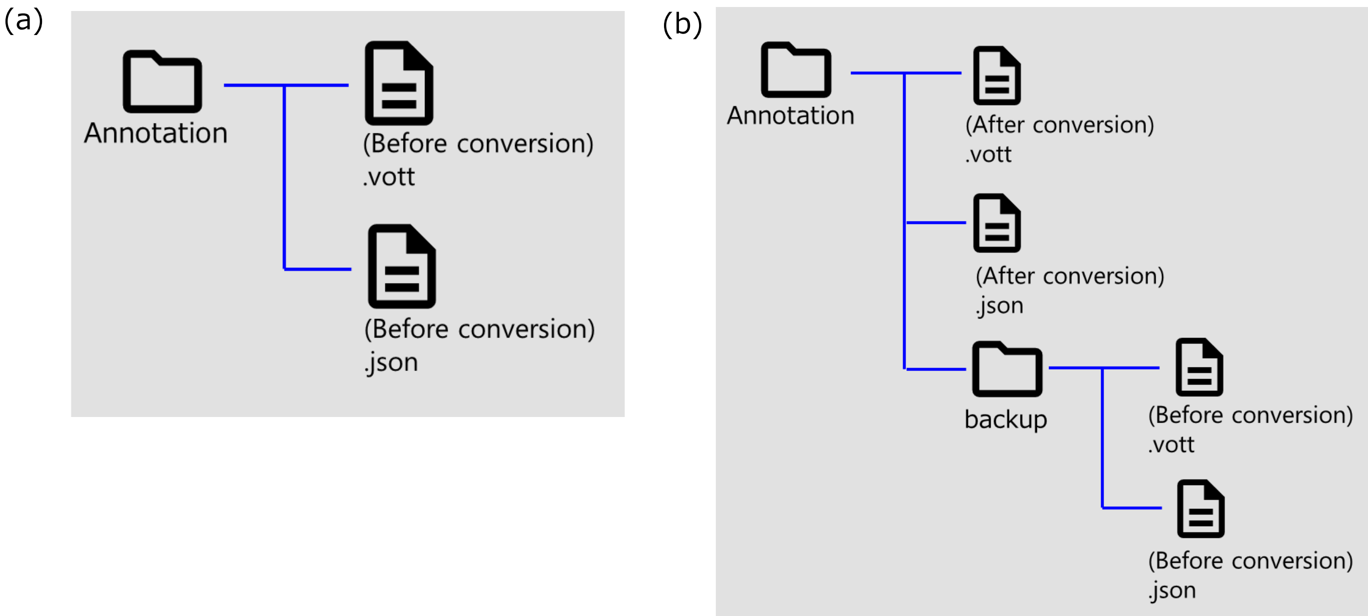


Fig. 4.17 Structure of Annotation folder (a) before replacing (b) after replacing.

(f) Confirm that the converted file is generated as shown in Figure 4.16 and close the command prompt.

(g) Move the data after path conversion to the destination (C:¥Working_folder) and use it.

5. Model training

5.1 Model training overview

An overview of model training and the software names used in each process are shown in Figure 5.1. First, create a training dataset and an evaluation dataset from the annotation data (Pascal VOC) prepared in Chapter 4. Then, a model is created by creating a parameter set for training and conducting the training using the training/evaluation data set and the parameter set. Finally, convert the created model (weight file) to a tflite file so that the microcomputer can read it.

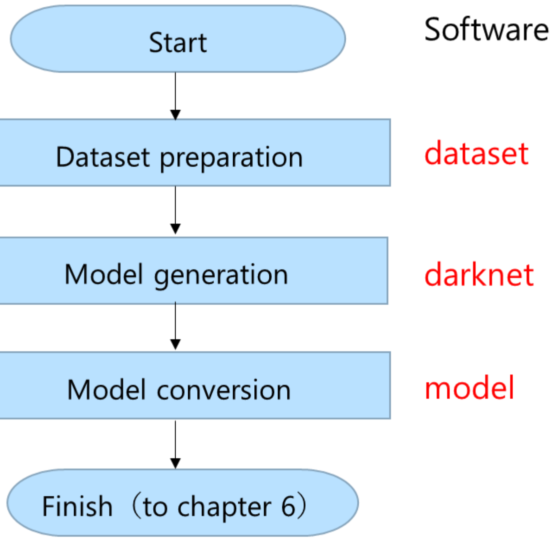


Fig.5.1 Model generation procedure

5.2 Software configuration

This chapter uses the Model training tool MTT-V002 provided by our company. Extract MTT-V002.zip to any location where the file path does not contain Japanese. The following description is based on (C: /MTT-V002) deployed directly under the C drive. The software used in model training is listed in Table 5.1. Any of the following software runs on Windows 10.

Table 5.1 MTT-V002 software configuration

Name	Overview	Folder
dataset	Convert annotation results to darknet format and then creates a training/evaluation dataset	C:¥MTT-V002¥1.dataset
darknet	Create weight files for heat source detection from training/Evaluation datasets	C:¥MTT-V002¥2.darknet
model	Change the file format of the training model from weight to tflite format. (Weight → h5 → tflite)	C:¥MTT-V002¥3.model

5.3 Training/evaluation dataset creation (dataset)

(a) Prepare data in the PascalVOC format created in Chapter 4 (Ex. PascalVOC-export_test). In the following explanation, it assumed to store in 4.data folder (C:¥MTT-V002¥4.data¥PascalVOC-export_test).

(b) Open the configuration file of dataset (config.ini) in C:¥MTT-V002¥1.dataset by Notepad or another text editor. Set each item as described in Table 5.2.

Table 5.2 List of items in the dataset configuration file

Item	Description
datasets	The annotation format. VOC fixed.
img_path	Folder path containing annotation image files. Set the -JPEGImages folder. (Example)C:¥MTT-V002¥4.data¥PascalVOC-export_test¥JPEGImages
label	Set the folder path containing the annotation result files. Set the ~ Annotations folder. (Example)C:¥MTT-V002¥4.data¥PascalVOC-export_test¥Annotations
img_type	Set the file extension of the annotation image file (image stored in img_path) with dot (.). (Example)　.png
cls_list_file	Sets the file path for the class list file (obj.names). The content of obj.names is explained in Figure 4.2. (Example)C:¥MTT-V002¥1.dataset¥obj.names
manifest_path	Include the output path of manifest file. If stored in a 1.dataset, no changes are required. (Example)　./
output_path	Include the folder path of all output files. (Example)C:¥MTT-V002¥1.dataset¥output Output the following to this folder: -darknet style annotation data (output to ¥train_dataset¥data¥obj) ・Class list file (obj.names) ・Training dataset File (train.txt) ・Evaluation dataset file (test.txt)

(c) In obj.names (C:¥MTT-V002¥1.dataset¥obj.names), enter the tag that you set when you annotated. The default setting is “Human” as shown in Figure 5.2. If you are not sure what kind of tags will be included in the annotation data, open it with notepad or something and check it out because it is listed in C:¥MTT-V002¥4.data¥PascalVOC-export _test¥pascal_label_map.pbtxt.

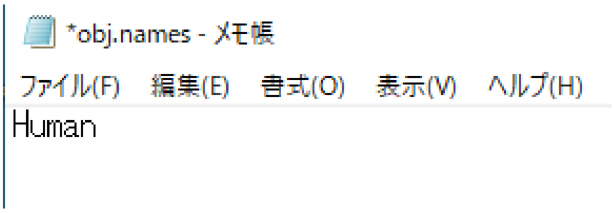


Fig. 5.2 Example configuration for obj.names

(d) Double click the executable file (dataset.exe) in C:¥MTT-V002¥1.dataset. Note that this application takes a long time to start, and it may take nearly 20 seconds before you can operate the GUI.

(e) When the screen shown in Fig. 5.3 appears, click (config.ini) in the upper right corner and select the configuration file of dataset. You can also directly key in the text display area of the file path.

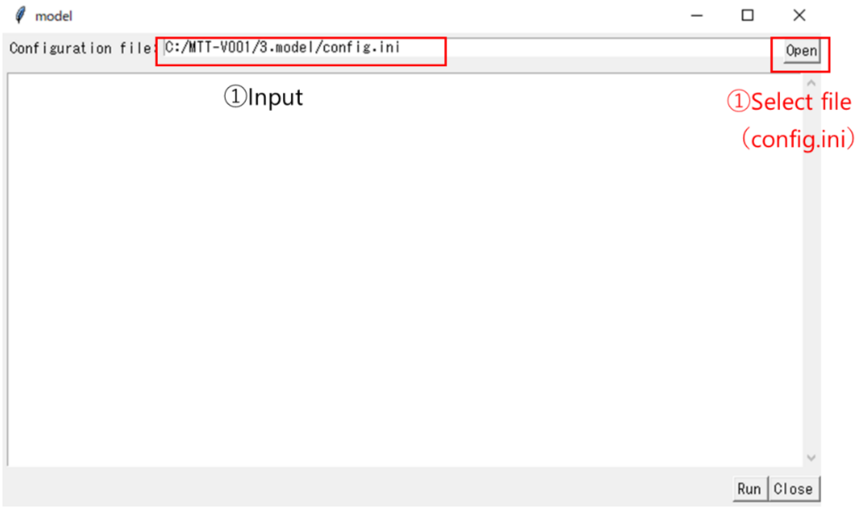


Fig. 5.3 Selecting the config file

(f) Run the dataset by clicking Run in the bottom right as shown in Figure 5.4.

Fig. 5.4 Running dataset

(g) When the log screen shows “Training/Validation data split end. Appears” as shown in Figure 5.5, the status is end. Click "Close" at the bottom right to exit.

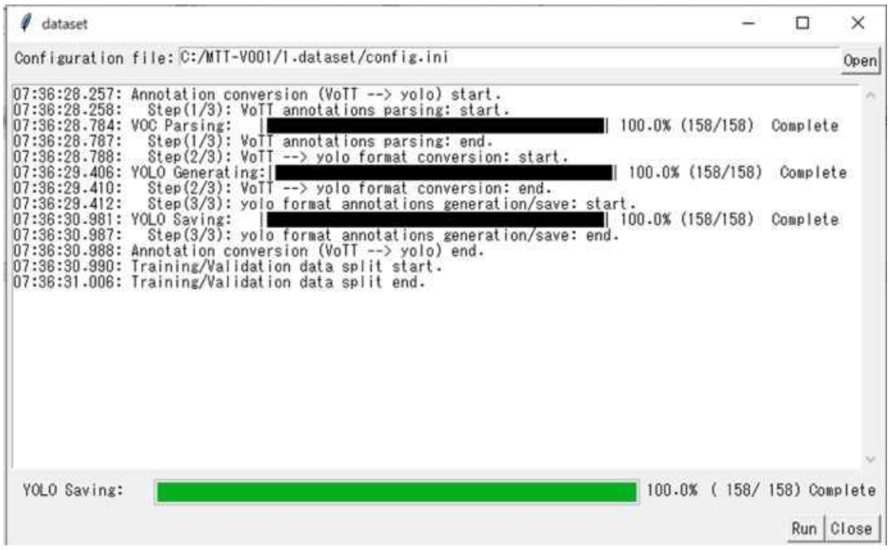


Fig. 5.5 Result of running dataset

Table 5.3 List of output files for dataset

Name	File (folder) name	Description
Manifest files	manifest.txt	List file containing paths of images (annotation image files) converted to darknet format.
annotation data of darknet style	¥output¥train_dataset¥data¥obj	A set of darknet format annotation files (*.txt) and input annotation images that convert the input annotation results to darknet format.
Training dataset file	train.txt	A file containing data extracted for training from darknet style annotation data. Only the annotation image file path is given.
Evaluation data set file	test.txt	A file containing data extracted from darknet style annotation data for evaluation. Only the annotation image file path is given.
Class list file	obj.names	A copy of the input class list file into the destination folder.

5.4 Creating a model

・Building an execution environment

(a) Click on Windows at the bottom left of OpenCV-3.4.4 from the following HP and download opencv-344-vc14_vc15.exe. https://opencv.Org/releases/page/4/

(b) Double-click opencv-344-vc14_vc15.exe to expand the folder. You can leave the folder expanded to the default location.

(c) Click “Extract” and unzip to create a folder called “opencv”. Please rename this "opencv" folder to "opencv-3.4.4".

(d) Copy "opencv-3.4.4" to a convenient place. However, do not include Japanese in the folder path. In the following, we will assume that you have created a folder called C:¥local and saved the "opencv-3.4.4" folder directly under it. (C:¥local¥opencv-3.4.4)

(e) Click "Windows + R", type "SystemPropertiesAdvanced.exe" and hit OK to open System properties. (Fig. 5.6)

(f) Click on Environment variables. (Fig. 5.7)

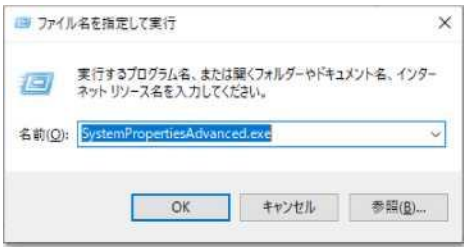


Fig. 5.6 Run with file name

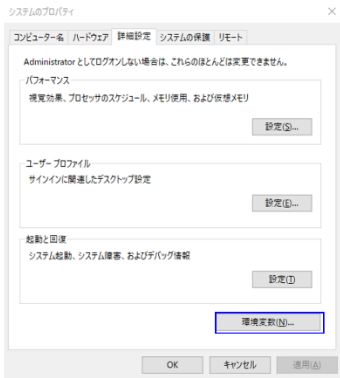


Fig. 5.7 System properties

(g) Select the system environment variable path and click Edit. (Fig. 5.8)

(h) Pass through by clicking "New" and adding the following two: (Fig. 5.9)

C:¥local¥opencv-3.4.4¥build¥bin

C:¥local¥opencv-3.4.4¥build¥x64¥vc15¥bin

(i) Click OK and close the screen.

Fig. 5.8 System environment variables

Fig. 5.9 Editing environment variable names

(j) Click on New in the system environment variable, and set the variable name to "OPENCV_DIR" and the value to “C:¥local¥opencv-3.4.4¥build” (Fig. 5.10).

(k) Click OK twice and close the screen.

Fig. 5.10 New system variables

(l) Press "Windows + R", type "cmd" and click OK.

(m) “Type where 0pencv_version” and press Enter. If the error is not displayed, the path is passed. (Figure 5.11)

If you see an error, restart your PC and check again to make sure the path is working.

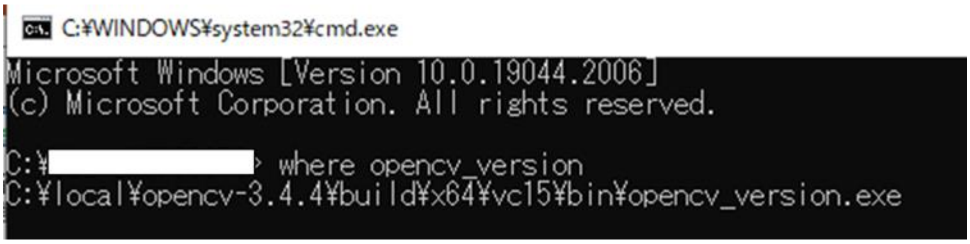


Fig. 5.11 Command prompt confirmation

Calculate anchor parameters by using darknet

Here, we calculate the anchor parameters to be used for training. Anchor parameters are reference detection frames for detecting heat sources, and by setting them, models can be trained efficiently.

(a) Open C:¥MTT-V002¥2.darknet¥obj.data in a text editor such as Notepad and set the file path for each item as described in Table 5.4. Note that only some items can be used when calculating anchor parameters, but other items can also be used for later model training, so set them here.

Table 5.4 Configuration of obj.data file

Item	Description
classes	Set the number of classes for detection (Not used when calculating anchor parameters)
train	Set the path for train.txt. (Example) C:¥MTT-V002¥1.dataset¥output¥train.txt
valid	Set the path for test.txt. (Not used when calculating anchor parameters) (Example) C:¥MTT-V002¥1.dataset¥output¥test.txt
names	Set path for obj.names. (Not used when calculating anchor parameters) (Example) C:¥MTT-V002¥1.dataset¥output¥obj.names
backup	Set a storage location for trained models, including in-progress training. (Not used when calculating anchor parameters) (Example) backup/

(b) Open the 2.darknet folder as shown in Figure 5.12, type “cmd” in the address bar and press Enter.

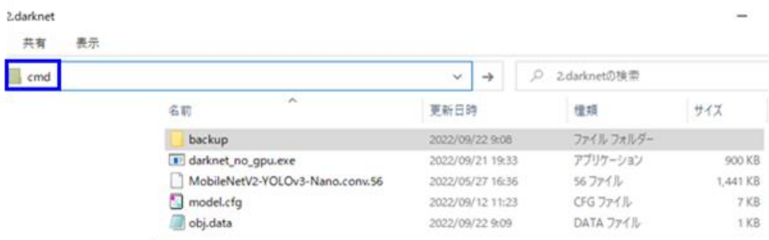


Fig. 5.12 Command prompt startup

(c) Type the following command at the command prompt and press Enter.

```
darknet_no_gpu.exe detector calc_anchors obj.data -num_of_clusters 6 -width 96 -height 96 -show
```

Table 5.5 Configuration of darknet (when calculating anchor parameters)

Name	Description
Executable files	darknet_no_gpu.exe
Argument	First argument: detector → Set detector mode Second argument: calc_anchors → specifies anchor calculation mode Third argument: data file path → Set path for obj.data file Fourth argument: -num_of_clusters AA → Set the number of anchors to AA Normally, AA should be 6 Fifth argument: -width BB → Set image width at detection as YY Normally, BB should be 96. Sixth argument: -height CC → Set the image height at detection as CC Normally, CC is 96 Seventh argument: -show → Show anchor calculation results
Import File	obj.data

Figure 5.13 shows examples of anchor parameter calculation results. Close the cluster screen shown in Fig. 5.14 when calculating. It creates cloud.png and anchors.txt in the “2.darknet” folder.

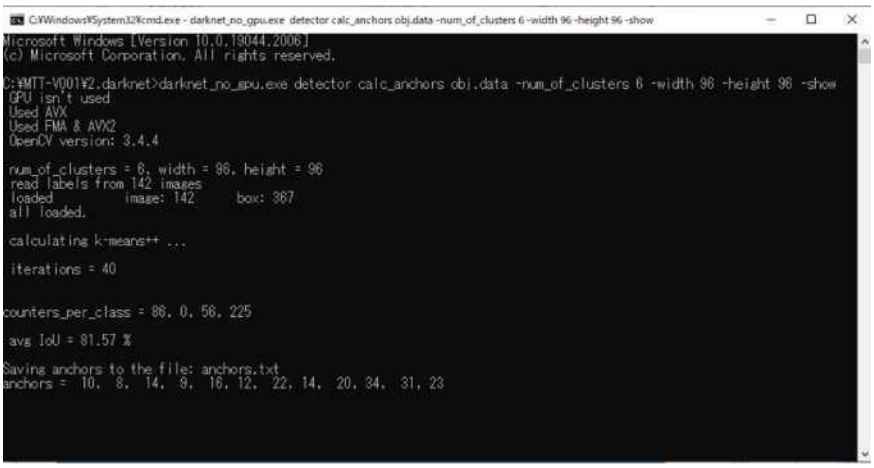


Fig. 5.13 Anchor parameter calculation results

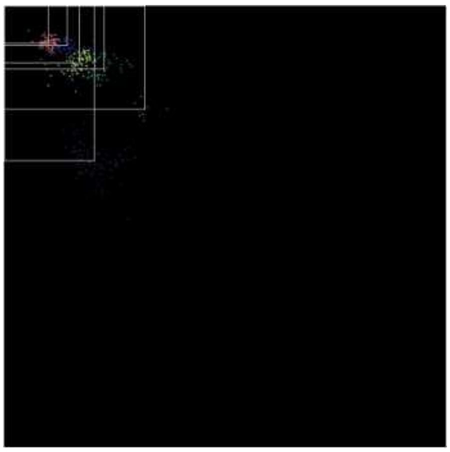


Fig. 5.14 Clusters

< Supplementary information>

If you get an error that says "MSVCR 100.dllj not found" when you run darknet_no_gpu.exe,

Install Microsoft Visual C++ 2010 Service Pack.

https://www.microsoft.com/download/details.aspx?id=26999

·Setting up the training parameter configuration file (model.cfg)

(a) Open C:¥MTT-V002¥2.darknet¥model.cfg by a text editor such as Notepad.

(b) Refer to Table 5.6 for setting. **Especially in line 554 - 638, which is written in red, make the setting every time.**

For setting points, also refer to the section on the settings in model.cfg in Section 5.6.

Table 5.6 Configuration of model.cfg

Lines	Item	Description
3	batch	Number of batches Number of data used for single error backpropagation Lower value if not working by default
4	subdivisions	Number of splits in batch count Increasing the number of splits allows you to increase the number of batches even if the memory size is small
5	width	Image width when training Set to 96
6	height	Image height when training Set to 96
16	burn_in	Number of times of prior training Set about 5% of max_batches
17	max_batches	Number of times trained Varies with data and number of classes Start with about 10,000, and increase the number if it doesn't converge
19	steps	Timing of batch numbers to reduce training rate Set about 90%, 95% of max_batches
20	scales	Rate of decrease in training rate
554	**filters**	**Number of filters** **Change according to the number of classes** **filters = (classes + 5) x 3** **Example: When the number of classes is 1, set (1 + 5) X 3 = 18.**
560	**anchors**	**Anchor size** **Copy all the values in 2.darknet/anchors.txt**
561	**classes**	**Number of classes**
632	**filters**	**Number of filters** **Change according to the number of classes** **filters = (classes + 5) x 3**
638	**anchors**	**Anchor size** **Copy all the values in 2.darknet/anchors.txt**
639	**classes**	**Number of classes**

・Note: Other settings in model.cfg

batch : Set values such as 1, 2, 4, 8, 16, 32, 64
The higher the value, the shorter the training time, but depending on the PC specs, it may not work or the detection rate may be worse because of local solutions.

burn_in : The training rate is set as a value that indicates the amount of weight change in a single training. Train at a small training rate up to the number of times set by burn_in, gradually increasing the training rate.

max_batches : The standard value is 10,000 times, with a minimum of 2,000 times multiplied by the number of classes. If Loss does not drop enough, increase number of max batches.

・Model training by darknet

(a) Start the command prompt as shown in Figure 5.12.

(b) Type the following command at the command prompt and press Enter.

```
darknet_no_gpu.exe detector train obj.data model.cfg MobileNetV2-YOLOv3-Nano.conv.56
```

You can stop training in the middle by typing Ctrl + C from the keyboard.

(c) The trained models (model_final.weights) are stored in the backup folder listed in obj.data.

It also stores a weight file for every 1000 steps and the model of the last step (model_last.weights).

Table 5.7 Configuration of darknet for training

Name	Description
Executable files	darknet_no_gpu.exe
Argument	First argument: detector Set one detector mode Second argument: train Specify one training mode Second argument: data file path: Set path of data file Fourth argument: cfg file path: cfg Set file path Fifth argument: Weight file path: Set Weight file Path
Import File	obj.data model.cfg Weight file (MobileNet V2-YOLov3-Nano.conv.56)

An example of the training result (Chrat_model.png) is shown in Figure 5.15. The Training keep until the loss on the Y-axis is no longer decreasing, with the aim of achieving a loss of less than 0.1%. If Loss increases from the middle of Iteration, it is possible that overtraining has occurred, increasing the number of thermal images to be annotated, or increasing the number of training times, or reduce (max_batches) to stop training early.

For the other arguments to darknet, see reference material [2]. Section 5.6 also contains some examples of commands.

Also, a description of YOLOv3, the object detection algorithm used, can be found in Reference [3].

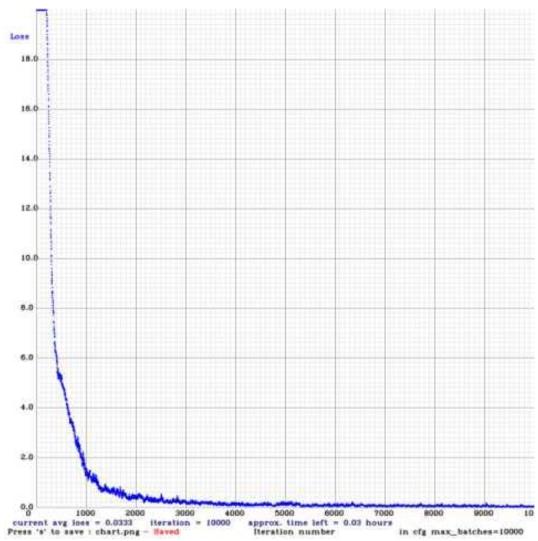


Fig. 5.15 Example of training result

5.5 Model conversion

Convert the created weight file to tflite format, which can be read into a microcomputer.

First, configure the configuration file (config.ini) used for model conversion.

(a) Open C:¥MTT-V002¥3.model¥config.ini in a text editor such as Notepad.

(b) Refer to Table 5.8 for setting.

Table 5.8 Configuration of config.ini

Item	Description
cfg_path	Set the path to the cfg file used for training. (Example) C:¥MTT-V002¥2.darknet¥model.cfg
weight_path	Sets the path for the weight file generated by training. (Example) C:¥MTT-V002¥2.darknet¥backup¥model_final.weights
input_image_size	Set input image size for training. Normally, set at 96.
quantize_mode	Set the quantization mode. "float 16", Set either "int8." Normally, int8 and set.
input_type	Set input data format. Set to int 8, uint 8, or float 32. Normally, uint8 and setting.
output_type	Set the output data format. Set to int 8, uint 8, or float 32. Normally, float 32 and set.
input_size	Set the size of the input image. Normally, set at 96.
calib_dataset_path	Set the path of the calibration image list file. (Example) C:¥MTT-V002¥1.dataset¥output¥train.txt
output_tflite_path	Set the destination file name for the converted tflite file. (Example) C:¥MTT-V002¥3.mode¥model_final.tflite

(c) Double click on C:¥MTT-V002¥3.model¥model.exe. Note that this application takes a long time to start, and it may take nearly 60 seconds before you can operate the GUI.

(d) When the screen shown in Fig. 5.16 appears, click "Open" in the upper right corner, and select the configuration file (config.ini) of the model you just created. You can also directly key in the text display area of the file path.

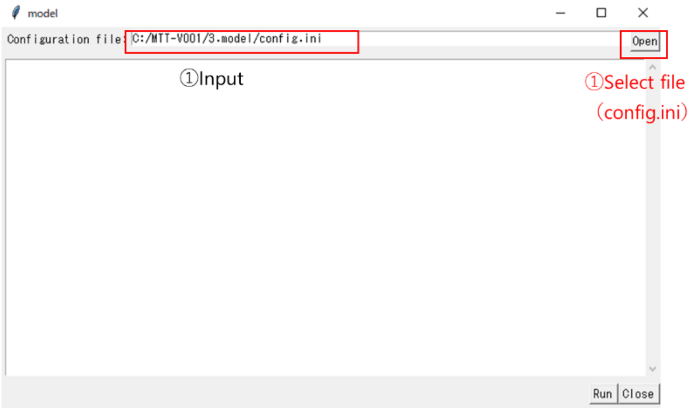


Fig. 5.16 Selecting the config file

(e) Click Run in the lower right to run the model as shown in Figure 5.17. We need to wait a while.

Fig. 5.17 Selecting Run

(f) When “convert_tflite (.h5 -- > .tfile): end.” appears on the log screen as shown in Figure 5.18, the process is complete. Click "Close" at the bottom right to exit.

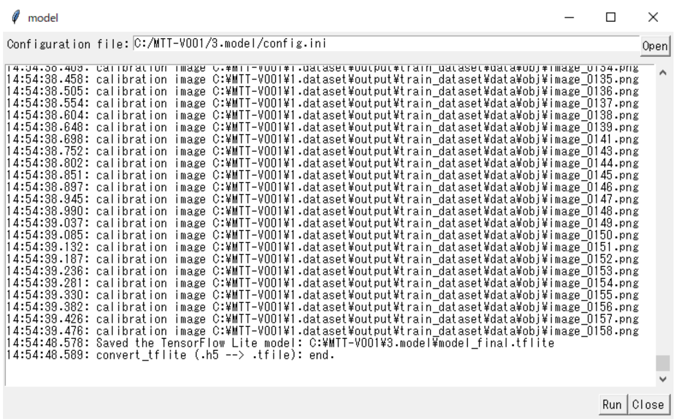


Fig. 5.18 Execution result

5.6 Appendix

Additional information about MTT-V002 is included. Please refer to it if necessary.

·Error messages

Table 5.9 shows the error factors in the configuration files for dataset and model, and examples of error messages.

Table 5.9 Error factors and error message examples

No.	Error Factors	Example error message
1	No configuration file specified	No configuration file specified.
2	The specified configuration file does not exist (Bad path)	Specified configuration file is not found. Please specify the path of the configuration file correctly.
3	The configuration file is written in a character code other than UTF -8	Configuration file error: 'c:¥tdir¥cfg¥cfg.ini'. The configuration file must be UTF-8 text file.
4	Other exceptions occur when reading configuration files	Configuration file error: 'c:¥tdir¥cfg¥cfg.ini'. Display an error message in response to the exception that occurred
5	A given key does not exist in the configuration file	Configuration file error: 'c:¥tdir¥cfg¥cfg.ini'. Key 'cfg_path' is not exist.
6	Directory does not exist when the key in the configuration file is a key that points to a directory	Configuration file error: 'C:¥TDIR¥cfg¥config.ini'. Directory not found. -> Key 'img_path': 'C:¥TDIR¥image'
7	A configuration file does not exist if its key is a key that points to a file	Configuration file error: 'C:¥TDIR¥cfg¥config.ini'. File not found. -> Key 'cls_list': 'C:¥TDIR¥obj.names'

•Darknet command example

Here are some examples of commands that are useful for training and checking models on darknet. (Reference: [2])

(a) mAP calculation

The training progress graph (Chrat_model.png) displays mAP (mean Average Precision) of object detection metric. In case of lower spec PC, PC may slow down and shut down.

```
darknet_no_gpu.exe detector train obj.data model.cfg MobileNetV2-YOLOv3-Nano.conv.56 -map
```

(b) Retraining

You can make them train from what follows if Loss doesn't go down enough. Set the path to the weight file you want to train from model_final.weights.

```
darknet_no_gpu.exe detector train obj.data model.cfg backup/model_final.weights
```

(c) Detect video and export the video of the detection result.

Set the path of the video file to be input in test01.mp4. Set the path of the file to output to test01_out.mp4.

If both input and output are to be saved in the "2.darknet" folder, set the file name with extension as shown in the example below. For files in formats other than .mp4, .wmv can be set for the input video format, but .wmv cannot be used for the output video format.

```
darknet_no_gpu.exe detector demo obj.data model.cfg backup/model_final.weights test01.mp4 -
ext_output -out_filename test01_out.mp4
```

(d) Weight file performance check

You can check the detection results and accuracy values for each class in a specified weight file. Set the weight file you want to check in model_final.weights.

```
darknet_no_gpu.exe detector map obj.data model.cfg backup/model_final.weights
```

・model.cfg settings

The contents of model.cfg after line 525 are as follows: The yellow hatching points correspond to the red areas in Table 5.6.

```
----------------------------------------------------------------------------------------
[convolutional]
filters=288
size=3
groups=288
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
filters=96
size=1
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
filters=384
size=1
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear


[yolo]
```

```
mask = 3,4,5
anchors =     9,  8,  13,  9,  18, 11,  15, 14,  22, 15,  20, 33
classes=1
num=6
jitter=.1
ignore_thresh = .5
truth_thresh = 1
random=0
################
scale_x_y = 1.1
iou_thresh=0.213
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
nms_kind=greedynms
beta_nms=0.6

################################
[route]
layers= 64

[upsample]
stride=2

[route]
layers=-1,47
###############################
[convolutional]
filters=80
size=1
stride=1
pad=1
batch_normalize=1
activation=relu
```

```
[convolutional]
filters=288
size=1
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
filters=288
size=3
groups=288
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
filters=192
size=1
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
filters=288
size=1
stride=1
pad=1
batch_normalize=1
activation=relu

[convolutional]
size=1
stride=1
pad=1
```

filters=18

activation=linear

[yolo]

mask = 0,1,2

anchors = 9, 8, 13, 9, 18, 11, 15, 14, 22, 15, 20, 33

classes=1

num=6

jitter=.1

ignore_thresh = .5

truth_thresh = 1

random=0

#################

scale_x_y = 1.05

iou_thresh=0.213

cls_normalizer=1.0

iou_normalizer=0.07

iou_loss=ciou

nms_kind=greedynms

beta_nms=0.6

----

6. Model conversion/writing to microcomputer

6.1 Preparation

This chapter describes the procedure for writing the AI model created in Chapter 5 to the MCU. For general incorporation method of AI model using by STM 32 microcomputer, see Reference [4].

・Downloading and installing software

Install the following two software programs provided by STMicroelectronics.

(1) STSW-LINK009 (version 2.0.2)

It can be downloaded from the following HP.

https://www.st.com/en/development-tools/stsw-link009.html

(2) STM32CubeIDE (version 1.8.0)

Go to the following HP and set the version to 1.8.0 at the right end of STM32CubeIDE-Win to download. The following steps cannot be performed outside of version 1.8.0.

Note that this file is very large at more than 800 MB, so download it under an environment with stable communication. Also, when downloading, if a message such as "HTTP Status 400 - Bad Request" appears, close the browser and download again. If you still can't download the file after repeated attempts, it may be set to prevent you from downloading large files. Contact your network administrator.

https://www.st.com/en/development-tools/stm32cubeide.html

・Preparing the Reference code

(a) Save the reference code (SAPM-MD01V7.zip) to any location. There is no need to unzip the folder.

Do not save any files other than the reference code in the zip file.

Also, do not change the folder name in the hierarchy below SAPM-MD01V7.

・Installing the CubeIDE plugin package

(a) Double-click stm32cubeide.exe to start Cube-IDE.

(b) Set the Workspace location to your preference and click Launch.

If necessary, set the proxy in steps (c) - (e) below. If you don't need it, go to (f).

Contact your network administrator for information about how to set proxy settings.

(c) Click “Window”-”Preferences” in the upper right corner of the workspace screen shown in Figure 6.1.

(d) In the Preferences screen shown in Figure 6.2, select “General”-“Network Connections”.

Set Active Provider to “Manual”, double-click HTTP, HTTPS, and SOCKS, and enter the Host and Port settings.

(e) Click “Apply and Close” to close the screen.

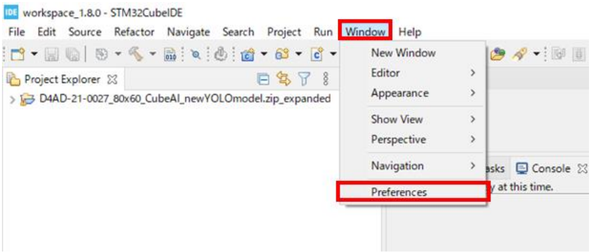


Fig. 6.1 Workspace screen

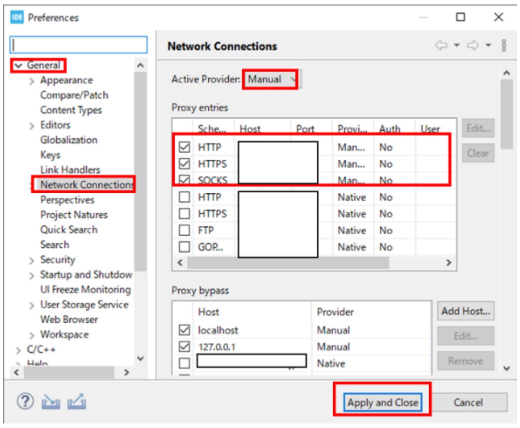


Fig. 6.2 Preferences

Install the plug-in package in the following process.

(f) Click “Help” - “Manage Embedded Software Packages” at the top of the workspace screen in Figure 6.3.

(g) Open the “STM32Cube MCU Packages” tab in the Embedded Software Packages Manager shown in Figure 6.4.

(h) Select version 1.9.0 of "STM32Cube MCU Packages for STM32H7 series" under STM32H7.

(i) At the bottom, click Install Now. If a download error appears, follow steps (c) through (e) above to make sure the proxy is set correctly.

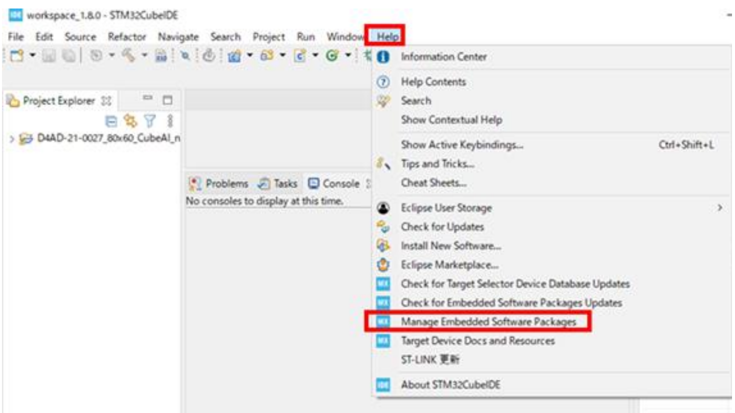


Fig. 6.3 Workspace screen

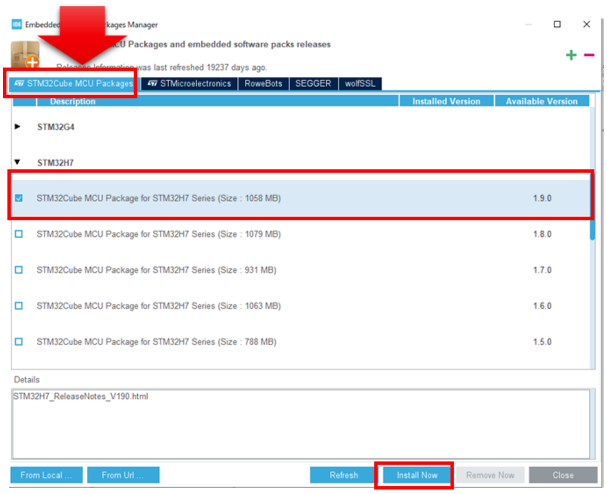


Fig. 6.4 STM32Cube MCU packages tab

(j) Click “STMicroelectronics tab” as shown in Figure 6.5. In X-CUBE-AI, select “Artificial Intelligence” version 7.0.0 and click “Install Now”.

(k) When the installation is complete, close the Embedded Software Packages Manager screen.

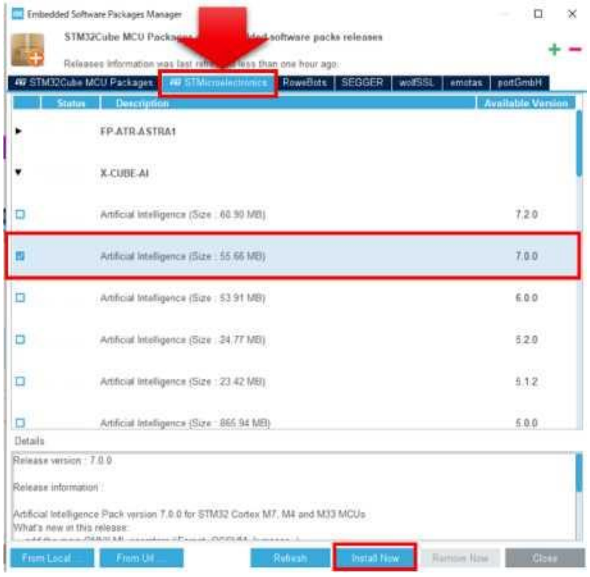


Fig. 6.5 STMicroelectronics tab

6.2 Converting AI models to MCU codes

・Reading Reference code

(a) Double-click stm32cubeide.exe to activate CubelDE.

(b) Set the location of your Workspace and click Launch.

(c) Select "File"-"Import" in the upper left of the screen, and click "Next"

(d) Select "Projects from Folder Archive" and click "Next" as shown in Figure 6.6.

(e) Select "Archive", as shown in Figure 6.7, and select SAPM-MD01V7.zip.

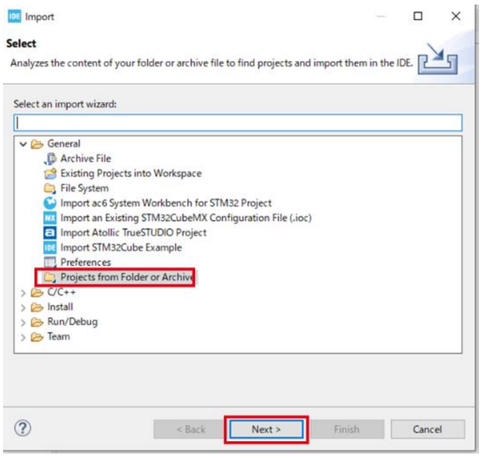


Fig. 6.6 Import wizard

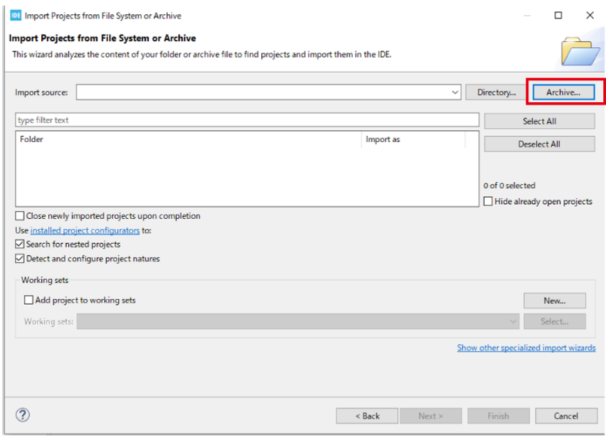


Fig. 6.7 Import file selection

(f) In Fig. 6.8, only the lower folder of the 2 folders surrounded by the red box is checked and "Click Finish".

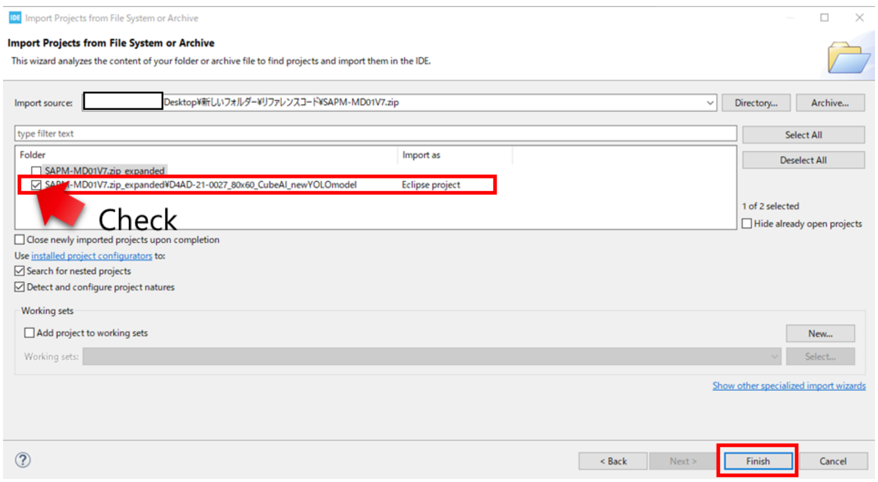


Fig. 6.8 Import file selection (after selection)

(g) Make sure that the imported file is displayed in Project Explorer on the left as shown in Figure 6.9. If Project Explorer is not displayed on the left side of the screen, you can display after select “Window”-“Show View”-“Project Explorer”.

Do not change the project name (D4AD-).

(h) Select "D4AD-21-0027_80x60_CubeAI_newYOLOmodel" in the second line of the Project Explorer shown in Figure 6.9 and click "Build".

(i) In the Console tab area, 6.10 type Build Finished. Wait until you see 0 errors, 1 warnings. One Warning appears below, but you can safely ignore it. ('ai run' defined but not used)

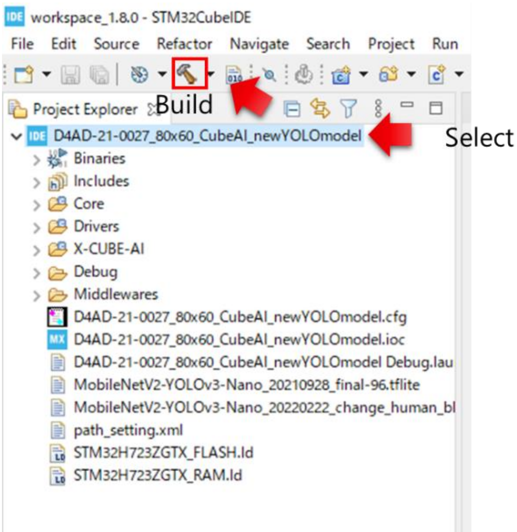


Fig. 6.9 Workspace screen

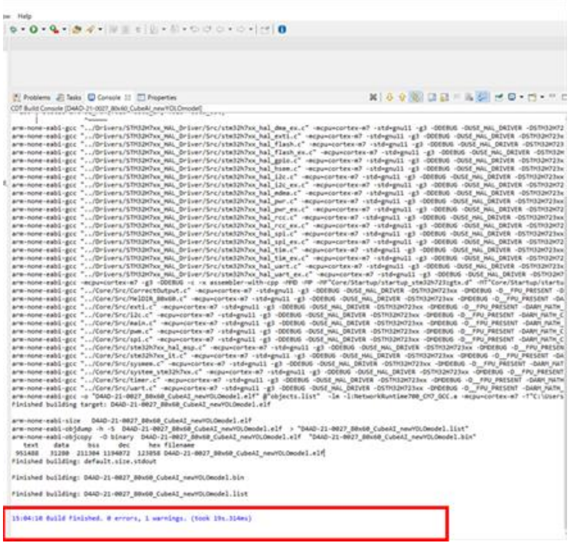


Fig. 6.10 Build result

Manual of MTT-V002 for training AI model

CTSA-231012-02

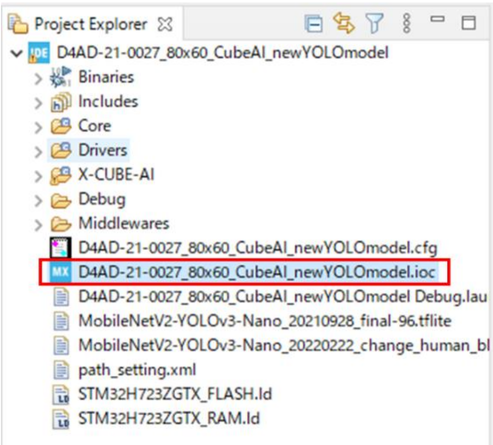
・AI model conversion

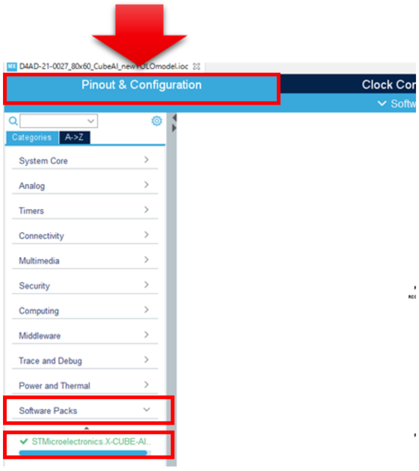
(a) Double-click "D4AD-21-0027_80x60_CubeAI_newYOLOmodel.ioc" in Project Explorer, as shown in Figure 6.11.

(b) Select Pinout & Configuration and click “Software Packs”–“STMicroelectronics X-Cube-AI” in the lower left, as shown in Figure 6.12.

(c) If a window of Figure 6.13 appears, click No. In case of Yes, it will affect communication with MelDIR.

(d) Select “Network” tab and select the AI model (tflite file created in Chapter 5, in example C:¥MTT-V002¥3.model¥model_final.tflite) from “Browse”, as shown in Figure 6.14. Then click “Analyze” button.





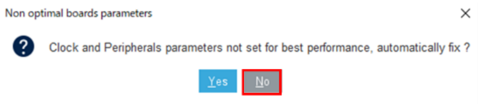


Fig. 6.13 Alarm of automatically clock

Fig. 6.11 Project explore

Fig. 6.12 Pinout & Configuration

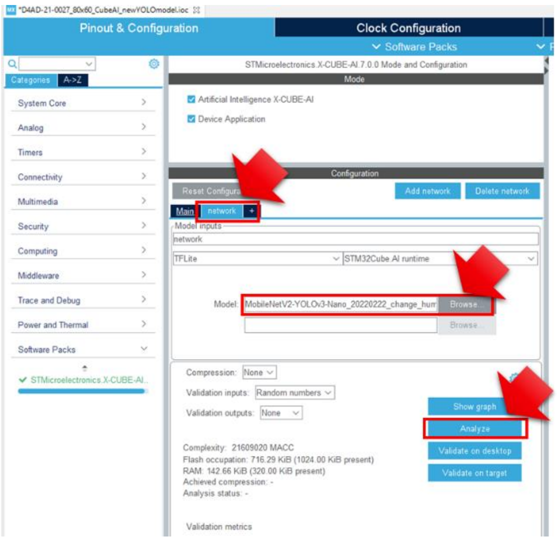


Fig. 6.14 Selecting and analyzing a tflite file

(e) The analysis results are shown as in Figure 6.15, so make sure that weights (ro) is less than 1 MB.

```
model_name          : MobileNetV2YOLOv3Nano_20220222_change_human_blur_and_cat_blur_with_20210728_final96
model_hash          : ea49717ff4f0417523d95de549073834
input               : input_1 [27648 items, 27.00 KiB, ai_u8, scale=0.003921568859368563, zero_point=0, (1, 96, 96, 3)]
inputs (total)      : 27.00 KiB
output              : conversion_71 [189 items, 756 B, ai_float, float, (1, 3, 3, 21)]
output              : conversion_80 [756 items, 2.95 KiB, ai_float, float, (1, 6, 6, 21)]
outputs (total)     : 3.69 KiB
params #            : 706,474 items (716.29 KiB)
macc                : 21,609,020
weights (ro)        : 733,480 B (716.29 KiB)
activations (rw)    : 114,656 B (111.97 KiB)
ram (total)         : 146,084 B (142.66 KiB) = 114,656 + 27,648 + 3,780
```

Fig. 6.15 Analysis result

・Code generation

(a) Click the code generation button indicated by the arrow in Figure 6.16. If the pop-up shown in Fig. 6.17 appears, click "Yes".

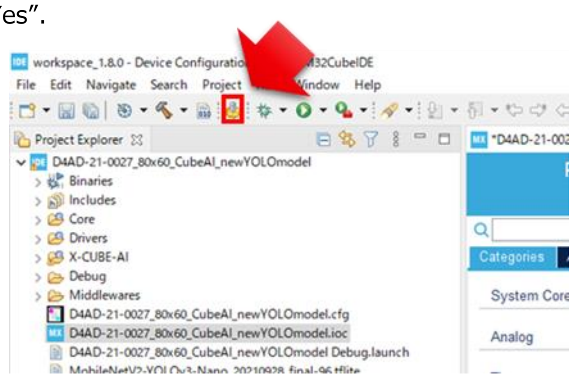


Fig. 6.16 Code generation

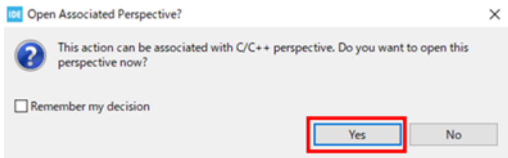


Fig. 6.17 Pop-Up of C/C++

(b) Click "File"-"Properties" in the upper left of the workspace screen as shown in Figure 6.18.

(c) Click the arrow to the right of Location from "Resource" in properties as shown in Figure 6.19.

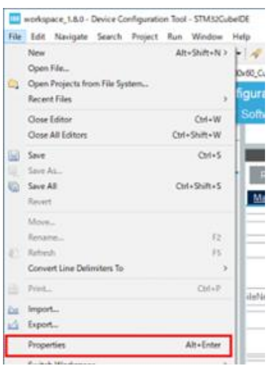


Fig. 6.18 File tab

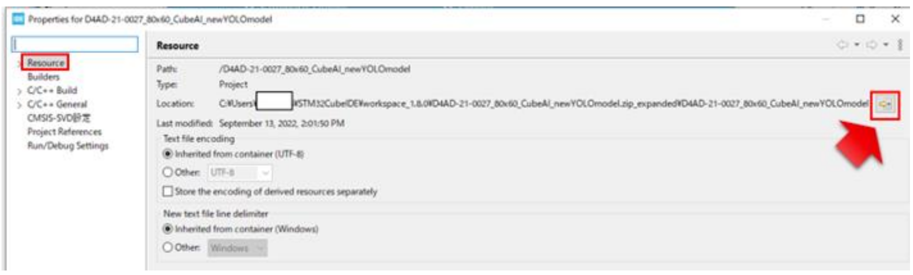


Fig. 6.19 Properties tab

(d) After opening the workspace folder, open the "Xcube-AI" App folder shown in Figure 6.20.
¥workspace_1.8.0¥D4AD-21-0027_80x60_CubeAI_newYOLOmodel.zip_expanded¥D4AD-21-0027_80x60_CubeAI_newYOLOmodel¥X-CUBE-AI¥App

Fig. 6.20 X-CUBE-AI/App folder

(e) Copy “app_x-cube-ai.c” from MTT-V002 folder to the above folder. Although warning about overwriting will appear, please continue.

(f) Return to the CubeIDE screen and double-click “X-cube-AI/App/app_x-cube-ai.c” in Project Explorer as shown in Figure 6.21.

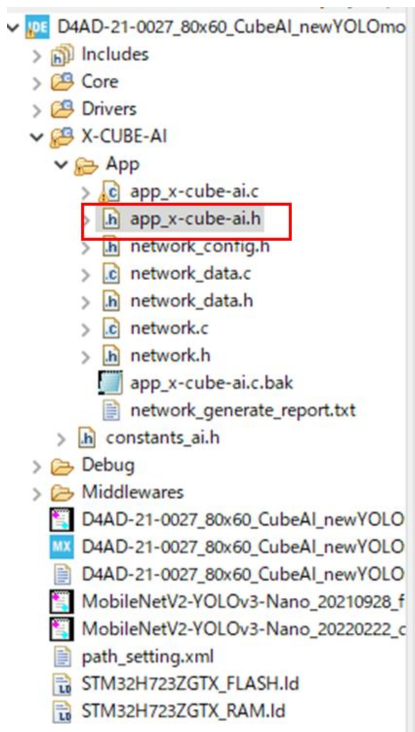


Fig. 6.21 Project explorer

(g) anchors.txt created in Chapter 4 on line 547 - 548 in Figure 6.22

(Figure 6.23, for example C:¥MTT-V002¥2.darknet¥anchors.txt). The numbers in blue in Figures 6.22 and 6.23 indicate the order of entry. Note that line 542 in Figure 6.22 is filled with ④ - ⑥ of anchors.txt.

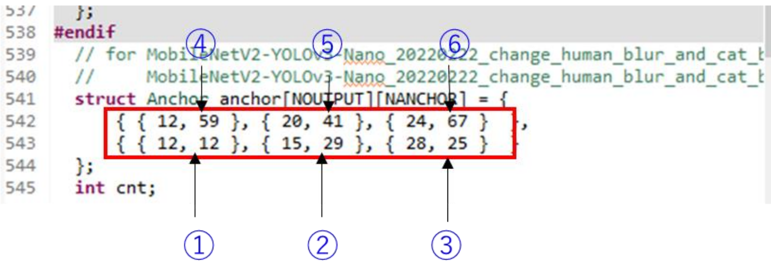


Fig. 6.22 app_x-cube-ai.c

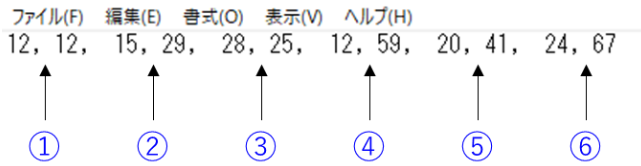


Fig. 6.23 anchors.txt

(h) In Project Explorer, double-click “X-cube-AI¥App¥app_x-cube-ai.h” to open it.

(i) Set the number to the right of NCLASS on line 110 of Figure 6.24 to the desired number of classes. (The default is 2, and in the example in Figure 6.24, the number of classes is 1.)

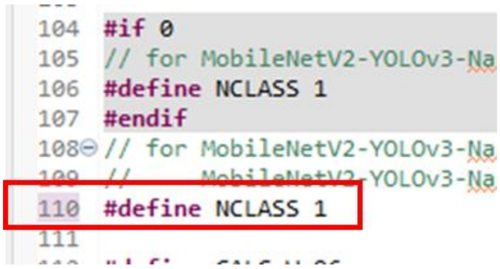


Fig. 6.24 app_x-cube-ai.h

(j) In Project Explorer, double-click “Core¥Src¥main.c” to open it.

(k) Line 1409 as shown in Figure 6.25

RCC_OscInitStruct.HSEState = RCC_HSE_BYPASS;.

should be changed as follows

RCC_OscInitStruct.HSEState = RCC_HSE_ON;

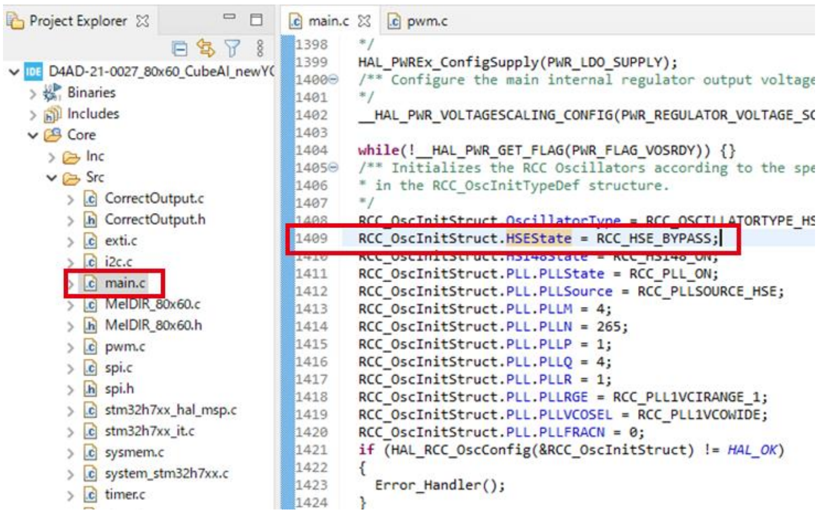


図 6.25 Quartz Crystal Setting

6.3 Writing to the MCU

(a) Connect a demo kit and a PC with a USB cable. For connection, use a USB Type-C cable for data communication.

(b) Select Middlewares in the Project explorer and click RUN in the upper right, as shown in Figure 6.26.

(c) When the ST-LINK firmware update shown in Fig. 6.27 appears, click "No".

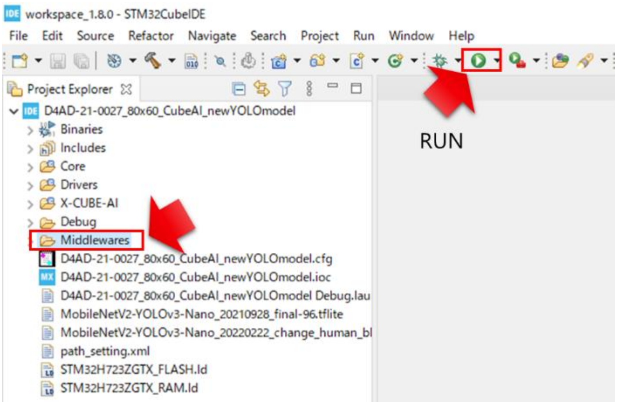


Fig. 6.26 Execution of writing

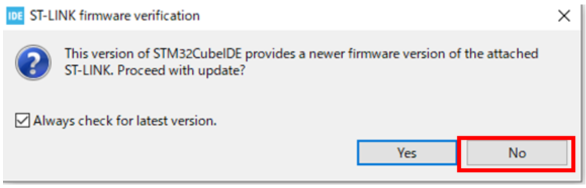


Fig. 6.27 Confirming ST-LINK firmware updates

(d) When the message shown in Fig. 6.28 appears, writing is complete.

(e) Once the USB cable is removed from the demo kit and reconnected, thermal imaging using the demo kit becomes possible.

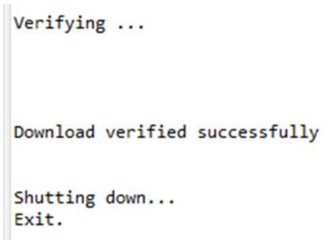


Fig. 6.28 Writing result

・Precautions for taking thermal image

As explained on page 7, the ceiling direction of the image taken by MelDIR must match the ceiling direction of the AI model (annotation image). Be careful of the orientation of MelDIR when taking thermal image.

7. References and related materials

[1] Microsoft VoTT Instructions

https://github.com/microsoft/VoTT

[2] Description of darknet YOLOv3

https://github.com/AlexeyAB/darknet

[3] YOLOv3 literature

J. Redmon et al., "YOLOv3: An Incremental Improvement https://arxiv.org/pdf/1804.02767.pdf

[4] Description of embedded AI using STM 32 microcomputer

https://www.st.com/content/st_com/ja/support/learning/stm32-education/stm32-moocs/Introduction_to_STM32CubeAI_MOOC.html

Trademarks

Manual of MTT-V002 for training AI model

CTSA-231012-02

Revision histories

No.	Date	Revision	
		Pages	Points
1.0	Nov. 2023	—	New

Manual of MTT-V002 for training AI model

CTSA-231012-02

Keep safety first in your circuit designs!

Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi Electric Semiconductor product best suited to the customer’s application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- The training tool in this document are provided for making AI model by using thermal image datasets from MelDIR. The training tool and manual are prohibited to use for datasets from the other sensor except for MelDIR.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party’s rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Electric Semiconductor product distributor for the latest product information before purchasing a product listed herein.
  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Electric Semiconductor home page (https://www.mitsubishielectric.com/semiconductors/).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to Evaluated all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Electric Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
  Any diversion or re-export contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Electric Semiconductor product distributor for further details on these materials or the products contained therein.