



# AI-Core Platform

# MANUAL

**VERSION: 2025 SE**

© Akkodis Germany Consulting GmbH, Sindelfingen 2025 - all rights reserved

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, eletric, mechanical, photocopying, recording, or otherwise, without the prior written permission of Akkodis Germany Consulting GmbH.

---

Akkodis Germany Consulting GmbH  
Flugfeld-Allee 12  
D-71063 Sindelfingen  
Akkodis Germany Consulting GmbH  
Tel.: +49(0)7031 686-3333

---

# Table of Contents

1. Installation Instructions .....	2
1.1. System Requirements .....	2
1.1.1. Hardware Environment Configuration .....	2
1.1.2. Software Environment Configuration .....	3
1.2. Installation Procedure .....	9
1.3. License .....	11
2. Overview .....	14
2.1. Machine Learning Process .....	16
2.2. Plugin-based Framework .....	18
2.3. Work Process .....	19
3. Configurations .....	24
3.1. Project Management .....	25
3.1.1. Create New Project .....	25
3.1.2. Remove Project .....	26
3.1.3. Rename Project .....	27
3.1.4. Import Project .....	27
3.1.5. Load Project .....	27
3.2. System Information .....	28
3.2.1. About .....	28
3.2.2. Account .....	29
3.2.3. Dependencies .....	30
3.3. Help .....	31
3.4. Test Mode Setting .....	32
3.4.1. General .....	32
3.4.2. Test Mode Settings .....	32
4. Training Mode .....	34
4.1. Create or Load Training Project .....	35
4.1.1. Create Training Project .....	35
4.1.2. Load Training Project .....	36
4.2. Prepare Video .....	37
4.2.1. Create New Video .....	37
4.2.2. Using Reference Points .....	40
4.2.3. Marker based Perspective Transformation .....	41
4.2.4. Merge From Images .....	42
4.2.5. Synthetic Data Generation .....	43
4.2.6. Load Video File .....	49
4.3. Prelabeling .....	51
4.4. Classification .....	53

4.4.1. Create/Delete Class Set .....	54
4.4.2. Create/Delete Class .....	54
4.4.3. Check Labeling Status .....	54
4.4.4. Copy Class Set from Other Projects .....	55
4.4.5. Concatenate Class Sets .....	56
4.5. Labeling .....	57
4.5.1. Add/Delete/Rename Class .....	58
4.5.2. Add/Remove Label .....	59
4.5.3. Assistant Functions .....	60
4.5.4. Auto Backup .....	61
4.6. Training .....	62
4.6.1. Training Presets .....	63
4.6.2. Training Run .....	66
5. Test Mode .....	70
5.1. In General .....	71
5.1.1. New Testing Project .....	72
5.1.2. Load Testing Project .....	73
5.1.3. Testing Setup .....	74
5.1.4. Select Stream Source .....	74
5.1.5. Model Files .....	80
5.1.6. Run Test .....	82
5.2. HDF5 .....	87
5.2.1. Overview of HDF5 .....	87
5.2.2. AI-Core HDF5 Implementation .....	87
5.2.3. Save Extracted Text Information .....	87
5.2.4. Save Image Results from PixelTrace and ONNX module .....	88
5.2.5. Opening HDF5 file .....	89
5.3. Icon Detection .....	89
5.3.1. Icon Detection Settings .....	90
5.3.2. Test Result Schema .....	91
5.3.3. Detection Filter .....	91
5.4. Color Detection .....	93
5.4.1. Color Detection Settings .....	93
5.4.2. Color Definition .....	95
5.4.3. Test Result Schema .....	96
5.5. Text Detection .....	97
5.5.1. Text Detection Settings .....	98
5.5.2. Test Result Schema .....	100
5.5.3. Train with Customized Font Styles .....	100
5.5.4. Detection Filter .....	102
5.6. Needle Detection .....	104

5.6.1. Needle Detection Settings .....	106
5.6.2. Test Result Schema .....	107
5.7. Line Detection .....	108
5.7.1. Line Detection Settings .....	108
5.7.2. Line Detection Color Filtering .....	110
5.7.3. Installing GIMP Portable .....	113
5.7.4. Test Result Schema .....	117
5.8. Audio Matching .....	119
5.8.1. Audio Matching Settings .....	119
5.8.2. Test Result Schema .....	123
5.9. Screen Analysis .....	124
5.9.1. Screen Analysis Settings .....	124
5.9.2. Test Result Schema .....	125
5.10. Pixel Trace .....	127
5.10.1. Pixel Trace Settings .....	127
5.10.2. Test Result Schema .....	129
5.11. Template Matching .....	130
5.11.1. Recording Templates .....	130
5.11.2. Template Matching Settings .....	132
5.11.3. Test Result Schema .....	134
6. Sequencer .....	135
6.1. Sequencer Definition .....	135
6.1.1. Triggers .....	135
6.1.2. Actions .....	137
6.1.3. Summary .....	137
7. Communication .....	139
7.1. Configuration of AI-Core .....	140
7.2. PROVEtech:TA .....	144
7.2.1. Configuration in TA .....	144
7.2.2. Test .....	147
7.3. MQTT .....	151
7.4. XIL .....	154
7.4.1. Overview of XIL .....	154
7.4.2. AI-Core XIL Implementation .....	155
7.4.3. AI-Core Client-Server XIL Setup .....	159
Prerequisites .....	159
Finding the Host IP Address .....	160
Reconfigure AI-Core .....	160
Copying Required Files to the Client PCs .....	161
Starting the Test Script .....	161
Troubleshooting .....	162

7.5. REST .....	163
7.5.1. Overview of REST .....	163
7.5.2. AI-Core REST Implementation .....	163
7.5.3. AI-Core using REST and XIL .....	163
7.5.4. AI-Core RemoteLauncher .....	164
Appendix A: Supported NVIDIA GPUs .....	165
Appendix B: Troubleshoot .....	167
B.1. Labeling & Training Issues .....	167
B.2. Communication Issues .....	168
Appendix C: Supported Commands .....	172
C.1. General Controls .....	172
C.2. Video Controls .....	174
C.3. Camera Setting .....	175
C.4. Icon Detection .....	175
C.5. Color Detection .....	176
C.6. Text Detection .....	176
C.7. Needle Detection .....	177
C.8. Line Detection .....	177
C.9. Audio Matching .....	178
C.10. Screen Analysis .....	178
C.11. Pixel Trace .....	179
C.12. Template Matching .....	180
Appendix D: Dependencies .....	181
Appendix E: Contact .....	186
Appendix F: Version Migration .....	187
F.1. Migration from UI:TestAId 2022 to AI-Core 2023+ .....	187
F.2. Migration from UI:TestAId 2021 to AI-Core 2023+ .....	188
Appendix G: ArUco Markers .....	190

# Chapter 1. Installation Instructions

This chapter will show you how to install AI-Core in your computer. It consists of three sections:

- **System Requirements:**

This section introduces the prerequisite of the installation. Please read it before installation carefully to make sure that your system is compatible to AI-Core.

- **Installation Procedure:**

AI-Core offers several installation options to meet different customers' requirements. You will be familiar with the whole procedure of installation and make your own installation decision.

- **License:**

AI-Core is a commercial application. It is managed by a license mechanism. Read this section, and you will learn more details of it.

---

## 1.1. System Requirements

Currently, AI-Core only support Windows 10 64bit. In the future releases, a Linux version is planned.

The core application and most of the detection plugins have no specific hardware and software requirements. However, considering the detection performance, the following configuration is recommended:

- CPU ≥ Intel i7 7. Generation
- RAM ≥ 16GB

In order to achieve real-time general detection capability of objects in video streams, AI-Core applies neural network algorithms in training and icon detection plugin. In these plugins, the neural network algorithms can be accelerated by NVIDIA graphics card. In the training plugin, it is obligatory that the training computer must be equipped with a NVIDIA graphics card, such as GeForce RTX 2080 Ti. In other detection plugin, NVIDIA graphics card is optional. However, only with the GPU acceleration, AI-Core can reach higher performance.

In the training plugin, the neural network needs to be loaded in graphics card.

### 1.1.1. Hardware Environment Configuration

NVIDIA graphics card is originally used to generate a feed of output images to a display device. Only with [CUDA](#) and its GPU-accelerated libraries, one can then apply neural networks to the GPU. Not every NVIDIA graphics card is compatible for AI-Core. The GPU compatibility depends on NVIDIA compute capabilities and memory.

## Application (UI:TestAId)

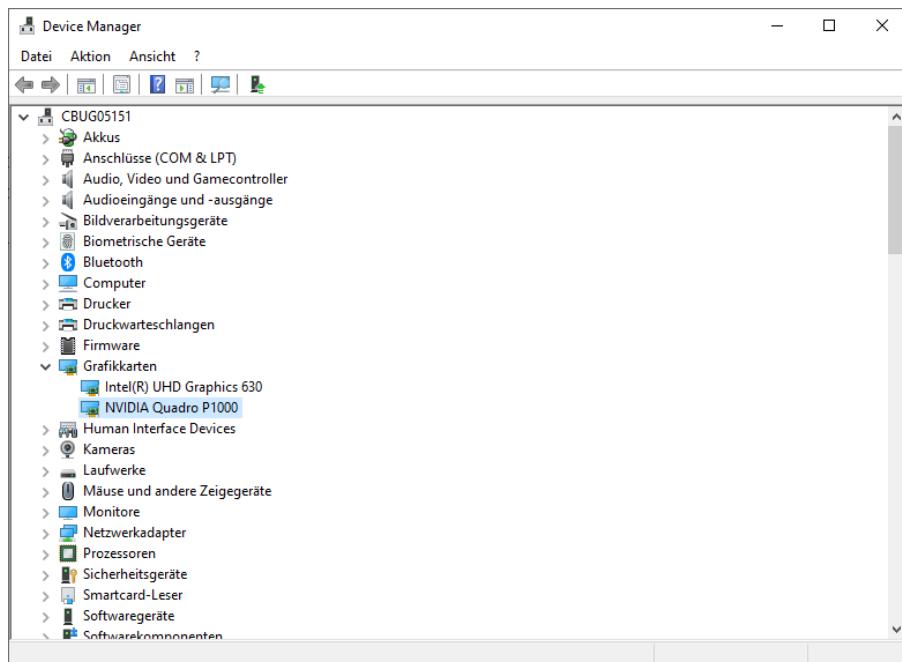
CUDA Driver, CUDA Runtime, CUDA Library

CUDA-Enabled NVIDIA GPUs

- The supported NVIDIA graphics card compute capabilities are 5.0, 5.2, 6.0, 6.1, 7.0, 7.5, 8.0 and 8.6.  
To looking for the compute capability for your GPU, you can check [link](#) or the table in appendix [Supported NVIDIA GPUs](#).
- GPU memory 4GB~11GB (for Learning) or  $\geq$  2GB (for Runtime)

If you cannot decide which GPU you should purchase, please contact [PROVEtech support](#).

If you are not sure which GPU is equipped in the computer, you can find the information in Device Manager > Display Adapters.



### 1.1.2. Software Environment Configuration

Besides of the hardware requirements, you should additionally install following software and libraries:

- NVIDIA graphics card driver  $\geq$  460.xx
- CUDA = 11.8
- cuDNN = 8.9.7 for CUDA 11.x

According to the license agreements of NVIDIA software component distribution, you need to

download the installation files and libraries by yourself. AI-Core does not provide redistribution.

The concrete steps are:

- Search the NVIDIA graphics card driver from this [page](#). Please select 64-bit version and the corresponding GPU series.

## NVIDIA Driver Downloads

### Official Advanced Driver Search | NVIDIA

**Product Type:**

NVIDIA RTX / Quadro

**Operating System:**

Windows 10 64-bit



**Product Series:**

Quadro Series

**Windows Driver Type:**

DCH



**Product:**

Quadro P1000

**Language:**

English (US)

**Recommended/Beta:**

All



**Search**

**Name**



**Version**

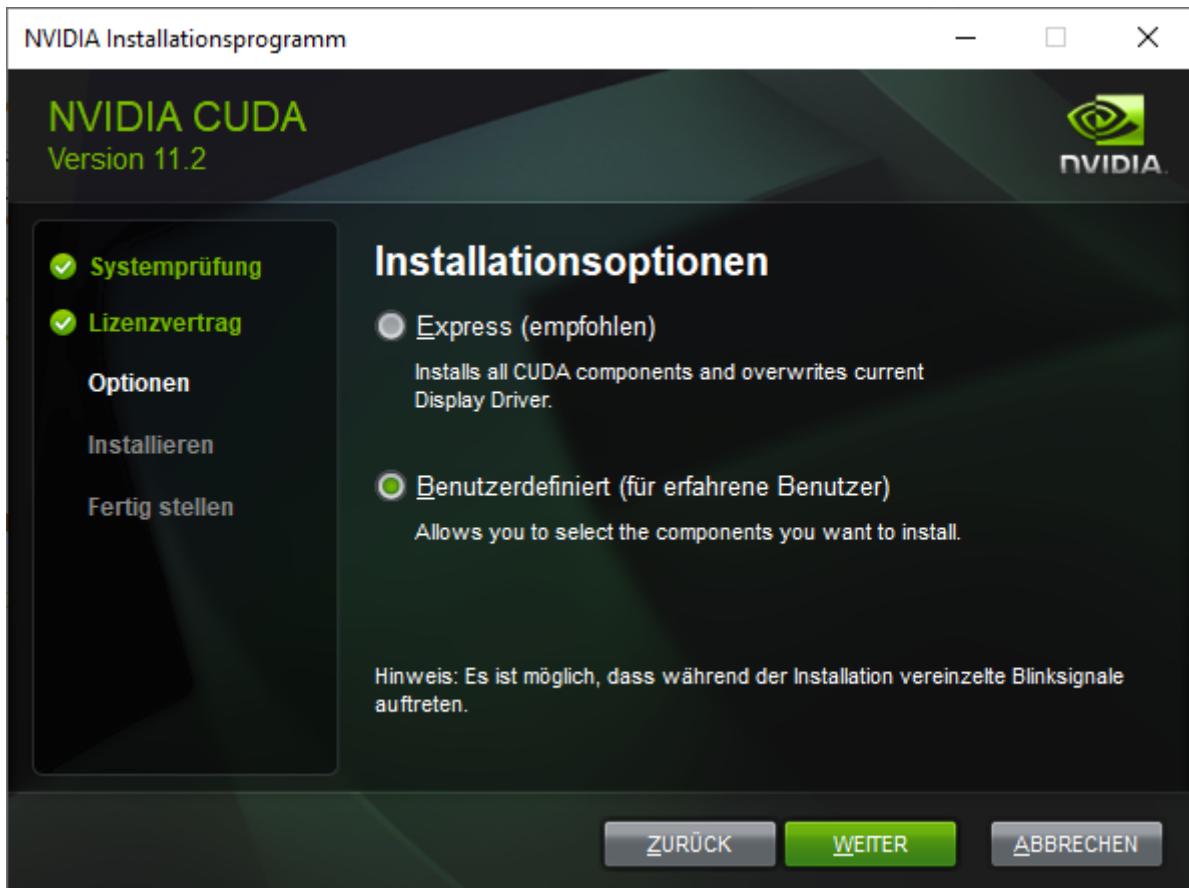
**Release Date**

<a href="#">+NVIDIA RTX / Quadro Desktop and Notebook Driver Release 525 WHQL</a>	R525 U6 (528.89)	March 30, 2023
<a href="#">+NVIDIA RTX / Quadro Desktop and Notebook Driver Release 515 WHQL</a>	R515 U7 (518.03)	March 30, 2023
<a href="#">+NVIDIA RTX / Quadro Desktop and Notebook Driver Release 470 WHQL</a>	R470 U13 (474.30)	March 30, 2023

Then, download the driver file and install it with the administrator right. The installation settings can be left as default.

- The second step is to download and install the CUDA library. Please download the installation file by [link](#). After downloading, double click the executable file with administrator right as well.

Select "Custom (Advanced)".



Only check option "Development" and "Runtime".



Then, continue the installation and you will then see the CUDA installation directory located in

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8

- Finally, you need to add cuDNN library files into the CUDA installation directory.

Go to the [page](#) and download the cuDNN library. Then, select "Archived cuDNN Releases".

If you are Windows 10/11 user, you can download the cuDNN library directly through [cuDNN 8.9.7 Windows](#).

If you are Linux user, the download [cuDNN 8.9.7 Linux](#).

# cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPL

For more information, refer to [the cuDNN Developer Guide](#), [Installation Guide](#) and [Release Notes](#)

[Download cuDNN v8.9.0 \(April 11th, 2023\), for CUDA 12.x](#)

[Download cuDNN v8.9.0 \(April 11th, 2023\), for CUDA 11.x](#)

[Archived cuDNN Releases](#)



Unpack the cuDNN zip file and copy all data inside to the CUDA installation directory. This step will also require administrator right!

File Explorer showing the contents of the CUDA v11.2 bin directory:

<input type="checkbox"/> Name	Änderungsdatum	Typ	Größe
crt	01.08.2022 14:30	Dateiordner	
nvrtc-prev	01.08.2022 14:31	Dateiordner	
bin2c.exe	15.02.2021 13:38	Anwendung	201 KB
compute-sanitizer.bat	29.01.2021 19:05	Windows-Batchda...	1 KB
cu++filt.exe	15.02.2021 13:38	Anwendung	180 KB
cublas64_11.dll	15.02.2021 08:07	Anwendungserwe...	104.815 KB
cublasLt64_11.dll	15.02.2021 08:07	Anwendungserwe...	171.588 KB
cudafe++.exe	15.02.2021 13:38	Anwendung	4.576 KB
cuda-memcheck.exe	15.02.2021 13:38	Anwendung	366 KB
cudart32_110.dll	15.02.2021 13:38	Anwendungserwe...	385 KB
cudart64_110.dll	15.02.2021 13:38	Anwendungserwe...	454 KB
cudnn_adv_infer64_8.dll	24.05.2021 08:48	Anwendungserwe...	127.480 KB
cudnn_adv_train64_8.dll	24.05.2021 09:01	Anwendungserwe...	85.049 KB
cudnn_cnn_infer64_8.dll	24.05.2021 09:09	Anwendungserwe...	622.421 KB
cudnn_cnn_train64_8.dll	24.05.2021 09:26	Anwendungserwe...	166.441 KB
cudnn_ops_infer64_8.dll	24.05.2021 08:27	Anwendungserwe...	267.465 KB
cudnn_ops_train64_8.dll	24.05.2021 08:36	Anwendungserwe...	57.124 KB
<input checked="" type="checkbox"/> cudnn64_8.dll	24.05.2021 08:18	Anwendungserwe...	219 KB
cufft64_10.dll	15.02.2021 13:38	Anwendungserwe...	183.888 KB
cufftw64_10.dll	15.02.2021 13:38	Anwendungserwe...	253 KB
cuinj64_112.dll	15.02.2021 13:38	Anwendungserwe...	1.284 KB
cuobjdump.exe	15.02.2021 13:38	Anwendung	2.811 KB
curand64_10.dll	15.02.2021 13:38	Anwendungserwe...	59.207 KB
cusolver64_11.dll	15.02.2021 13:38	Anwendungserwe...	387.009 KB
cusolverMa64_11.dll	15.02.2021 13:38	Anwendunaserwe...	208.820 KB

Finally, you should see files "C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin\cudnn\_xxx.dll"

File Explorer showing the contents of the CUDA v11.2 directory:

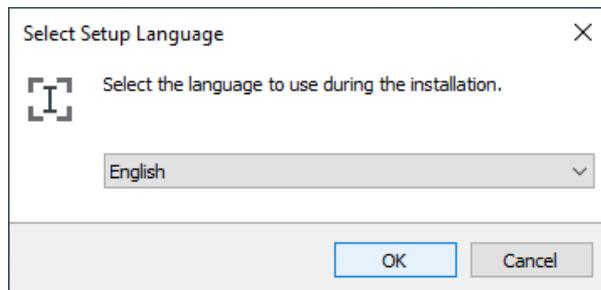
<input type="checkbox"/> Name	Änderungsdatum	Typ	Größe
bin	01.08.2022 14:53	Dateiordner	
compute-sanitizer	01.08.2022 14:30	Dateiordner	
extras	01.08.2022 14:31	Dateiordner	
include	01.08.2022 14:53	Dateiordner	
lib	01.08.2022 14:31	Dateiordner	
libnvvp	01.08.2022 14:30	Dateiordner	
nvml	01.08.2022 14:30	Dateiordner	
nvvm	01.08.2022 14:30	Dateiordner	
nvvm-prev	01.08.2022 14:30	Dateiordner	
src	01.08.2022 14:30	Dateiordner	
NVIDIA_SLA_cuDNN_Support.txt	04.05.2021 09:12	TXT-Datei	23 KB
version.json	26.02.2021 21:38	JSON-Datei	3 KB

After all, the software environment is set up. If you have installed AI-Core before, you will find that

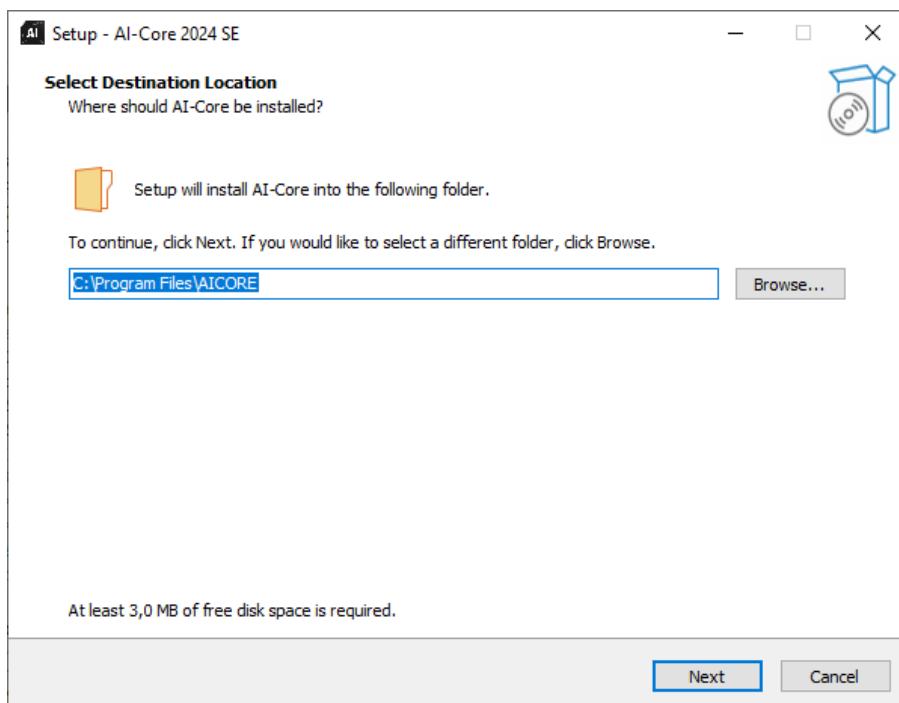
all authorized GPU-accelerated plugins will be activated now.

## 1.2. Installation Procedure

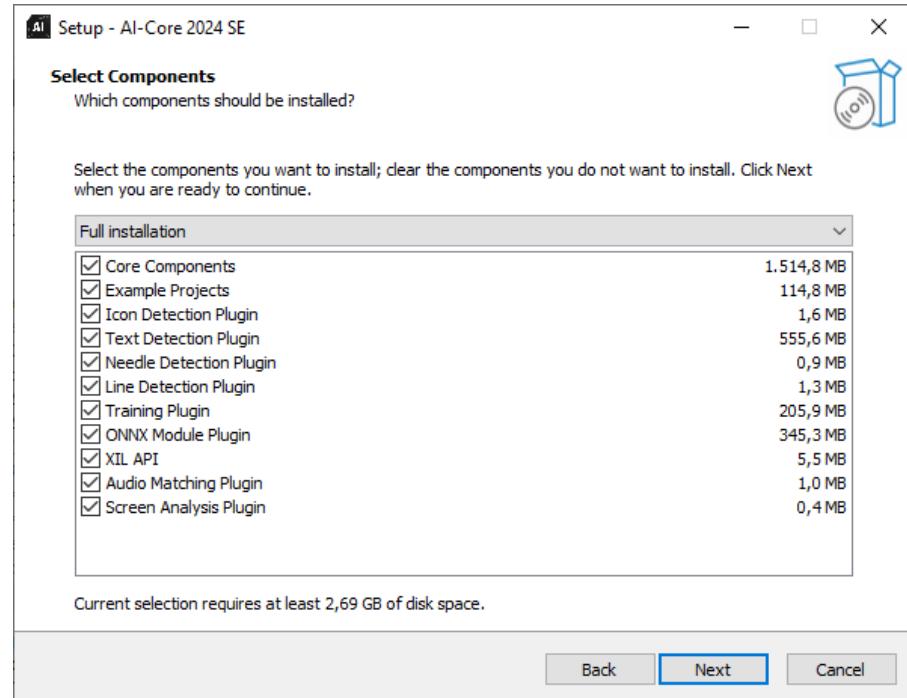
To install AI-Core, you should make sure that the system is Windows 10 64-bit version. Start "UITestAIdInstaller.exe". After starting you might need to enter the password of the Windows administrator. The first step is language selection.



Then, click "OK". The dialog will lead you configuring the installation folder.



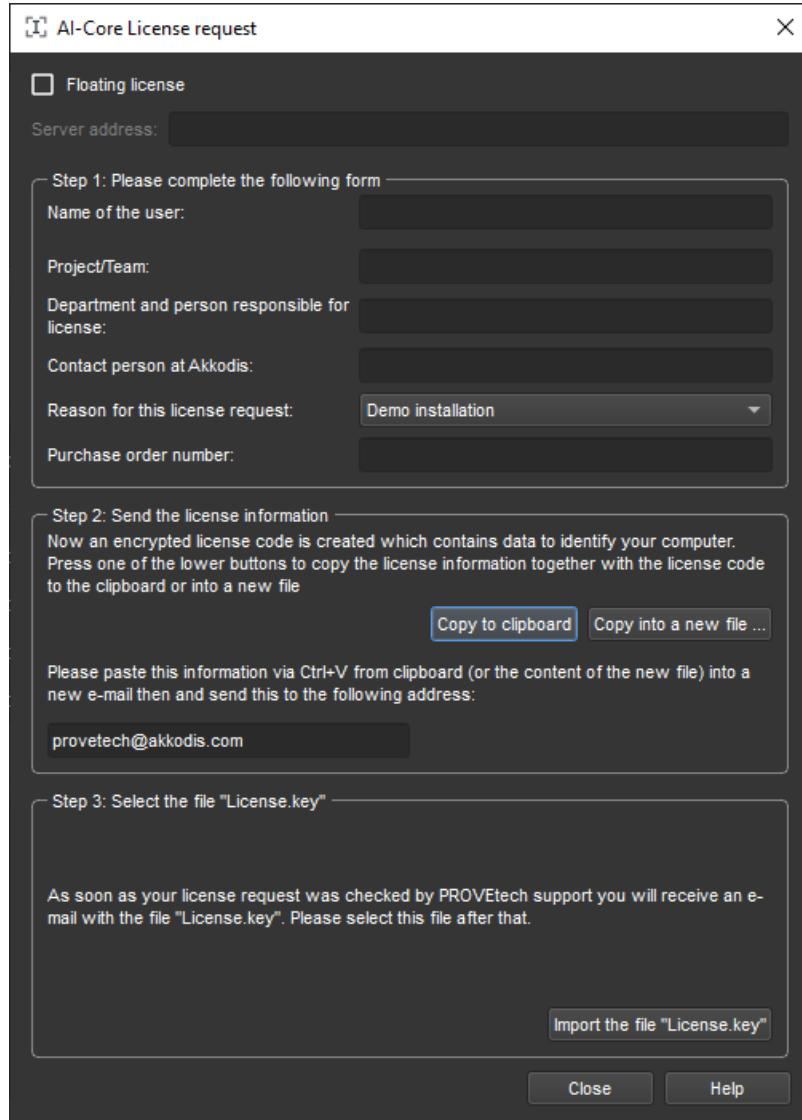
After confirming the installation path, click "Next". The dialog will show all available components that can be installed. Every detection function is integrated into a single plugin and works independently. You can choose your wished plugins to install. The core components are responsible for organizing all plugins and building communication with other applications. "Example Project" contains an example icon detection project. The project will be placed in the path "C:\Users\Public\Documents\AICORE\example".



Then, accept the creation of start menu shortcuts. AI-Core will be installed in your computer.

## 1.3. License

When you start AI-Core via the Windows desktop or the start menu for the first time, the following license dialog is displayed:



AI-Core shares the same license management with PROVEtech products. PROVEtech Support will thus assist in the license authorization of AI-Core. The same license authorization can also be accomplished in PROVEtech:TA for one time and license data will be shared between PROVEtech:TA and AI-Core. Just follow the instructions of the dialog. Depending on your license contract, you immediately receive the license file "License.key". So please do the following steps:

- Complete the form of "Step 1". The PROVEtech Support needs this information to synchronize the data of your license purchase order.
- As described in "Step 2", click on the button "Copy to clipboard" to copy the data and the corresponding license code as text into the clipboard. You can also click on the button Copy into a new file... if e.g. there is no e-mail program installed on the computer or there is no network access.

Important: Make sure that you have correctly set the computer time on the local computer. Otherwise, the PROVEtech Support cannot create a license file for you and you have to repeat the license process from start.

- Create immediately a new e-mail in your mail tool and set the subject, e.g. "License request".
- Paste the text (e.g. via hotkey Ctrl+V) from the clipboard into the e-mail or attach the currently saved file from step 2.
- Copy the e-mail address of the License dialog into the addressee field of the email.
- Send the e-mail.
- After your request has been reviewed by PROVEtech Support, you will immediately receive the license file "License.key" by e-mail. Save it on the hard disk.
- If you temporarily have finished, start AI-Core again and run "Step 3". Click the button Import the file "License.key" ... and choose the license file "License.key" that was just saved to hard disk. Now the license file is copied and later you can delete or archive the original file.
- After that AI-Core is starting and the licensing is finished.

In case the license dialog appears anyway, repeat the entire process. From the new license code of the e-mail, the PROVEtech Support is able to analyze your licensing problem. The file "License.key" specifies the expiry date, the activated PROVEtech products, the version binding and the activated modules of AI-Core.

Apart from a new installation, the following are reasons why the license dialog may appear at AI-Core start-up:

- New installation of AI-Core. AI-Core cannot be used immediately after installation. First, a valid computer-dependent license file "License.key" is needed in order to run AI-Core on this computer. This License.key can only be granted by the PROVEtech Support of Akkodis. Please note (for example, in the case of a customer's demonstration installation), that it may take some hours before you receive your file "License.key". The PROVEtech Support needs this time to verify the AI-Core license. The file "License.key" can also be generated a few days before AI-Core is installed. For this option, you should contact your Akkodis representative. However: The more detailed the information you provide, the quicker the license file can be generated.
- The license expired and has to be renewed. The AI-Core licenses are often set to expire at the end of the calendar year, after which they must be yearly renewed. To serve as a reminder and not to block the user, the license dialog starts appearing three weeks before the license expires. During this three week period AI-Core can still be started, despite the license dialog. Now you should renew your support agreements and request the new license file.
- The license file does not cover the installed AI-Core version. If you update AI-Core version then a new license file "License.key" could be required. You should synchronize this with PROVEtech Support before updating the PROVEtech:TA version.
- The license file "License.key" was accidentally deleted or overwritten by a different computer's license file within the last week. Sometimes a new license file is accidentally chosen on a computer for which it was not intended. It is therefore advisable that you create copies of your license files and carefully structure and manage them. However, a new file "License.key" can be requested from the PROVEtech Support at any time.
- The computer time was changed. This case could occur if your license file "License.key" has a global expiration date (in contrast to a version-bounded license file). If the computer time is

altered by more than a certain margin (e.g. set back by one day or a few hours), then the license dialog will appear. The license mechanism already accounts for different time zones as well as daylight saving time. In case you move the computer from one time zone to a different one, it is important that you should change the Windows time zone and not the system time.

- The Windows operating system was reinstalled. When you reinstall the operating system, a new AI-Core license file is needed. The same applies when the operating system is damaged and must be reinstalled on the computer.
- The hardware changed within the last week. The license file is bounded to your computer's hardware. When the hardware is altered AI-Core ignores the change for week thus accommodating temporary hardware changes. However, after the week is up the license dialog will start appearing. AI-Core recognizes a deactivation of hardware in Windows as a "hardware change". Permanent changes in your hardware require a new license file "License.key" which can be requested from PROVEtech Support.

# Chapter 2. Overview

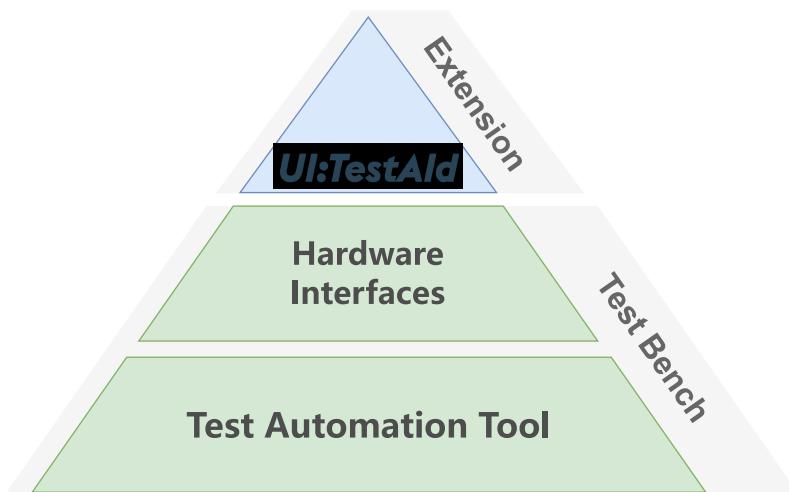
## Challenge in HMI Testing

In each industrial field, testing is always an important and inevitable process accompanying with system development. In the past, the testing is incredibly laborious especially when the system is complex and covers numerous fields. To simplify the testing but meanwhile improve the testing quality, many test benches have been developed in pursuit of testing automation. In present, test benches involve test automation tool (e.g. PROVEtech:TA) and hardware interfaces (e.g. PROVEtech:RE).

However, the present tools have no sound solution for the testing of human machine interfaces (HMI). Such kind of systems usually can only capture and output raw data, such as a video stream, but the key information which is valuable for testing is not extracted. For example, the testing needs to know whether a specific icon was shown or not.

The main challenges includes:

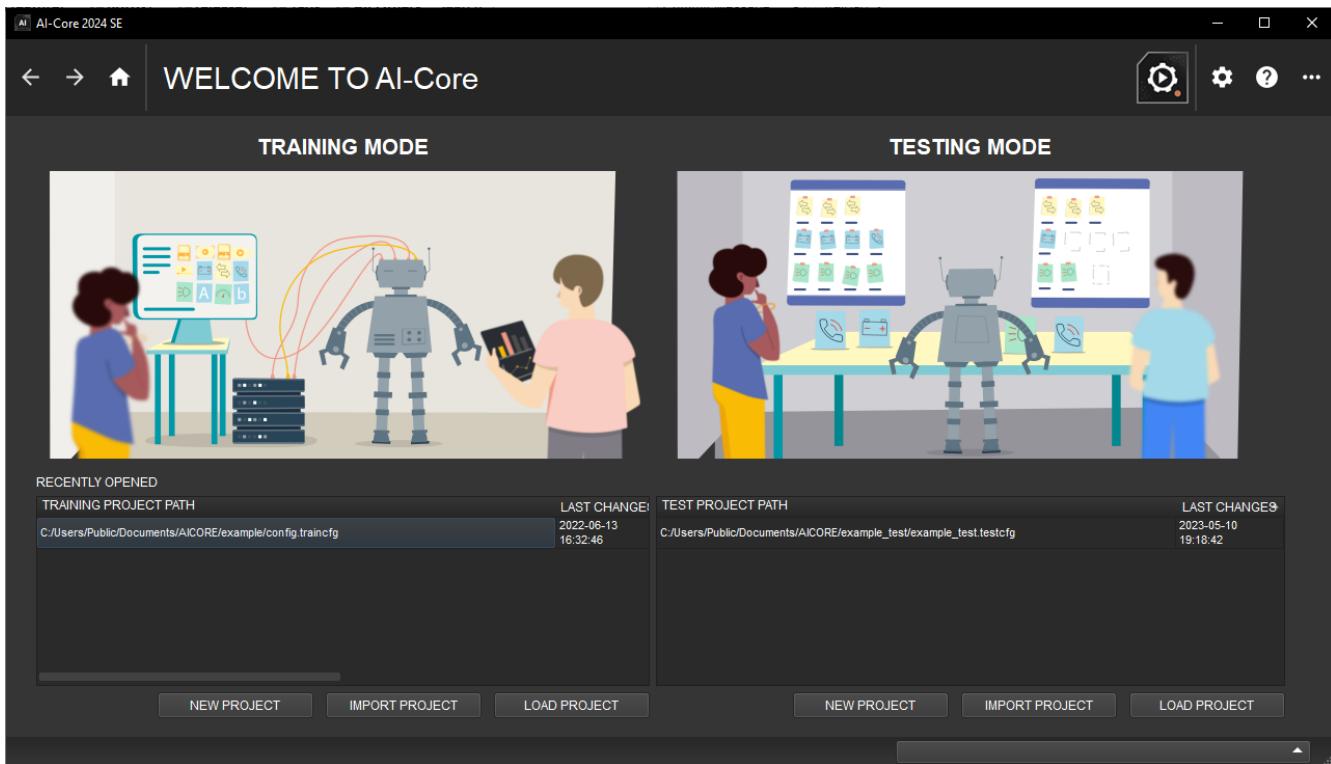
- High efforts due to frequently changes of icons/designs during the development.
- Testing video stream is not directly accessible. The developers sometimes do not want to leave video outputs for testing.
- Classical computer vision-based approaches cannot handle complex light conditions and even minor differences of the objects.



## *Introduction of AI-Core*

Facing the practical challenges, AI-Core team developed AI-Core, an artificial intelligence-powered perception tool. Its goal is to extract visual and auditory information from the raw data streams. AI-Core provides multiple artificial-intelligence (AI) models and training approaches to automatize the process. During a product development procedure or a productive industrial scenario, AI-Core helps customers find product faults and failure during runtime or execution or locate object positions.

AI-Core can not only work standalone but also cooperate with other test automation tools as an extension. The customers can access the forwarded extraction result and perform customized tasks.



AI-Core comes completely integrated with machine learning process from labeling, training to testing (inference). Even if you have no background knowledge of machine learning, you can still be guided to train your own AI model and validate your products within a short time.

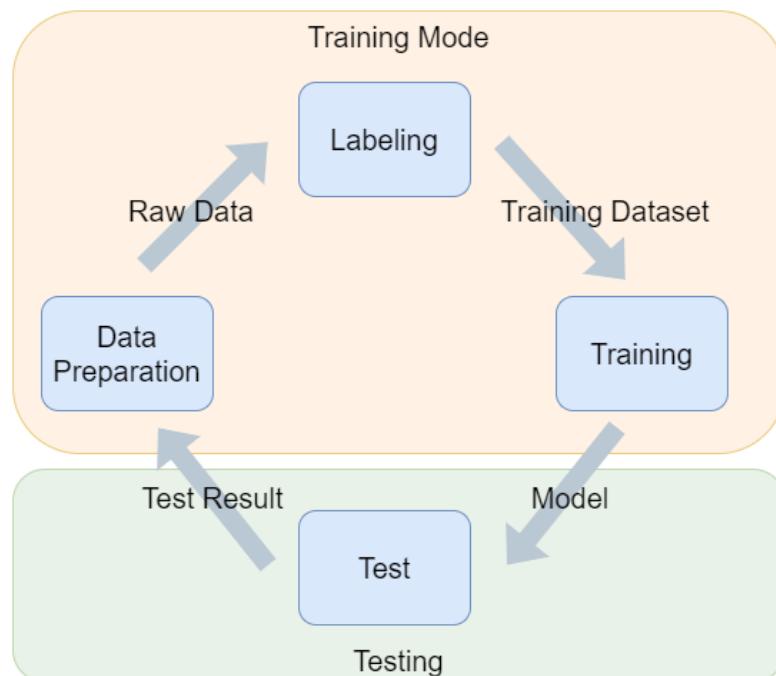
Compared to other similar perceptual tools, AI-Core can handle more intractable test cases. Even if the visual source is from an ordinary webcam with a low quality, AI-Core can also precisely distinguish trained objects from the pictures. Besides, it offers more object types that can be detected, such as text, lines, and so on.

## 2.1. Machine Learning Process

Machine learning can be categorized into three main classes: supervised, semi-supervised, and unsupervised. They differ in training data requirement.

Supervised learning is commonly used in various commercial and industrial applications due to its major deterministic results and high performance. A supervised learning algorithm typically analyses a manually-labeled training dataset. Through training a probabilistic model, it produces an inferred function, which can be used for classification or regression.

In order to ease the usage and enable non-professional users to acquire their own models, AI-Core has integrated the whole process into one tool. Roughly, the process can be separated into two parts: training and testing. The training part can further be divided to three steps: data preparation, labeling, training. Since the training has much higher requirement on hardware, such as high-capacity graphic card for parallel computing, AI-Core offers two distributions (Learning and Test) to satisfy different users' needs.



Data preparation is the first step. You need to record so-called raw data such as a video. The raw data should include as many scenarios as possible which users wish AI-Core detect.

The second step is labeling. Only with raw data, AI-Core still cannot understand the context and semantic information of it. You need to teach the tool by labeling the relevant objects. For example, for the icon detection, users need to mark the positions and categories of icons. The group of labeled data is called training dataset.

Next, AI-Core will learn the labeled training dataset and provide a model file. This step is fully automatized. users just need to wait for a while and set some simple configuration based on the local hardware performance.

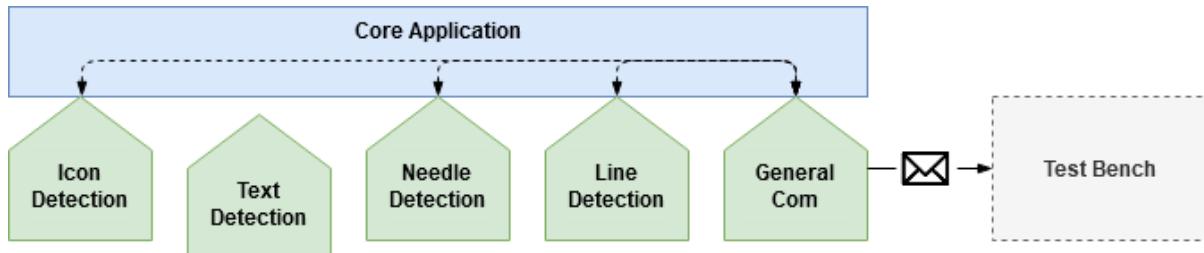
Finally, with the help of the trained model, AI-Core can seamlessly conduct online/offline testing. In this procedure, users can choose your own test automation tool such as PROVEtech:TA as

cooperation to automatize the test cases or use AI-Core standalone and save the results in wished formats.

The last but not least, since there are plenty of pretrained models for different languages, and font styles, the text detection algorithm can already handle most use cases. Currently, AI-Core does not offer the training tools for text detection. If users have special requirements, please contact AI-Core team. We will provide necessary technical support.

## 2.2. Plugin-based Framework

Different users have their own various tasks in the testing based on their specifications. When more and more tasks are involved in AI-Core, the system will be bloated and lack of extensibility. To solve this problem and fulfill the custom requirements, AI-Core implements plugin-based frameworks.



This framework consists of a core application and multiple plugins. You can decide which plugins you want to load on the core application. Each plugin works independently, but still be able to communicate with other parts of the system. The advantages are

- Users are allowed to decide to use each single plugin. It will reduce the workload of the computer when unnecessary plugins do not join the testing.
- Each plugin can be maintained separately, which increase the stability and maintainability of the system.
- Algorithms in different detection types have their own detection speed and limitation. Independent working will maximize the performance of each plugins.
- For users, you can customize the purchase of licenses.

Currently, AI-Core has five plug-ins: Icon Detection, Text Detection, Needle Detection, Line Detection and GeneralCom. In core application, the license mechanism will check the authorization of each plugin according to the contract, and decide which plugins are allowed to be loaded.

### Notice:

Icon Detection has two versions. One supports GPU acceleration. It will increase the speed of icon detection. The other one is CPU-only version, which will only use CPU to conduct icon detection algorithm. The core application will check the compatibility of your computer. If it finds supported GPU or CUDA library, it will automatically load GPU version in default. Otherwise, it will switch to CPU version.

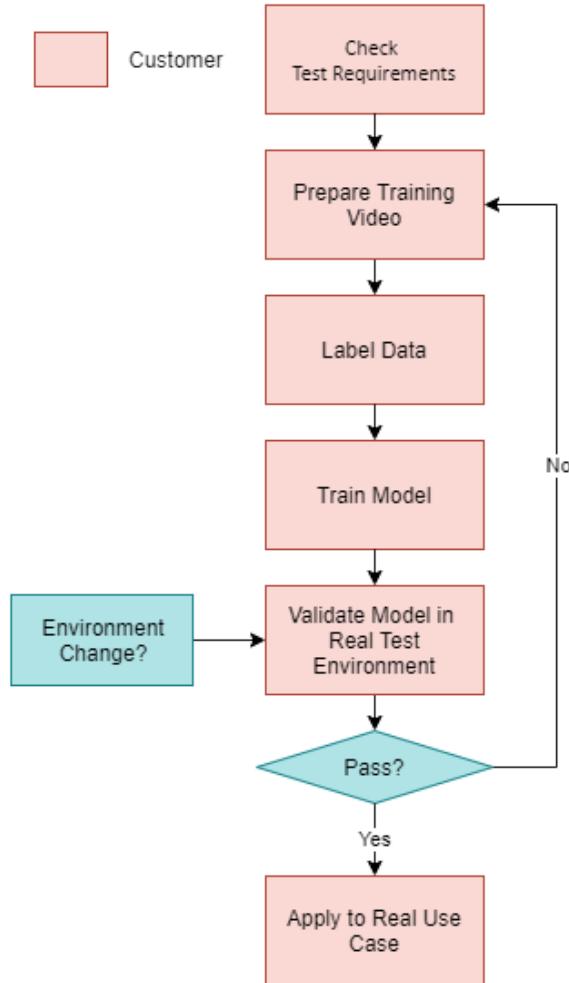
## 2.3. Work Process

AI-Core is a machine-learning-based tool which provides visual and auditory detection in the testing. Machine learning algorithm has a lot of advantages such as automatical adaption to different use cases, no human intervention needed, robustness to noise.

The final result highly depends on the training data, labeling, and training process. In the ideal case, the training and test should record under the same conditions. For example, if you record a video under only one single camera exposure as your training data but test under other dramatically different configuration, you might get bad results in the real testing. To guarantee the detection quality, you should follow a work process. To adapt different requirements of users, the process has two different variants.

AI-Core allows you to accomplish the whole process by yourself. You should do following steps indicated in Figure below:

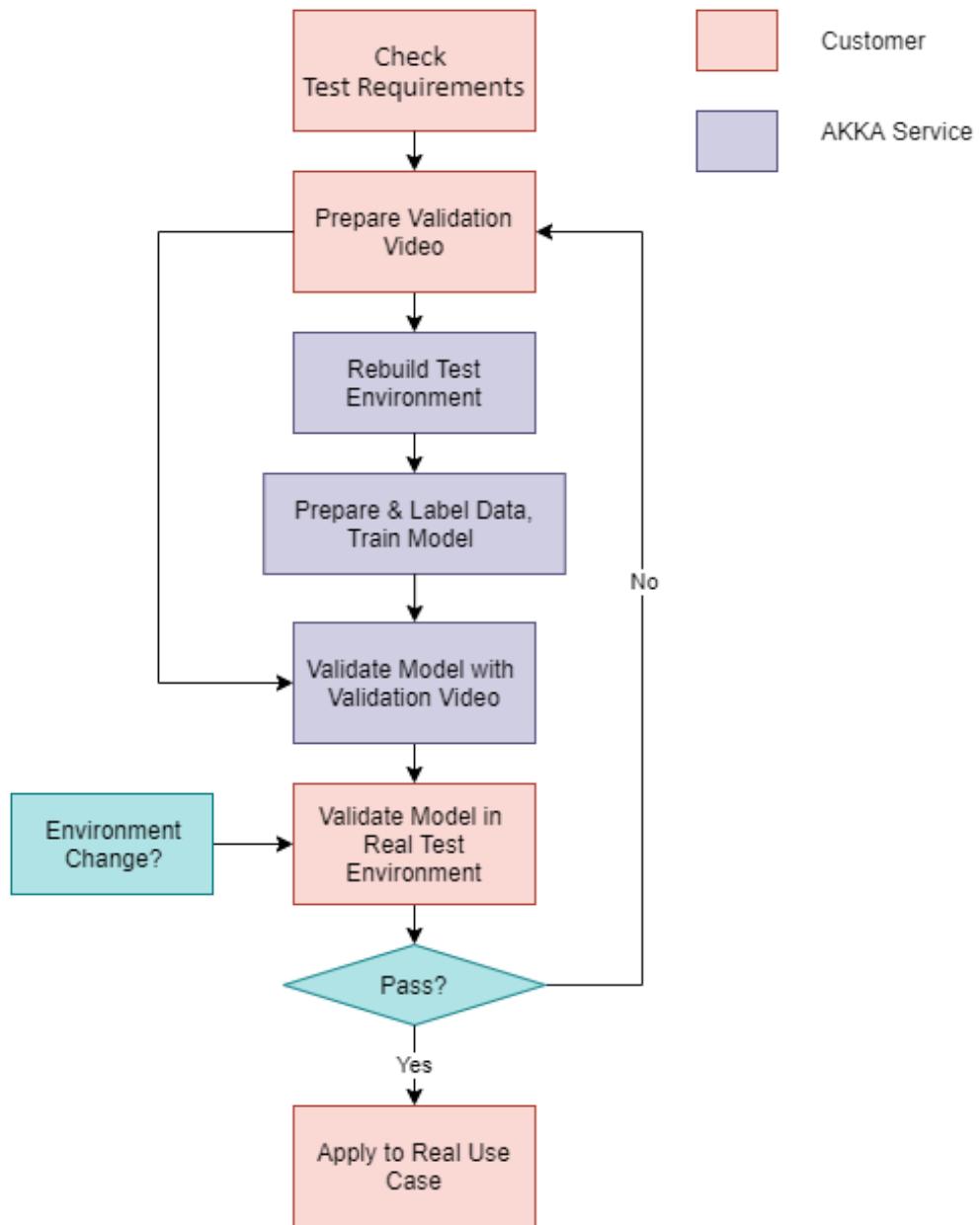
- Define your test requirements, such as in which light condition you want to conduct, whether the camera will later be moved.
- You should record a training video of your test procedure and label it.
- Launch the training program to get a machine-learning model.
- To make sure the trained model is qualified for the testing, you should run a probe test with the model. If it passes in the test, then you can apply the model to your actual test, otherwise you need to check your training data again, and find what is going wrong.



When you could have no sufficient infrastructure for the whole process or do not want to invest resources in labeling or training, Akkodis also can offer a professional service team. The team will be responsible for the labeling and training. At the end, the team will produce a trained model to the customers and help them integrate AI-Core to their test procedure.

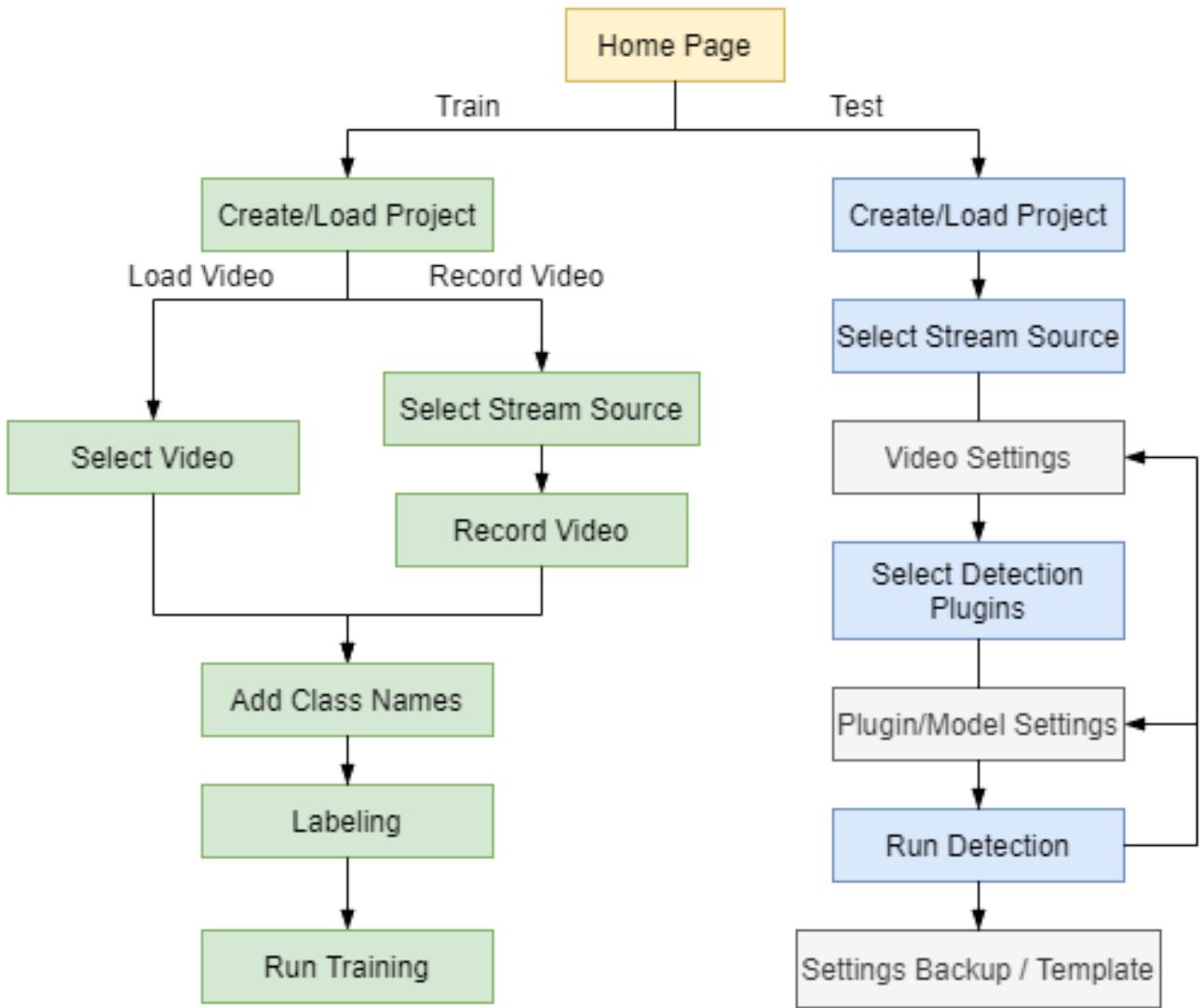
Different from the single team mode, the dual teams often conduct the training and test in different places. It will cause the inconsistency between the training and test environments, such as different ambient lighting, camera settings and so on. The final consequence could be that Akkodis team gains perfect detection results on its own environment, but the customer cannot reproduce it. To avoid this case, it is necessary to synchronize the both environments with the help of validation video.

Validation video can only be recorded by the customer under the real test environment before AKKA service team starts recording training data and labeling. The validation video implies all information such as ambient light source, camera settings, movements of device under test etc. According to it, Akkodis service team will rebuild the environment and also validate the trained model at the end.



To guide users accomplishing the whole process, AI-Core has a friendly user interface design. As previously mentioned, a machine learning process consists of two phases: training and testing. Thus, AI-Core separates the whole workflow to "Train" and "Test" parts.

In the following, we present an overview of the structure of our program, with its possible paths. Each step is visualized with a page. Following the hint of each page, you will easily finish all steps.



## *Home Page*

When the program starts, the main page is presented in 3 main parts. The home page gives you informational and configuration options. From here, it guide you to train a new machine-learning model or start a test. Then the workflow will be split to Training Mode and Testing Mode.

## *Train Flow Overview*

After you have created or loaded a training project, you have the option to load an existent video ("Load Video") or to record a new one ("Record Video"). For the "Record Video" path, they need to firstly select which stream source they want to use (USB Camera, IP Camera, Web/Screen Gruber). And proceed with with the recording.

Then we are ready to begin creating class names of the icons that you want to classify in the videos. And afterwards labelelging the objects on the intended frames, by placing a bounding box around them. It is important to note that the labeling step can be skipped, in case the labeling has previously been done.

Finally, the training can begin, using the selected video as input and the labels as outputs. This process can take a while. To give you some information about the progress of the training, we provide a comprehensive graphical interface.

## *Test Flow Overview*

After you have created or loaded a training project, you need to select the intended source stream. The settings of the selected source can be configured (with "Video Settings"), where one can control properties such as video brightness or zooming.

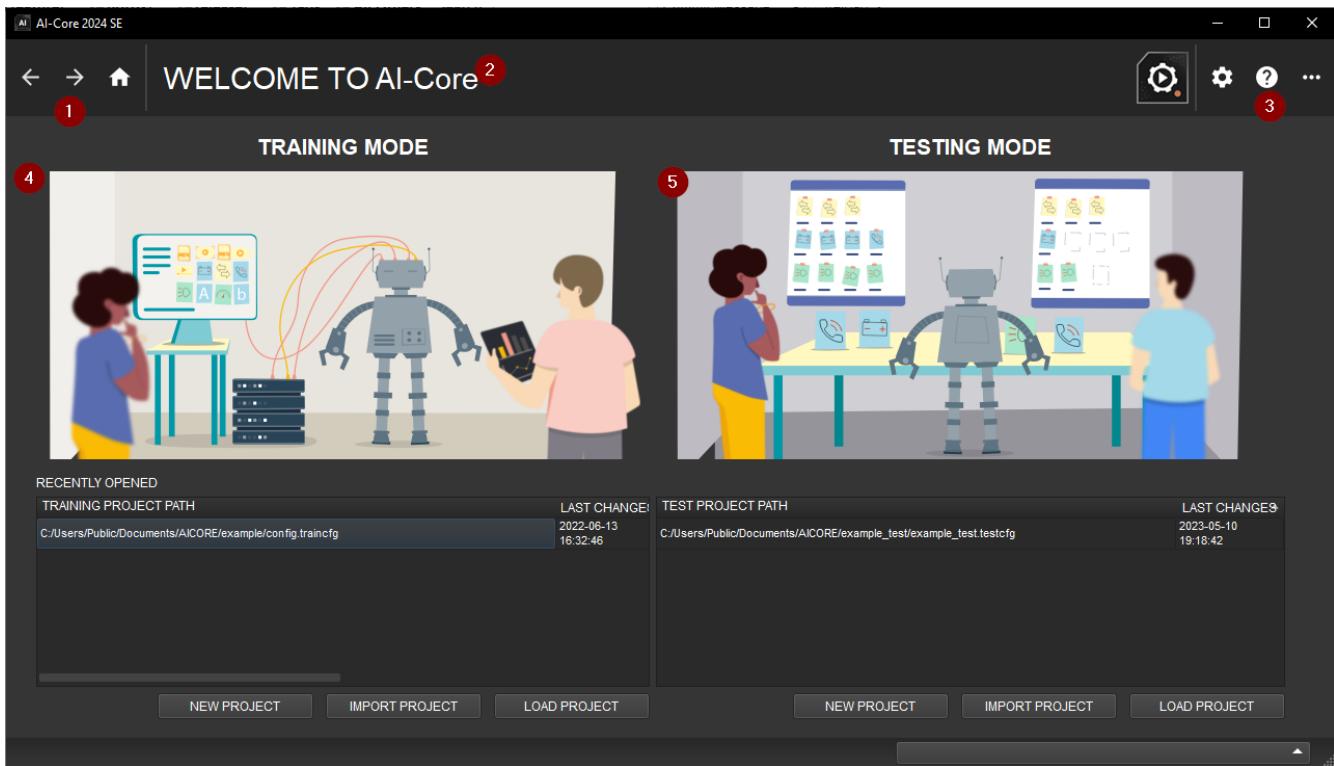
After selecting the source stream, you are presented with the available detection plugins (e.g. Icon Detection, Text Detection), and they are free to choose which ones to use in testing. Additionally, (in "Plugin/Model Settings") they can change the Plugin Settings, as well as choose the machine learning model for the case of Icon Detection.

Finally you can start running the test. In this stage, we still give the possibility to change any settings live (video and plugin settings). On top of that, it is possible to select multiple ROIs (regions of interest), determining where you want to apply the computer vision techniques. These ROIs can be saved as a template, and load in future times. As a final step you have the ability to save them for future use ("Backup Settings").

Later on, we will describe more details, as well as, show an illustrative guide and tutorial of each component. We will start with the home page and configurations, and then the training and testing Modes.

# Chapter 3. Configurations

Let us first have an overview of the home page.



- **Workflow Control**

The whole workflow is visualized in form of pages. Each page guides you to accomplish one step. You can click "Forward" and "Backward" button to jump to the previously visited pages. Or click "Home" to jump back to the home page.

- **Status Bar**

The status bar will show you the information of the current status. If you load a training/test project, it will also show the project name.

- **Tool Bar**

The tool bar allows you to check the account, configurations, license of AI-Core. It will be later introduced in detail.

- **Training Mode**

On the left side, you can find the section of training mode. It shows all information related to training projects, including all loaded training projects.

- **Testing Mode**

On the right side, you can find the section of testing mode. It shows all information related to test projects, including all loaded test projects.

In the following content of this chapter, you can see how to manage projects in section [Project Management](#), check system information in section [System Information](#), application and plugin settings [Test Mode Setting](#).

In section [Help](#), the manual contents are well organized. You can still read this manual from here.

## 3.1. Project Management

From AI-Core 2022, the application and projects are organized by four types of configuration files:

- **Application configuration file (.appcfg)**

used to save all current setting parameters of AI-Core, including preferences, plugin settings, last loaded training and test projects. It is equivalent to the old "settings.ini" file.\\ From AI-Core 2022, it should be always located in the path \\ "C:/Users/{user\\_name}\Documents/AKKA/UITestAId/cfg". The testing configuration will be saved in test project configuration file instead.

- **Training project configuration file (.traincfg)**

It saves the data structure of the training project including videos, class sets, class names, and paths of trained models.

- **Model configuration file (.modelcfg)**

It saves the configurations and history of one training. It will be later introduced in "Training Mode" in detail.

- **Test project configuration file (.testcfg)**

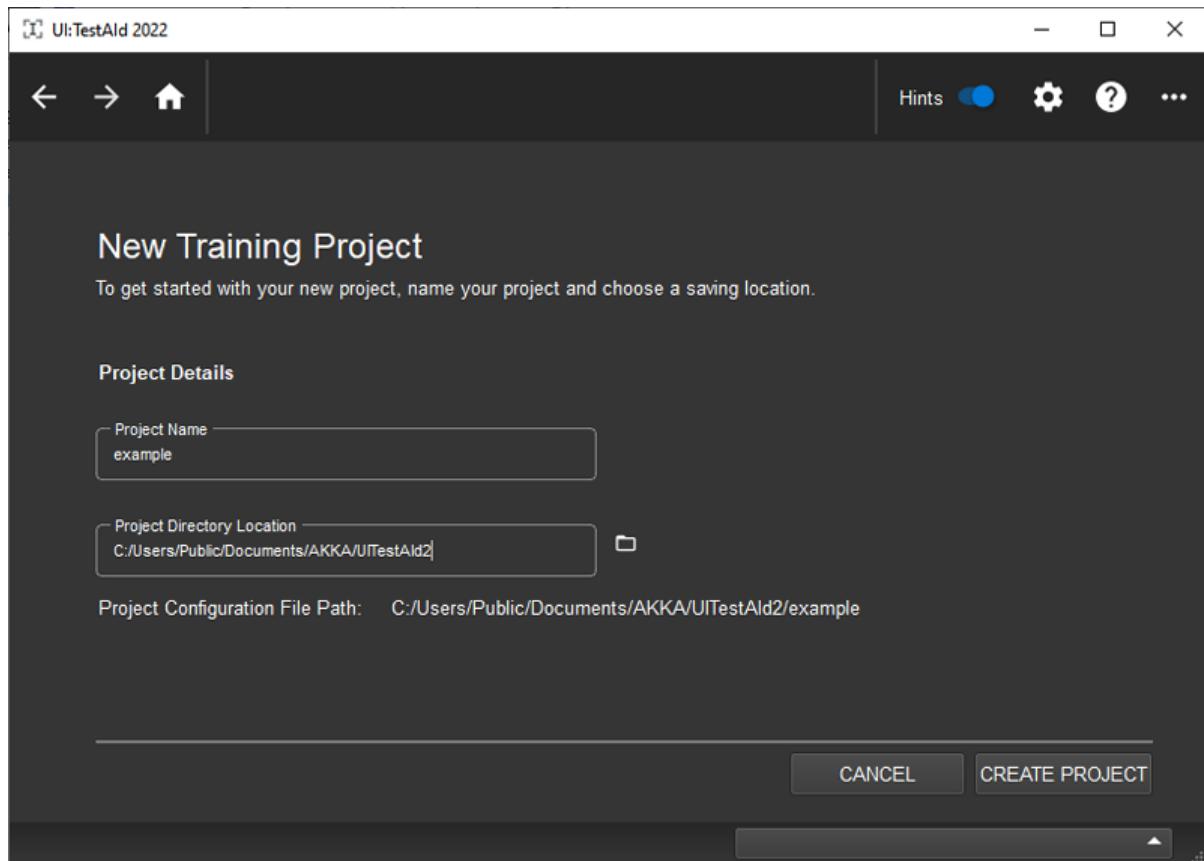
The test project configuration file saves paths of loaded machine learning models, snapshot of application configurations (Settings Backup), and templates.

The training and test projects are mostly managed in home page. It allows you to add, remove, rename, import and load a training/test projects.

These operations work the same for both the training and testing projects. Therefore, we will illustrate with examples from the training side.

### 3.1.1. Create New Project

To perform this action, you should click on "New Training Project". This will advance to another page, as seen in the figure below.



You should input a project name in "Project Name" field and specify the project directory. When you type the path and press return, the final project path will be shown underneath.

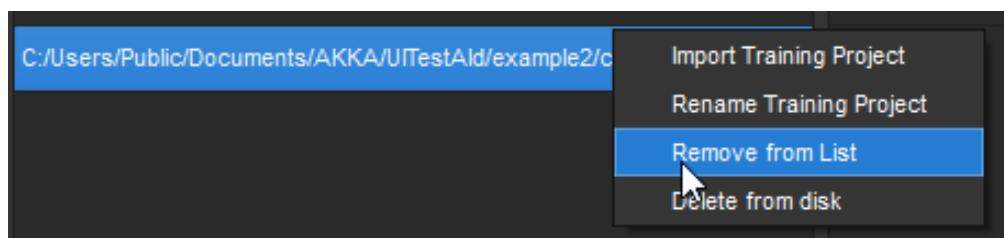
In case, this already exists, a warning will appear with red letters, stating that the path already exists. In this case, the user must change the Project Name to an unique one.



To finalize the project creation process, please click on "Create Project". After the creation of a project, you will be guided to the next step. In the home page, the table "Recently Opened" will also list the newly created project.

### 3.1.2. Remove Project

If you do not need one project any more, you can right-click in the home page, and a list of options will appear.



There are 2 ways to remove:

- **Remove from List**

This operation will just remove the project from AI-Core. However, the project still exists in the hard disk. You can add it back later through "Import Project".

- **Delete from Disk**

If you want to erase the project from the hard disk completely, you can select "Delete from Disk". After that, the project will be removed from both AI-Core and the system. This operation cannot be recovered any more!

### 3.1.3. Rename Project

If you want to rename the project, you simple right-click on the "Recently Opened" box, and press on the "Rename" option. This will show a dialog that allows the user to write the new name.

**Notice:** To prevent mismatching path in your test bench, the operation will not change any file or folder name. Instead of that, it just changes the name saved in the configuration file.

### 3.1.4. Import Project

If your project does not appear on the "Recently Opened" box, you can add it there by right-clicking on the box, and selecting the "Import Training Project" (or "Import Testing Project"). This will open a dialog where the user can navigate over the folders and select the appropriate file.

Note:

- This file must have the extension of ".traincfg" for the training, or ".testcfg" for the testing.
- If you are a user of AI-Core 2021 or elder versions, you might have multiple settings.ini to specify the test configuration. In this case, you should import the "settings.ini" als test project. AI-Core will convert it to the new version of testcfg and update all related training projects.
- You can also import the old training project file (config.yml) in the training project. However, AI-Core will automatically backup the old file and update it to the version.

### 3.1.5. Load Project

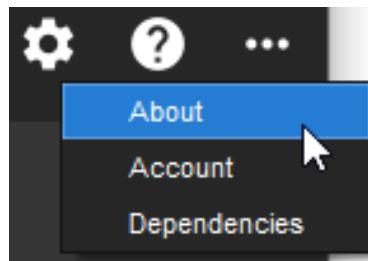
To load an existing project, that appears in the "Recently Opened" box, the user must select it by left-clicking it, and then pressing on the "Load Training Project" (or "Load Testing Project") button. AI-Core will then load the selected project and guide you to the next step.

## 3.2. System Information

On the top-right corner of the toolbar, there 3 buttons available.

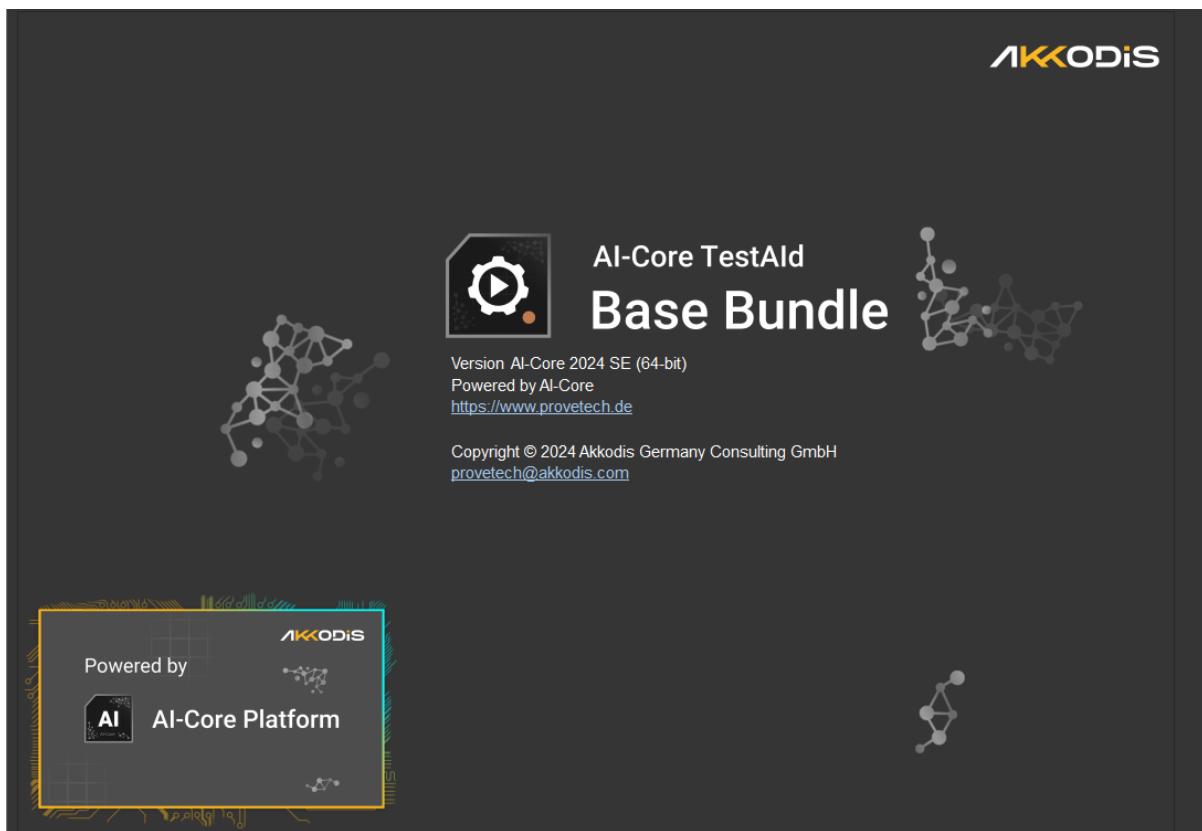


By clicking on the icon with the three dots, you can check the general system information, including about, account and dependencies.



### 3.2.1. About

In the "About" page, the program presents information about the current version.



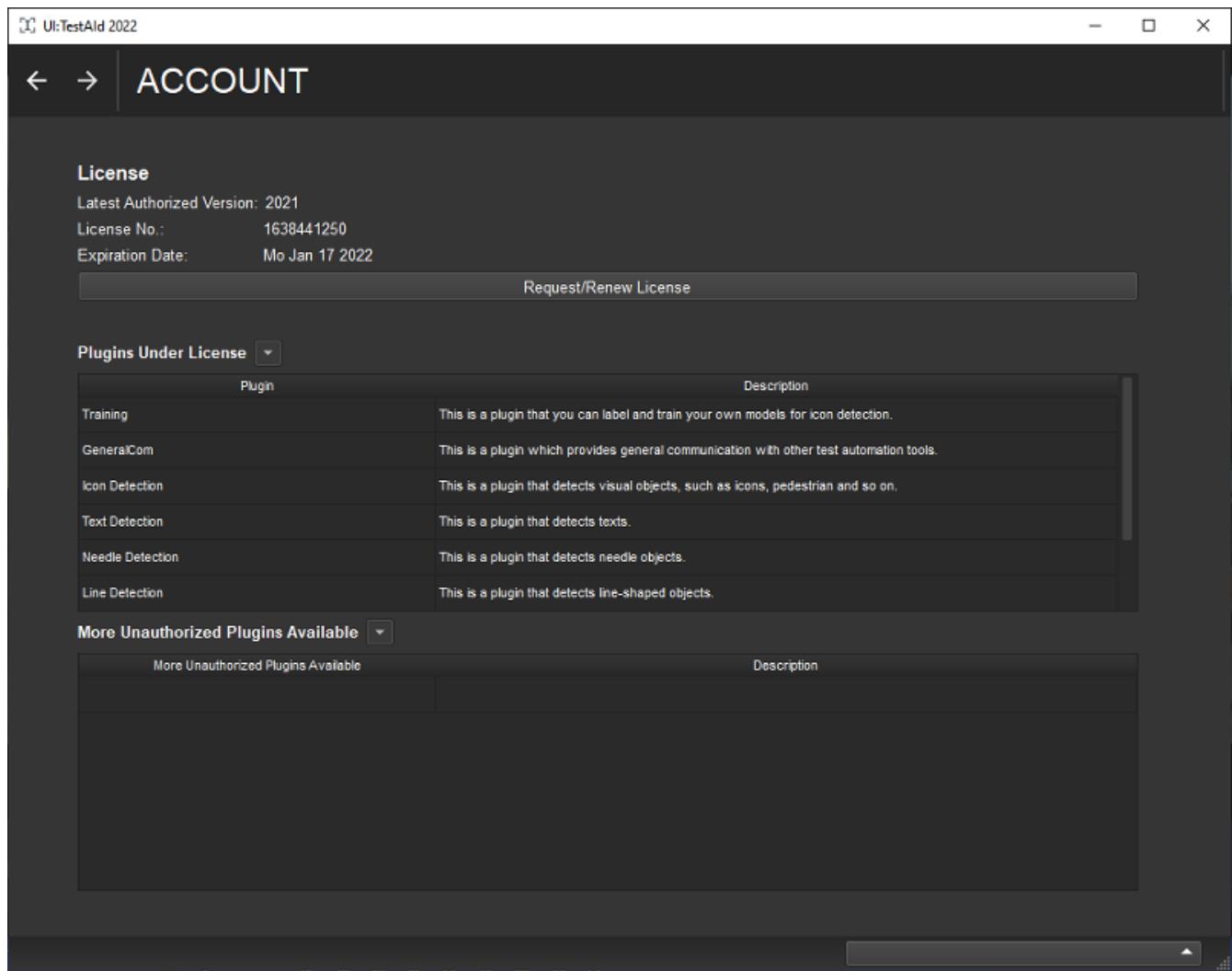
#### Version Name

You can find the version name. Usually, it starts with the year number. In the example, it is version 2022. AI-Core usually has two yearly version updates. The former one is released at the end of the previous year. For version 2021, it was released on 30.11.2020. The latter one is released in the middle of the current year. Since the release date is usually in Summer, the version name will be followed by "SE" (Summer Edition).

Of course, according to the feedback and bug reports, AI-Core team will also release several improvement version with suffix "SPx" (e.g. SP2, SP3 ...) aperiodically. You can find concrete information of version update in [link](#).

### 3.2.2. Account

The "Account" page shows the license information including latest authorized application version, license number, and expiration data.



#### *License*

The license has its own expiration date and identification number. If you do not update the license before the expiration, AI-Core will be locked up and show license dialog during each launching.

Click "Request / Renew License", the license update dialog will pop up. You can use it to generate system information for requesting new license or update your new license key.

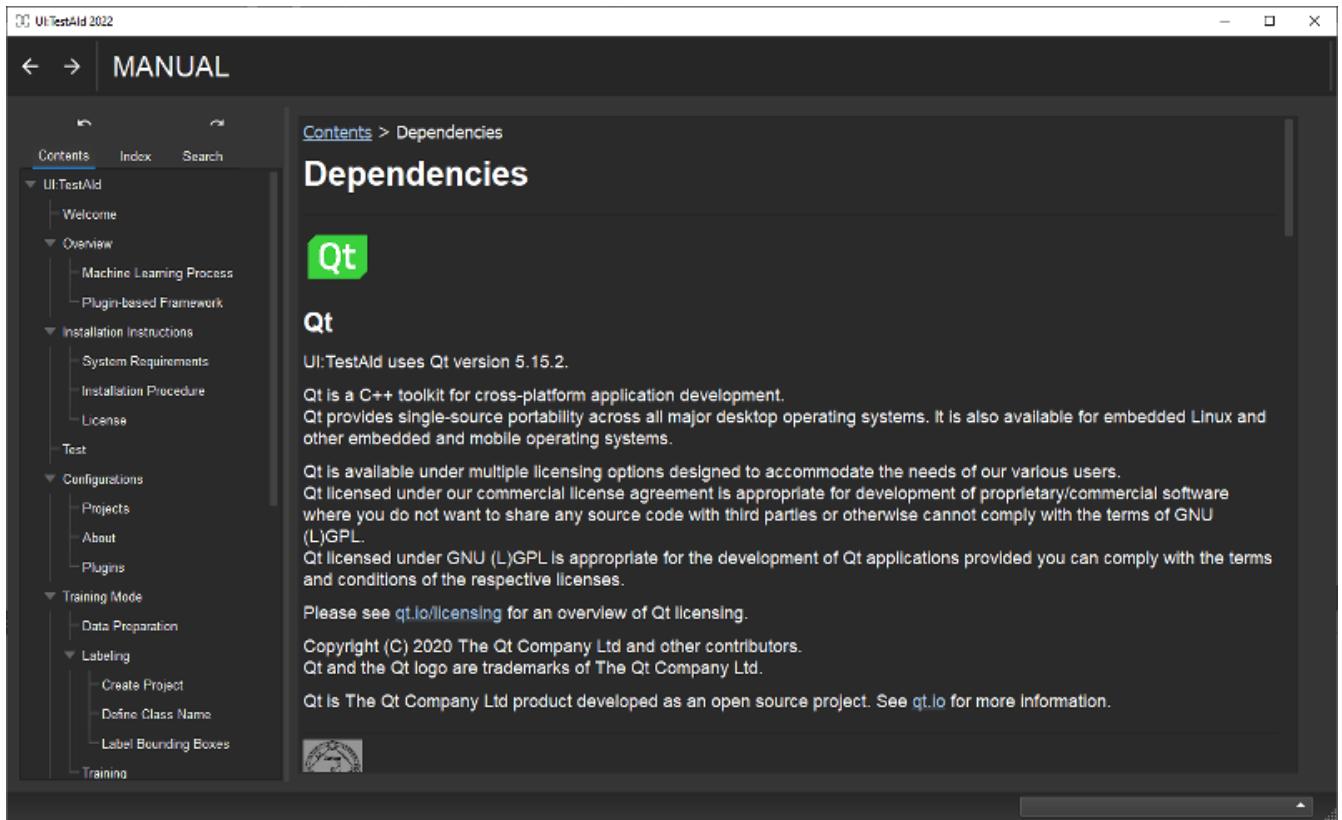
#### *Plugin Authorization*

Each function is managed in form of plugin in AI-Core. AI-Core team developed various functions to meet different requirements of customers. According to your contract, purchased plugins will be authorized within the license. The license key that you obtain from PROVEtech Support, also contains the information of the authorized plugins.

The available plugins with their descriptions are all listed in the table "Plugin Under License". The other plugins that AI-Core can provide, but are not currently authorized in the current license are showed in the table "More Unauthorized Plugins Available". In case you wish to acquire them, please contact PROVEtech Support.

### 3.2.3. Dependencies

This option will display all the existing Dependencies used in the program.



We also provide descriptive information regarding the libraries used. And the user is able to search for it using the search bar from above.

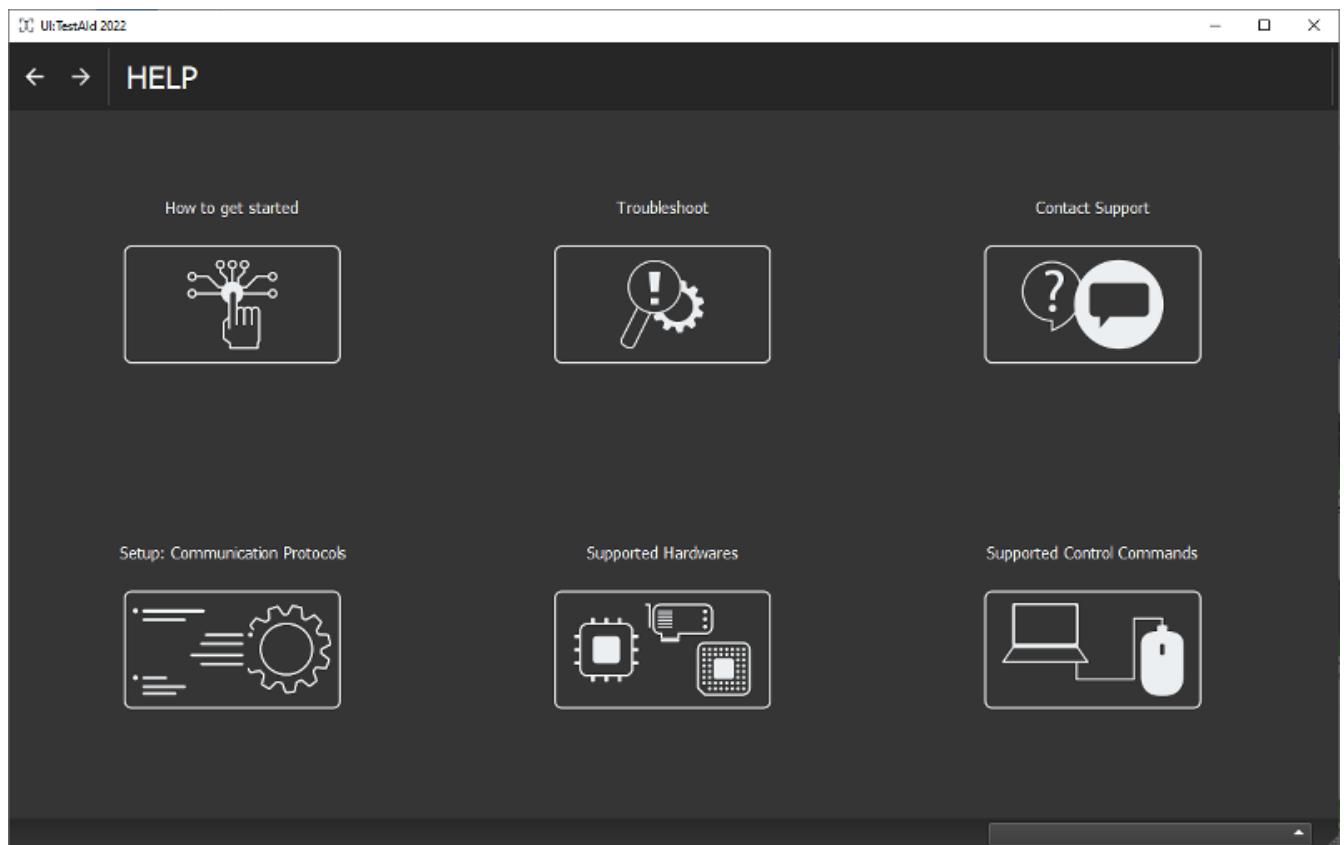
### 3.3. Help

The second button on the toolbar buttons is the help option.



This will provide the user with Help on 6 distinct themes:

- How to get started
- Troubleshoot
- Contact Support
- Setup: Communication Protocols
- Supported Hardwares
- Supported Control Commands



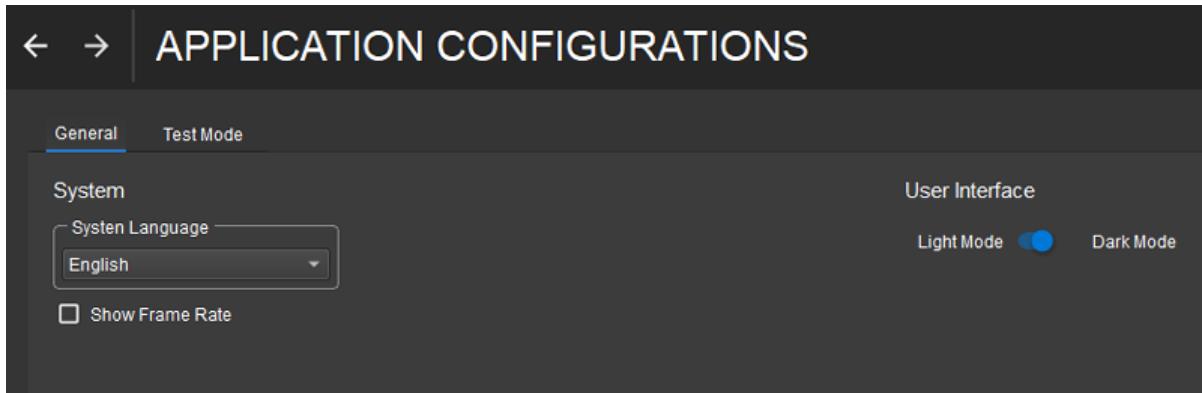
## 3.4. Test Mode Setting

The final option, is the Testing Mode Setting.



When clicking it, the user is presented with a tab bar containing 2 different options:

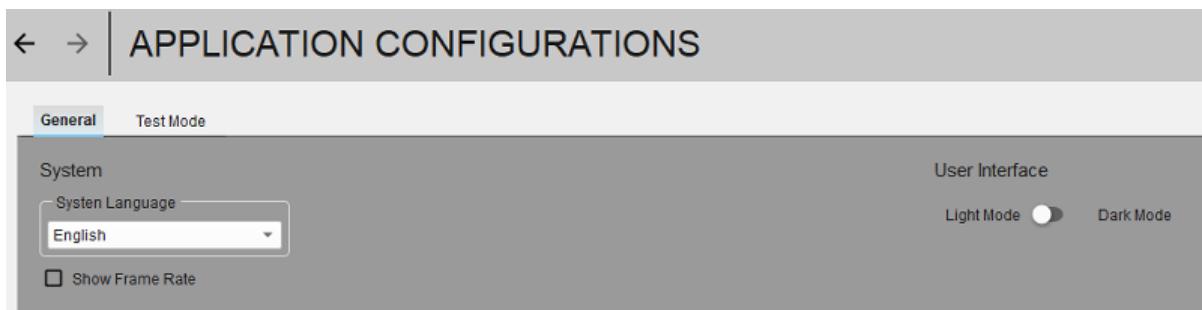
- General
- Test Mode



### 3.4.1. General

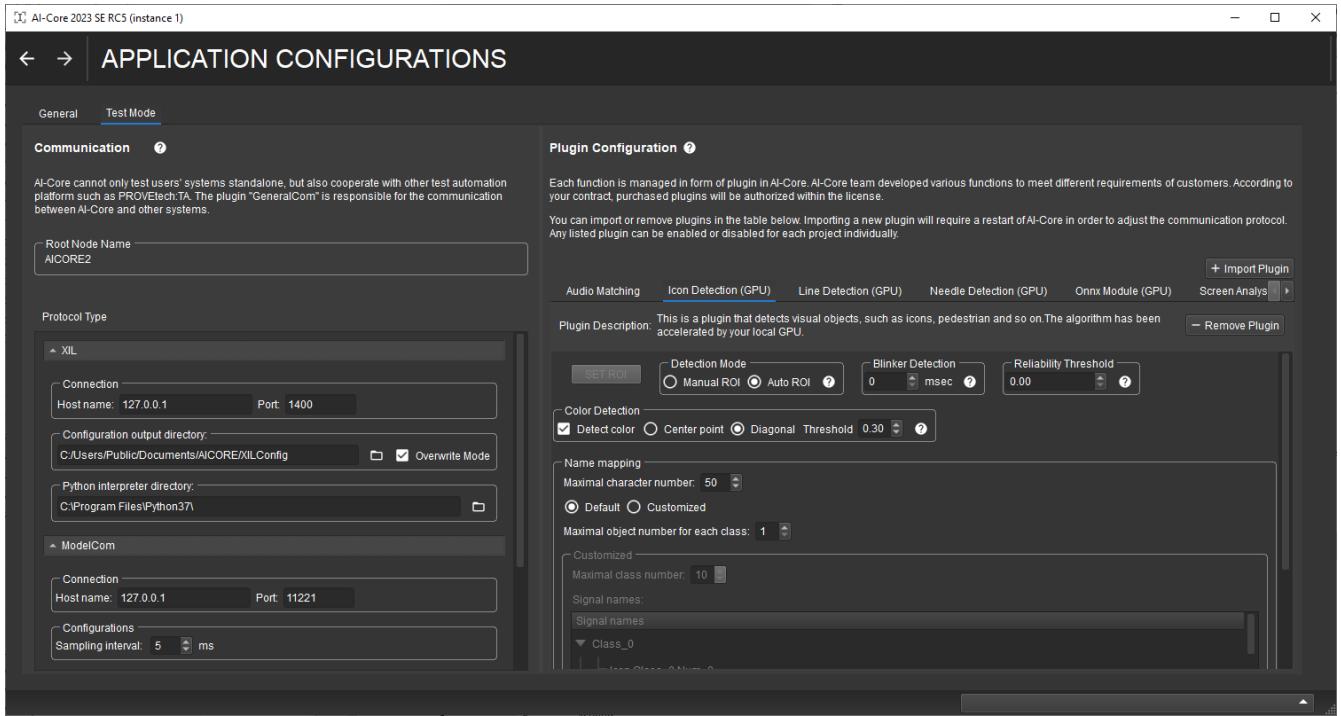
The "General" tab has 3 options:

- System Language: to switch the language (currently English, French, German and Chinese are supported).
- Show Frame Rate: even if you setup a target frame rate of the camera in the test mode, the system or the hardware might not be able to reach it. Enable this option to check the actual frame rate.
- User Interface: to switch between Light Mode and Dark Mode. Most of the examples provided are in Dark Mode. The figure below shows an example of Light mode.



### 3.4.2. Test Mode Settings

The test mode contains configuration options regarding: communication and plugin configuration.



## Communication

On the communication side we have access to the different communication protocols: XIL, ModelCom and MQTT. You are able to change their configurations here.

## Plugin Configuration

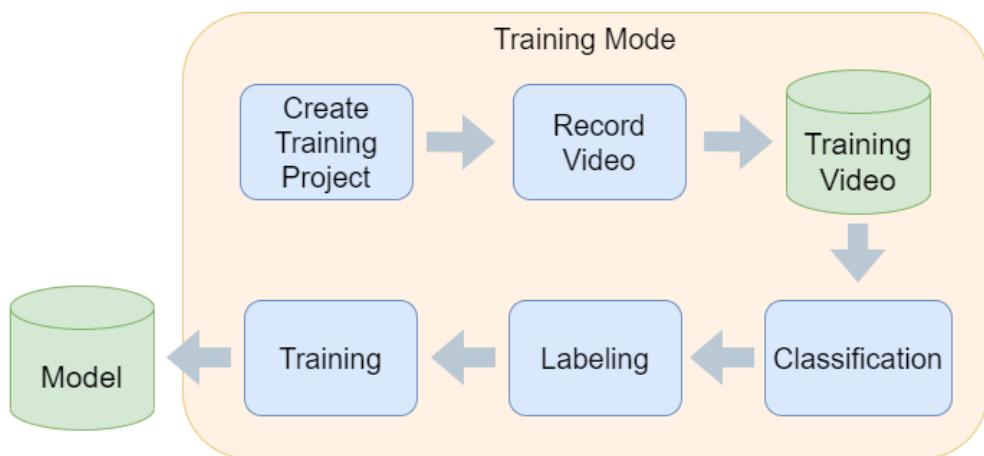
This part allows the user to change the settings of each detection plugin. On top of that, they can remove the plugin with "Remove Plugin", and even add a new with "Import Plugin". The latter will open the files on the user's personal computer, and add a new Plugin in case a valid file of type .dll was selected. It is important to note that it is necessary to restart the program, in order for the new added Plugins to take effect.

# Chapter 4. Training Mode

The training is a necessary process when you want to conduct a test with model-based test plugins, such as icon detection. The basic idea is that the computer will learn the visual features from a training dataset (e.g. a section of video) and try to perform a task (e.g. classification or detection of objects). The computer will finally persist the learning results through providing a machine-learning model. During the test, the computer will reuse the model and perform the learned task.

AI-Core has embedded the whole process in the training mode. You can record or import your own training video and obtain a learned model at the end with some visualized tools.

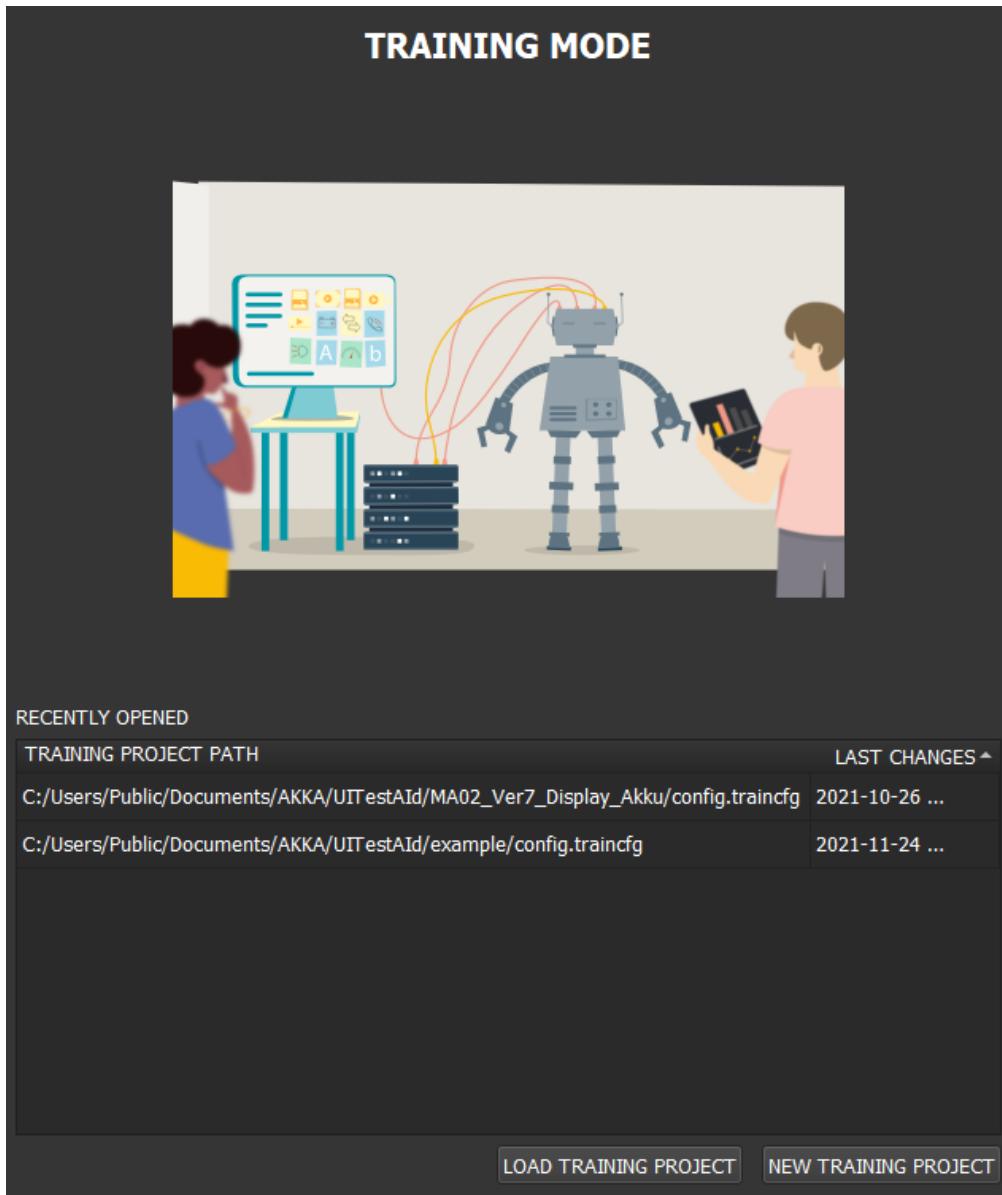
In the training mode, there are five steps: [create training project](#), [record video](#), [classification](#), [labeling](#), and [training](#). The final output of the process is a folder which contains a model configuration file, binary model file, and intermediate data file generated during training.



- Data preparation is the first step in which you need to record a sample video.
- As raw data, the recorded video still remains to be labeled. In the labeling step, you will need take time to mark all objects of interest by drawing boxes. AI-Core will assist you accomplishing the task efficiently. The output of this step is labeling information corresponding to the sample video. These information will be saved in project configuration files.
- At the final step, AI-Core will train a model based on the labeled video. You just need to set several simple parameters and wait until the training ends. Finally, you will have a trained neural network model. It is saved in a folder with the same of the model version. The folder is located in the "project\_directory/Models".

## 4.1. Create or Load Training Project

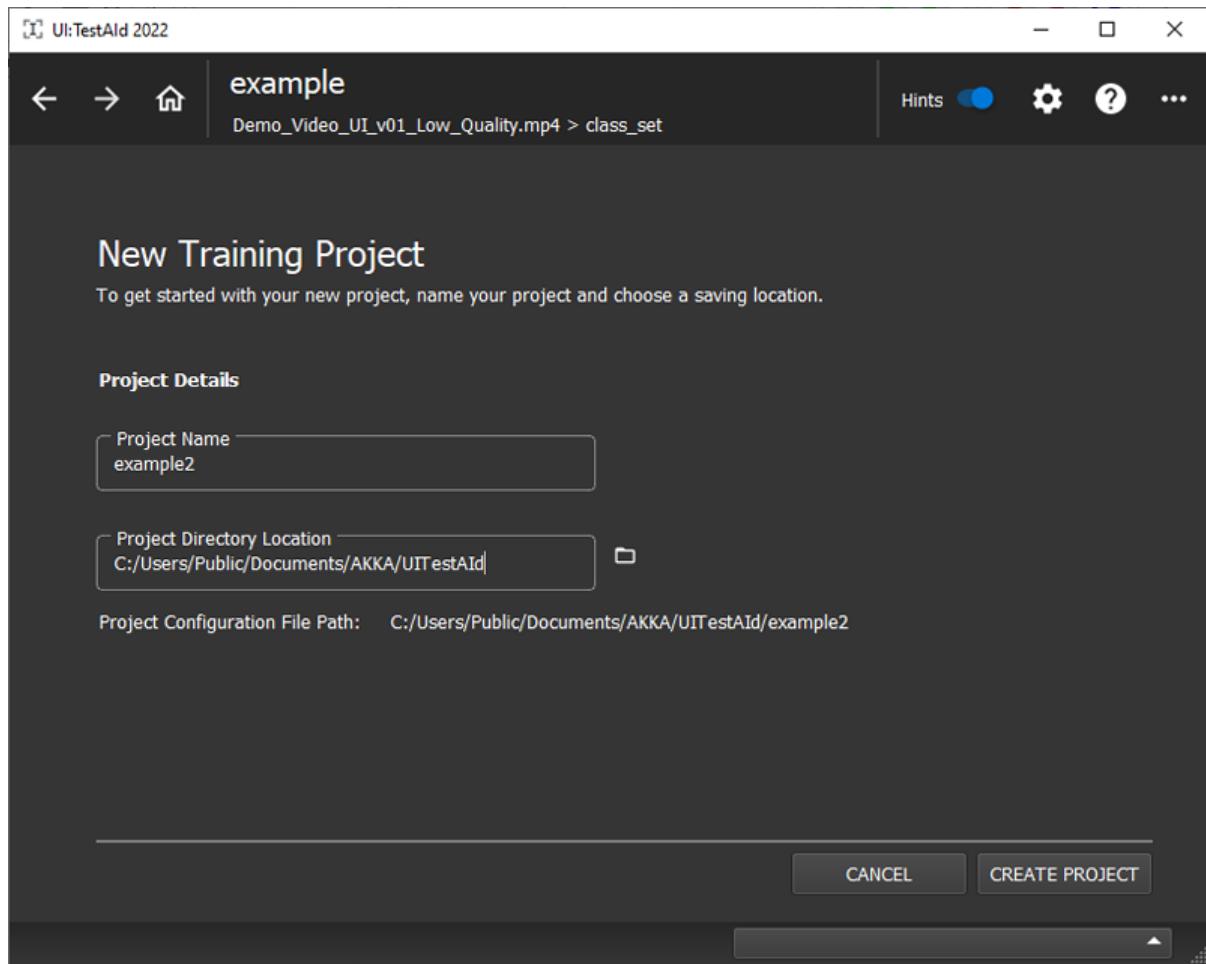
In the home page, you can see the training mode appears on the left side.



This page will allow you to either start a new project, or load an existing one.

### 4.1.1. Create Training Project

When the button "New Training Project" is pressed, the user is lead to the page in the figure below.



You need to name the new project, as well as to select one appropriate folder, and then click "Create Project".

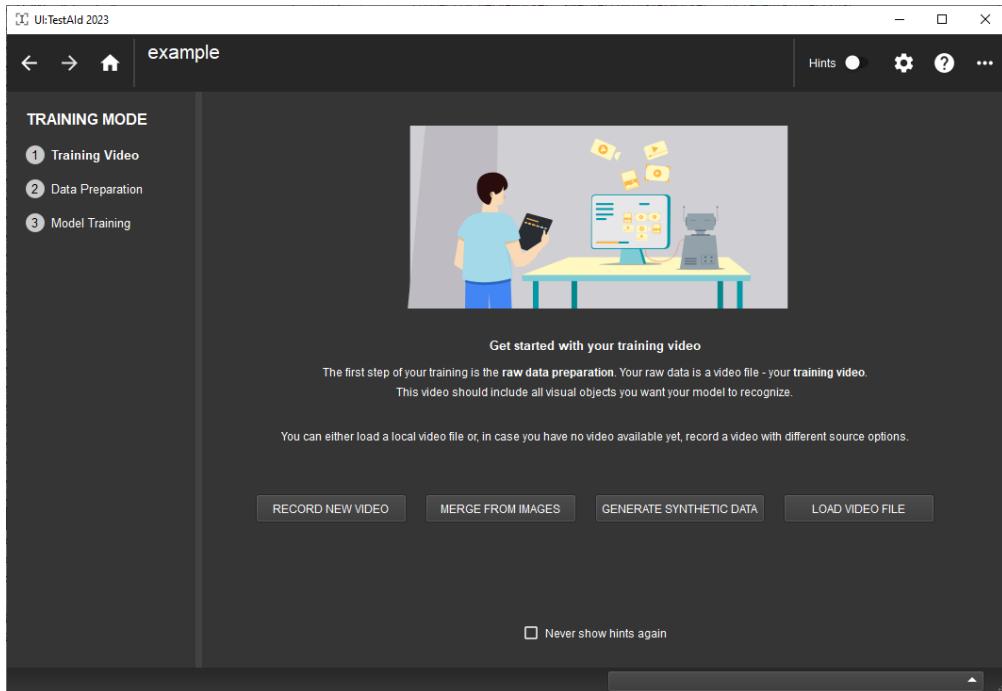
#### 4.1.2. Load Training Project

If you have already created one training project and want to load it again, you should double click the wanted training project in the home page list or click the button "Load Training Project".

## 4.2. Prepare Video

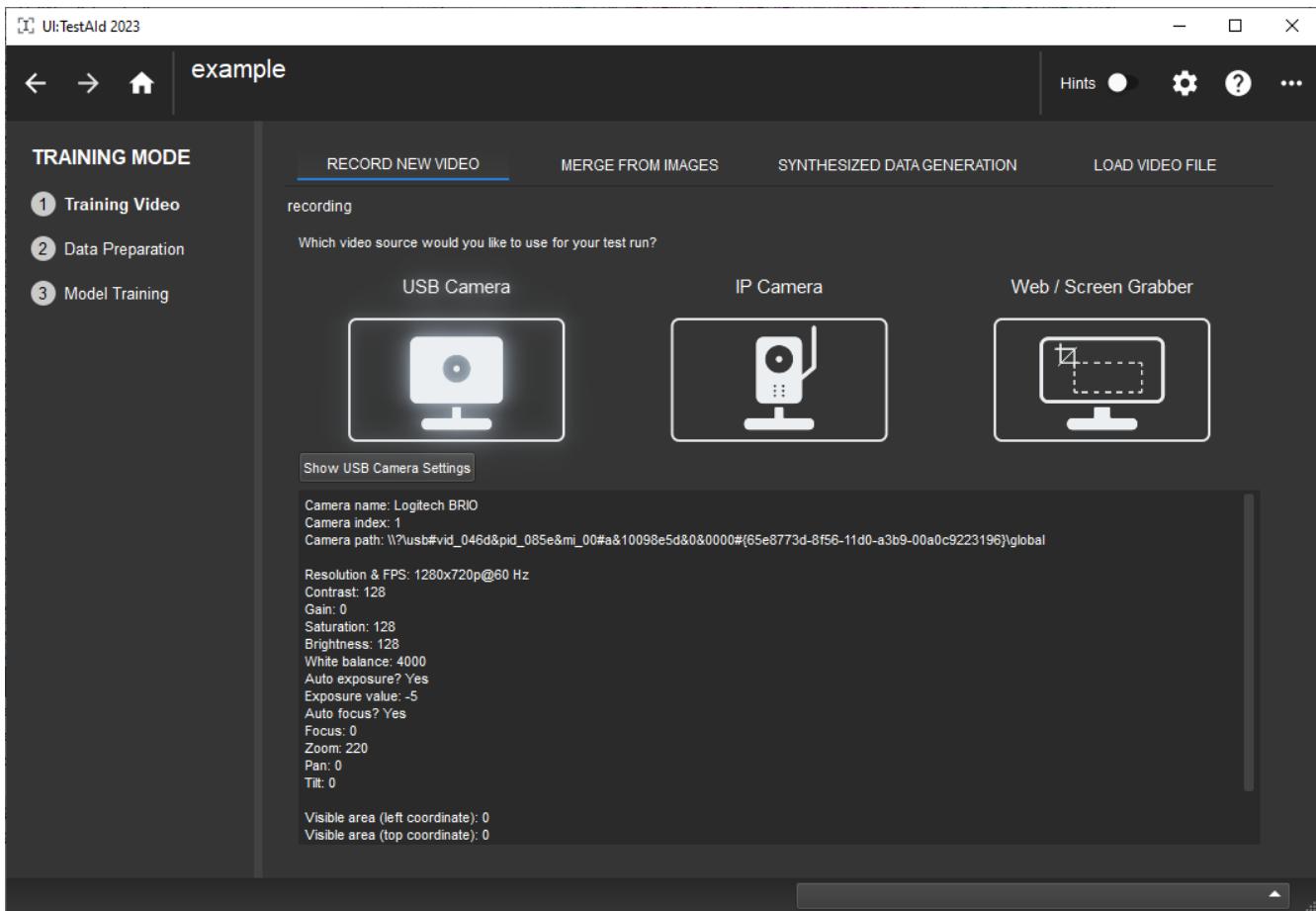
After the training project created, you should prepare the training videos. You can either record the video by yourself or with the help of the video recorder embedded in AI-Core.

Here, you should choose either to record a new video, or to load an existing one.



### 4.2.1. Create New Video

By clicking "Record New Video", you will be presented with the page in the figure below.



Here they can choose which source to use:

- USB Camera
- IP Camera
- Web/Screen Grabber

By clicking on the icon one time, the information about the settings will appear in the box below. To proceed test, one must double-click on the desired source.

It is also possible to change the settings, by clicking on the "Show ... Settings" button. This leads into a screen, where the user can modify them, while seeing the video live.

In the setting of USB Camera, you can access several camera types:

- Windows Media Foundation Camera
- Basler Camera
- Alvium Camera
- Daheng Imaging Camera
- Baumer Camera
- Intel RealSense Camera

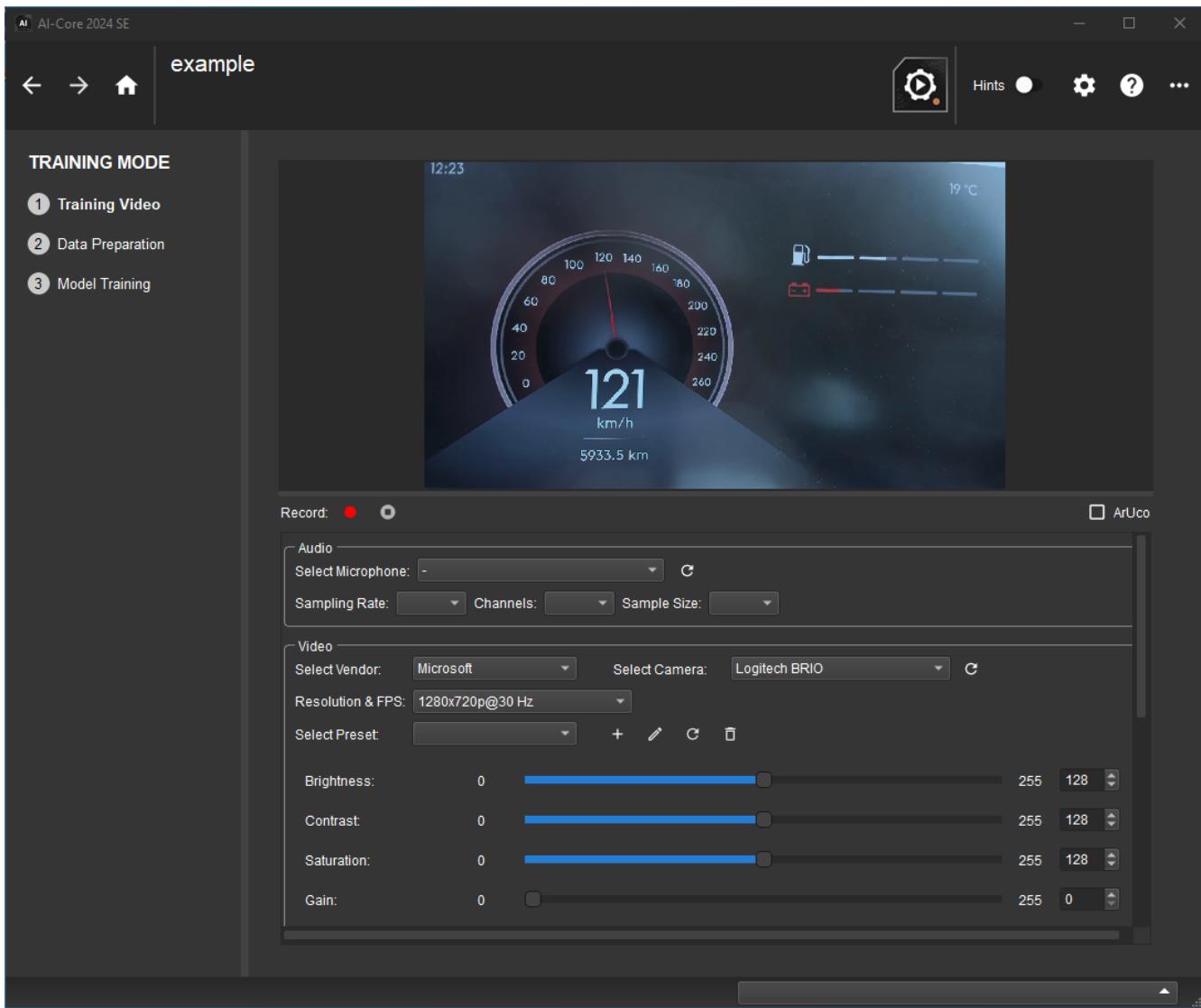
### **Notice:**

To show the options of Basler, Alvium, Daheng and Baumer camera, you need to install the corresponding SDK runtime libraries in your system. You can download them from the following

links. You might need to register an account at the camera manufacture website to get the real download link.

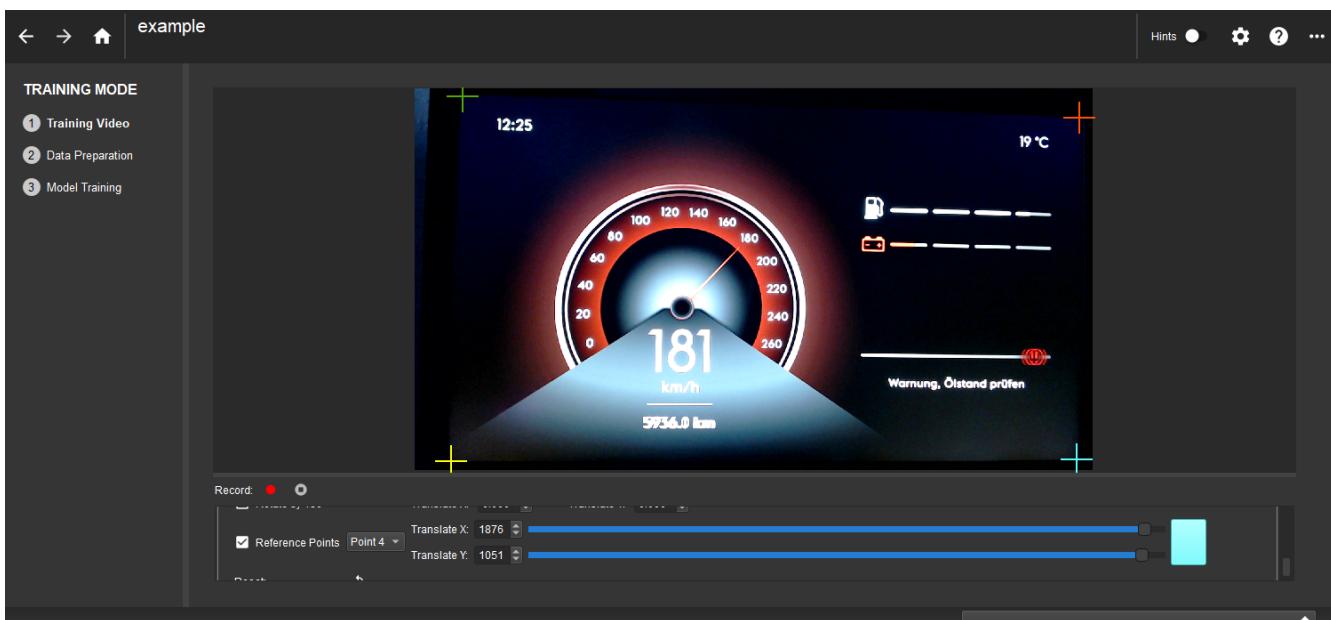
- Basler
  - Download [pylon 7.5.0](#) and install it
  - Add "C:\Program Files\Basler\pylon 7\Runtime\x64" to the environment variable "Path"
  - Restart your computer to let the driver and settings effective
- Allied Vision
  - Download [Vimba X 2024-1-Win64](#) and install it
  - Add "C:\Program Files\Allied Vision\Vimba X\bin" to the environment variable "Path"
  - Restart AI-Core
- Daheng Imaging
  - Download [Galaxy Windows SDK \(V2\)](#) and install all components
  - Make sure that the environment variable "Path" contains all SDK directories (this should be done by SDK installation program itself automatically)
  - Restart AI-Core
- Baumer
  - Download [Baumer GAPI SDK 2.14.1](#) and extract the zip file to a wished directory, e.g. "C:\libraries\Baumer\_GAPI\_SDK\_2.14.1\_win\_x86\_64\_cpp"
  - Add the extracted folder path, e.g. "C:\libraries\Baumer\_GAPI\_SDK\_2.14.1\_win\_x86\_64\_cpp\bin", to the environment variable "Path"
  - Restart AI-Core

Click the red recording button, AI-Core will start recording the video stream. Then, click the stop button to stop recording. A dialog will pop up to name the new video. After finishing recording, please switch to "Load Video File" to check the recorded video and proceed next steps.



#### 4.2.2. Using Reference Points

To guarantee the exact placement of a USB camera across multiple recording sessions, multiple reference points can be activated. To activate the reference points, press the corresponding checkbox inside the USB camera settings.



Four reference points will appear, each of which can be selected separately. The position can be freely changed within the bounds of the captured screen. To ensure that the reference points are clearly visible in all situations, the color for each point can be freely changed by clicking the button on the right.

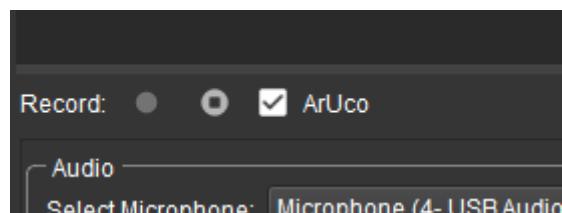
#### 4.2.3. Marker based Perspective Transformation

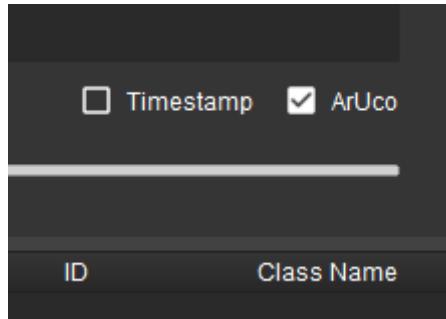
In case of hardly fixing the camera at an ideal position, AI-Core provides a function called marker based perspective transformation. It helps you keep the view always straight towards the test region.

You need to print four ArUco markers which can be found in appendix [ArUco Markers](#). and stick them at the four corners of the wished region as the following figure.



If you select the ArUco checkbox in camera configuration panel or in labeling manager,





AI-Core will automatically search for ArUco markers during test or recording videos. Once it finds four wished markers, AI-Core will treat the internal corners (marked by red circles in the above figure) as the boundary of the target region. The marked region will be cropped and transformed to the same resolution as the video source. The perspective will be rectified.

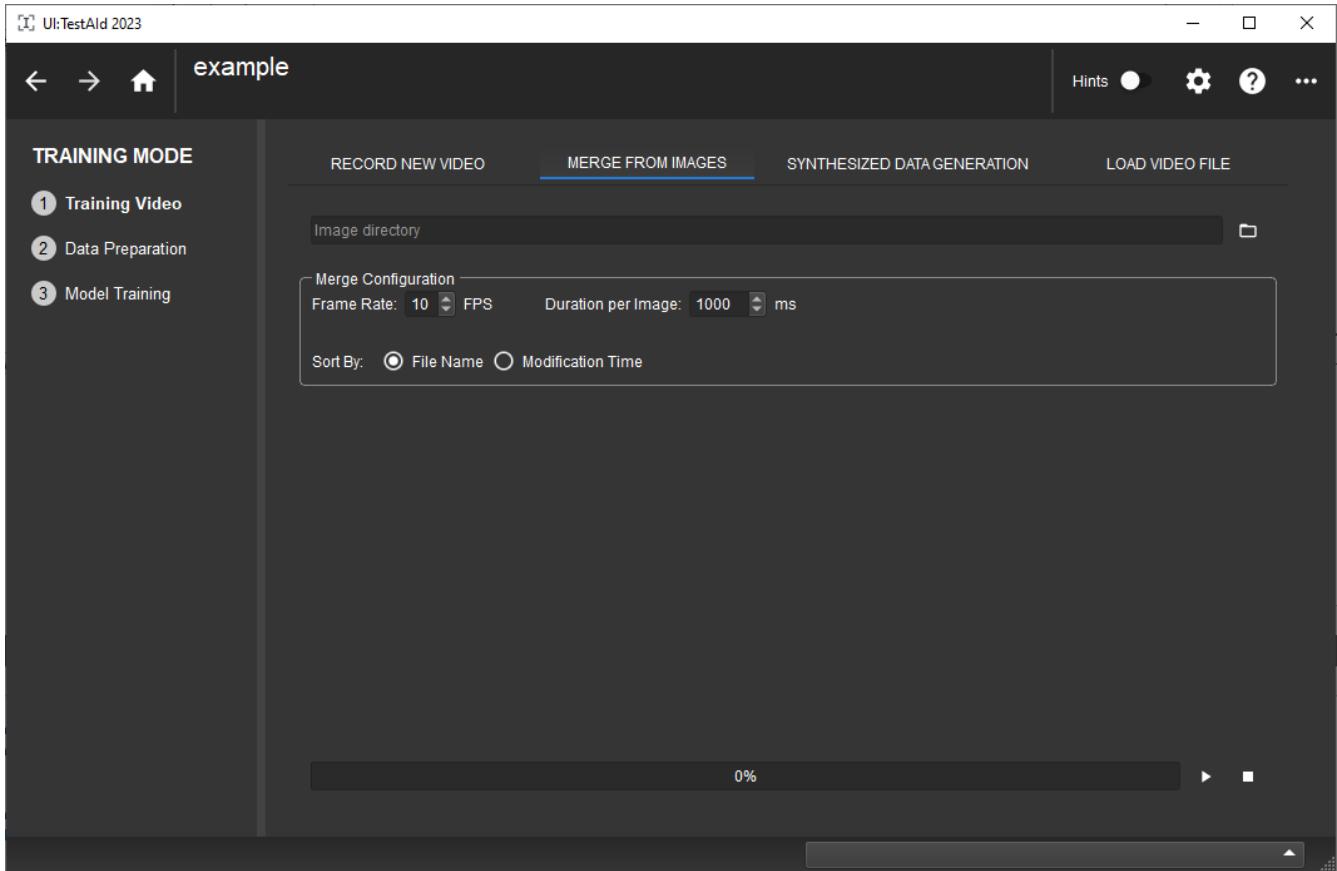
#### **Notice:**

- The order and orientation of the four markers do influence the final result.
- The scale of the markers is irrelevant for the detection. You can scale the markers according to your own test case.
- Make sure that the edge of all markers are clear. If the markers are stucked on a dark surface, please leave thin borders around the markers.

#### **4.2.4. Merge From Images**

If you have a group of sample images, you can merge them to one single training video. You should first select a directory which contains these sample images. Next you should set a bunch of parameters.

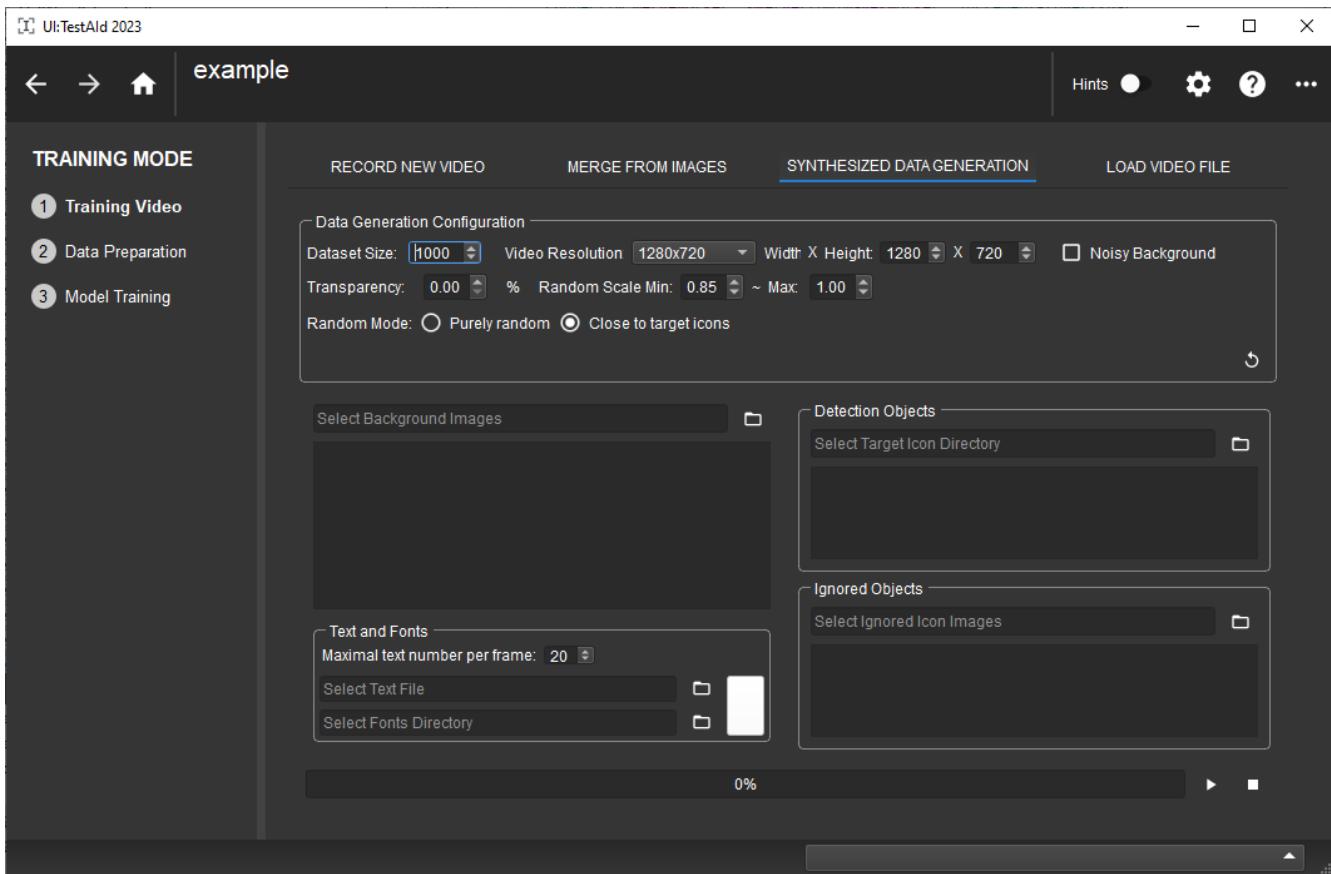
- **Frame rate:** defines how fast the video will be played.
- **Duration per Image:** defines how long each sample image will be displayed in the video.
- **Sort by:** defines the order of displayed images. You can either select sorted by file names or modification time of the image files.



After the configuration, press the play button to launch the image merging process.

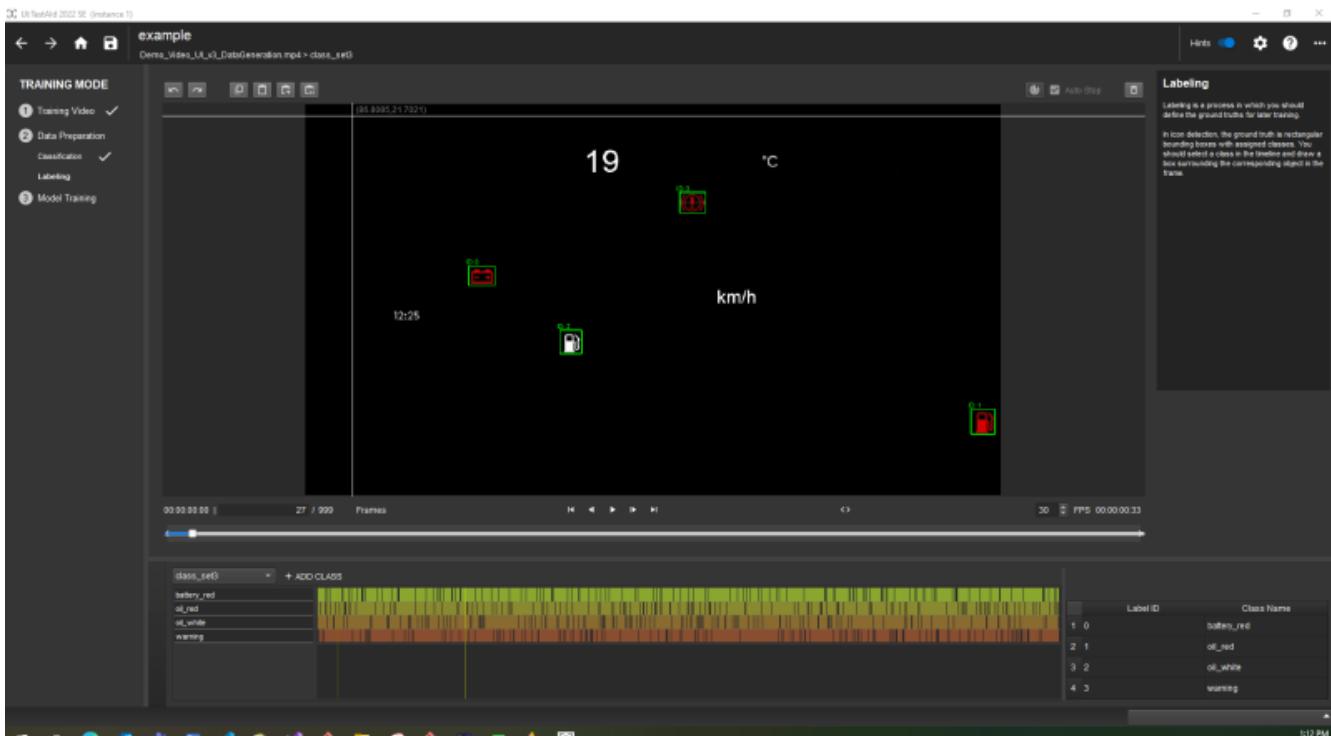
#### 4.2.5. Synthetic Data Generation

The data generation interface enables you to create a training dataset automatically as a video with the length of dataset size without labeling it manually, with each frame is a background with a set of icons on it. By clicking on "Generate Synthetic Data", you will be presented with the page in the figure below.



- **Data Generation Configuration:**

You can configure some parameters of Dataset generation in the figure below.



- Available settings are:

- Data Set Size: It can generate a video with the length or number of frames in the range of [500, 10000].
- Video Resolution: The resolution of videos supports all standard resolutions plus customized

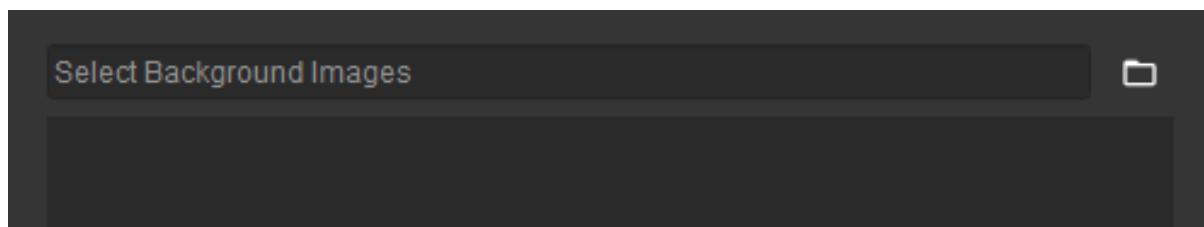
resolutions.

- Transparency: The transparency is a factor in range [0.0, 1.0] with 0.0 means the icons are completely visible on the background image and 1.0 means invisible.
- Random Scales: The size of icons on the backgrounds will be randomized in the range of given min and max scaling factors.
- Random Mode: Sometimes, there are a lot of objects close to the target icons, in order to have a more stable prediction, we can choose "Close to target icons" to generate more ignored texts and objects near the objects which we want to actually detect.
- Reset Symbol: By pressing on the reset symbol resets all parameters to their default values in the figure below.



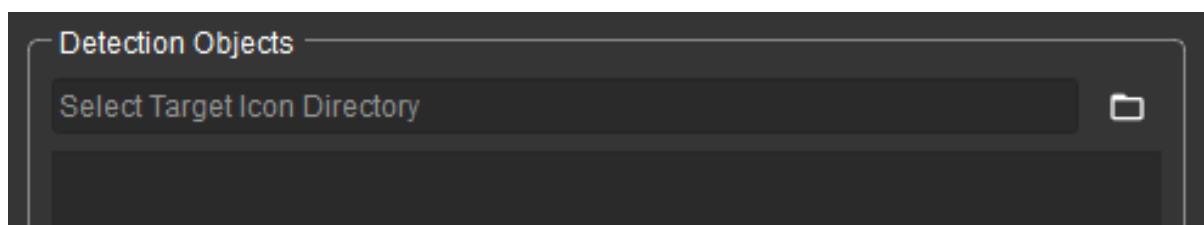
- **Select Background Images:**

The dataset generation is convenient when the background does not change a lot and icon files are available. By clicking on the folder symbol, you can choose your background image files in the formats of (\*.png \*.jpeg \*.jpg \*.jpe \*.bmp \*.dib \*.pbm \*.pgm \*.ppm \*.pxm \*.pnm \*.sr \*.ras \*.hdr \*.pic) in the figure below.



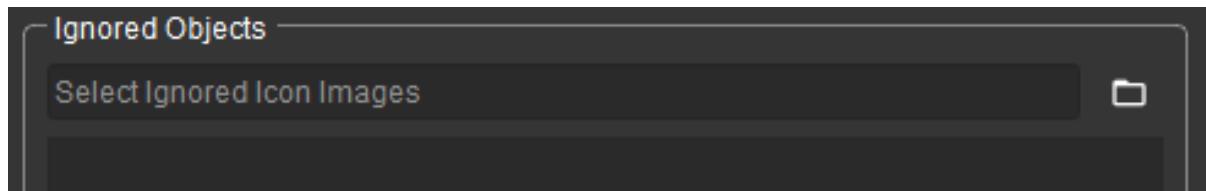
- **Select Target Icon Directory:**

The target icons are the ones to be trained and get labeled. There is also ignored icons, which are icons added only to aid the training process by adding them on the backgrounds, so that the training model be aware of them. By clicking on the folder symbol, you can choose your target icon directory. Inside the folder, it need to be at least 3 icons with the format in (\*.png \*.svg) in the figure below.



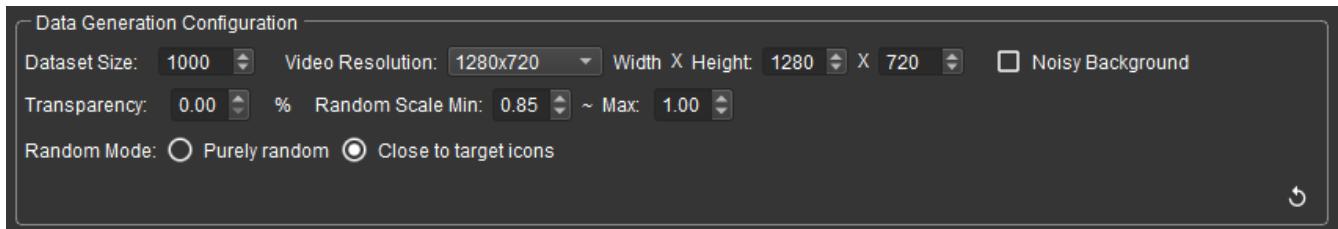
- **Select Ignored Icon Directory:**

In order to aid the training process, the ignored icons are those objects which should not be detected and ignored on purpose. By clicking on the folder symbol, you can choose your ignored icon files. The format of icons are in (\*.png \*.svg) in the figure below.



- **Add Text on the Backgrounds:**

In order to aid the training process more, you can add some customized text file, which is a set of user comma, new line and tab separated keywords. the format and style of text can be customized in the figure below.

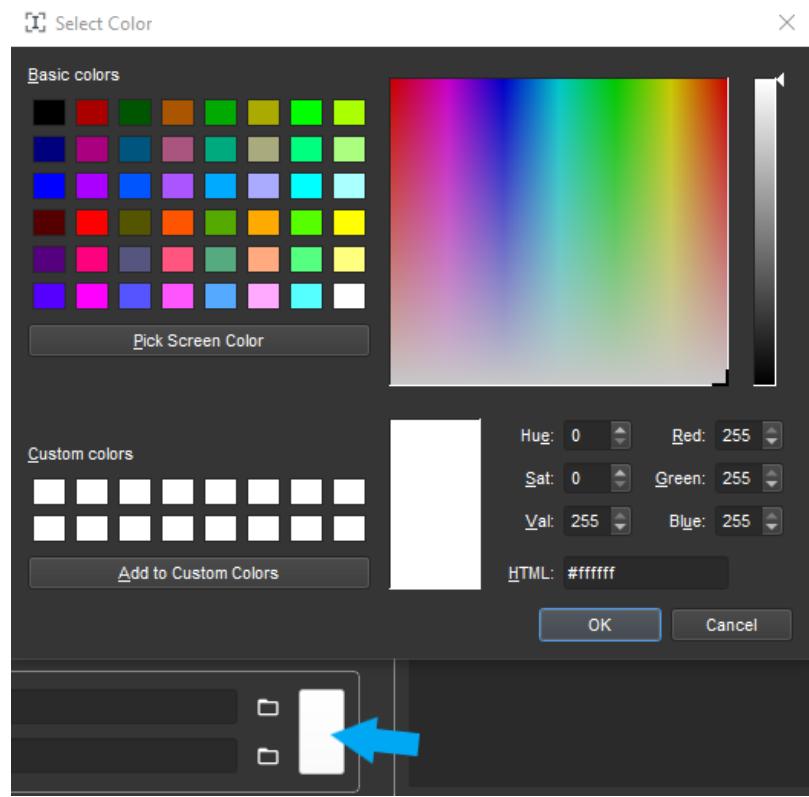


- Available settings are:

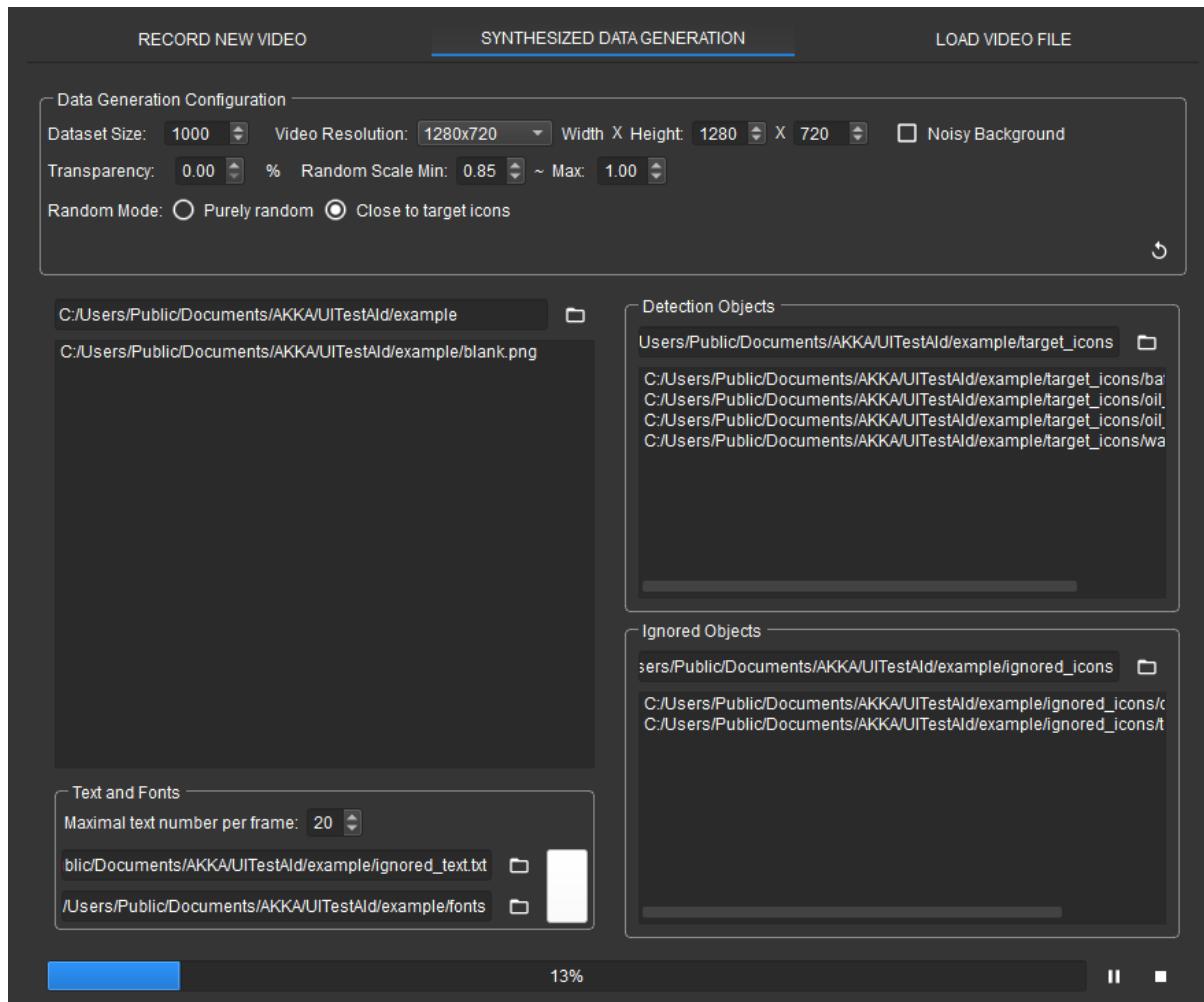
- Select Text File: By clicking on the folder symbol, you can choose your text file in the format of \*.txt. The text file is a set of keywords, which are separated with comma, new line, and tab in the figure below.

```
1 different New Articles page 23 a To Service a details DC that don't as 7 Date: #1 : AZ
2 12:30 2022.05.09 °C rider
3
4
5
```

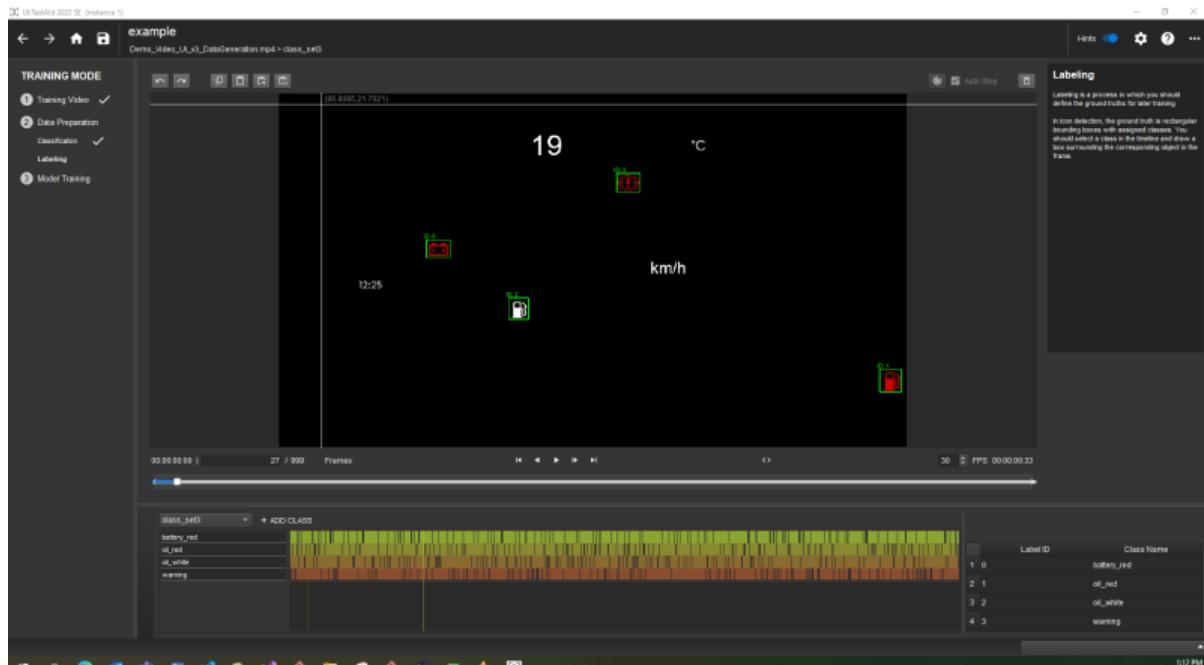
- Select Fonts Directory: By clicking on the folder symbol, you can choose your font directory with at least one font file inside it. The format of fonts are in (\*.ttf, \*.TTF). In the case of more than one font, it selects all fonts one by one.
- Choose Text Color: By pressing on the color picker symbol, it lets you to choose your customized color tab in the figure below.



- Run The Data Generation: After fill in all settings and input boxes, you can run the process with pressing on the play key. You can pause it, stop it, or let it complete the generation. After it completes, it goes automatically to "Load Video File" page for more processes in the figure below.



Now you can load the dataset and choose the right class set. Your dataset is labeled out of the box and ready for model training in the figure below.



The limitations of the synthetic data generation:

- In the actual test, the background should be in pure color or relatively static.
- The objects/icons should not be crowded or overlapped with each other.

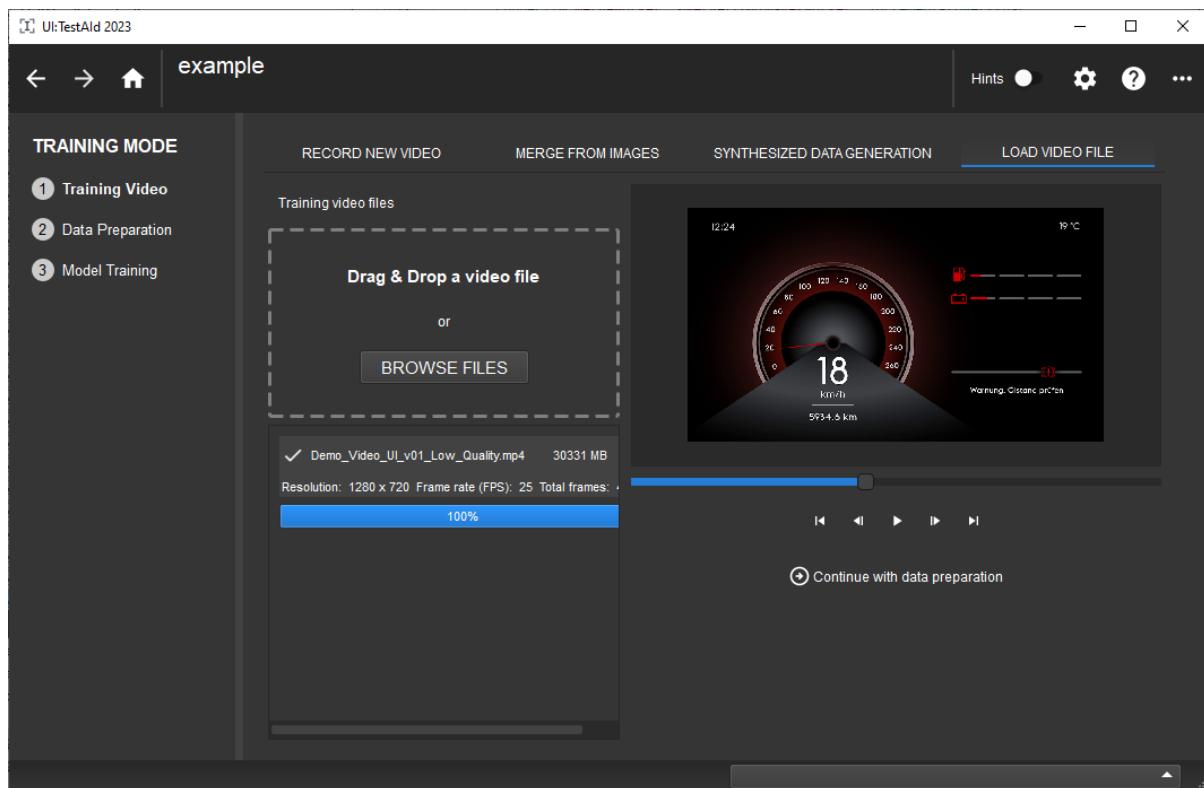
- If the final detection results are not ideal, you should still record actual videos and manually label them.

#### 4.2.6. Load Video File

In the page of "Load Video File", you can load existing video file to the training project. If you previously have recorded a video in AI-Core, it should also be automatically listed here.

AI-Core supports following video file formats:

- AVI: .avi
- MPEG-4 Part 14: .mp4
- MPEG-1: .mpeg
- Flash: .flv
- Windows Media Video: .wmv
- RealMedia Variable Biterate: .rmvb
- QuickTime File Format: .mov
- Matroska: .mkv



By simply dragging the intended video file, or selecting it by clicking on "Browse Files", it is possible to load a new video. Through selecting one of the listed video, the video will be played on the right side.

If you want to remove one of the video, please right click it and select "Remove Video". Please take care that if you delete the video, all labeling data under it will be removed as well.

Sometimes, the source video could have not been recorded or encoded correctly. It will lead wrong

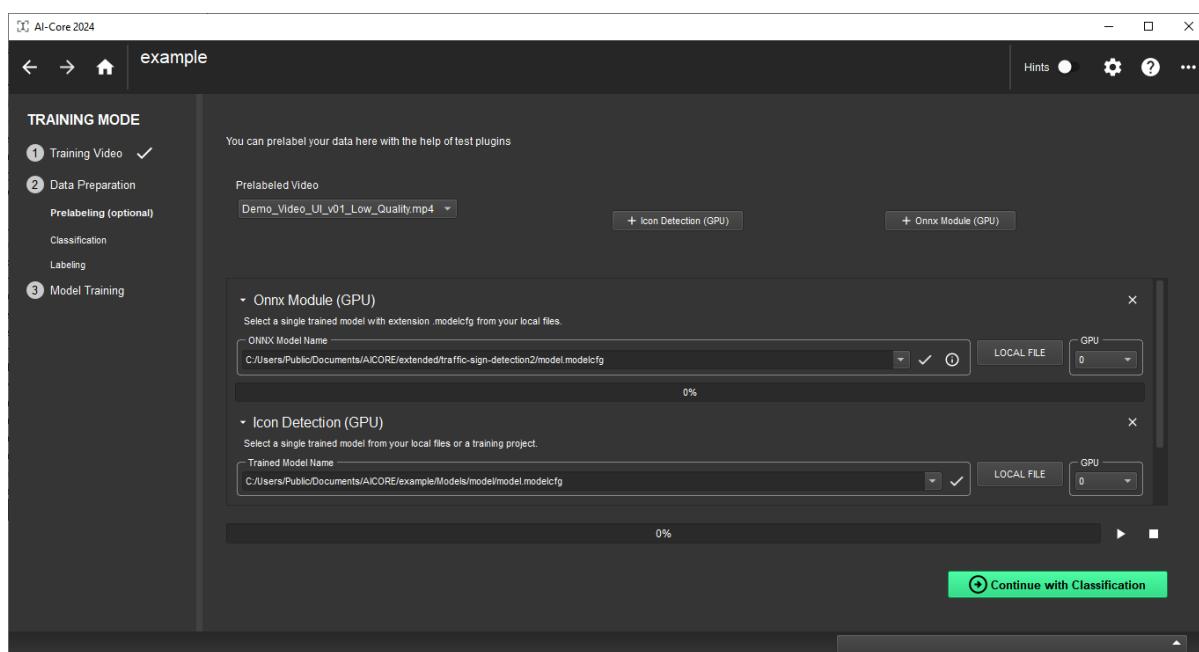
total frame number, mismatched frame indexing, or even loading failure. The transcoding option helps you fix and compress corrupt videos. If you activate this function, AI-Core will transcode the source code to H.264 and fix errors in the video format context.

## 4.3. Prelabeling

Since version 2024, AI-Core provides a new function in training mode. It is called "Prelabeling". Its purpose is to reuse your pretrained icon detection or Onnx Module network model to prelabel a new video. It is especially useful when you have a continuous project, in which you have more and more objects needed to be labeled.

The neural network models cannot be directly combined, because the model itself is a black box. With the prelabeling, you can however load one or more pretrained neural network models and let them label a specific video automatically. The models will create a new class set and detect the previously trained objects. If the class names are duplicated, the detected objects will be added to the same class.

If you do not have pretrained model for labeling, you can click the green "Continue with Classification" button and skip to the next step.



To conduct the prelabeling, you need to do following steps:

1. Select one of the videos which you have added in the last step "Prepare Video".
2. Create one or more model selectors by clicking the corresponding plus button.
3. Select the configuration file of your wished pretrained model (\*.modelcfg). Take care that the model should belong to the selected test plugins.
4. Click the play button to start prelabeling. A message box will pop up. You should give a name for the newly created class set. The video will be labeled by the selected neural network models one by one. You can stop the prelabeling by clicking the stop button.
5. After all, click the green "Continue with Classification" button and continue to the next step.

If you want to use Onnx Module to prelabel the data, you need to notice following points additionally:

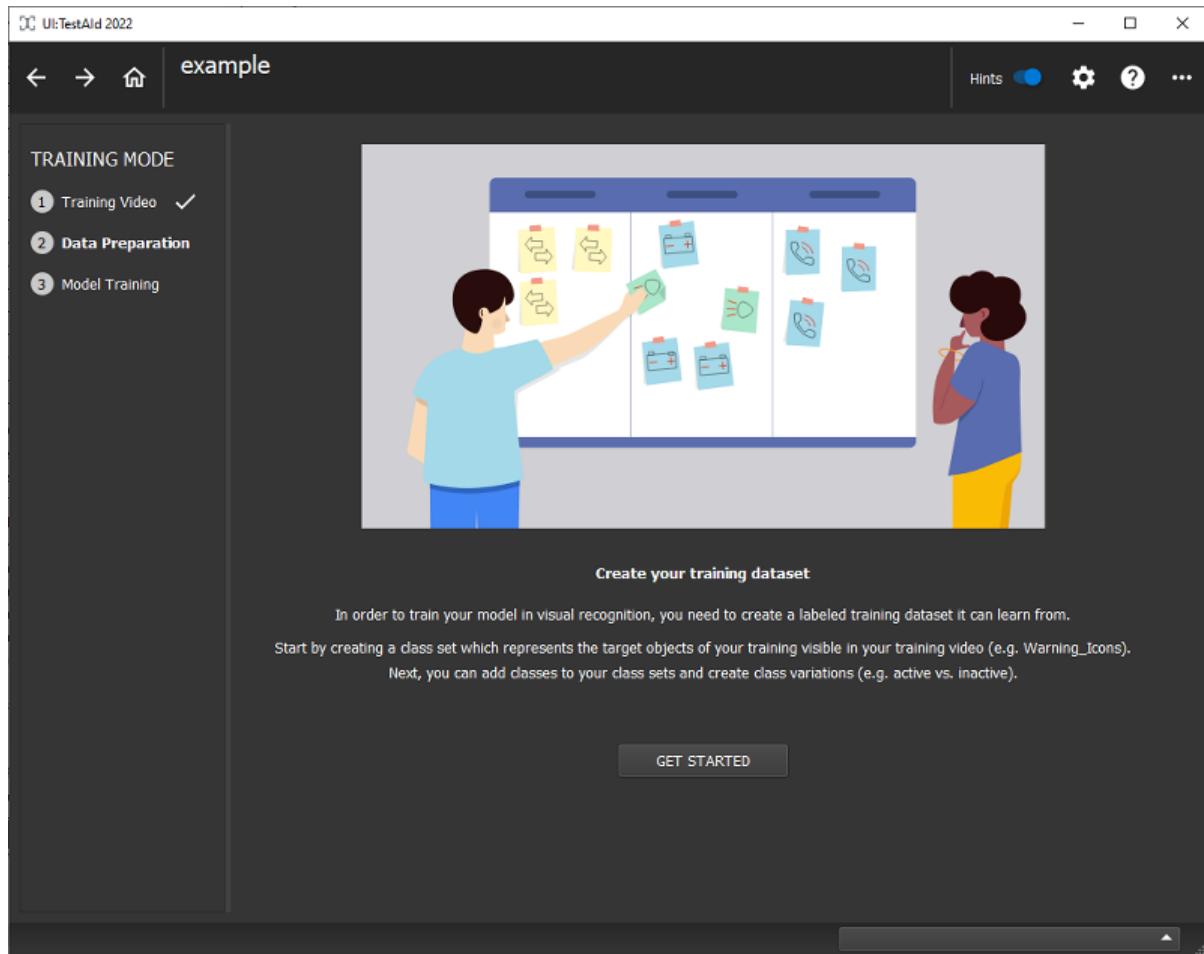
- Currently the prelabeling only supports models with object detection output type.

- The prelabeling is also influenced by the time and processing count limitation of the model. You can click the information button to check the remaining processing count number.

## 4.4. Classification

Click on "Continue with data preparation" in "Training Video", you can see the introduction page about the next phase "Create your training dataset".

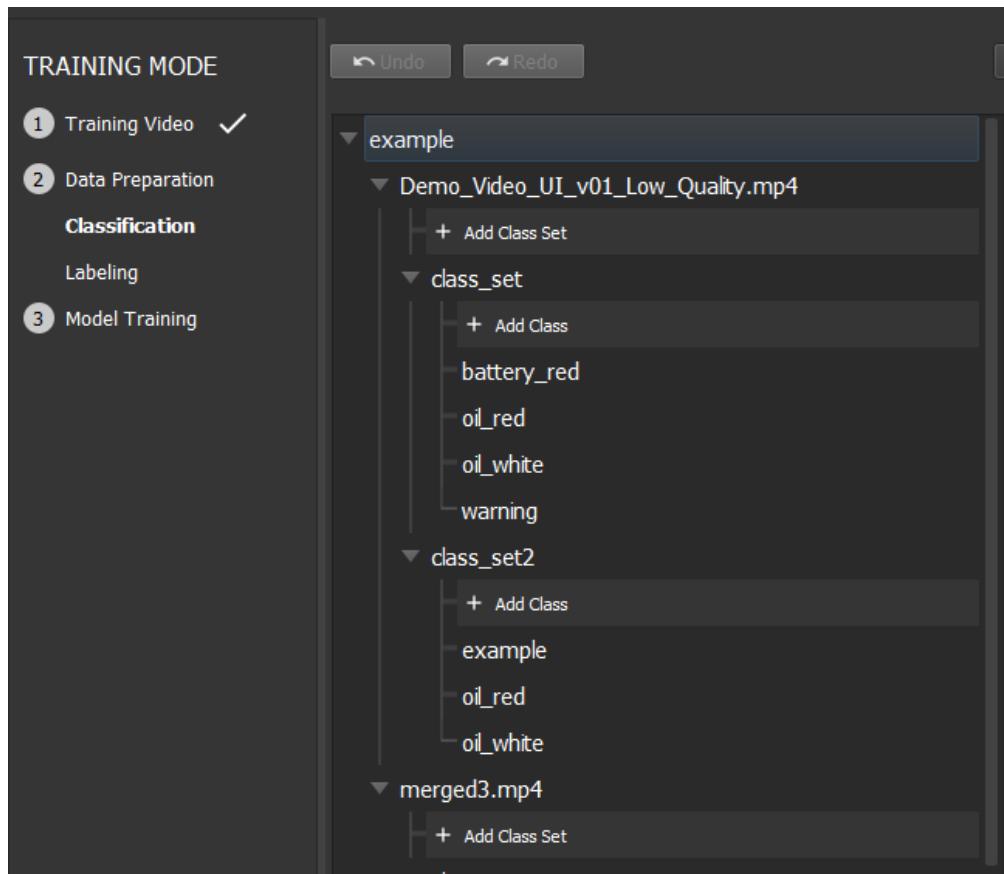
In order to train your model in visual recognition, the machine should understand and learn the target objects, such as warning icon. You need to first create a class set in which all target objects in the training video should be named (or classified).



To advance, simply press on the button "Get Started".

In the step "Classification", you can manage the class sets and their classes which will be used to train the model. You can see a tree view of the project information. The project is organized in form of a hierarchical structure. The first level is the imported training video. One class set always depends on a training video. Under a class set, you can define multiple classes. In the next step "Labeling", you are still allowed to modify your created class set.

You can click button "Expand All" and "Collapse All" to show or hide all details in the project structure viewer.



#### 4.4.1. Create/Delete Class Set

Click the option "Add Class Set" under a video and give a name, you can then simply create a class set for this video. If you want to cancel the creation, just leave an empty name, AI-Core will ignore the creation.

To delete a class set, you just right click it and select "Delete Class Set" or select it and press "Delete" key. Then, the selected class set and all classes under it will be deleted.

**Notice:** You can click button "Undo" to undo these operations. However, if you leave the classification page, the operations will be clean up. Namely, you cannot undo or redo these operations any more.

#### 4.4.2. Create/Delete Class

The operations of class creation and deletion are similar to the ones for class set.

Click the option "Add Class" under a class set and give a name, you can then simply create a class. If you want to cancel the creation, just leave an empty name, AI-Core will ignore the creation.

To delete a class, you just right click it and select "Delete Class" or select it and press "Delete" key. Then, the selected class set and all classes under it will be deleted.

#### 4.4.3. Check Labeling Status

When selecting a class set, the labeling status will be shown on the right side.

In so-called list view, all the classes, as well as their numbers of objects and labels will listed. The difference between object and label is that object indicates the same object instance with the same object ID in a video. It can appear in multiple frames. The label number indicates the total number labeled frame of a specific class. In the case of icon detection, you just need to take care about the label count.

To delete a class, you just right click it and select "Delete Class" or select it and press "Delete" key. Then, the selected class set and all classes under it will be deleted.

The screenshot shows the 'List View' mode of a labeling application. At the top, there are buttons for 'Expand all' and 'Collapse all'. To the right are icons for 'List View' (a grid with a camera icon) and 'Thumbnail Gallery' (a grid of squares). Below these are instructions: 'Drag & Drop a training config file' or 'BROWSE FILES'. A message below the table says 'Demo\_Video\_UI\_v01\_Low\_Quality.mp4 > class\_set'. The table displays the following data:

Class Name	Object Count	Label Count
battery_red	1	2022
oil_red	1	927
oil_white	2	3542
warning	2	3484

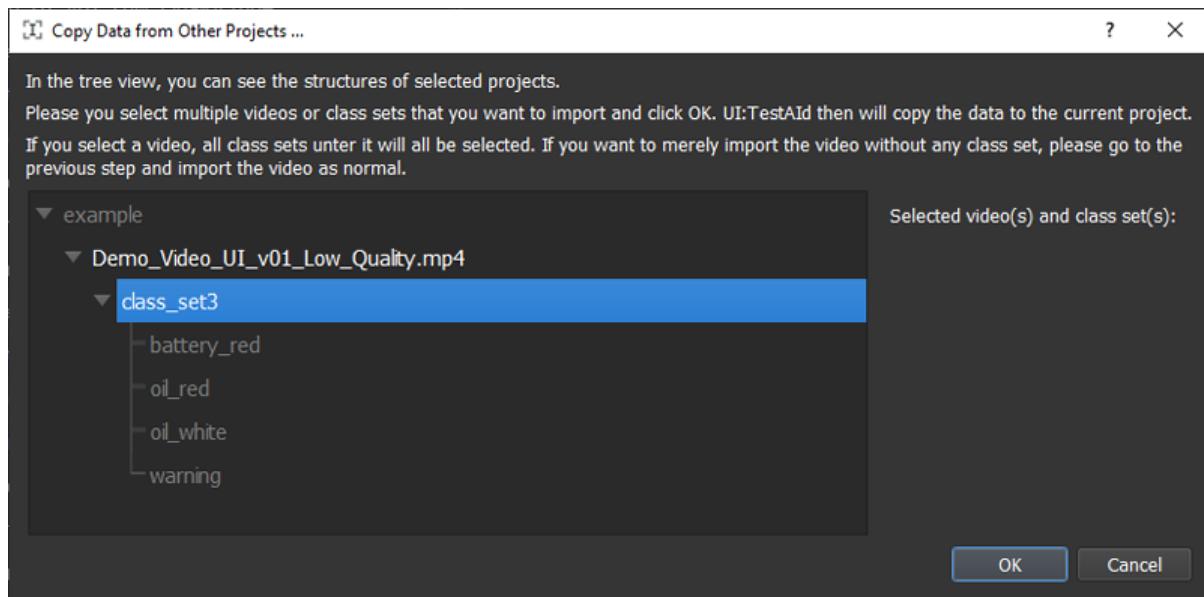
There is also an option "Thumbnail Gallery" (top-right corner). It will show a cropped image of each class.

The screenshot shows the 'Thumbnail Gallery' mode of the application. At the top, there are buttons for 'Expand all' and 'Collapse all'. To the right are icons for 'List View' and 'Thumbnail Gallery'. Below these are instructions: 'Drag & Drop a training config file' or 'BROWSE FILES'. The table from the previous screenshot is shown again, but now each row includes a thumbnail image. For 'battery\_red', there is a red battery icon with dimensions 54x39. For 'oil\_red', there is a red fuel pump icon with dimensions 42x50. The other rows are identical to the List View table.

#### 4.4.4. Copy Class Set from Other Projects

If you want to copy a labeled class set from another project to the current one, you can drag and

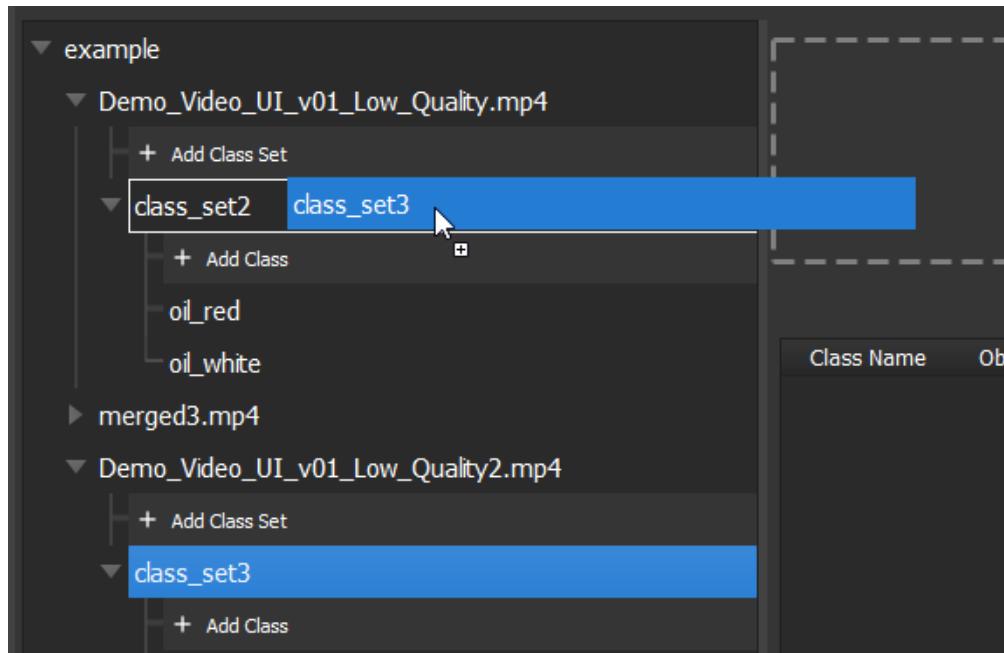
drop the traincfg file of the external project to the dashed-line area. AI-Core will pop up the following dialog to ask you which class set and its training video you want to copy.



After you select and confirm, AI-Core will copy it to your current project. If the name of the imported video is conflict with one in the current project, AI-Core might remind you of renaming it.

#### 4.4.5. Concatenate Class Sets

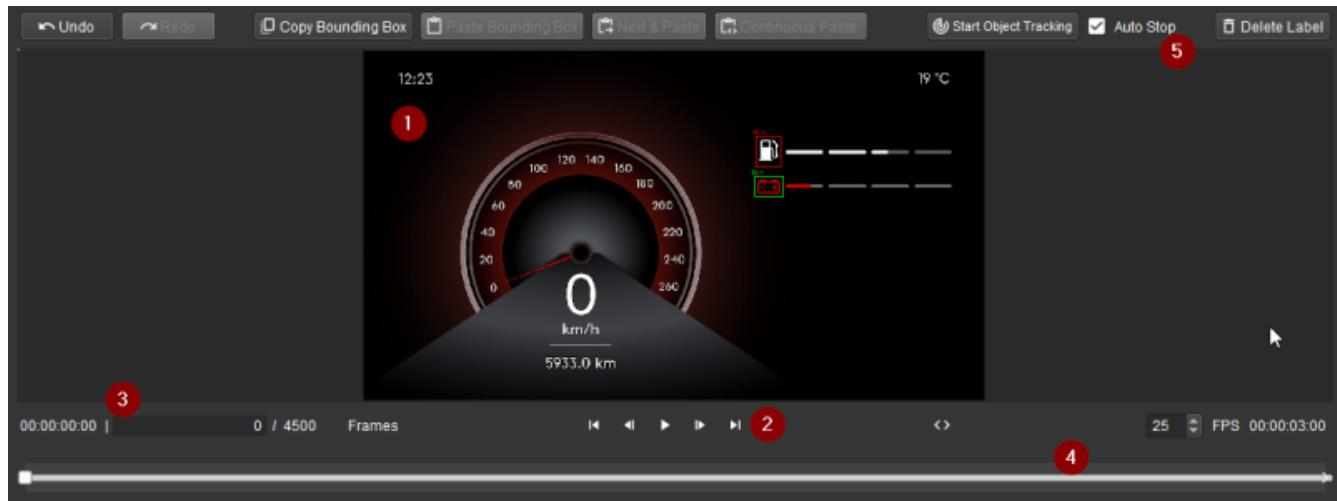
When your project has two labeled videos and want to concatenate them together, you can drag one of the class set and drop it to the other one. AI-Core will then concatenate the both videos to a new merged one. The class names will be fused together. If there are duplication, they will be considered as the same class.



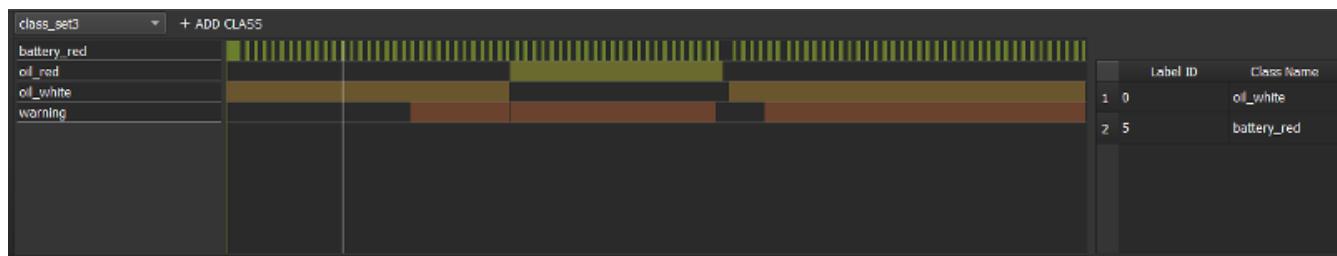
After the concatenation, you will see a new video and class set in the project structure view. The source videos and class sets will not be changed.

## 4.5. Labeling

After the class set configured, you should select one class set and then click "Continue with labeling", on the bottom right corner. AI-Core then will show the labeling tool.



1. The training video is shown in the center. By pressing Ctrl and scrolling the mouse wheel, you can zoom in and out. With right clicking and dragging, you can adjust the visible area.
2. You can use the buttons "play/pause", "forward/backward", "jump to the beginning/end" to control the video playing.
3. On the both sides of the control panel, you can see the current frame number and timestamp in the video.
4. With slider bar, you can drag the control handler to adjust the displayed frame. If you drag the range block, you can adjust the view range of the time line, which will be later introduced.
5. In the tool panel, you can see multiple labeling assistant functions. They will be introduced later.



Under the labeling tool, there is a dock widget. It can be docked in AI-Core's window or floating separately by dragging the right sidebar.

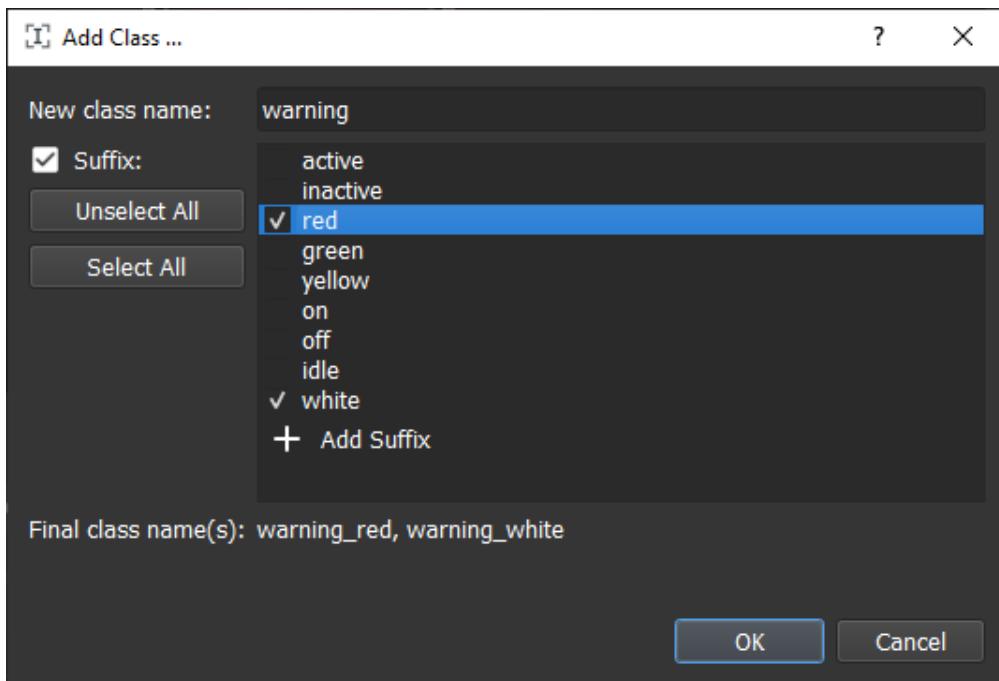
In the dock widget, there is a so-called timeline. With it, you can overview the labeling status. Each class has one single row with its individual color. From the timeline, you can find which section has been labeled or not. Through scrolling the mouse wheel in the area, you can adjust the view range. It is synchronized with the slider bar in the control panel.

On the right side of the dock widget, you can find the labeling information in the current frame.

#### 4.5.1. Add/Delete/Rename Class

##### Add Class

As introduced in the last section, in the labeling phase, you can still add and delete classes. By clicking the button "ADD CLASS" in the dock widget, AI-Core will show a dialog to add classes.

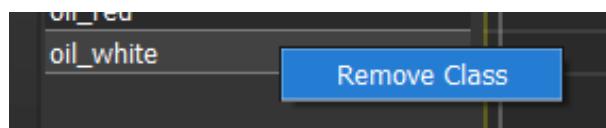


In the dialog, you can define the name for the new class. It often occurs that one object has multiple statuses. For example, an warning icon is while when it is deactivated. However, when there is a warning, it will becomes red. In this case, the both statuses should be treated as different classes, since they look different. The dialog allows you to create multiple classes with the same base name but different suffixes to distinguish them with each other. In the picture, the suffix is activated by the check box in the left top corner, "red" and "white" suffixes are selected. Thus, it finally generates two classes "warning\_red" and "warning\_while".

You are also allowed to customize your own suffixes by clicking "Add Suffix", and entering the new suffix name. The newly created suffixes will be saved in the application configuration. Next time, you can still reuse them. To remove one suffix from the list, you should right click it and select "Remove Suffix".

##### Delete Class

By right clicking a class name in the timeline, and selecting delete class, you are allowed to delete the class and its all labels.



##### Rename Class

When you double click the class name, you will activate the edition mode. Then, you can type the wished new class name to rename it.

#### 4.5.2. Add/Remove Label

##### Add Label

After defining class names, you can start labeling objects. Each object is labeled by a rectangular bounding box with an instance ID number. People can understand one instance as the identical object which could appear in a number of video frames. In AI-Core, each instance has its own ID.

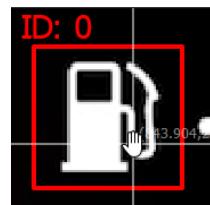
To create a single label, you need to first select a class in the timeline column.



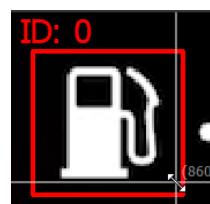
Then, draw the bounding box in the workspace by pressing left buttons and dragging the mouse from top left to right bottom of the box. Please take care that the smallest acceptable height and width of the box is 3 pixels. If the drawn box is too small, AI-Core will ignore it.



If the box is drawn wrongly, you can modify it again. Move the mouse inside of the box, then the mouse will become a hand. Press the left button and drag. The box will be moved.



If the mouse is moved closer to the boundary, the mouse will become a split cursor. Press the left button and drag. The corresponding boundary will be changed.



To delete one box, you should first select the corresponding box. If you want to select multiple boxes, press "Ctrl" key and select the boxes successively. Or, sometimes, there could be a lot of boxes needed to be removed in a specific region. You can also press "Shift" key and drag the mouse. In the workspace, AI-Core will show a selection box with yellow dashed boundary. All labels in the region will be selected.

Then, click "Delete" key. The selected boxes will be removed.

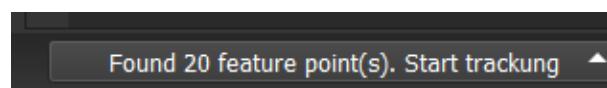
### 4.5.3. Assistant Functions

AI-Core offers a lot of useful assistant labeling function. These functions allow you to label a large number of frames in a short time if the labeled objects do not move or not change a lot. In the new version, they are integrated into some shortcut combinations:

Icon	Shortcut	Description
	Space	Play/Pause
	Left	Jump to the previous frame
	Right	Jump to the next frame
	PgUp	Jump to the first frame
	PgDown	Jump to the last frame
	Ctrl+Z	Undo
	Ctrl+Y	Redo
	Ctrl+C	Copy selected labels in the clipboard
	Ctrl+V	Paste labels in the clipboard to the current frame
	Ctrl+X	Jump and paste labels in the clipboard to the next frame
	Ctrl+Shift+V	Paste labels in the clipboard between the recorded frame and the current one
	Ctrl+F	Start object tracking
	Ctrl+F	The selected object is being tracked. Click the button again to stop the tracking
	Ctrl+T	Perform template matching on the current frame
	Escape	Cancel all operations and clear clipboard
	Delete	Delete one or multiple objects

- Object tracking

is a newly added function. It analyses one target object and follow its optical features. These features can be the textures, edges, or corners in the object region. After you press Ctrl+F, AI-Core should give you some hints about the tracking status:



Sometimes, when the target object is too small, blurring or lack of textures, AI-Core might not be able to track the object successfully. Please take care of the information "Failed to find enough feature points. Abort tracking." shown in the logging bar.

Usually, when an object disappears, the tracking will be automatically interrupted. However, if the object to be tracked is blinking, you can unselect the option  Auto Stop. Then, the tracking

will still be alive until the object shows up again.

- Template matching

If the training video is high-quality, for example recorded from frame grabber, you can use template matching.

To use this function, you need to

- first label at least one object for each class as templates.
- Then, jump to the frame that you want to label and click the "Template Matching" button .
- AI-Core will search the entire frame and compare with all templates. If one object matches one of the templates, it will be automatically labeled.

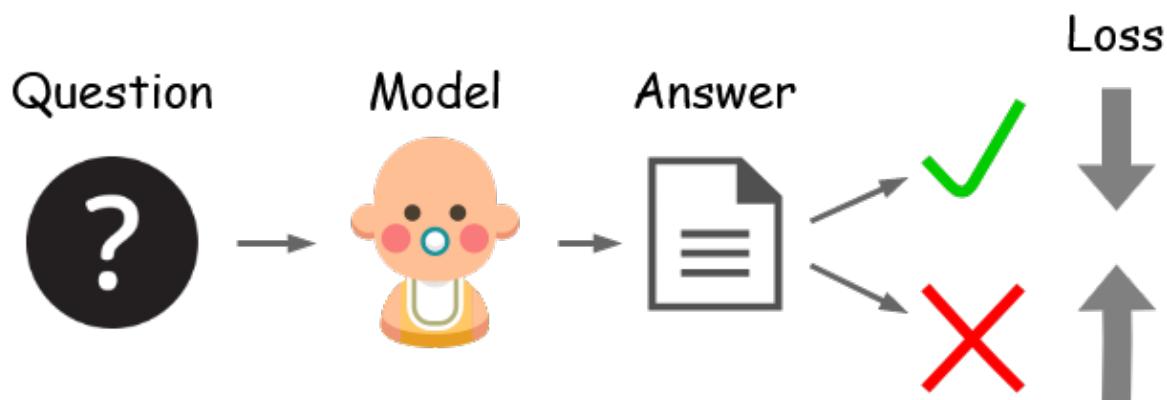
#### 4.5.4. Auto Backup

In case of unexpected interruption of the labeling, you can activate the auto backup by selecting  **Activate auto backup**. You can choose how often the data should be backup by setting the backup interval **Backup every 5.00 min**.

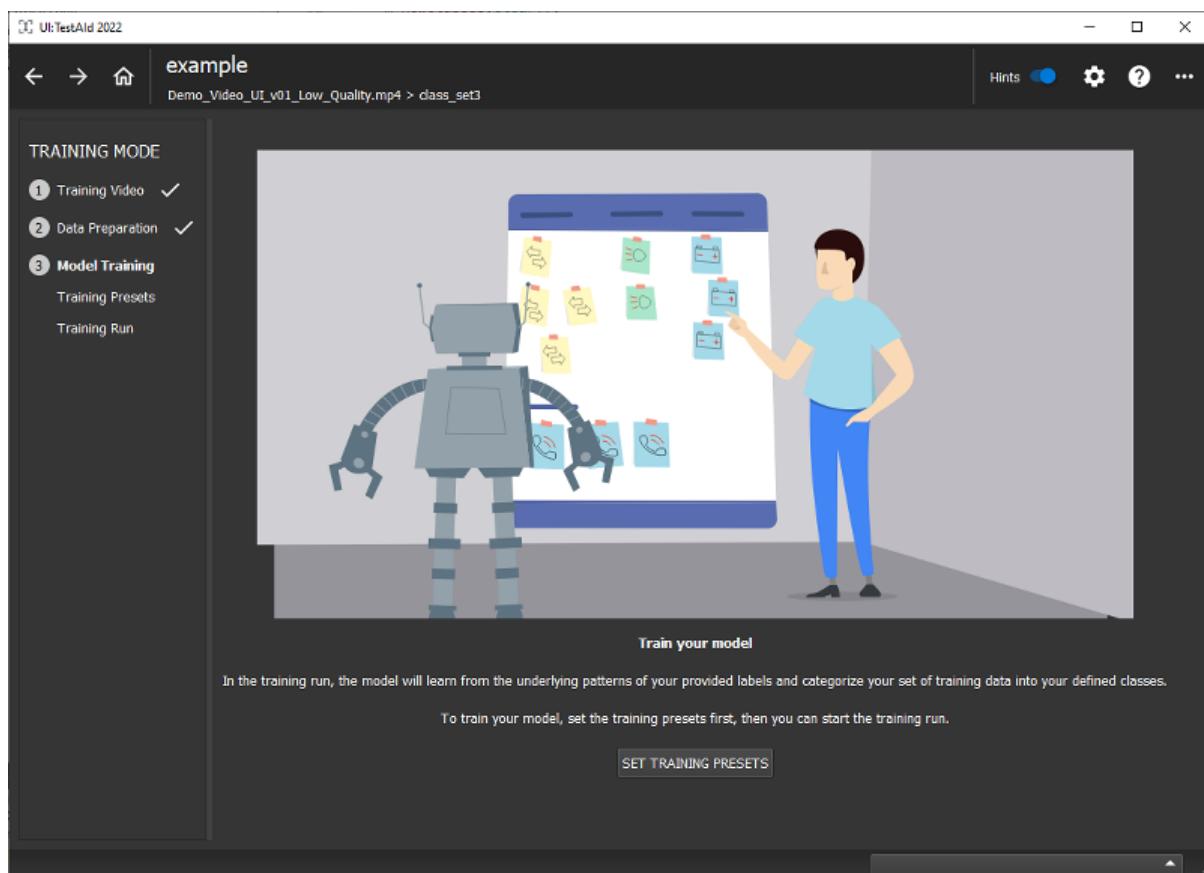
If you save the project, the backup will be automatically cleaned up. If AI-Core detects any backup data when you load the project, it will ask you whether you want to recover it.

## 4.6. Training

After finishing all labeling work, you can now train the detection model. People can imagine the detection model as a child. Labeled data is a teaching book. To teach the child to detect the objects, users need to train it. The training process will raise questions to the model successively. Once the model answers correctly, a loss value will reduce. Otherwise, the loss will rise. The model will be encouraged by a reducing loss and optimize itself.



To start training, you can click the button "Model Training" in the workflow on the left side. AI-Core might first pop up a dialog to ask you whether you want to save the labeling data. After saving, you will see the introduction page of training.

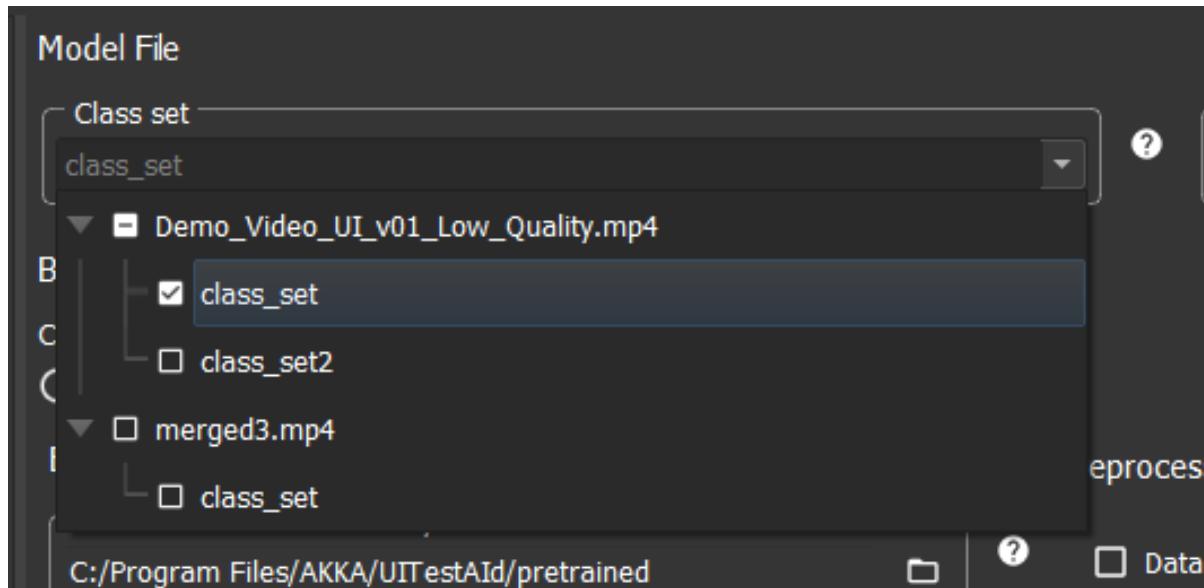


Click button "SET TRAINING PRESETS", you can go to the training configuration preset page.

#### 4.6.1. Training Presets

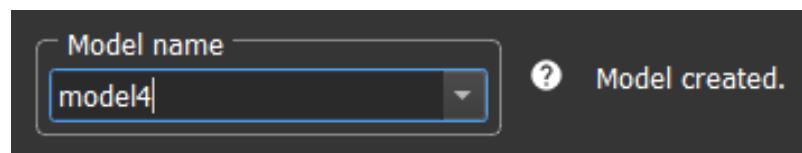
##### Select Class Set

To start the training, you should first select with which class set(s) you want to train.

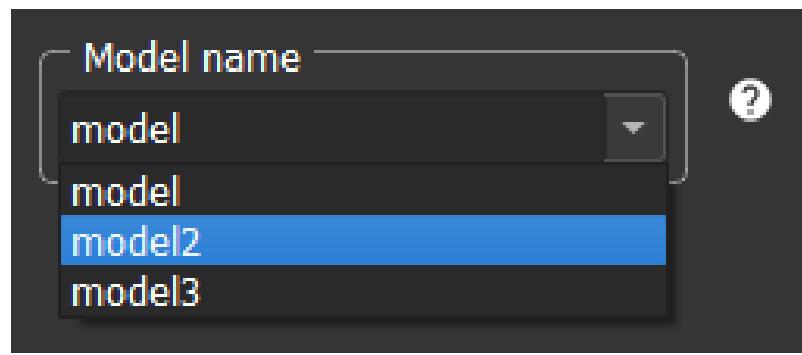


##### Select Model

For the same training data, you are allowed to conduct training for multiple times, in order to get the best result. To create a model version, you can directly type the wished model name in the field. Confirm by pressing "Return" key. Beside the field, AI-Core will show "Model created". In the Project Folder/Models, you will find a newly generated folder with the given name. The model configuration file and binary model file will be saved here.



If you want to reuse a previously created model, you can also select it from the combobox.

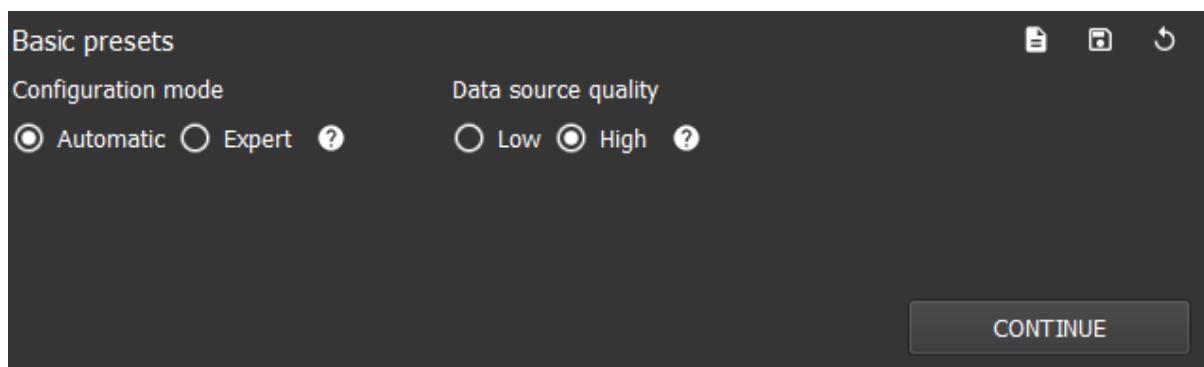


##### Set Training Configuration Parameters

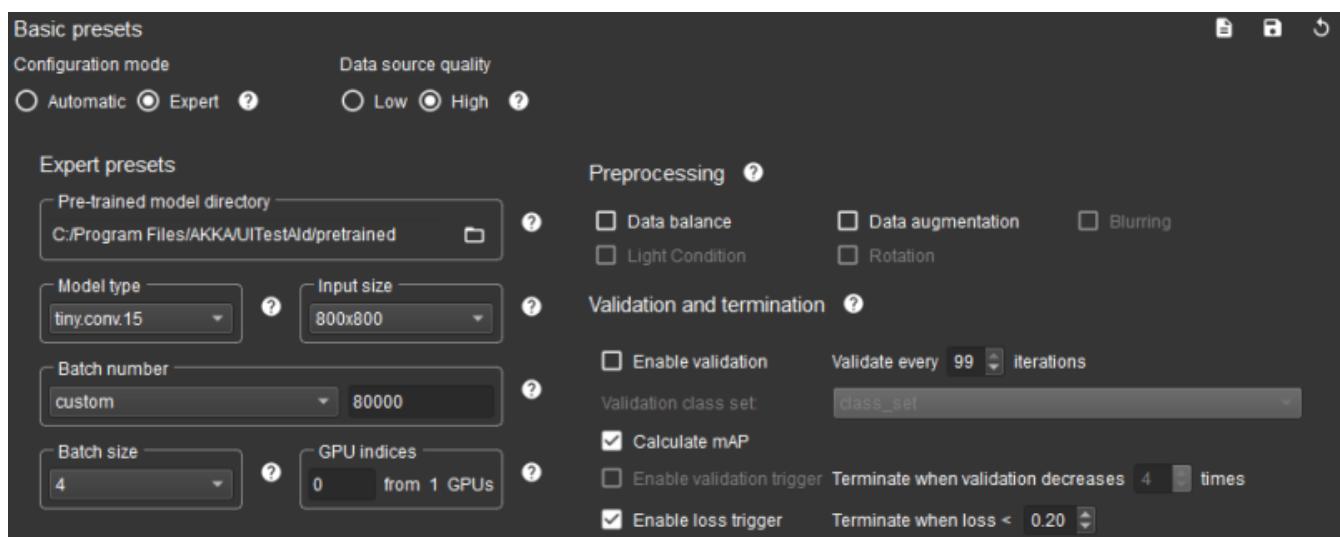
To simplify the usage for the new users, AI-Core provides two modes for users to setup the training configurations: automatic mode and expert mode. For new users, it is recommended to use the automatic mode. According to the available GPU memory and labeled data, AI-Core will automatically select the ideal input size, batch size, model type, GPU indices, and so on.

##### Set Video Quality

In automatic mode, you just need to select one binary option, namely data source quality. If your training and test video sources are captured by a camera or under complex light conditions, then you need to select "Low", so that AI-Core will perform data augmentation during training. Otherwise, if the source is always from a frame grabber or an artificial video, there is no need to undergo data augmentation process.



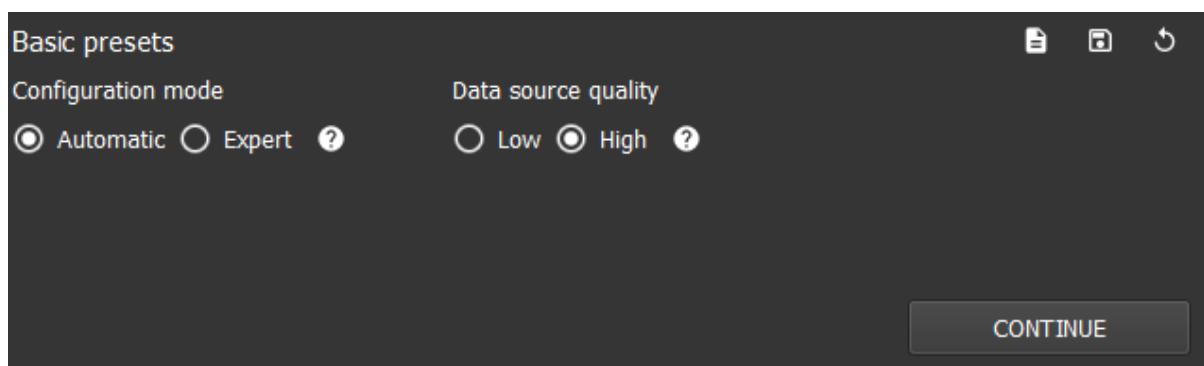
On the other side, in expert mode, you can completely control all training configurations. It gives old users which special requirements more flexibility.



Available settings are:

- **Pretained model directory**

To reduce the training time, we usually use a pretrained model. This setting tells AI-Core where the pretrained models are located.



- **Input size**

To adapt different frame size, all frame to be tested will first scaled to a fixed size. In default, we

use 416x416 pixel. However, when some objects in your training data are very small, which is smaller than 14x14 after the scaling, you should select a larger input size to achieve a high accuracy.

- **GPU indices**

When you have multiple GPUs, you can define which ones you want to use to train the model. For example, you have three GPUs and want to use the first and third one, you can type 0,2 in the field.

- **Batch number**

Control the training times. The larger batch number, the longer the model will be trained. However, according to the loss diagram, you can decide whether you want to terminate the training earlier.

- **Batch size**

Batch size defines how many questions AI-Core will raise for the model at one time. Higher batch size will increase the stability of training and probably better accuracy. However, a high batch size will increase the consumption of GPU memory and training duration. Usually, batch size is suggested setting to 8.

- **Model type**

There are several types of pretrained model. In most cases, "tiny.conv.15" is already sufficient for many challenges. It needs around 4~11 GB of GPU memory for training (depending on the selected batch size) and has fastest detection. If users have very hard detection tasks and one or more high-performance device (11 GB of GPU memory), it is also possible to train the more advanced "full53.conv.74". This model has better accuracy but needs more time to detect objects.

- **Preprocessing**

It often occurs that some rare classes are seldom labeled. Without optimization, the model will not learn these classes properly. If the option "balance data" is selected, AI-Core will replicate the labels from these classes. It will improve the detection performance of the model with respect to these rare classes. Sometimes, the actual testing environment is different from the sample videos which are used in training. For example, the light conditions or camera exposure could be changed. To mitigate these influences, users can choose the option "data augmentation". AI-Core will randomly apply some effects such as blurring or cropping to the images and simulate various situations. If you want to detect objects with rotations during test, you need to enable the rotation in data augmentation.

- **Loss-based termination trigger**

This setting can be changed during the training. It defines a loss threshold which will trigger the termination of training automatically. Usually, termination trigger is suggested setting to 0.25.

- **Enable validation**

During the training, the model will be optimized to achieve a better detection result iteratively. However, when the model learns too well for the training data, it will be overfitting, which means it can only detect the objects exactly appearing in the training data, but cannot handle other variants. In order to avoid it, we need to know when we should stop the training. It is recommended to create a separate dataset for validation. The validation dataset should have the same classes as your training one. You can choose one of your class set as the validation

dataset. If you do not have additional dataset, you can optionally choose your training dataset. AI-Core will first splits 2% samples randomly from your training data as validation set, and test the model during the training. Please notice that the splitting does not check the data distribution. If you have imbalanced dataset, in which some classes have only few samples, there is risk that these samples will not be trained or validated. The validation performance is quantified by the mean average precision (mAP) illustrated by a red line in the loss diagram. mAP takes both of classification and box positions into account.

- **Validation-based termination trigger**

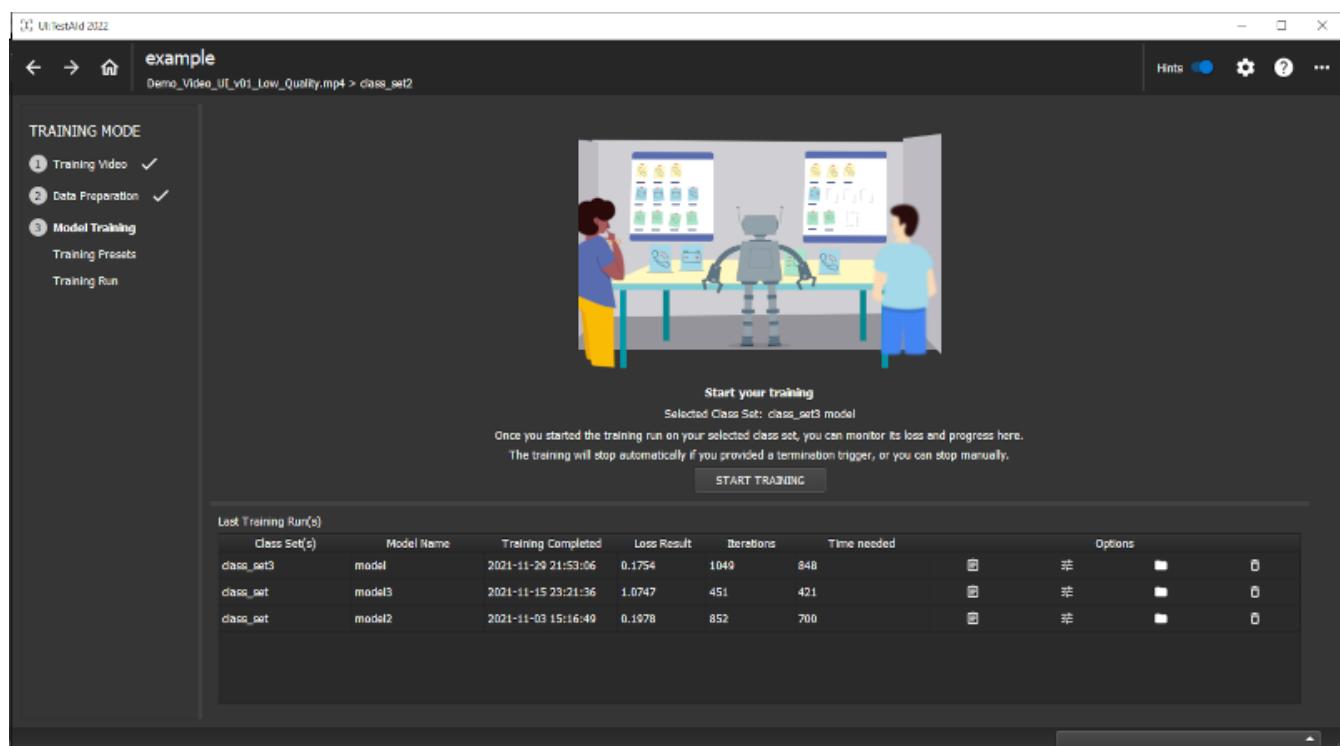
This setting is only active when you train the model with validation and changable during the training. It defines a trigger step number. As we know, the mean average precision (mAP) provided by validation will first increase due to better-trained model and decrease later because of overfitting. However, the randomness of training could also cause short-term fluctuation. The validation-based termination trigger accepts a temporary fluctuation (shorter than the trigger step) and terminates the training after a persistent descending.

- **Calculate mAP**

If you do not want to run the validation, however, still want to check the mean average precision (mAP), you can choose to activate the option. AI-Core will regularly calculate the mAP based on the training dataset. The difference between it and validation is that this function cannot detect overfitting.

## 4.6.2. Training Run

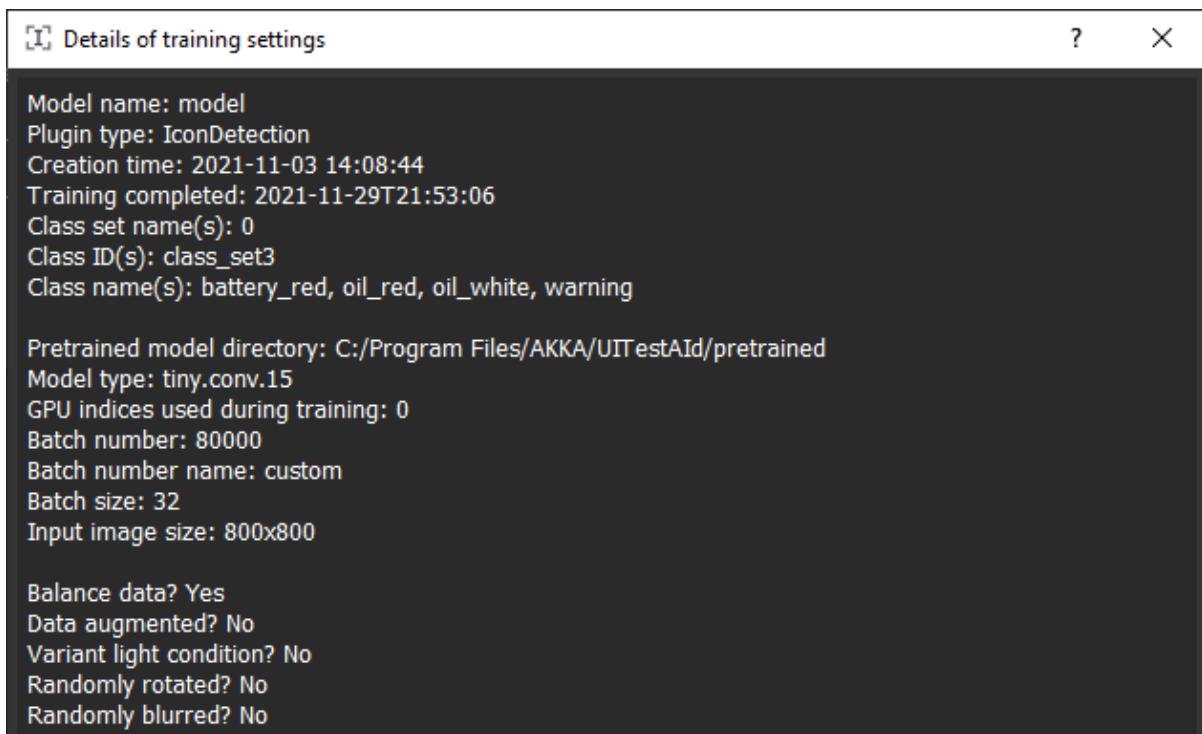
After you finish the configuration and make sure that one model has been selected, then click button "CONTINUE" to proceed to the "Start Your Training" page. On this page, you can already see the history details of your trained model, such as completion time of training, iteration number and so on.



The history list provides many functions, such as

- **Show details of training settings**

Every time when you start a training, all preset parameters will be saved with the model configuration files including the batch size, preprocessing settings. Click button  , the following dialog will pop up, in which all details of this training will be listed.



- **Recover the historical training settings**

Click  Then, AI-Core will jump back to the preset page and recover the selected settings. It allows you to adjust the training setting of one model, and train the model again.

- **Open Model Folder in File Explorer**

Click button  . AI-Core will open the model folder in file explorer.

- **Delete Model**

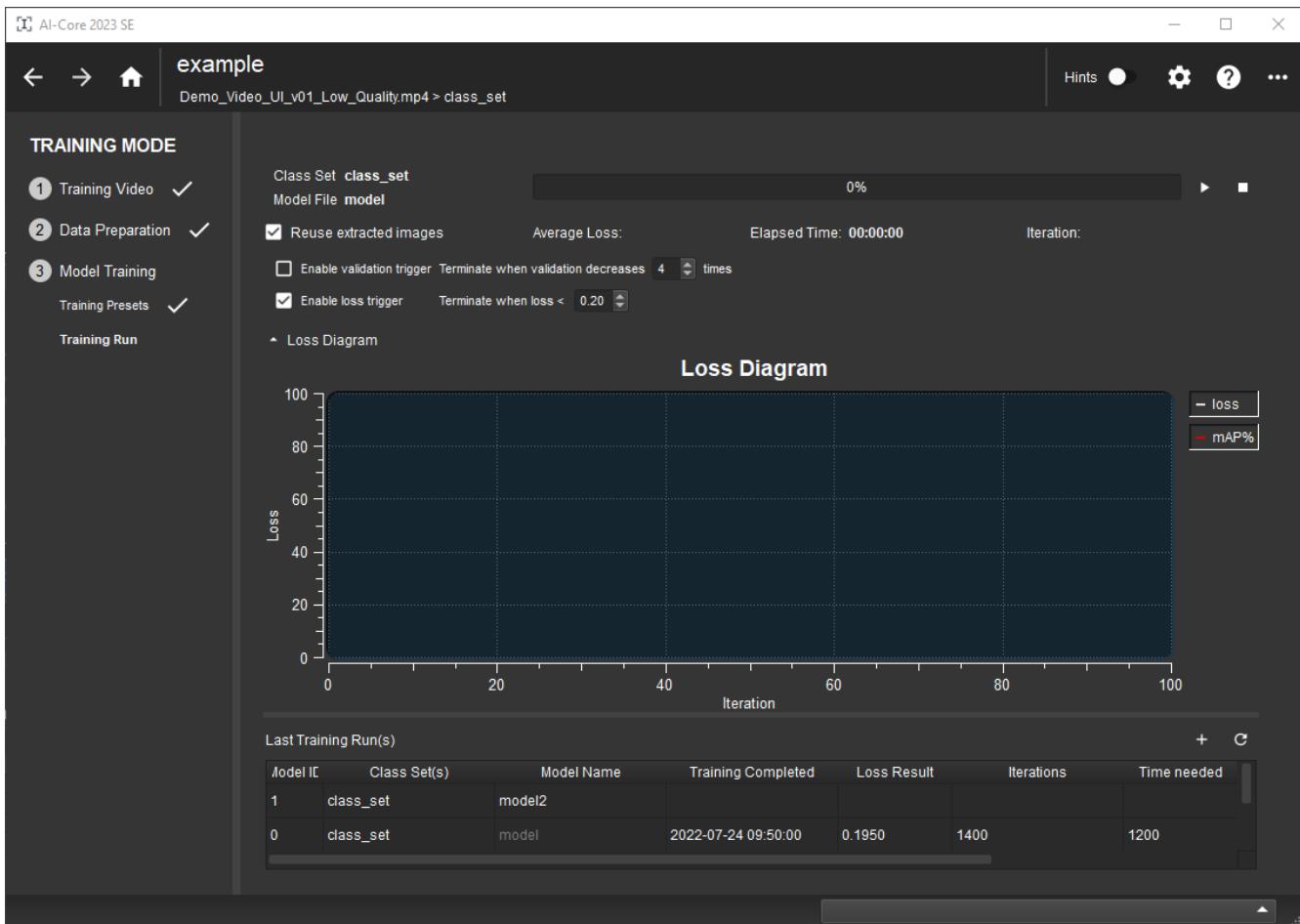
If you want to delete one historical model, you can click button  .

- **Reuse Extracted Images**

Before the real training, AI-Core will extract labeled frames to images and save them in **training\_project\_dir/temp**. By default, the previously extracted images will be removed first in order to prevent stale files, such as you have edited or changed the videos. However, if you are sure that there is no change in the training video, activating "Reuse extracted images" option

 **Reuse extracted images** can help you save time when you re-train the model next time.

Then, click button "START TRAINING" to the training dashboard.



Click the "Play" button , then you can launch the training.

The whole training process is divided to three steps: model configuration, extract frames, and training. The first two steps are preprocessing. Then, the training iterations will start. The loss diagram underneath will show the loss curve.



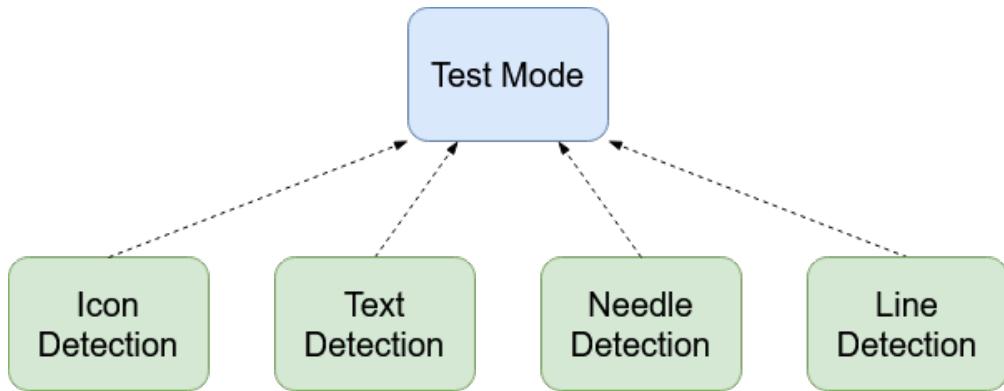
If you want to pause the training, click the "Pause" button . AI-Core will need some time to finish the task in hand and pause the training as soon as possible.

When the model is already well trained, users should terminate the training by clicking . If the model is trained too much, it could correspond too closely or exactly to the training data, and may therefore fail to predict future observations reliably [overfitting](#)). If you have no experience in the field of machine learning but there is advanced GPU (>6GB), it is recommended to use the validation-based termination trigger with step number 4.

When the training is accomplished, you can then switch AI-Core to test mode to check the result.

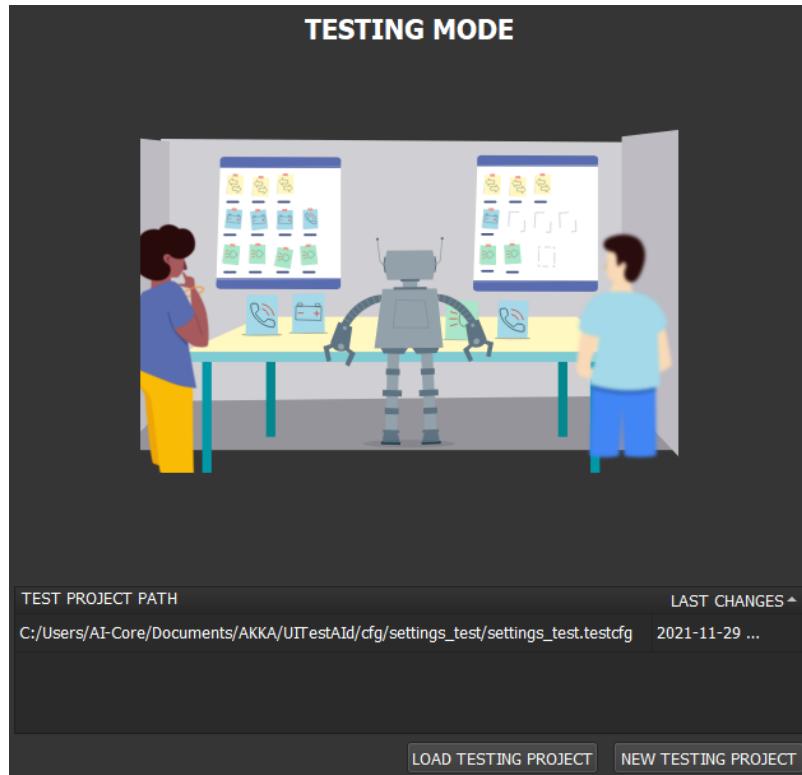
# Chapter 5. Test Mode

In test mode, users can conduct the testing for your own products or systems. AI-Core supports detecting various kinds of objects from online/offline visual data, such as icons, text, and so on. These detection functions are modularized into independent plugins. According to users' demand, you can load, unload, (de)activate each plugins individually. These plugins will work independently during the test. The details of plugin mechanism can be found in chapter [Plugin-Based Framework](#).



## 5.1. In General

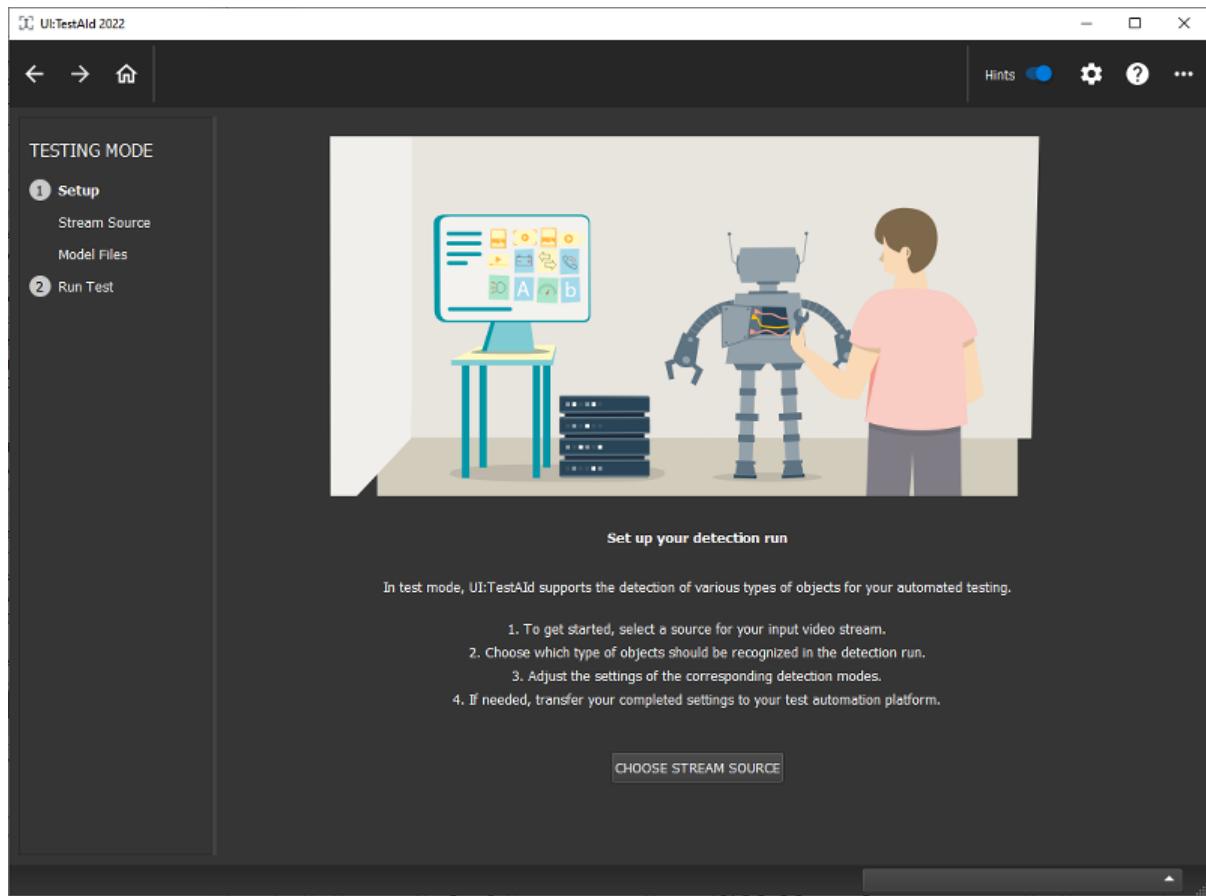
In this subsection we will show how the testing mode works. We will show all steps, and guide you with illustrations, in order to give an in-depth display of how the testing mode works, in the practical sense.



Here we have 2 buttons:

- Load Testing Project: load an already existing project
- New Testing Project: create a new project

After Creating>Loading the test project, you will be presented with the following page.



### 5.1.1. New Testing Project

When clicking on the "New Testing Project" we are presented with the following page. On the field "Project Name" we can choose which name we want to give to our new project. Below in "Project Directory Location", you can select the directory location by inserting the name, or by clicking on the icon on the right, selecting the appropriate one. The full path will then be displayed below. After this step, you can proceed by clicking "Create Project".

## New Test Project

To get started with your new project, name your project and choose a saving location.

### Project Details

Project Name  
example

Project Directory Location  
C:/Users/Public/Documents/AKKA/UITestAId

Project Configuration File Path: C:/Users/Public/Documents/AKKA/UITestAId/example

The path has already existed. Please change the project name or root directory

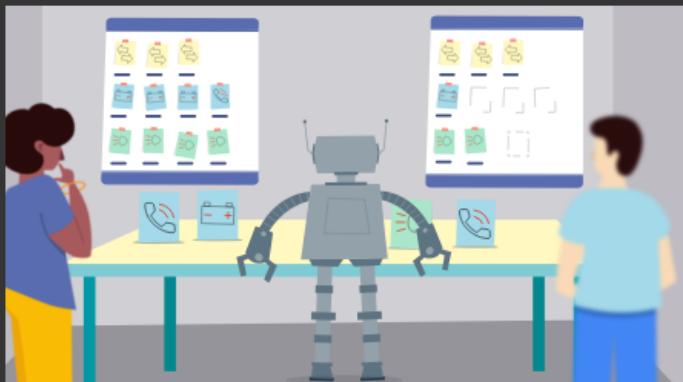
CANCEL

CREATE PROJECT

## 5.1.2. Load Testing Project

In case you want to load an existing project, you need to select the intended path, and press "Load Testing Project" or simply double click the intended path.

**TESTING MODE**



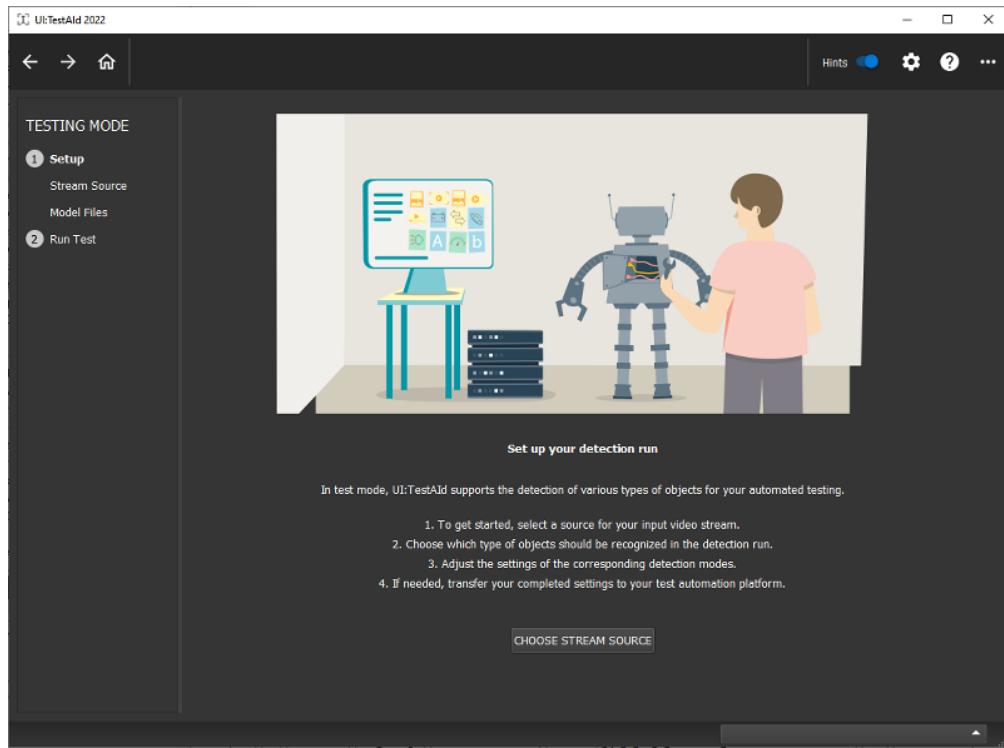
TEST PROJECT PATH LAST CHANGES ▾

C:/Users/AI-Core/Documents/AKKA/UITestAId/cfg/settings_test/settings_test.testcfg	2021-12-01 22:10:24
---	---------------------

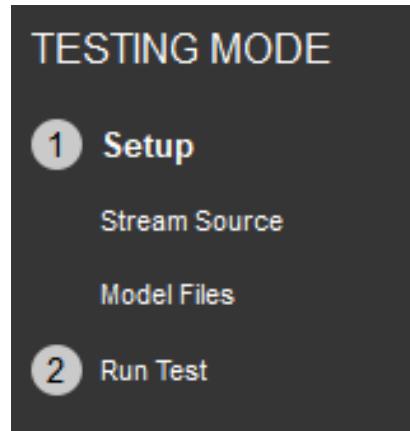
LOAD TESTING PROJECT NEW TESTING PROJECT

### 5.1.3. Testing Setup

After Creating/Loading the test project, you will be presented with the page presented in following figure:

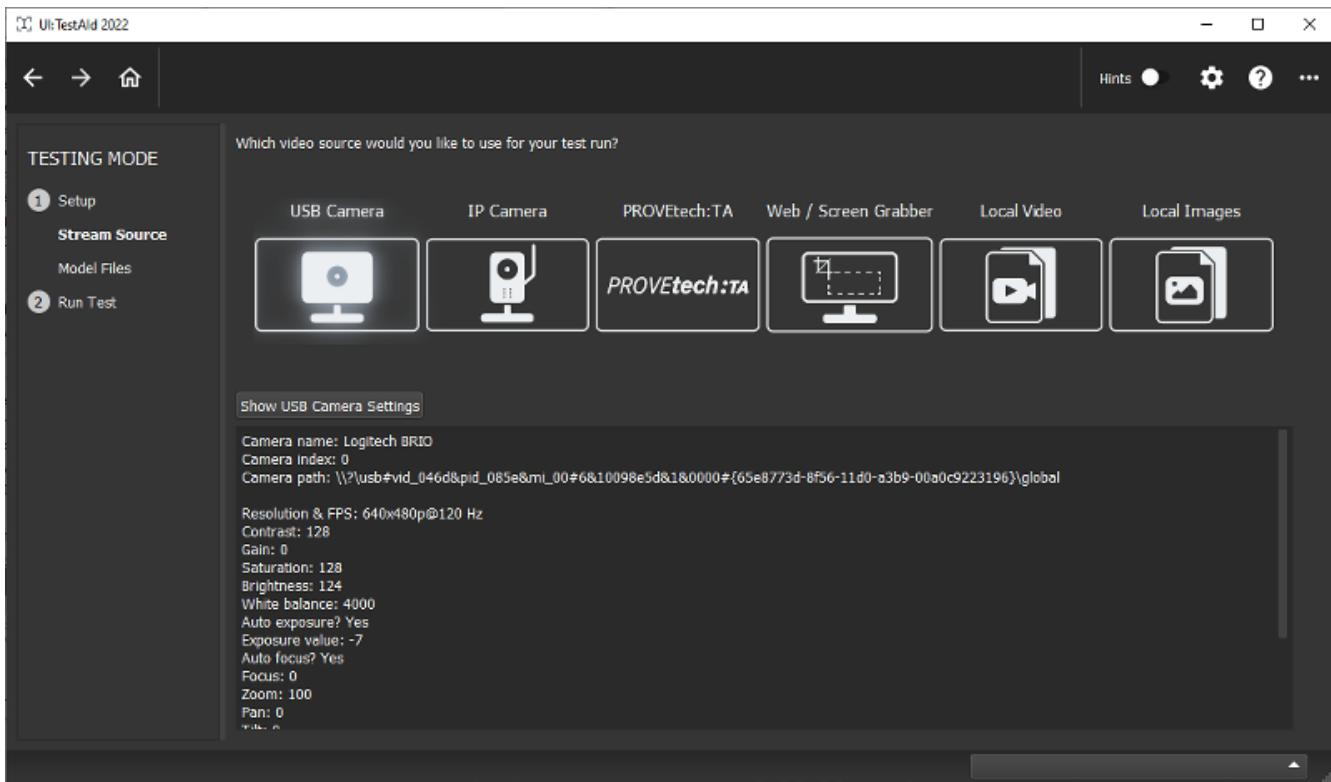


To proceed, simply click on the button "Choose Stream Source". Additionally, on the left side, there is a workflow bar, which gives you the possibility to skip steps, by clicking on the different texts. Thus, this is an alternative way to continue, and you can simply use it as a visual guide.



### 5.1.4. Select Stream Source

In the stream source selection menu, you can choose which type of source you want to test.



To choose the source, simply click twice on the desired one. If you click just once, they will present their information and current settings in the text box below. In case you want to change video source settings, please click on the "Show ... Settings" button.

The supported source streams are:

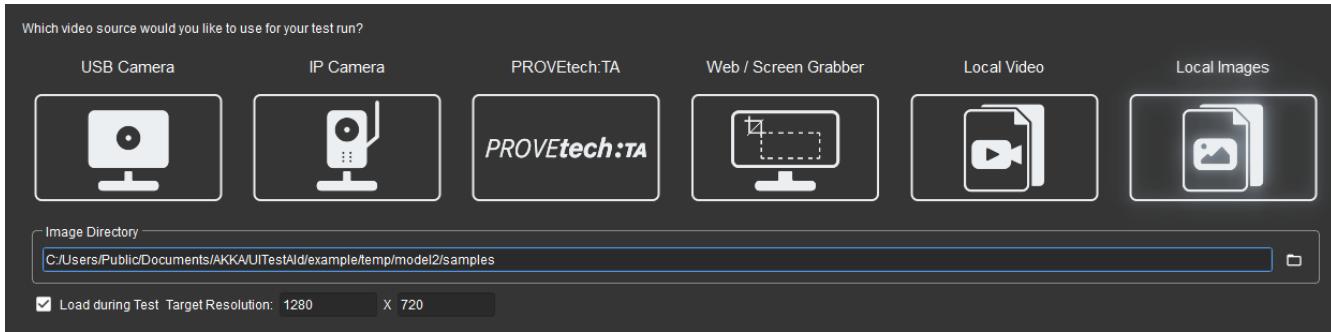
- USB Camera
- IP Camera
- PROVEtech:TA
- Web/Screen Grabber
- Local Video
- Local Images

### ***Local Video***

AI-Core uses FFmpeg as the backend for handling video. The supported video formats can be found in link. Usual video formats, such as mp4, avi, have all been When you specify the video path, AI-Core will memorize it and recover it during launching at the next time.

### ***Local Images***

Besides videos, AI-Core also supports detecting from images. In this case, you need to select a folder which contains the images needed to be detected.



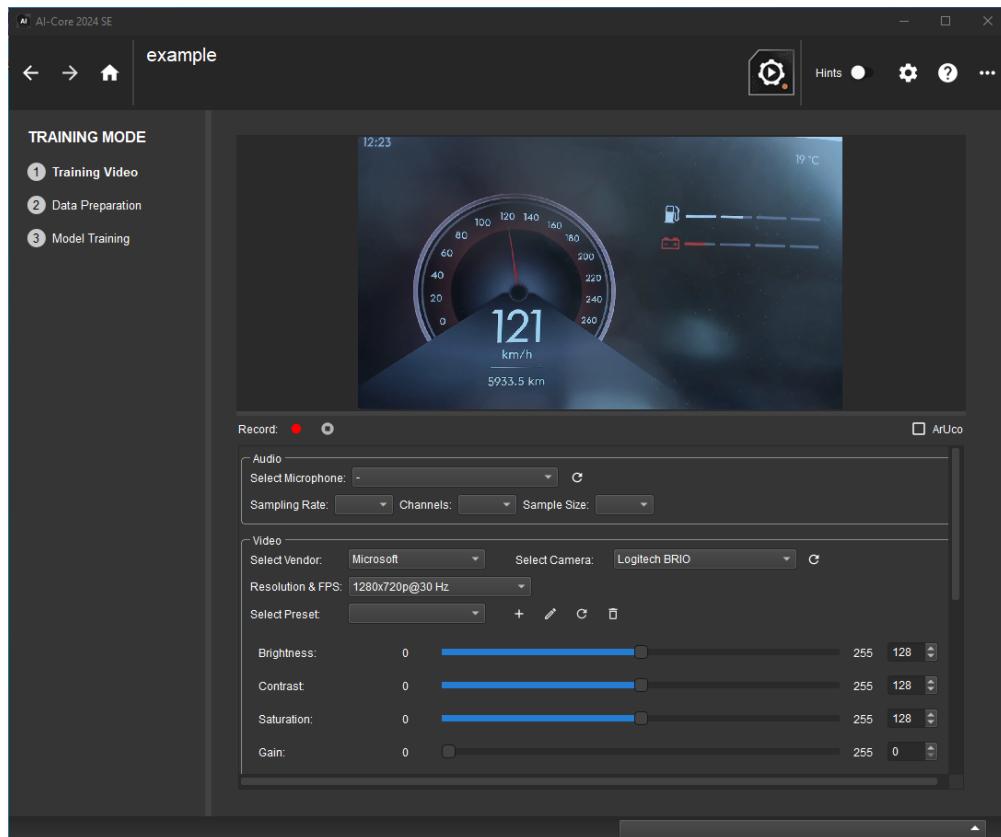
If you select "Load during Test", AI-Core will not load all images from the assigned folder at the beginning. Instead, it will just show a black screen. Then, you should call "LoadNewImage" command in your test automation tool, such as PROVEtech:TA to dynamically query specific images.

If you send a relative paths of images, AI-Core will assume that these images are located in the predefined image directory. In opposite, if you send absolute image paths, AI-Core will directly load images based on the path regardless of the predefined image directory.

In the mode of "Load during Test", you should predefine the target resolution as well, because the image directory could be empty at the beginning. Based on the target resolution, AI-Core can then reserve memory and deploy detection algorithms. If the images do not have the same resolution as predefined, they will be scaled.

### *USB Camera*

AI-Core supports all local USB cameras which support Media Foundation on Windows Platform. We can adjust the settings of the source, and observe the changes happening live on the video player above.



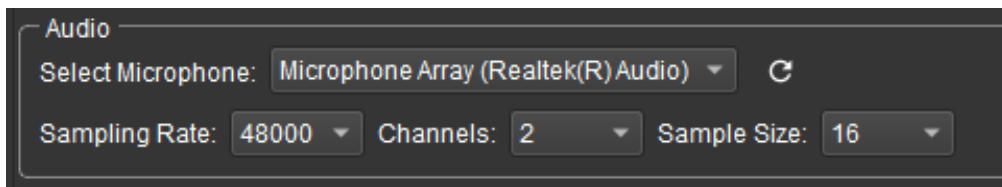
AI-Core reads the name of each device and their identification path. Every time when you load a

camera, AI-Core will record its name, identification path, and corresponding configuration parameters. If you change the parameters, the values will be automatically saved in the application configuration file.

At the next launch, AI-Core will try to load the same camera by its name and recover the configuration parameters.

### **Microphone**

If you want to test auditory stream, you also need to set up the microphone. You can choose the microphone by the name, the sampling rate, the number of channels, and the sample size.



- **Sampling rate:** This parameter gives the number of samples taken in one second. Some standard sampling rates are 16 kHz, 44.1 kHz, which is the format of an audio CD, and 48 kHz.
- **Channel number:** One channel for mono format and two channels for stereo sound.
- **Sample size:** The resolution of each audio signal sample and the number of bits of information in each sample. Some standard values are 16-bit integers or 32-bit floats.

### **Notice:**

- **Multiple camera scenario:** AI-Core finds the last loaded camera by its name. If there are two identical camera devices which share the same vendor and product model, then AI-Core will attach the first 5 characters of their serial code to the camera name in form of "Camera Name (xxxxx)" to identify them from each other. This will make sure that you can always load the correct hardware device.

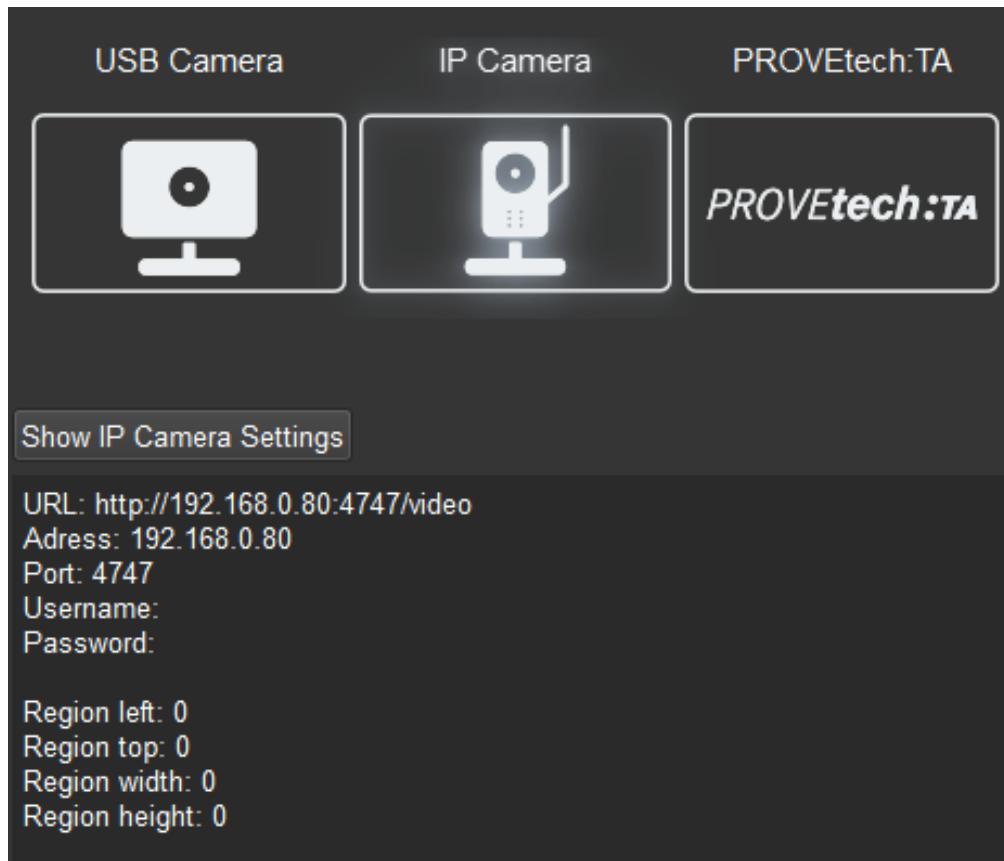
However, if you connect one camera and next time you change another device with the same model (there is always only one device connected to the computer), AI-Core will treat them as the same device and recover the last configuration including their camera name.

- **Actual frame rate:** The actual camera frame rate could be slower than the setting. It depends on many factors, such as hardware performance, video decoding, and so on.
- **Brightness vs Gain:** Brightness is a postprocessing parameter in the camera setting. It simply increases each pixel value linearly. When you choose a large brightness parameter value, the image will become white and lose the color information. Gain is a hardware parameter. It decides how much the camera should amplify the electrical signal from the camera sensor. Relatively, the gain can increase the brightness and meanwhile keep the color information better.
- **Relationship among exposure time, frame rate, and gain:** The exposure time, frame rate, and gain are correlated. When you select a large exposure time, the actual frame rate and brightness will decrease. You can select a higher gain value to increase the image brightness. However, large gain value will also increase the image noise at the same time.

### **IP Camera**

AI-Core is able to connect IP Camera under either HTTP or RTSP protocol. Through selecting "IP Camera" in the combo box under "Sources" configuration panel, you can switch the video source to

IP camera. You can see its configuration parameters in the panel.



First of all, You just need to define the camera address and port. If the connection needs authorization, you should additionally give the username and password. After you finish editing and press "enter" key, AI-Core will generate the final URL link used for connection. If your IP camera requires special edition of the url configuration, you can just change the link value alternatively. As other configurations, the parameters of IP camera are also saved in appcfg file. The password and URL are specially encrypted. However, please still not share appcfg file to other people.

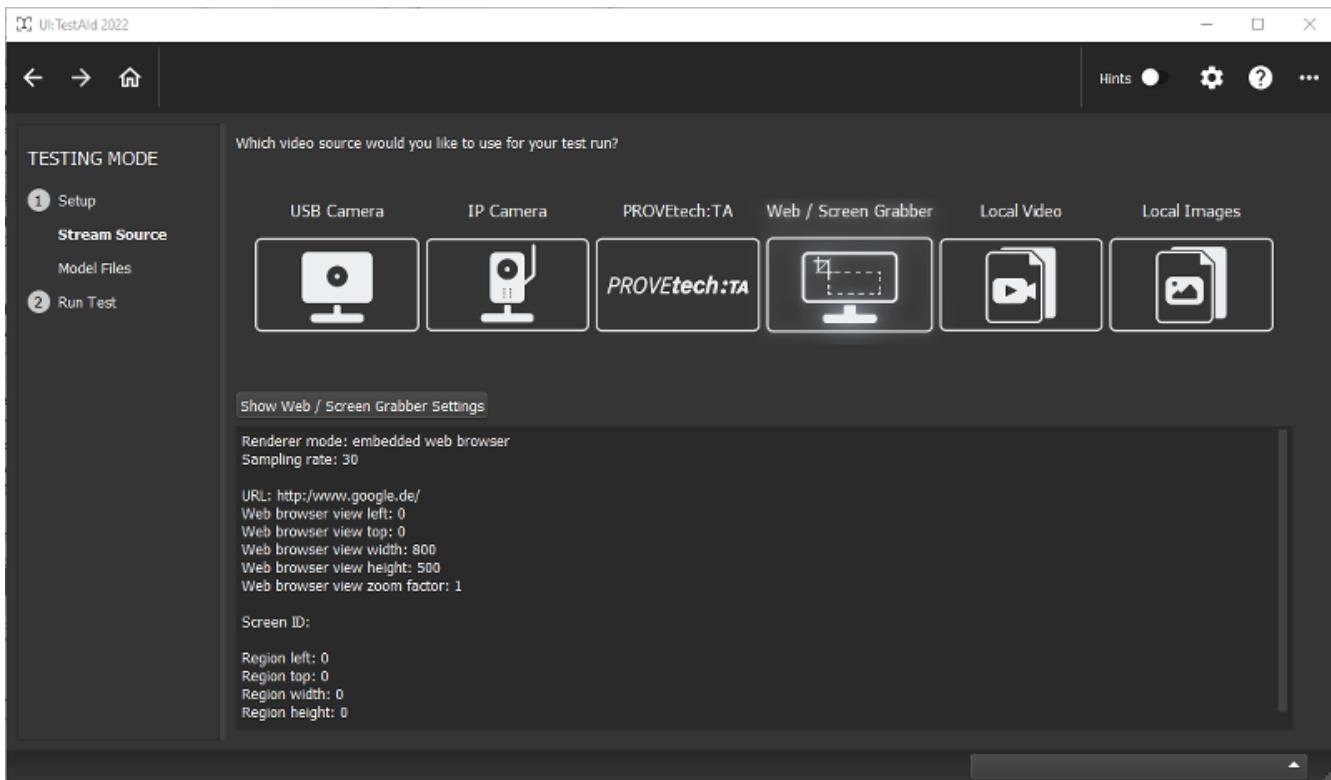
### ***PROVEtech:TA***

AI-Core supports video transfer with PROVEtech:TA. With the help of the seamless cooperation between AI-Core and PROVEtech:TA, it is allowed that the test video stream is captured and recorded in PROVEtech:TA, but meanwhile shared with AI-Core for object detection. The details of the connection building can be found in section [PROVEtech:TA](#)

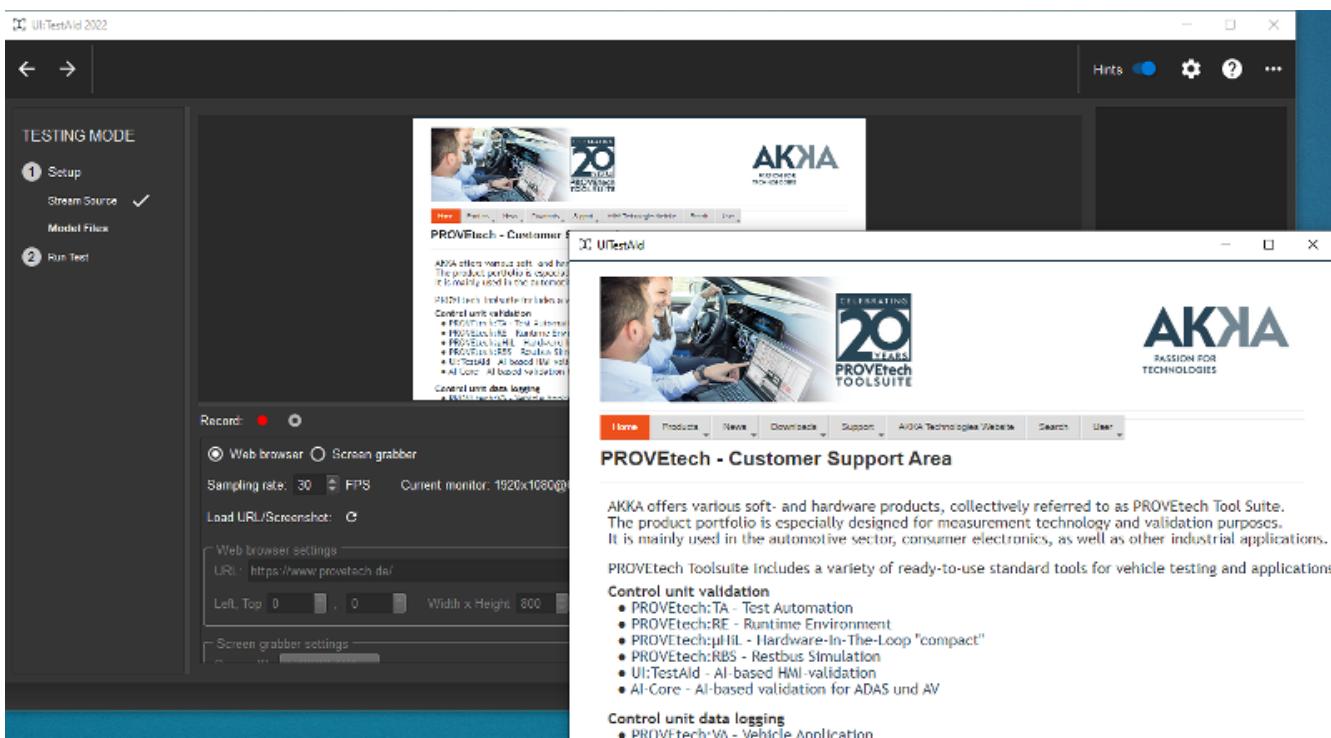
### ***Web Renderer***

From 2021 SE SP1, AI-Core newly supports web renderer and screenshot as video input for testing. It enables users to load webpages via URLs and conduct testing. Its principle is to render webpages in an embedded web browser and capture screenshots of the browser periodically. If you want to use a specific web browser, such as Firefox or Chrome, you can alternatively choose to capture the entire screen and select a region of interest as video input.

Through selecting "Web Renderer" in the combo box under "Sources" configuration panel, you can switch the video source to web renderer. You can see its configuration parameters in the panel.



Click "Configure" button, AI-Core will show the web renderer configuration page.



There are some configuration options:

- Capture mode:** as previously mentioned, you can switch the capturing mode between web browser and screen grabber. When you switch the mode during test, the loaded resources will be freed.
- Sampling rate:** it defines the suggesting frame rate in which AI-Core should capture the web browser/screen. However, the actual reached frame rate depends on many factors, such as system workload, monitor refreshing rate. Beside the combo box, AI-Core shows the

information of the monitor where the web browser is located in as well, including resolution and physical refreshing rate. Usually, you do not need to set a dramatically higher sampling rate than your monitor, because the contents cannot change faster than the monitor.

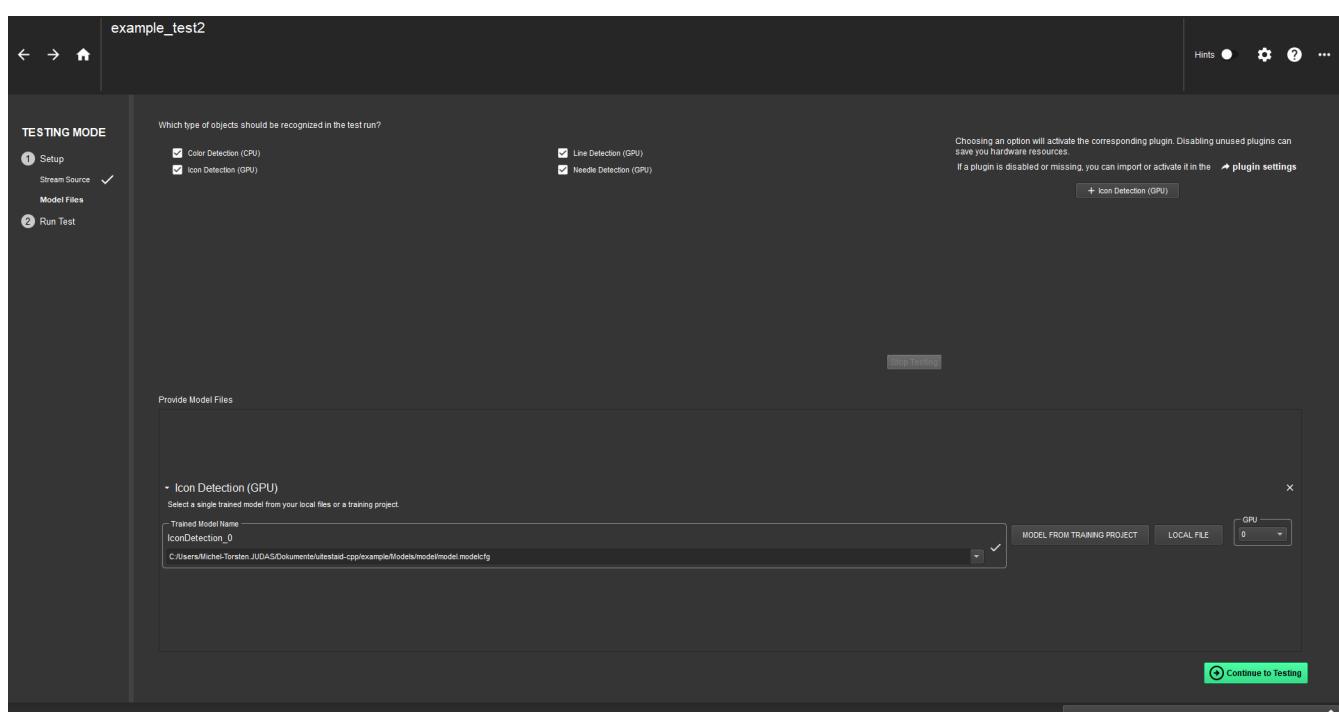
- **URL:** In the web browser settings, you should assign the URL which you want to load in the browser. It should start with "https://" or "file://". The browser does not automatically validate the link.
- **Left, Top, Width, Height, Zoom:** you might be only interested in one specific region of the webpage. When you scroll or zoom the browser, AI-Core will memorize these information and recover it when you load the page again.
- **Screen ID:** when you have multiple monitors, you can choose which one you would like to capture in screen grabber mode via the screen ID option.
- **Record/Stop:** Press the red record button, you can create a video. When you want to stop the recording, press the black stop button. After you configure all the parameters, you can press the green loading button. When you select the web browser mode, a browser window will pop up. You can adjust the region and make interaction with it.

### Notice:

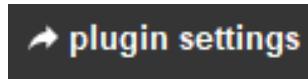
- In web browser mode, you should never minimize the browser, because when you minimize it, the browser will stop rendering the webpage internally. The captured frame will become a black-background image.
- In web browser mode, AI-Core usually can reach much higher actual sampling rate than in screen grabber.

## 5.1.5. Model Files

After selecting the source stream, you are presented with the following page:



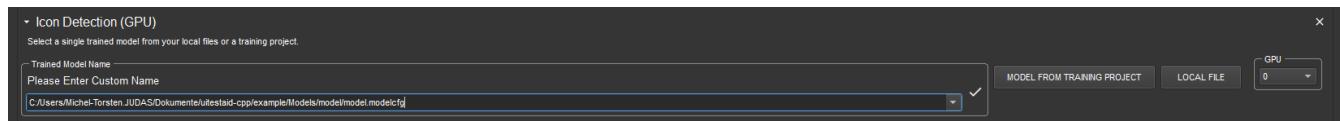
On the left side, all available test plugins will be listed. You can select or unselect to use in the testing. If you want to change the plugin settings, please click "plugin settings" button.



This will lead you to the test mode settings page, which we described in the subsection [Testing Mode Settings](#).

Below this, you will find several buttons that can be used to add multiple entries in the model selection part at the bottom of the page. These buttons will only appear if plugins are loaded that require an already trained model. 'IconDetection' would be such a plugin. In the case that one such plugin is loaded but appears deactivated on the left side, then the respective button will appear disabled as well.

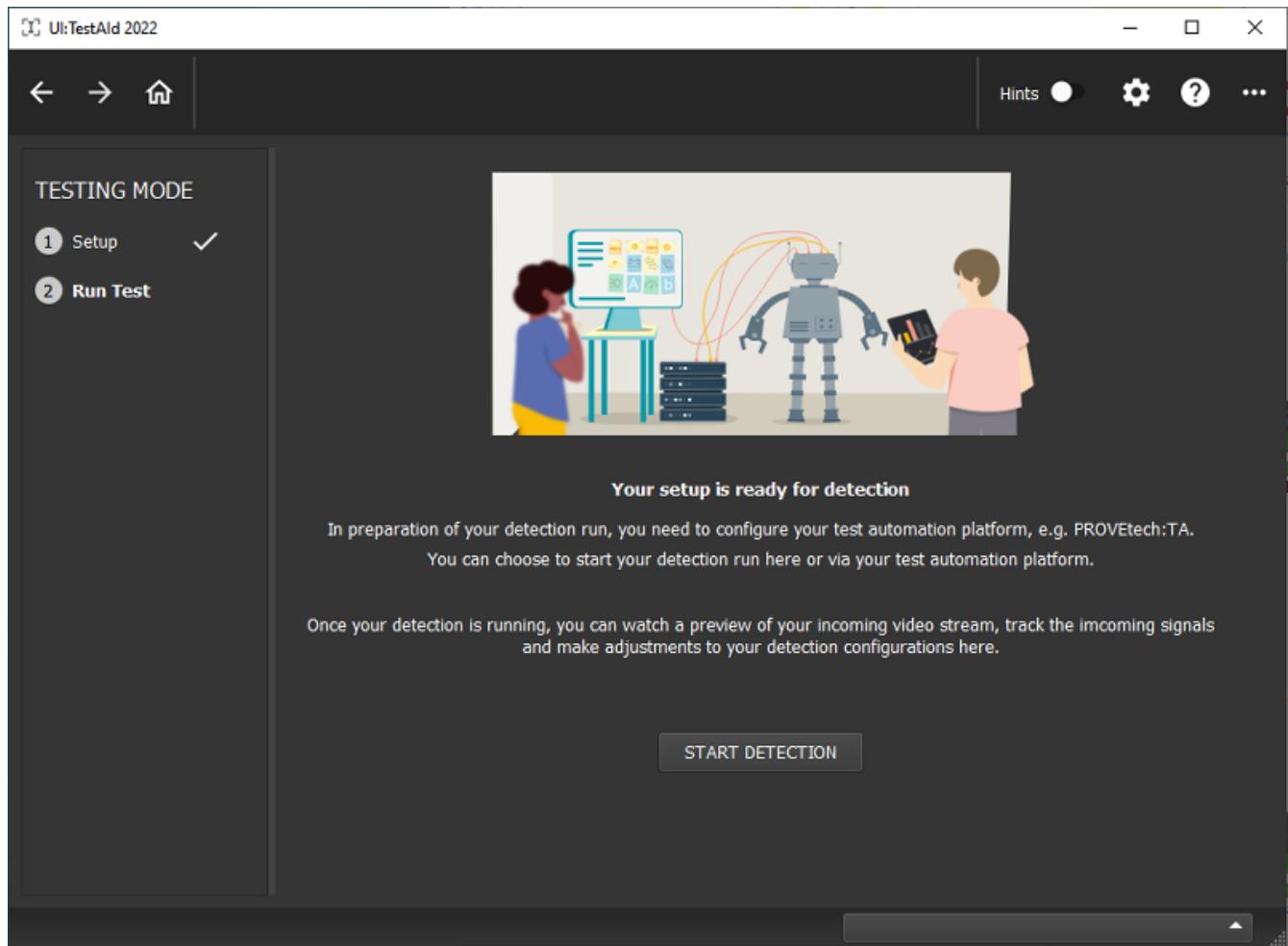
At the bottom of this page, you will find a list of multiple items like this:



In the case that only models that don't require a pre-trained model are loaded, then no items will be displayed, and none can be added. Within these items, you can specify which model should be loaded for the respective plugin. Additionally, you can give this item a custom name to distinguish between multiple models later on. Lastly, you can also select the desired GPU for the testing process if at least one is available.

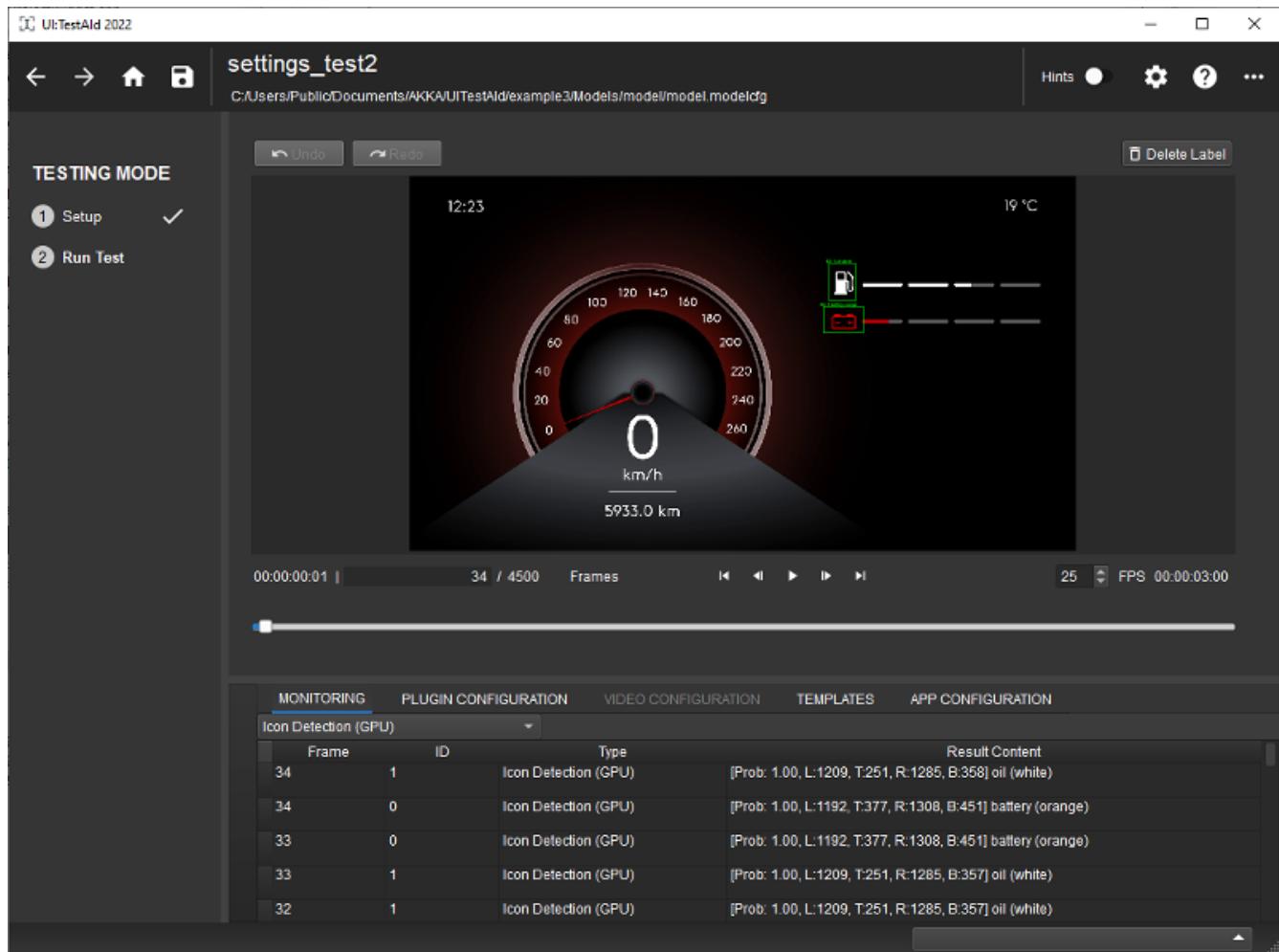


After clicking "Next", an informative page will be presented. Press on "Start Detection" to continue the testing.



### 5.1.6. Run Test

This page contains all the needed tools for testing. We will describe the different available tools.



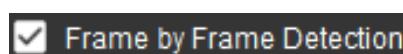
In the middle area, there is a video player which contains the selected source stream. You can easily interact with it. For that we provide the buttons shown in the following figure:



Where you have the following options:

- First Frame: jump to the first frame.
- Previous Frame: jump to one frame backwards.
- Pause/Play: play or pause the video (valid for both offline video or live video stream).
- Next frame: jump to the next frame.
- End frame: jump to the last frame.

Right next to the video control buttons, you will find a checkbox that activates the frame-by-frame detection mode.



This feature aims to provide reliable results for each frame of the provided video. You will notice that by activating this feature, the video runs slower than usual in the video player. This is because the program now waits for each plugin to finish its detection process before moving to the next frame.

Additionally, you are able to move the video control bar, and choose to which part of the video you want to go.



Below the video player, there is a widget containing 4 tabs: Test Results, Adjust Configuration, Templates, Video Settings.

### Test Results

The history of the last detections of the plugins will be listed in the test result table. This table contains 4 columns: frame, ID, type and result content.

- Frame: Video frame of the detected object.
- ID: ID number. You can find the corresponding box with the same ID in the video frame.
- Type: Type of detection plugin.
- Result Content: The concrete detection results. The contents vary for different plugins.

More details of result content can be found in the chapter of each plugin. You can also select which plugins they want to visualize, by clicking on the button above the table.

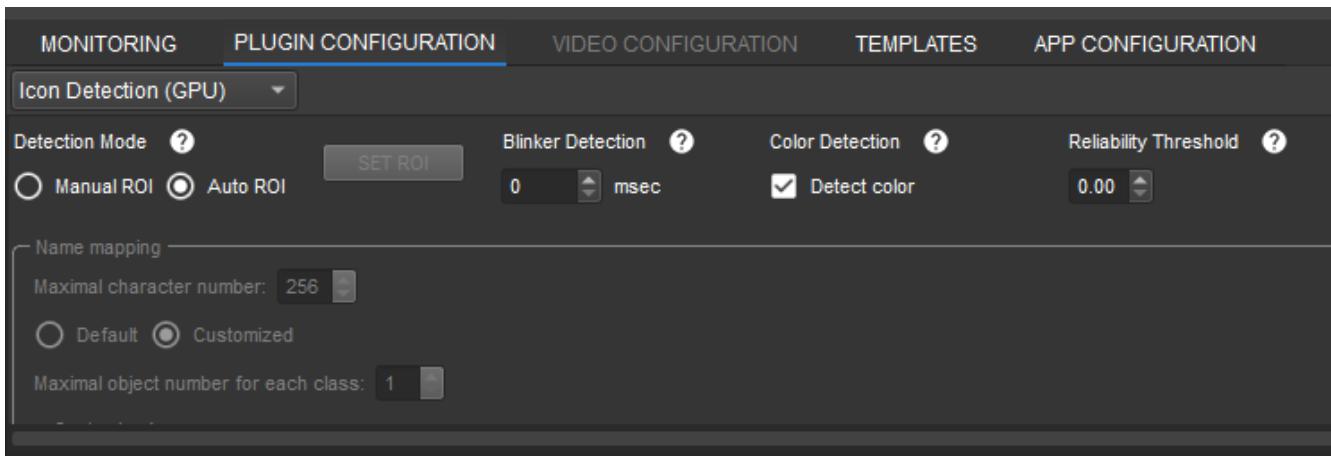
MONITORING	PLUGIN CONFIGURATION	VIDEO CONFIGURATION	TEMPLATES	APP CONFIGURATION
<input checked="" type="checkbox"/> Icon Detection (GPU)				
<input type="checkbox"/> Line Detection (GPU)	Type			Result Content
<input checked="" type="checkbox"/> Needle Detection (GPU)	detection (GPU)			[Prob: 1.00, L:1209, T:251, R:1285, B:358] oil (white)
<input checked="" type="checkbox"/> Text Detection (GPU)	detection (GPU)			[Prob: 1.00, L:1192, T:377, R:1308, B:451] battery (orange)
33	0	Icon Detection (GPU)		[Prob: 1.00, L:1192, T:377, R:1308, B:451] battery (orange)
33	1	Icon Detection (GPU)		[Prob: 1.00, L:1209, T:251, R:1285, B:357] oil (white)

Additionally, the detection results can all be saved into a JSON format. This is done by clicking on the save icon on the top-left corner, press Ctrl+S.



### Adjust Configuration

During the test, it is still allowed to change the plugin settings while testing. You can either adjust the settings directly through the "ADJUST CONFIGURATION" tab, or by clicking the settings button in the tool bar in the top-right corner.



Each plugin can be selected by clicking on the button on the top-left corner, like with the "Test Results" tab.

The plugins contain an option called "Set ROI", to set a manual region of interest. When clicked, the video is automatically paused, and you can draw the region, which determines the area to apply the detection. You can set as many boxes as wished. After adding ROIs the video will be paused, you need to click on the Play button to continue.

### **Templates**

The option "Templates" is divided in 2 parts: Templates and Setting Backups. The "Templates" side (on the left) has the functionality of allowing you to save and load ROI templates. To avoid repeatedly manual drawing operations, one template saves the bounding boxes drawn over the video for detection. Next time, if you want to test with the same template, you are allowed to load saved templates with manual loading or sending commands from test bench.

TEMPLATES				
Template Name	Relevant Plugins	Creation Time	Last Used Time	
example3	Icon,Text	2021-12-12 22:57:44	2021-12-15 09:32:44	
template2	Icon	2021-12-12 22:33:31	2021-12-13 09:28:03	
template1	Icon,Line,Needle,Text	2021-12-12 22:30:43	2021-12-14 23:37:41	

To manually load a template, just double click the wished one in the template list. You can also save a new one, overwrite an existing one, and delete. These three functionalities can be performed with the 3 icons on the top-right corner of the "Templates" box.

### **Setting Backups**

This functionality is used to save all settings of the current project including:

- Application preference such as language, dark/light mode.
- History of loaded training and test project.
- Video source settings such as camera parameters, IP camera address and so on. Plugin settings.
- Configurations of communication with other test benches.

If you click the save button, AI-Core will create a snapshot of the current application configuration file and copy to the test project folder. In the future, the copy can be loaded with double-clicking in

the table or sending commands.

MONITORING	PLUGIN CONFIGURATION	VIDEO CONFIGURATION	TEMPLATES	APP CONFIGURATION
Default	Application Configuration Name		Creation Time	Last Used Time
<input type="checkbox"/>	video_setting2		2021-12-05 09:02:33	2021-12-16 11:32:57
<input type="checkbox"/>	video_setting		2021-12-10 09:58:46	2021-12-16 11:32:55
<input type="checkbox"/>	provetech_setting		2021-12-13 13:06:48	2021-12-16 11:23:00
<input checked="" type="checkbox"/>	camera_setting		2021-11-26 10:32:32	2021-12-16 11:23:04

### Video Settings

If you are using live video stream such as USB camera, the video settings tab will be activated. It allows you to adjust the video settings while testing. Each source has its unique settings.

MONITORING	PLUGIN CONFIGURATION	VIDEO CONFIGURATION	TEMPLATES	APP CONFIGURATION
Select Camera:	Logitech BRIO	C		
Resolution & FPS:	640x480p@120 Hz			
Brightness:	0		255	124
Contrast:	0		255	128
Saturation:	0		255	128
Gain:	0		255	0

## 5.2. HDF5

### 5.2.1. Overview of HDF5

HDF5 (Hierarchical Data Format version 5) is a versatile data model and file format designed to efficiently store and organize large amounts of data, making it widely used in fields such as scientific computing, engineering, and data analysis. It features a hierarchical structure that allows data to be organized in a tree-like format with groups and datasets, supporting a wide range of data types, including integers, floats, and complex structures. HDF5 is scalable, capable of handling datasets that may not fit into memory, and is portable across different platforms, ensuring compatibility when sharing data. It also offers compression to reduce file sizes, chunking for improved I/O performance, and supports concurrent access for multiple processes. With components like groups, datasets, and attributes for metadata, HDF5 is commonly utilized in scientific research, machine learning, geospatial data management, and engineering applications, and is supported by a rich ecosystem of libraries and tools in various programming languages, including C/C++, Python, MATLAB, and R.

### 5.2.2. AI-Core HDF5 Implementation

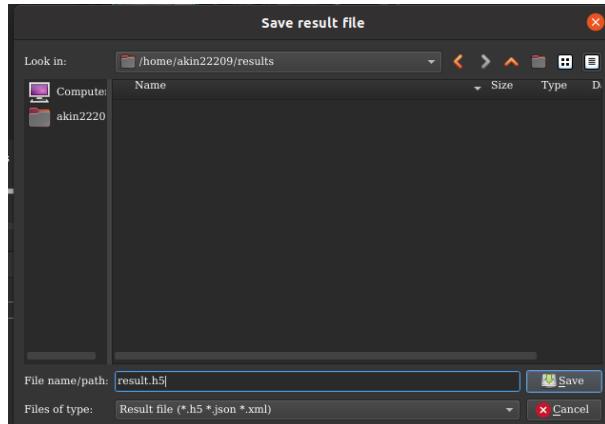
AI-Core is a visual analysis tool specifically designed to extract information as result from raw video streams, utilizing multiple plugins to handle complex data. To ensure organized storage of the extracted results, HDF5 is implemented as an effective solution. HDF5 not only accommodates a variety of data types but also supports the storage of detected image results. Currently, this functionality is available for two specific plugins: PixelTrace and the ONNX module, while detected text data can be saved from the other plugins. The history of extracted results is maintained in a temporary HDF5 file, which is created once the in-memory records exceed 5,000 rows. Users can view the most recent 50 rows of extracted test results in the monitoring tab. On Windows systems, this temporary file is stored in the directory specified by the TEMP or TMP environment variable (typically C:\Users<Username>\AppData\Local\Temp), while on Linux systems, it is located in the /tmp directory. This structured approach ensures that all extracted results are efficiently stored and readily accessible for further analysis.

### 5.2.3. Save Extracted Text Information

To save the extracted information displayed in the "Monitoring" tab of AI-Core, click the save icon located in the top left corner.



This will open the "Save Result File" window, where you can browse to the desired path, provide a name for the file, and select the ".h5" file format (e.g., results.h5).



## 5.2.4. Save Image Results from PixelTrace and ONNX module

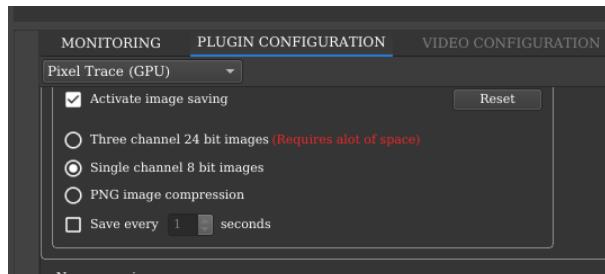
For the PixelTrace and ONNX plugins, along with extracted text data and image data can be saved in the resulting HDF5 file in three formats:

1. Three-channel 24-bit images
2. Single-channel 8-bit images
3. PNG image compression

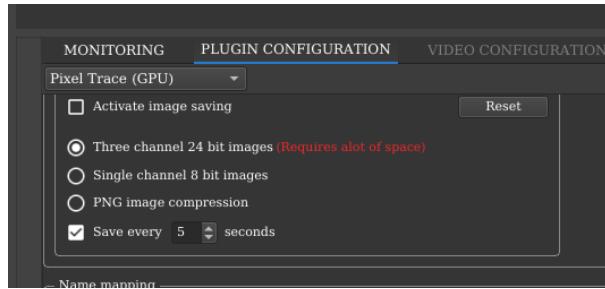
A three-channel image, often referred to as a 24-bit image, is a type of digital image that uses three color channels to represent color information. Each channel corresponds to one of the primary colors of light: red, green, and blue (RGB). A single-channel image, often referred to as a gray scale image or an 8-bit image, is a type of digital image that contains only one channel to represent intensity or brightness. Where as "PNG image compression" refers to a method of reducing the file size of an image while preserving its quality. It uses lossless compression techniques to encode the image data, allowing for efficient storage, transmission, or embedding in text-based formats without converting the data into a textual representation.

Please note that selecting the 24-bit format will result in significantly larger file sizes, consuming more disk space. To manage this, AI-Core offers a feature that allows you to save detected image data at specified intervals (every X seconds). This functionality helps optimize storage by capturing only the relevant image data over time. To save images, please select one of the three available formats and choose an option for saving: either "Activate image saving" or "Save every X seconds"

To use this feature click on "Plugin Configuration" tab of AI-Core, select desired plugin and browse for "HDF5 Result" option.



Above configuration option will save all resulting images in single channel 8 bit format.



Above configuration option will save resulting images after every 5 seconds interval in three channel 24 bit format.

### 5.2.5. Opening HDF5 file

To open the resulting HDF5 file, use the HDF5View application. This tool allows you to easily explore and visualize the contents of HDF5 files, enabling you to access the stored data and metadata efficiently.

## 5.3. Icon Detection

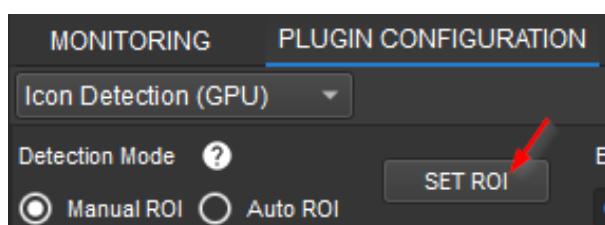
Icon detection is a plugin for real-time general visual object detection. It can detect the categories, positions, colors of an object in a frame. The highest detection speed can reach around 60 fps (with GPU acceleration). Typically, you can use this plugin to detect icons, LED indicators, even persons or vehicles from video streams.

Since the objects which need to be tested are highly customized, such as unique icons in newly-developed UI, it is hard to solve such general detection tasks with one single pretrained model. Therefore, icon detection needs you train your own models first, and then conducts the detection. In the previous chapters, the training has been introduced. Now, it is assumed that you already have a trained model. Next, it will show how to use this model.

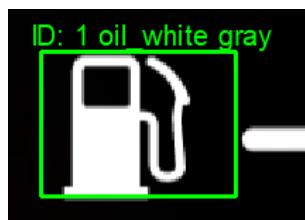
Icon detection reads and analyze frames in form of images. The results are visualized to green rectangular bounding boxes with an ID number in the workspace and also the textual details in the result board. The ID number is used to help you distinguish boxes from each other within a frame. Please notice that icon detection cannot track objects, which means one object could be assigned by different ID numbers in two frames.

There are two detection modes in icon detection. In auto ROI mode, AI-Core automatically detects objects and their bounding box positions. In manual ROI mode, you need to draw the regions of interested manually. AI-Core will only check these regions. If nothing found in one region, the region will be labeled as "NotFound". Otherwise, it will be labeled with the detected class name.

To draw the manual icon detection ROI, you should first click "SET ROI" in the settings panel of icon detection.

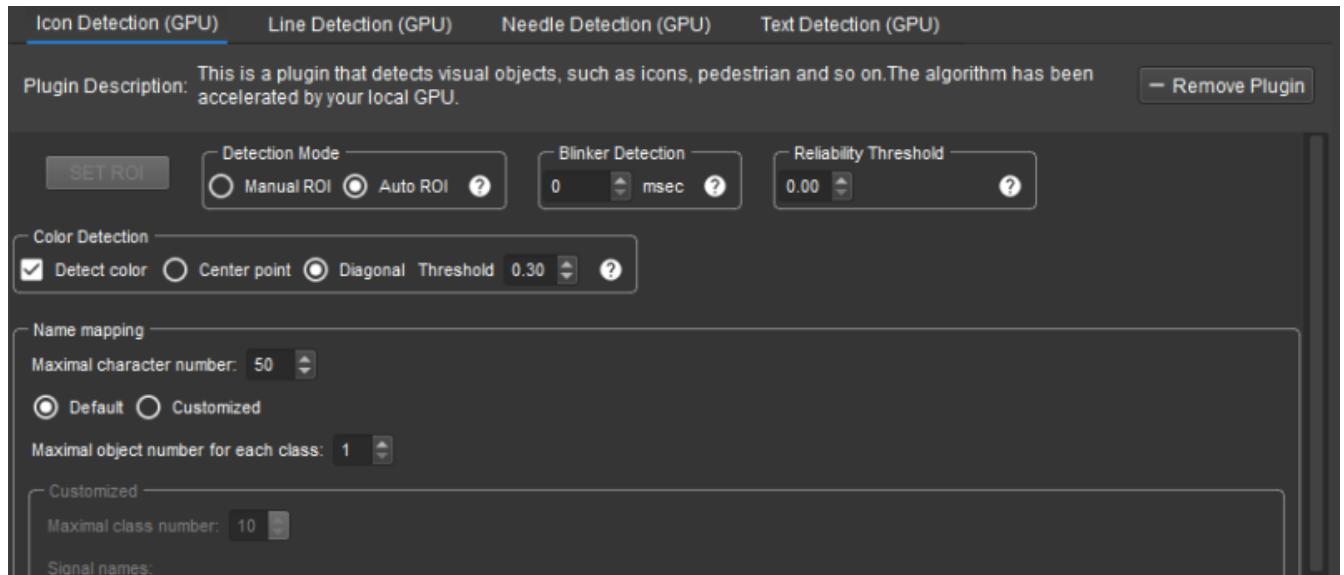


Then, you can draw a rectangular region surrounding the interesting object.

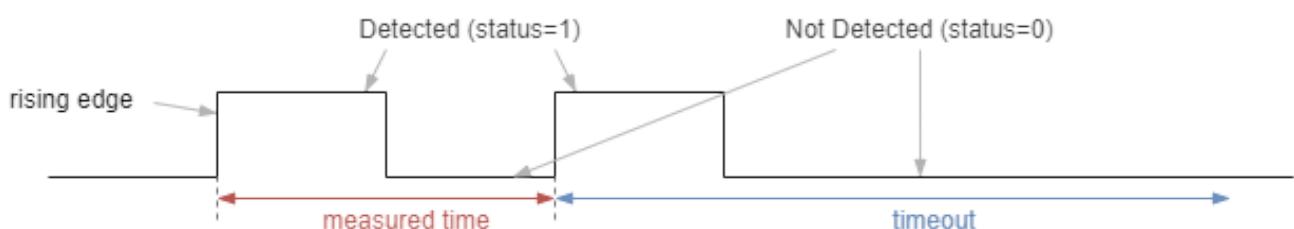


### 5.3.1. Icon Detection Settings

You have control over multiple settings for icon detection.



- **Auto/Manual ROI:** Switch between auto and manual icon Detection mode.
- **Reliability threshold (0.00 ~ 1.00):** When AI-Core is connected to PROVEtech:TA, you will see that there is an integer detection flag (0: not found, 1:found, 2:unreliable) Reliability threshold defines the detection reliability and controls the detection flag.  
For example, if the threshold is set to 0.5, when the probability is 40% (smaller than 0.5), the detection will be treated as unreliable. The detection flag will be set to 2.
- **Blinker timeout (msec):** In many test case, the occurrence frequency of some blinking objects, such as LED indicator, is very important and needed to be detected. In manual mode, AI-Core can also detect the blinking frequency. Blinker timeout defines the longest waiting time for the calculation of blinking frequency. If the region does not change for such a long time, the blinking frequency will be set to 0.



- **Detect color:** This option controls whether AI-Core should detect icon color. AI-Core detects color in HSV color space like the color detection plugin. For more details please refer to [color](#)

detection.

- **Name mapping:** The name map defines how the detection is shown in the signal tree in PROVEtech:TA. After connected with PROVEtech:TA, the signal tree will be unchangeable. Therefore, you need to predefine the signal name mapping before the connection.
- **Maximal object number:** defines the maximal number of objects from each class which can be shown in PROVEtech:TA. If AI-Core detects more objects from the same class than the maximum, the detection result will be discarded. If you want to customize your own signal names, you can define them in this configuration group. Especially, when you want to switch among different class sets or projects, you need to set the name map to "Customized", because the class names will be changed. The old signal names cannot be reused, which will lead connection failure between AI-Core and PROVEtech:TA.

From PROVEtech:TA, you can send commands to change the configurations of icon detection plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.3.2. Test Result Schema

After AI-Core detects the icons, the results will be shown in the test result table. In the following, all concrete components of detection results are listed:

- Prob: The probability to be the detected class (0.0 ~ 1.0). The highest is 1.0, which means the detected object is 100% similar to the class in the class set. If the value is 0.6, it means the object has 60% similarity to the labeled class, but the light condition or video quality might not be perfect.
- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.
- Content: The detected class name will be shown behind the coordinations. In the manual mode, if there is no object detected in the drawn ROI, it will show "Not Found".
- Color: In the brackets at the end, AI-Core will show the detected color, if you activated the color detection.

Following results are only available in the test automation tool:

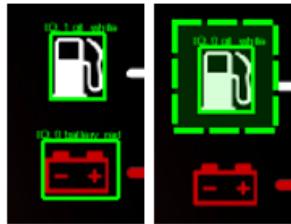
- ColorId: The ID of detected color. 0,1,2 are white, gray, black respectively. Depending on the color definition, the ID increases with the color number. See the details from the following [color definition table](#).
- ColorHue, ColorSaturation, ColorValue: The hsv component values of the detected color.

### 5.3.3. Detection Filter

On the top right side, you will find the following button:



By pressing this button, you are able to draw additional Regions of Interest (ROIs), which specify in which regions of the image you want the detections to happen. The following image shows the effect of these new regions of interest:



You can see that in the right image, the additional region of interest suppresses the detection of the red icon, which is useful for cases in which you are only interested in the values of certain icons.

[.notice] Notice: The regions of interest can be drawn at any time, but they only affect the icon detections while in Automatic Mode. Rectangular regions drawn in Manual Mode are not affected by the regions of interest.

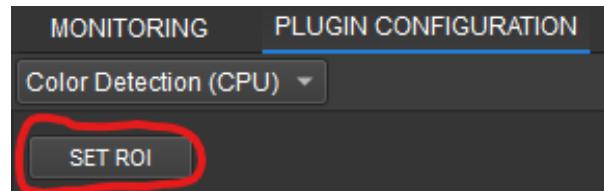
## 5.4. Color Detection

Color detection is a plugin for real-time color detection. It can detect colors of objects inside a frame.

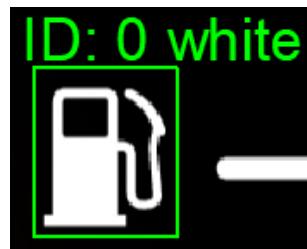
Color detection reads and analyse frames in form of images. The results are visualized to green rectangular bounding boxes with an ID number in the workspace and also the textual details in the result board. The ID number is used to help you distinguish boxes from each other within a frame.

You need to draw the regions of interested manually. AI-Core will check these regions and label them with the detected color.

To draw the color detection ROI, you should first click "SET ROI" in the settings panel of color detection.

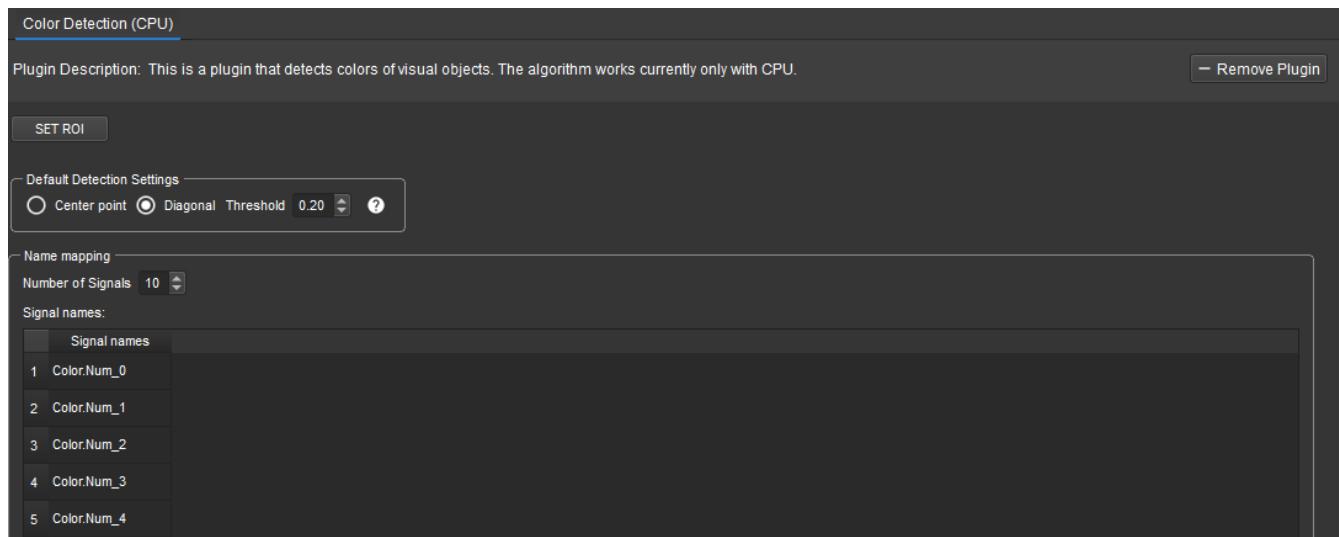


Then, you can draw a rectangular region surrounding the interesting object.



### 5.4.1. Color Detection Settings

You have the control over multiple settings for color detection.



- **Color Extraction Mode** AI-Core detects object color in two extraction modes.

- **Center Point:**

extract the color value from the center point of each detected or drawn ROI.

- **Diagonal:**

extract the pixel values on a diagonal line over each image from the diagonal line from left-top to right-bottom corner. It separates two different colors out of these pixels by calculating the maximum difference between pixels. All pixels of the line are assigned to one of the colors foreground color or background color.

AI-Core ignores those pixels on the diagonal if their values are close to the background color. The color threshold (0~100%) defines the ignored range regarding to the background color. If the threshold is high, then more pixels will be ignored.

**Notice:** since some objects have colorful background but black texts such as the following image. We are interested more in the background color. Thus, in this case (text color is black), AI-Core will treat the background color as the detected color.



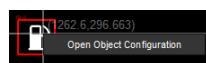
- **Contour:**

This method employs contour curves that connect all continuous points with the same color or intensity. It is utilized because it provides valuable information about the boundary between the background and the object. Additionally, this method is used to extract both the background and object colors. The background color is then used to disregard pixels with similar hues, further enhancing the accuracy of color detection.

- **Name mapping:** The name map defines how the detection is shown in the signal tree in PROVEtech:TA. After connected with PROVEtech:TA, the signal tree will be unchangeable. Therefore, you need to predefine the signal name mapping before the connection.
- **Maximal object number:** defines the maximal number of objects which can be shown in PROVEtech:TA. If AI-Core detects more objects than the maximum, the detection result will be discarded. If you want to customize your own signal names, you can define them in this configuration group. Especially, when you want to switch among different class sets or projects, you need to set the name map to "Customized", because the class names will be changed. The old signal names cannot be reused, which will lead connection failure between AI-Core and PROVEtech:TA.

From PROVEtech:TA, you can send commands to change the settings of color detection plugin remotely. The concrete commands are listed in [supported commands](#).

**Individual color settings:** If you right-click the region box, you will see a context menu. Select 'Open Object Configuration' from the context menu, and a dialog will pop up.

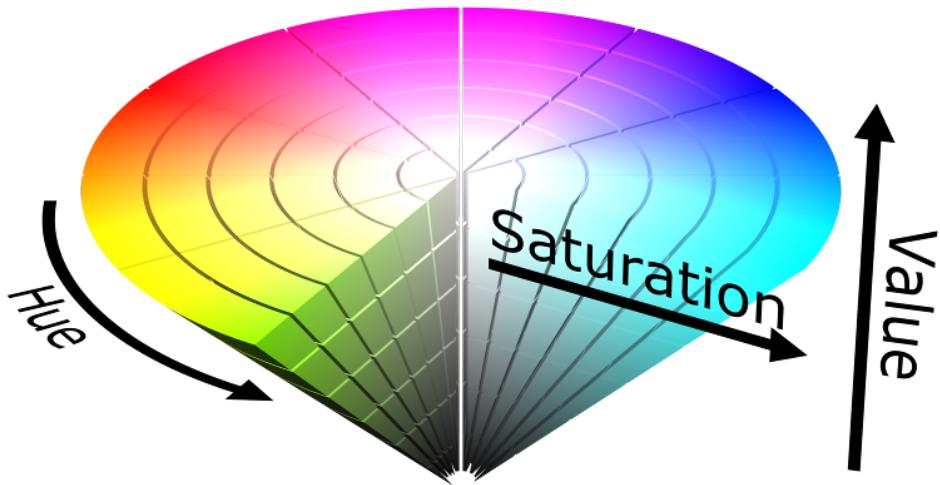


In the dialog, you can adjust the color detection settings for this region box. The settings will only apply to this region box. The settings will reset regularly if you leave the testing. If you don't want to reconfigure everything, you can save your settings by creating a template.



#### 5.4.2. Color Definition

AI-Core detects color in HSV color space.



HSV color space has three dimensions: hue, saturation, and value. Hue defines the appearance of color in value range of 0 ~ 360. Saturation defines the colorfulness of a color in value range of 0 ~ 255. Any color with saturation 0 will turn to gray. Value defines the brightness of a color in value range of 0 ~ 255.

AI-Core will classify any pixel value in 11 color with names: white, gray, black, red, orange, yellow, green, cyan, blue, violet, magenta. If you are using other test automation tools, such as PROVEtech:TA, you will find that AI-Core will send three signals of color: Each color has a string name, a unique integer ID, and the value of its three HSV channels.

If you are using other test automation tools, such as PROVEtech:TA, you will find that AI-Core will send three signals of color: Each color has a string name, a unique integer ID, and the value of its three HSV channels.

The concrete definition is listed below.

Color ID	Color Name	Value Range
0	white	saturation < 102 and value > 229
1	gray	saturation < 102 and 50 < value ≤ 229
2	black	value ≤ 50
3	red	(0 ≤ hue < 15 ) or hue > 310 (value ≥ 50 and saturation ≥ 102)
4	orange	15 ≤ hue < 40 (value ≥ 50 and saturation ≥ 102)
5	yellow	40 ≤ hue < 70 (value ≥ 50 and saturation ≥ 102)
6	green	70 ≤ hue < 160 (value ≥ 50 and saturation ≥ 102)

Color ID	Color Name	Value Range
7	cyan	160 ≤ hue < 200 (value ≥ 50 and saturation ≥ 102)
8	blue	200 ≤ hue < 266 (value ≥ 50 and saturation ≥ 102)
9	violet	266 ≤ hue < 290 (value ≥ 50 and saturation ≥ 102)
10	magenta	290 ≤ hue < 310 (value ≥ 50 and saturation ≥ 102)

Since AI-Core 2023 SE, you are allowed to define the color definition by yourself. In the appcfg, you can find following parameters under "Color" (or "Icon" for icon detection plugin) section:

```
blackThreshold=50
whiteThreshold=229
saturationThreshold=102
hueValueList=15, 40, 70, 160, 200, 266, 290, 310
hueNameList=red, orange, yellow, green, cyan, blue, violet, magenta
```

Black, white, gray are three obligate colors, which is defined by blackThreshold, whiteThreshold, and saturationThreshold. If a color does not match the rule described in the above table and corresponding thresholds, it will be classified as one of hueNameList by checking the hue range.

If the hueNameList is empty or its length does not match hueValueList, the color will be classified as gray by default. If the parameters do not work any more, reset color detection configuration, then you can recover the default definition.

### 5.4.3. Test Result Schema

After AI-Core detects the colors, the results will be shown in the test result table. In the following, all concrete components of detection results are listed:

- Color: In the brackets at the end, AI-Core will show the detected color, if you activated the color detection.
- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.

Following results are only available in the test automation tool:

- ColorId: The ID of detected color. 0,1,2 are white, gray, black respectively. Dependig on the color definition, the ID increases with the color number. See the details from the above color definition table.
- ColorHue, ColorSaturation, ColorValue: The hsv component values of the detected color.

## 5.5. Text Detection

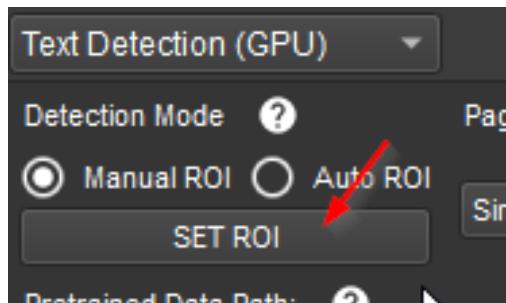
Text detection is a plugin which is aimed to detect texts from video streams. It supports multiple languages, such as Arabic, Chinese, English, French, and so on. As a prerequisite, you need to first install language packages in your system. Otherwise, some special characters could lead to false decoding.

Although the text detection is based on both tradition optical character recognition technology and neural networks, which requires supervised learning, a package of pretrained models will be installed with the plugin too. That means, you do not need to train the text model by yourselves, if the detected texts are printed in ordinary font styles, such as Arial, Calibri.

If you need to detect texts in custom font styles, such as handwritten fonts, you need to send the technical support a font library, supporters will generate a pretrained model based on the font library as soon as possible.

From 2021 SE, AI-Core supports both manual and auto mode of text detection. You can switch between them with the radio button in the configuration panel.

In manual mode, it requires you first to draw rectangular detection region. To create a detection region, you need to first click "Text Detection" button.



Second, draw a rectangular bounding box by pressing and dragging the left button. Then, click play "Play" button to check the detection result.

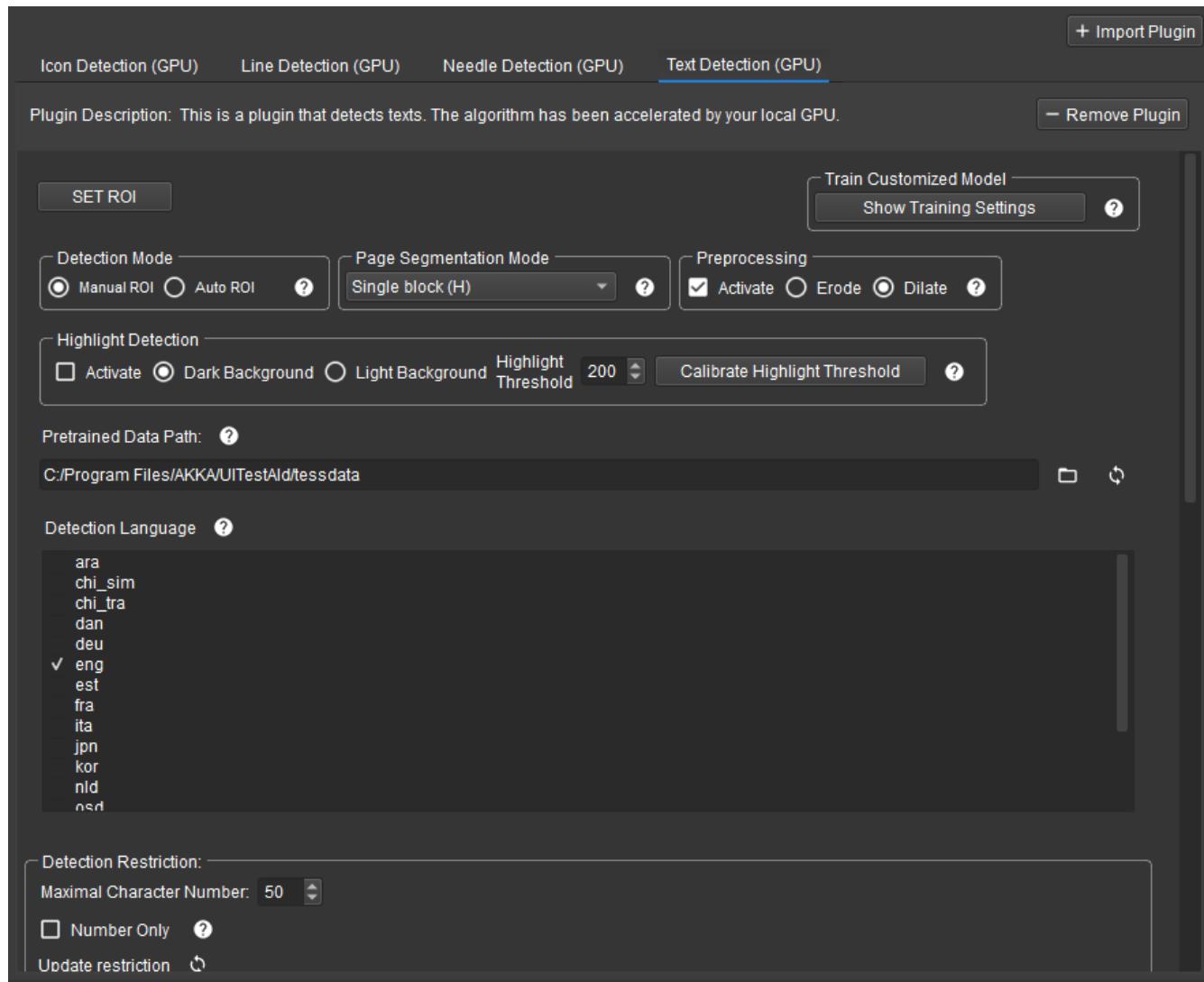


In auto mode, you do not need to draw the detection region box manually. AI-Core will detect all texts and detect the contents of them. In spite of its convenience, you should still be aware of its limitation:

- Currently it only supports English and Chinese. German or other languages from Latin family can also be detected, but some special characters could not be detected precisely, such as ä, ü, ö.
- The detection speed of text region is around 3~5 FPS, which will lead the delay of box position update. The text content recognition is not influenced by the region detection. Therefore, if the

text positions in your test case does not frequently change, the slow position detection will not have much effect on performance.

### 5.5.1. Text Detection Settings



The text detection has several parameters. Correctly selected parameters will increase the detection accuracy significantly.

- **Manual/Auto Mode:** it switches between manual and auto detection mode.
- **Pretrained data path:** This parameter defines the directory path of pretrained models. The directory should contain all model files with extension of ".traineddata". In default, the pretrained data directory (named by "tessdata") is located under the installation position of AI-Core, but you can also define its path according to requirements.
- **Detection language:** It specifies the languages needed to be detected. You can select maximal 5 languages at the same time.
- **Page segmentation mode:** The text alignments could vary a lot. The text to be detected could be an article, a line, or a single word. The text detector will first segment the input image (detection region) to characters, and then group the detection of each character to the final result. Thus, if there is assumption of the text alignment type, the segmentation result will be optimized.

Supported segmentation types are listed in the following:

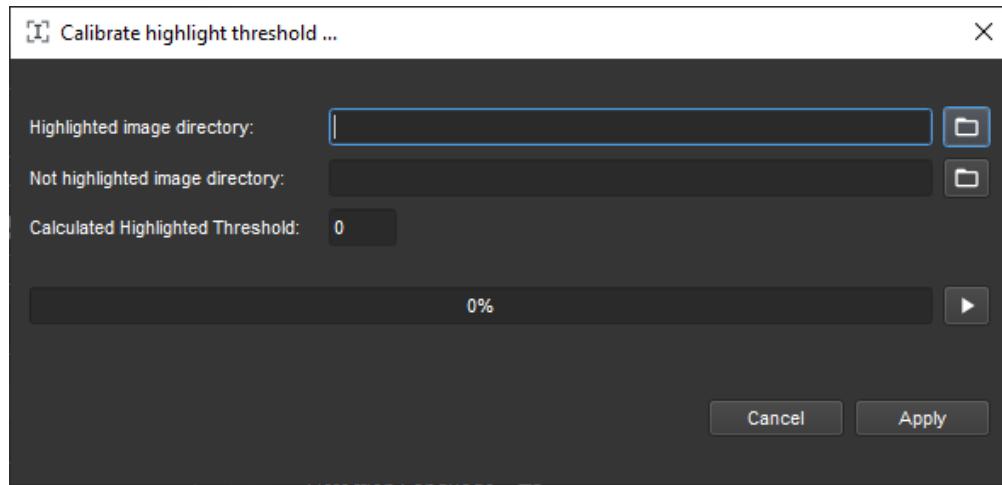
- Auto OSD: Automatic page segmentation with OSD and OCR.
- Auto only: Automatic page segmentation, but no OSD, or OCR.
- Auto: Fully automatic page segmentation, with OCR, but no OSD.
- Single column: Treat the image as a single column of text of variable sizes.
- Single block (V): Treat the image as a single uniform block of vertically aligned text.
- Single block (H): Treat the image as a single uniform block of text.
- Single line: Treat the image as a single text line.
- Single word: Treat the image as a single word.
- Circle word: Treat the image as a single word in a circle.
- Single char: Treat the image as a single character.
- Sparse text: Find as much text as possible in no particular order.
- Sparse text OSD: Sparse text with OSD

**Note:**

- Orientation and script detection (OSD) is an algorithm which detects the orientation and type of the text alignment.
  - Automatic page segmentation with OCR will optimize the segmentation based on the result of optical character recognition.
  - You might need try different options to achieve the best result.
- **Number only:** Select this option when the texts to be detected only contain numbers.
  - **Black list:** The black list defines the set of characters that should be ignored. After edition finished, click "Update restriction" button to update the black list.
  - **White list:** The white list defines a set of characters. The detected texts can only be composed of this set of characters. After edition finished, click "Update restriction" button to update the white list.
  - **Preprocessing:** In some cases, such as small font sizes, thin font styles, noisy background, the text detection does not work very well. The dilation and erosion enable you expand bright and dark texts. Meanwhiles, the background noise will be removed as well.
  - **Highlight detection:** when people select a component on a GUI component, the texts are often highlighted. Usually, the brightness of the font color will change. Based on this assumption, you can set a threshold (0~255) to filter these highlighted texts. If the background is dark, the selected text usually becomes brighter. Then, all texts with higher brightness will be considered as highlighted. In contrast, if the background is light, the selected text often becomes darker. Then, the threshold will be the highest acceptable brightness of the highlighted color. All darker texts will be treated as highlighted.

To estimate the threshold, AI-Core provides a calibration tool. It needs several example images

of highlighted and non-highlighted texts. The calibration tool converts the example images to grayscale and takes the pixel values on a diagonal line over each text image. Meanwhile, it eliminates the background to get the median color value of that text. The middle color value between the smallest median value of all given highlighted text examples and the biggest of the non-highlighted ones is calculated as the threshold.



- **Name mapping:** is the setting group for the signal generation in the communication with PROVEtech:TA. In this group, you can define the maximal number of detection regions which can be shown in PROVEtech:TA and the name of each signal. If the number of text detection regions are more than the maximal signal number, the rest of detection results will not be sent to PROVEtech:TA.

From PROVEtech:TA, you can send commands to change the configurations of text detection plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.5.2. Test Result Schema

The results of text detection will be shown in the test result table. In the following, all concrete components of detection results are listed:

- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.
- Content: The detected contents will be shown behind the coordinations.

### 5.5.3. Train with Customized Font Styles

The performance of text detection depends on the font style, font and background colors, and layouts. If you have a special font style needed to be detected, you should train a customized model with the extension of .traineddata.

The process of the training can be separated into several steps roughly:

- AI-Core first creates a group of example text images in the customized font style with different

font and background colors. These images are saved in a .tif file. The content of these example texts are defined by a .training\_text file.

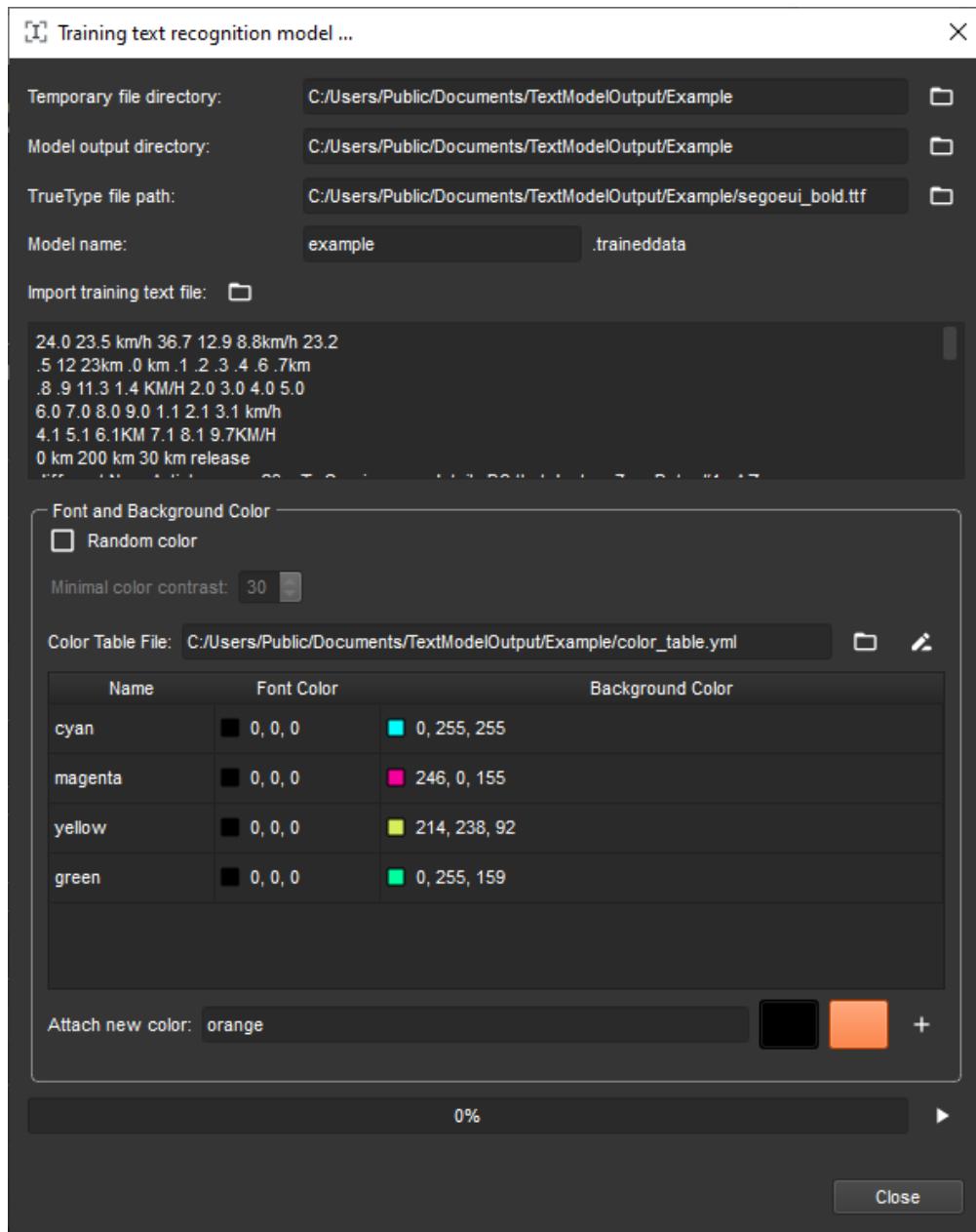
- Based on the images, it generates further a label file ended with .tr, in which the position and content of each character or punctuation will be saved.
- The training program will then train a model according to the images and label files.
- Finally, AI-Core will generate a .traineddata file. Users should copy this file to the pretrained data path, so that AI-Core can detect it.

To achieve all the steps, you need to first prepare a TrueType file, which defines your font style. These files are typically located in path "C:/Windows/Fonts".

After that, open the text detection training dialog window by clicking button **Train customized model** in the configuration panel.

In the text detection training dialog, you can see following parameters:

- **Temporary file directory:** During training, AI-Core needs to generate some temporary files. These files are useful for the later estimation of the training. Thus, you need to assign a place to keep these files. After you are sure that the training is satisfying, you should manually delete all generated files in this folder.
- **Model output directory:** This folder will hold the .traineddata file. Please take care that the default pretrained data directory is located in the path: "C:/Program Files/AKKA/UITestAId/tessdata", which requires administrator right to access. If you define it as the output directory, the write operation might fail.
- **TrueType file path:** Defines the position of your TrueType file.
- **Model name:** Defines the output name of .traineddata file.
- **Import training text file:** This option allows you to import a predefined .training\_text file for training. AI-Core has offered some example in the path "C:/Program Files/AKKA/UITestAId/tessdata/langdata/{language\_name}". However, you can still edit the training contents according to your requirement.
- **Random color:** By default, the font and background colors are randomly generated. You can want to generate texts and backgrounds with specific colors, you can unselect the random color check box.
- **Minimal Color Contrast:** If the color of the font and the background is randomly selected, there is risk that the both colors are too close to distinguish the texts from each other. To avoid this case, we can set a threshold of minimal color contrast. Only if the color contrast is larger than the threshold, the text images can then be accepted and generated.
- **Color Table:** If you want to choose specific font colors and background colors, you can add new color combinations to the color table. To reuse the configuration, AI-Core allows you to save and load color table from a yaml file.



After you have configured everything, just press the "Play" button. AI-Core will automatically proceed the training procedure. Please wait for several minutes until the progress bar reaches the end.

#### 5.5.4. Detection Filter

On the top right side, you will find the following button:



By pressing this button, you are able to draw additional Regions of Interest (ROIs), which specify in which regions of the image you want the detections to occur. This works the same way as it does for icon detection. Therefore, with this feature, you can suppress certain text detections.

[.notice] Notice: The regions of interest can be drawn at any time, but they only affect the text detections while in Automatic Mode. Rectangular regions drawn in Manual Mode are not affected

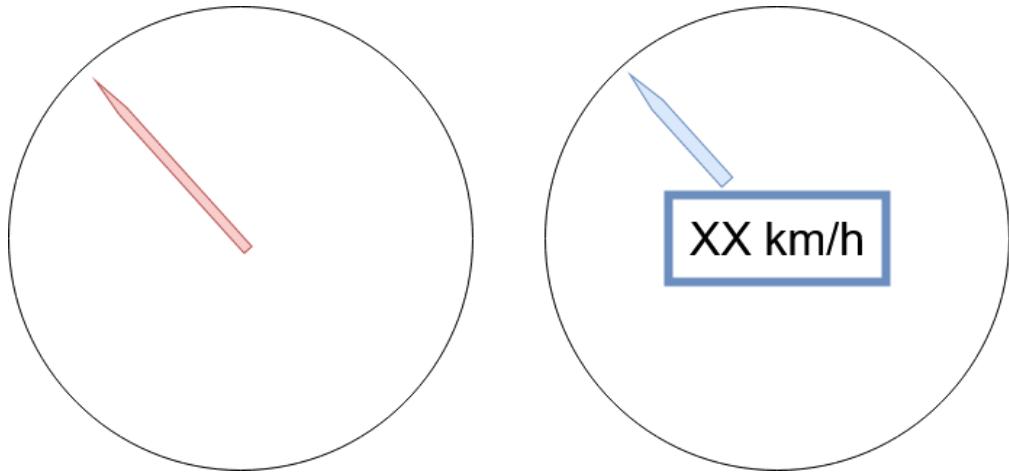
by the regions of interest.

## 5.6. Needle Detection

Especially in industrial applications, you need to detect line-shaped needles in analog clocks or instruments. Needle detection is a plugin which is aimed to detect such kind of needles from video streams. The detection is based on some simple assumptions:

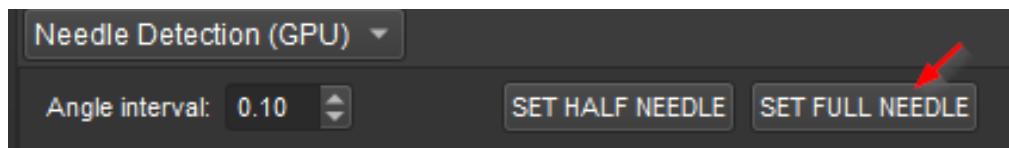
- The needle rotates around a fixed center.
- The color of the needle does not change a lot.
- The needle can be distinguished from the background easily.

AI-Core has two needle detection modes: "Full Needle" & "Half Needle". The following diagram has depicted the both cases:

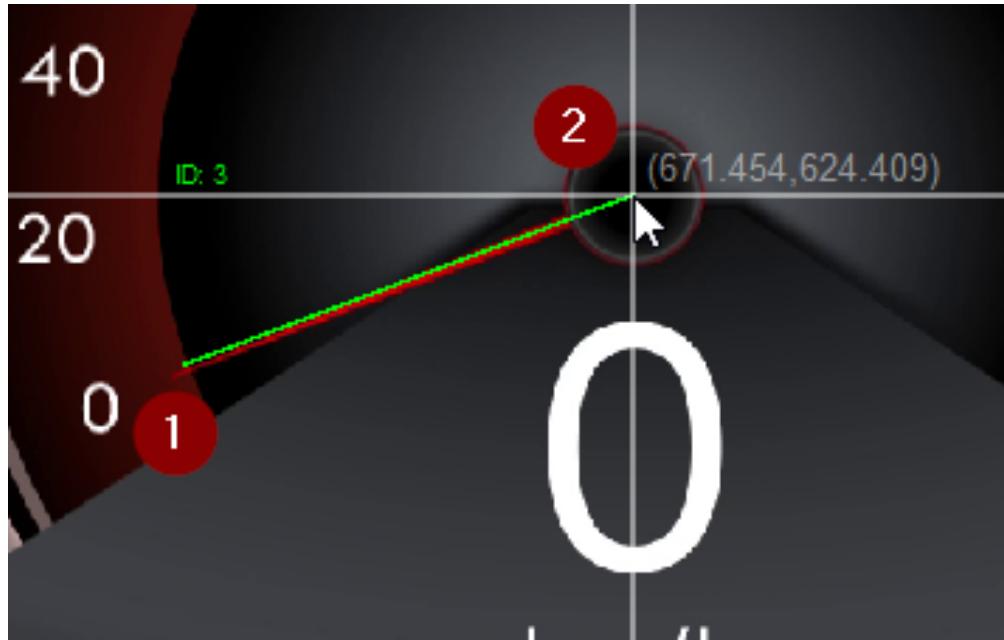


In "Full Needle" mode (left), the needle starts from the rotation center and completely visible. In "Half Needle" mode (right), the needle does not start from the rotation center and is only partially visible.

In "Full Needle" mode, you need to first click "SET FULL NEEDLE" button in the settings panel of needle detection.

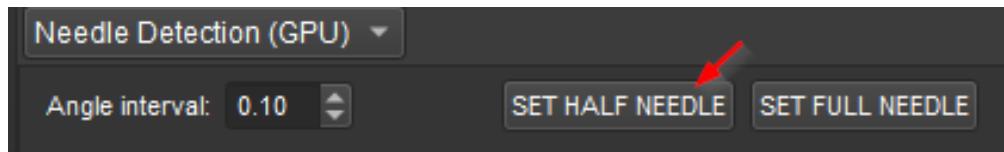


Then, draw a line along the needle from the outer end to the center by pressing and dragging the left button. After all, click play "Play" button to check the detection result.



If you are not satisfied with the drawing, you can either delete the line by pressing "Delete" key, or adjust the position by dragging the both ends of the needle line.

In "Half Needle" mode, you need to first click "SET HALF NEEDLE" button in the settings panel of needle detection.



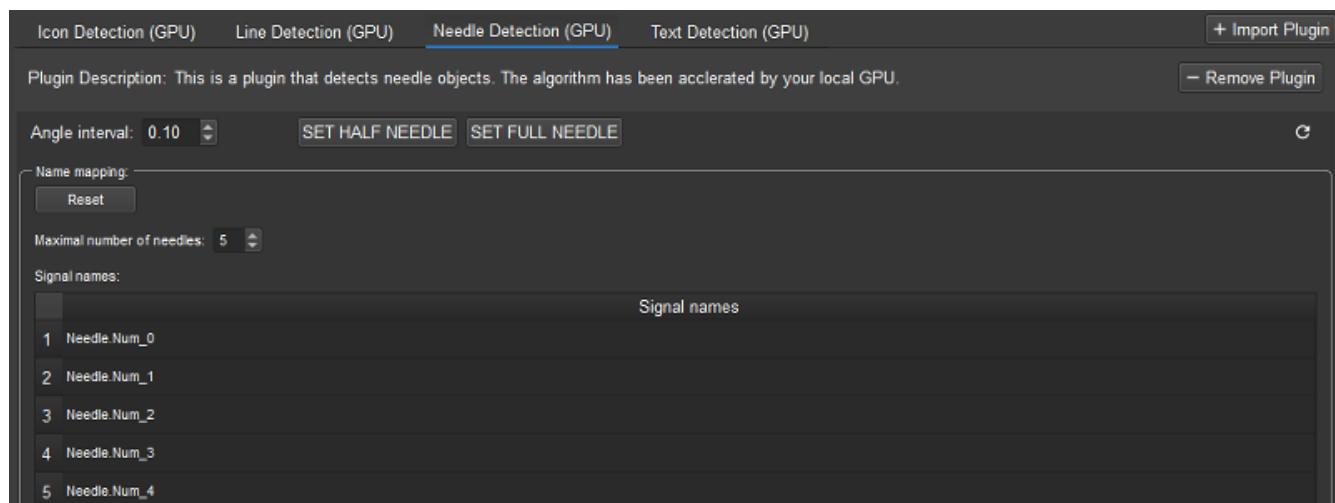
Second, draw a line along the needle from the outer by pressing and dragging the left button. Third, when the mouse is moved to the inner visible end, release the left button. Fourth, move the mouse, and a dashed-line circle will appear and follow the mouse. When the mouse should be moved to the position so that the center point is located at the rotation center of the needle. Then, click the left button to finish the drawing. Finally, click play "Play" button to check the detection result.



Similar to "Full Needle" mode, in "Half Needle" mode, you are also allowed to adjust the needle after drawing. You can either delete the needle or adjust the needle and circle by dragging the both ends of the needles and the yellow handle point. If you drag the internal part of the circle, you can also translate the whole plot.

### 5.6.1. Needle Detection Settings

The detection parameters can be found in settings panel of needle detection (ADJUST CONFIGURATION > NEEDLE DETECTION) or in the testing mode settings from tool bar.



There is only one detection parameter: angle interval. It defines how accurate the needle angle can be measured. A low angle interval will increase the angular resolution of the needle detection, but reduce the detection speed.

Furthermore, you can find the setting group "Name mapping". In this group, you can define the maximal number of detection regions which can be shown in PROVEtech:TA and the name of each signal. If the number of needles are more than the maximal signal number, the rest of detection results will not be sent to PROVEtech:TA.

From PROVEtech:TA, you can send commands to change the configurations of needle detection plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.6.2. Test Result Schema

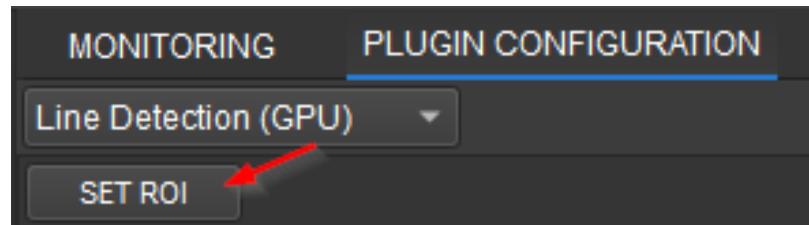
After AI-Core detects the needles, the results will be shown in the test result table. In the following, all concrete components of detection results are listed:

- CX (full needle): the x-coordinate of the central needle end.
- CY (full needle): the y-coordinate of the central needle end.
- EX (full needle): the x-coordinate of the outer needle end.
- EY (full needle): the y-coordinate of the outer needle end.
- IX (half needle): the x-coordinate of the inner needle end.
- IY (half needle): the y-coordinate of the inner needle end.
- OX (half needle): the x-coordinate of the outer needle end.
- OY (half needle): the y-coordinate of the outer needle end.
- Angle: the angle of the needle.

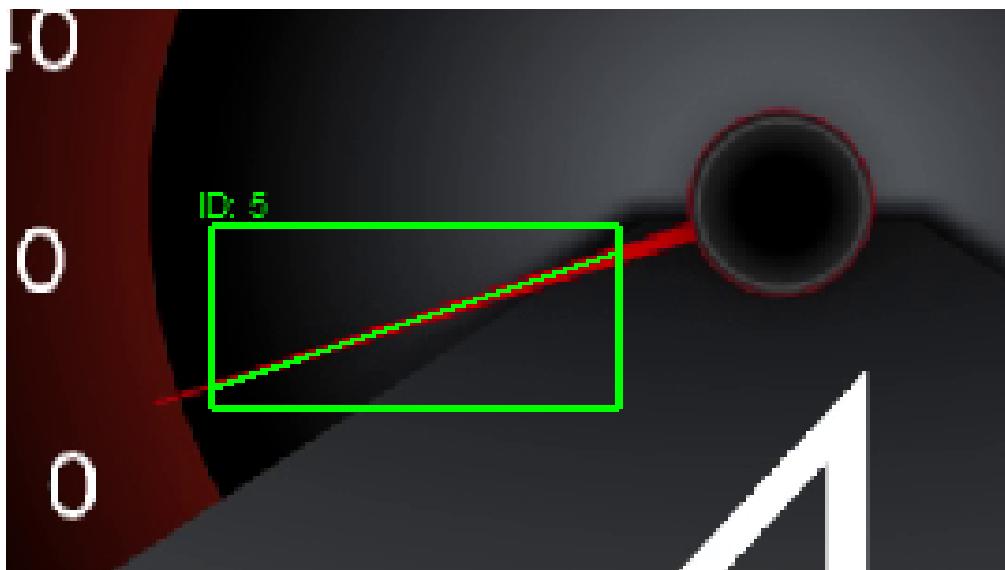
## 5.7. Line Detection

Line detection is a plugin which is aimed to detect a line in a specific rectangular region. Line detection plugin can not only provide the length and angle of the detected line, but also give the precise position of the line. The detection does not need any training process and is able to feedback real-time results.

To create a detection region, you need to first click "SET ROI" button in the setting panel of line detection.



Second, draw a rectangular bounding box by pressing and dragging the left button. Then, click play "Play" button to check the detection result.



The line detection algorithm is based on [Canny edge detection](#) and [Hough transformation](#). Since test cases could vary a lot, such as blurring, noise etc., some parameters which control the detection sensitivity cannot adapt all these situations. Therefore, to obtain the optimal results, you need to adjust these parameters based on the actual use case.

### 5.7.1. Line Detection Settings

The parameters can be divided to three groups: preprocessing, detection settings, and name mapping. Preprocessing settings includes four parameters:

- **1. & 2. threshold:** The line detector first finds all edges. Strong edges usually have obviously higher gradient values. The edges are detected by filtering gradient values with respect to each pixel.

Canny edge detection algorithm ignores the pixels whose gradient values are lower than 1. threshold. Then, it checks whether there are edges on which the pixels have higher gradient

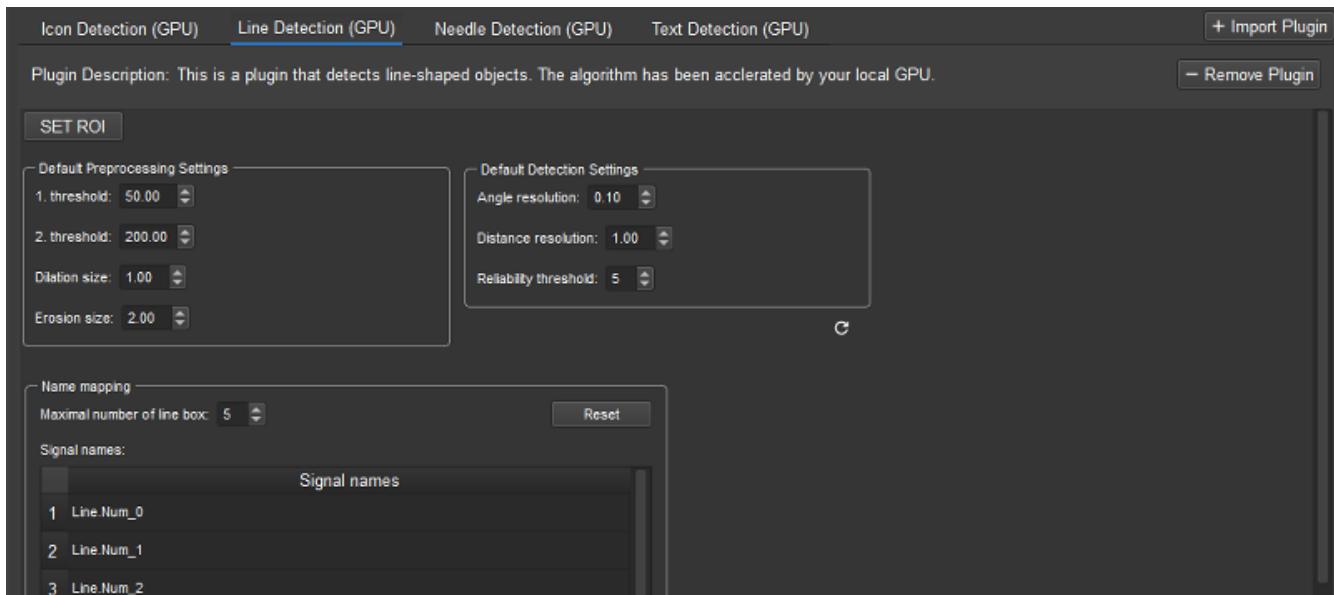
values than 2. threshold. Once it finds any edges, the connected pixels whose values range between 1. and 2. thresholds will also be considered as a part of the detected edges.

In brief, when the edges could be fuzzy, you should select lower 1. and 2. threshold values (e.g. 10 and 12). When the edges are sharp, you should select higher 1. and 2. threshold values (e.g. 50 and 100) in order to ignore unwished minor edges.

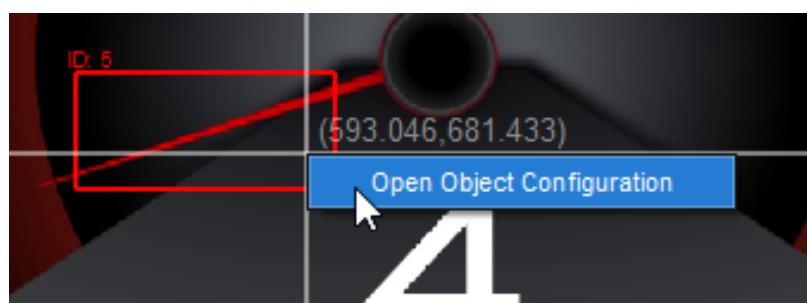
- **Dilation & Erosion size:** Sometimes, some trivial edges are clear and sharp. However, they are broken and distributed sparsely. The double threshold strategy is still not sufficient to filter them out. The algorithm [morphological transformations](#) will further eliminate those small and broken edges or noise from images.

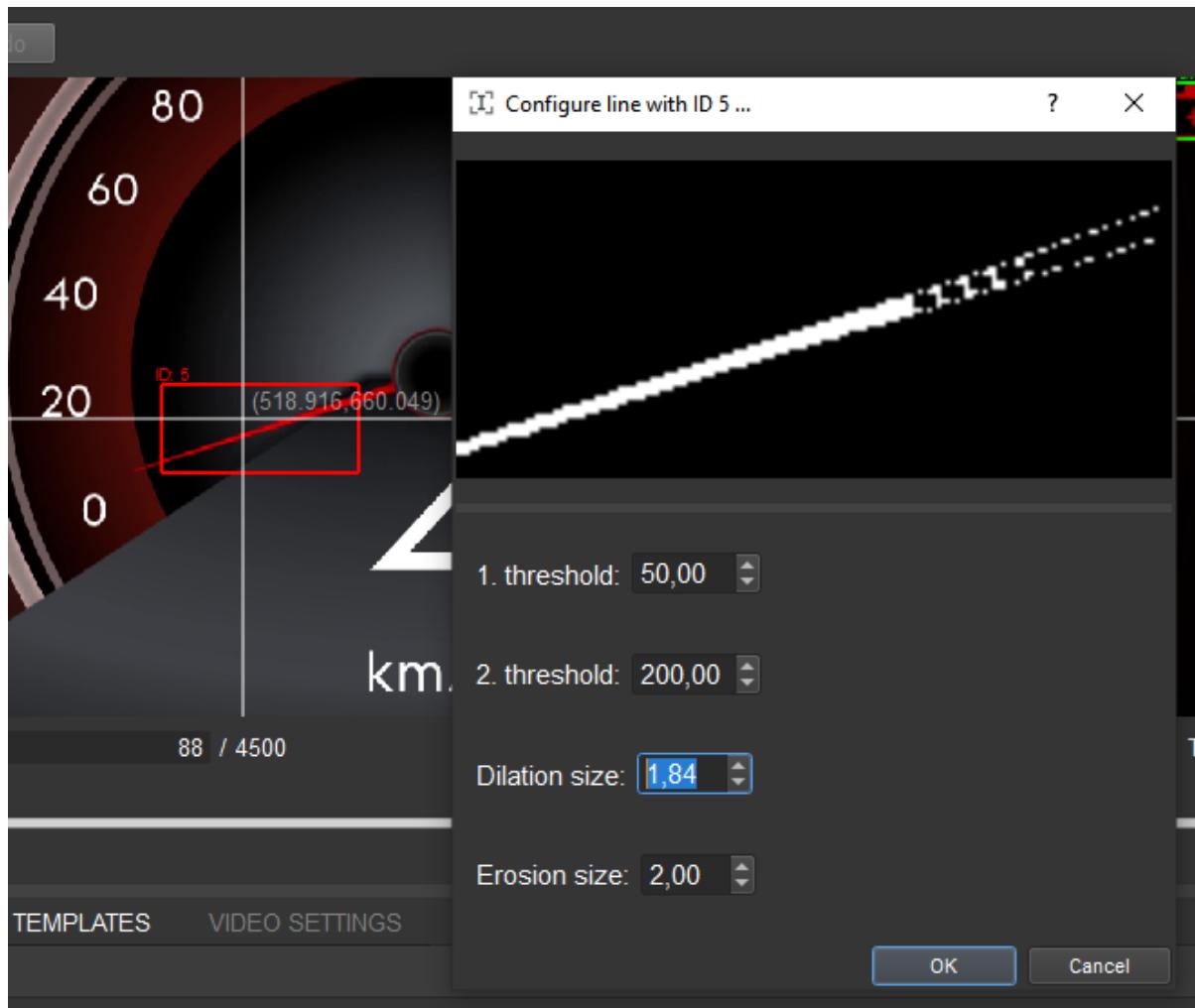
When you set a larger dilation size and a smaller erosion size (e.g. 2.0 and 1.0), the line detector will intend to connect more close broken edges but remove isolated small objects.

In contrast, when dilation size is smaller than erosion size (e.g. 1.0 and 2.5), more small and broken edges will disappear. Only those thick and obvious edges will remain.



If right clicking the region box, you will see a context menu. Select "Show processed image" in the context menu, and then a dialog will pop up. In the dialog, you can check the result of preprocessing and adjust the preprocessing parameters.





Detection settings includes five parameters. They decide which edges can compose a line.

- **Angle & Distance interval:** After preprocessing, the image of extracted edges will be mapped into a 2-dimensional space. The two dimensions represent angle and distance. the line detector will search lines in the space. Angle and distance interval define the search resolution in the space. The smaller interval you set, the more precise results it will turn out. However, the small interval will also increase the computational effort, and thus reduce speed.
- **Reliability threshold:** This is an integer parameter which should be larger than or equal to 1. During searching lines, the line detector may find multiple candidates. A higher reliability threshold will filter out those unreliable results.

Name mapping is the setting group for the signal generation in the communication with PROVEtech:TA. In this group, you can define the maximal number of detected lines which can be shown in PROVEtech:TA and the name of each signal. If the number of line detection regions are more than the maximal signal number, the rest of detection results will not be sent to PROVEtech:TA.

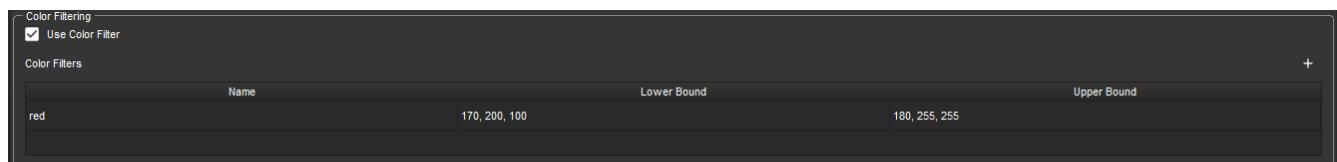
From PROVEtech:TA, you can send commands to change the configurations of line detection plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.7.2. Line Detection Color Filtering

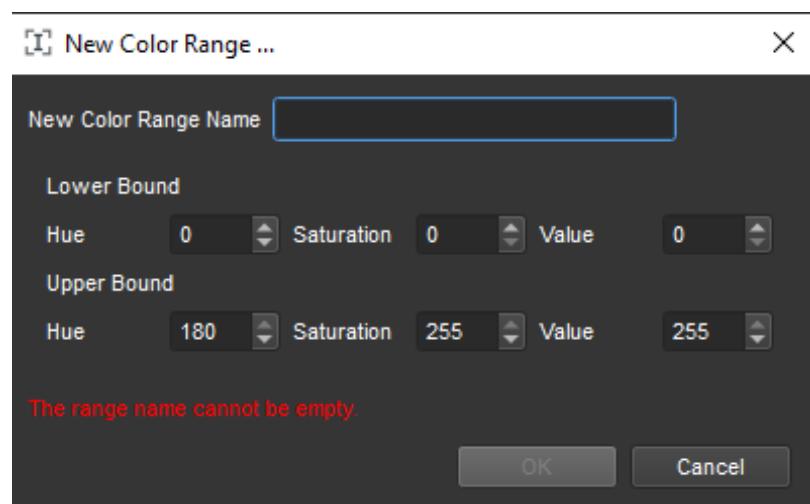
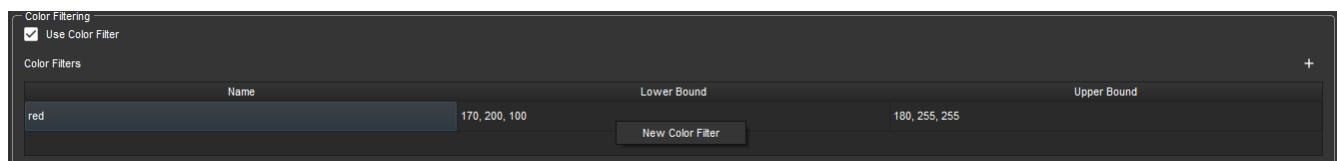
An enhancement to the existing settings now enables the detection of lines based on their color.

While the earlier settings facilitated the detection of lines within a specified Region of Interest, there are scenarios where it is necessary to identify a line based solely on its color. For instance, if the line alternates between red and green, this feature would allow the system to detect the line only when it is red.

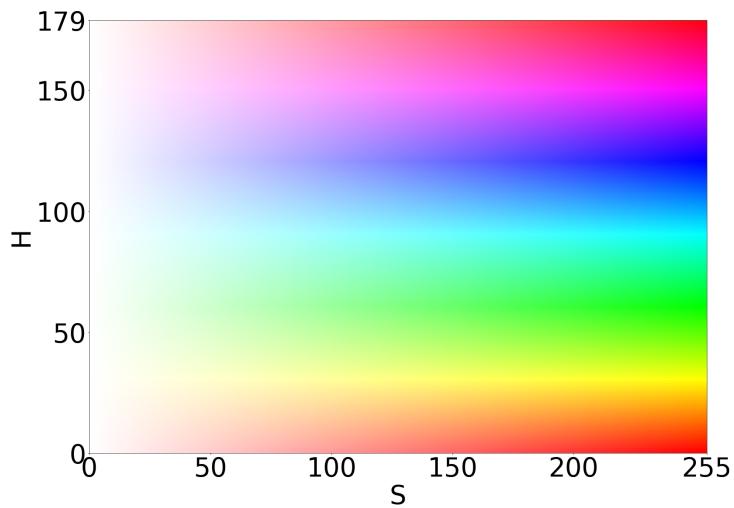
This feature incorporates a selective filtering system that disregards pixels falling outside a user-defined color spectrum, focusing exclusively on those within the specified range. The identified line can then be refined using existing settings such as edge detection and dilation.



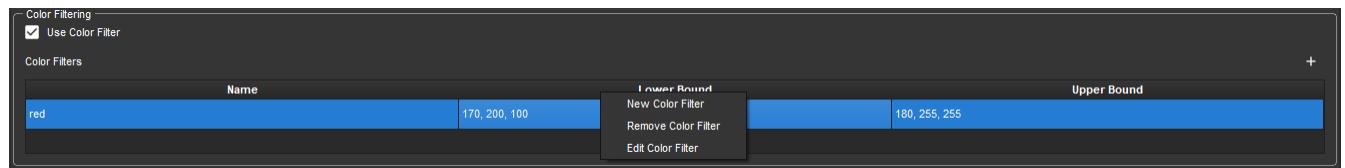
The Color Filter can be turned on and off by enabling/disabling the "Use Color Filter" Check Box on the left side. Beneath this checkbox is a table where the desired color ranges for filtering are defined. Right-click in an empty area of the table to bring up a context menu, and select "New Color Filter." or press the plus icon on the right side to open a new window to specify the color range.



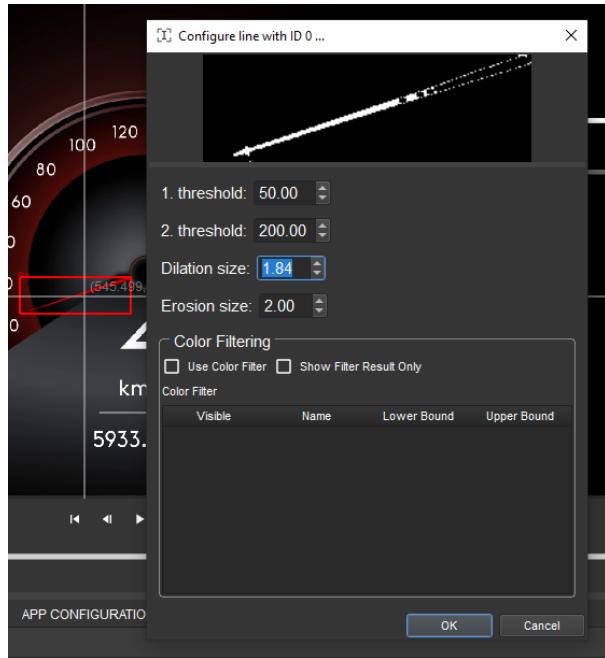
At the top of this window, you can name the color range, which can be any label of your choosing. To accurately set the color range, you must establish lower and upper bounds for the color you wish to include. AI-Core operates in the HSV color space.



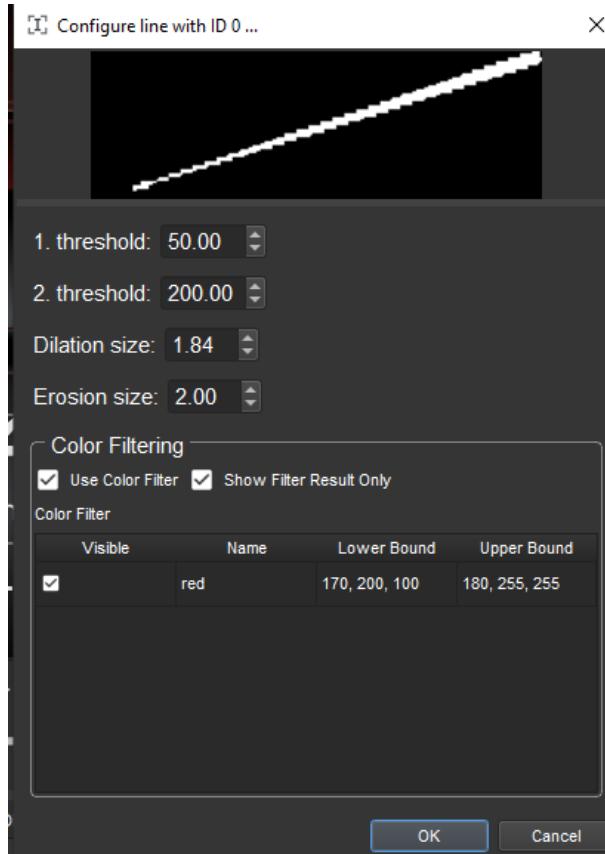
In the HSV color spectrum, hue represents the type of color and ranges from 0 to 180. Saturation indicates the intensity of the color, with a range of 0 to 255, where a saturation of 0 results in gray. Value measures the brightness of the color, also ranging from 0 to 255. To ascertain the color range for your target color, an external tool is necessary. We suggest using the free software GIMP Portable [install GIMP Portable](#). The HSV values extracted from GIMP Portable can help determine your color range's bounds. For the hue, take the 'H' value from GIMP and add or subtract 5 to set the upper and lower bounds, respectively. For saturation and value, either adjust the GIMP values by plus or minus 30 or default to the standard bounds of 0 and 255. By clicking 'Ok,' you add a new entry to the table. Right-clicking on an existing entry in the table opens a context menu that allows for the deletion, editing, or creation of a color filter.



Similar to other settings, the outcomes of the algorithm can be displayed. To visualize them, right-click within the region box and select "Show processed image" from the context menu that appears. The Color Filter settings include two checkboxes:



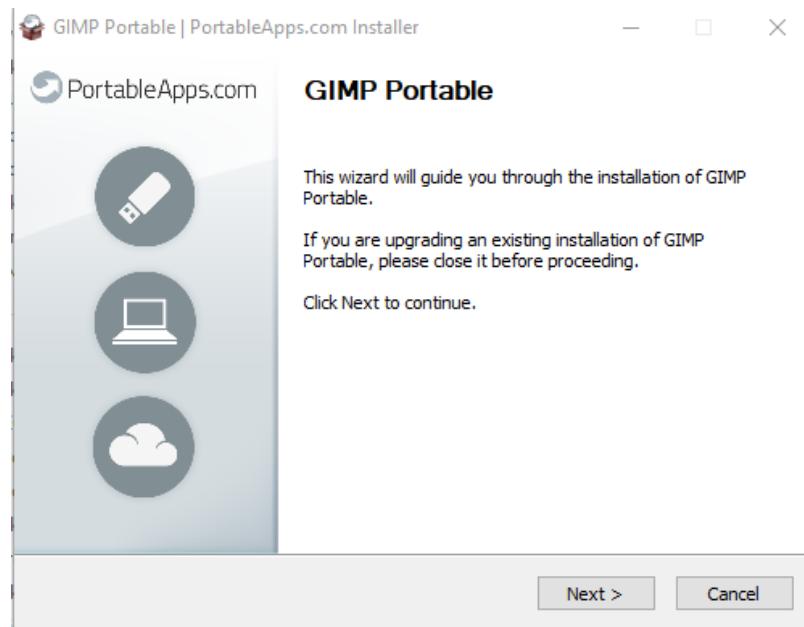
The "Use Color Filter" checkbox toggles the color filter on or off, while the "Show Filter Results Only" checkbox determines whether to apply only the color filter or to include other settings like dilation and erosion. Below the checkboxes, there is a table where you can set the color ranges as previously described. When the color filter is accurately configured, the object of interest will be displayed as white in the preview window. This confirms that only the intended object will be recognized.



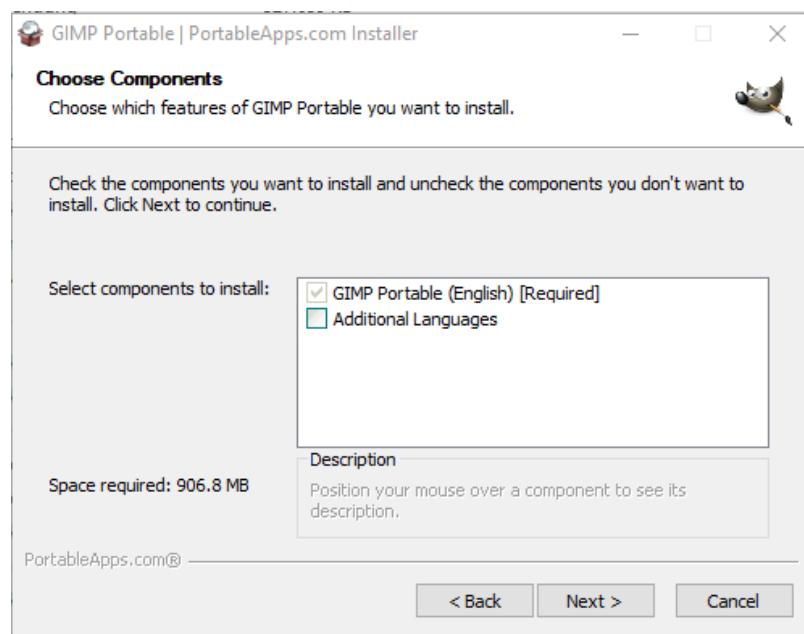
### 5.7.3. Installing GIMP Portable

You can download the GIMP Portable application from the following website:

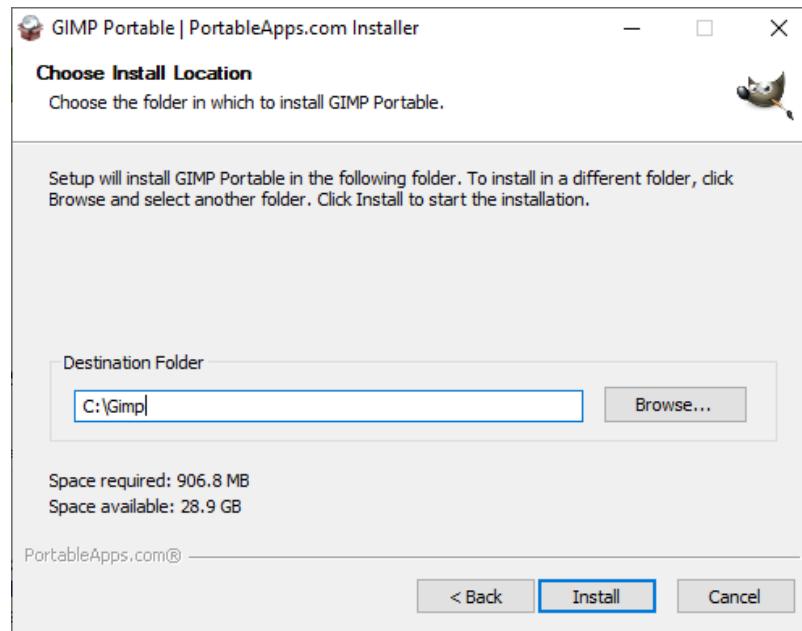
[https://www.gimp24.de/gimp\\_download\\_portable.php](https://www.gimp24.de/gimp_download_portable.php). Once the download is complete, double-click the executable file to initiate the installation process.



An installation window will pop up. Click 'Next' to proceed to the subsequent window:



At this stage, you have the option to install additional languages. For our purposes, extra languages are not necessary. Click 'Next' once more, and you will be prompted to choose a destination folder for GIMP Portable. The choice of installation location is yours to make.

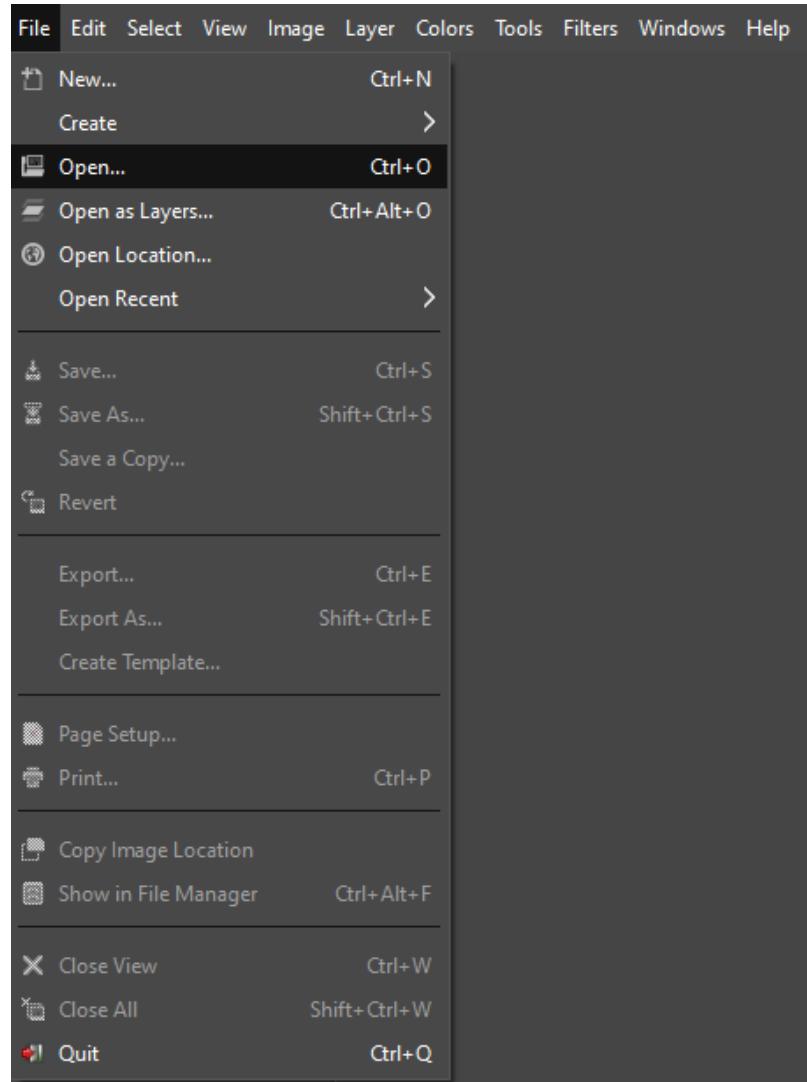


After selecting the desired folder, click the 'Install' button and wait for the installation to complete. Once finished, click 'Finish' to conclude the installation process. Remember, no desktop shortcut will be created, so you will need to navigate to the folder where GIMP Portable is installed and launch GIMPPortable.exe manually.

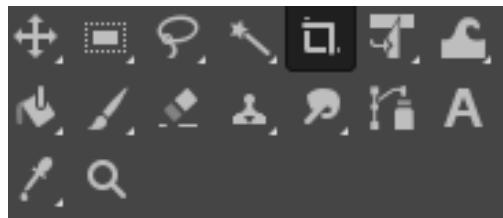
Upon opening the application, you should see the main window.



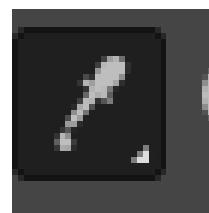
In the top left corner, click on 'File' to access a dropdown menu, then select 'Open' to use the file explorer to locate and open your image file.



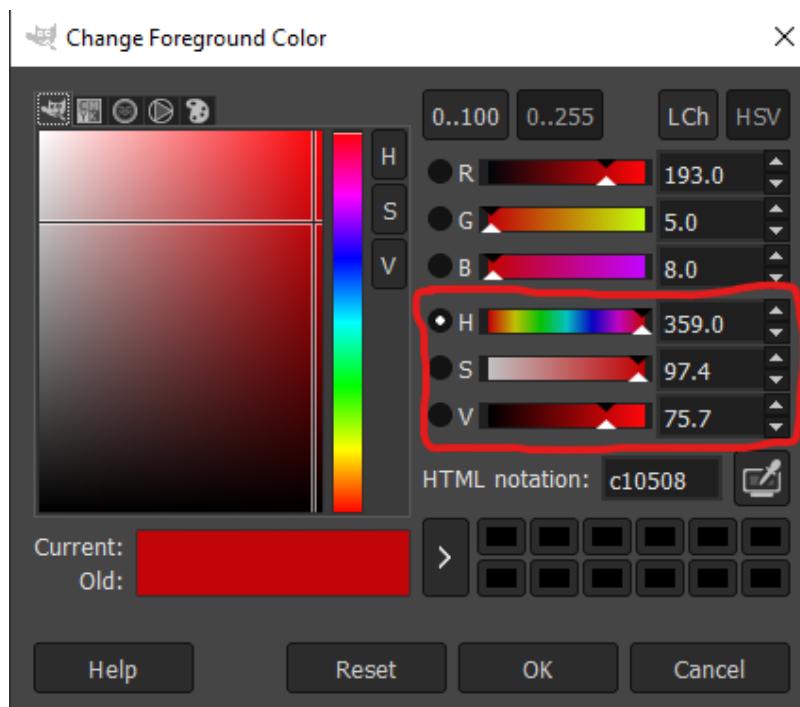
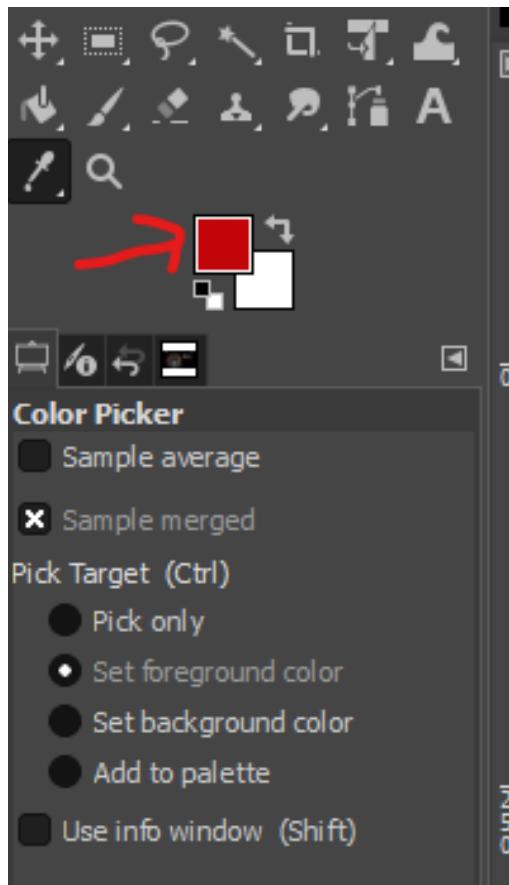
Now, turn your attention to the left side of the application, where you will find various icons.



Locate and click on the 'Color Picker' icon:



Then click on the area of the image where you want to select the color. After selecting the color, click on the colored square to bring up a new window:



Here, on the right side, you will find the HSV (Hue, Saturation, Value) values for the color of the selected object. Note that the HSV values are scaled differently compared to those used in AICore. To convert, divide the 'H' value by 2, and multiply both the 'S' and 'V' values by 2.55.

#### 5.7.4. Test Result Schema

The results of line detection will be shown in the test result table. In the following, all concrete components of detection results are listed:

- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.
- P1: The start point coordination (X,Y) of the detected line.
- P2: The end point coordination (X,Y) of the detected line.
- Angle: The angle of the detected line. The angle is defined in counterclockwise orientation, varying from 0° (at 3 o'clock) to 180° (at 9 o'clock).
- Length: The length of the detected line in pixel.

Additionally, when the Color Filter is applied:

- color: This represents the hue of the line that has been identified.
- colorID: This is the associated identifier for the color of the line detected.

## 5.8. Audio Matching

Since 2023, AI-Core supports audio matching in the test mode. You can give some audio reference files as input and configure your microphone to stream audio signals for matching.

Audio Matching is a plugin that aims to find the optimal match by comparing the similarity between the source audio signal, which comes from the microphone, and the target reference audio files. The plug-in accepts some reference audio files in the format of wav as input and processes in real time each part of the source audio signals in the form of frames.

It is based on digital signal processing (DSP) and dynamic time warping (DTW), which is an algorithm for measuring similarity between two given series of feature vectors. With the help of DSP, it calculates and extracts all relevant audio features from raw audio signals, for example, MFCC, which is one of the main features used for the audio matching. On the other hand, DTW compares the features of the source audio signal, which comes from a microphone, with the features of each of the given reference audio signals. Then, it detects and returns the most similar reference audio signal to the source audio signal.

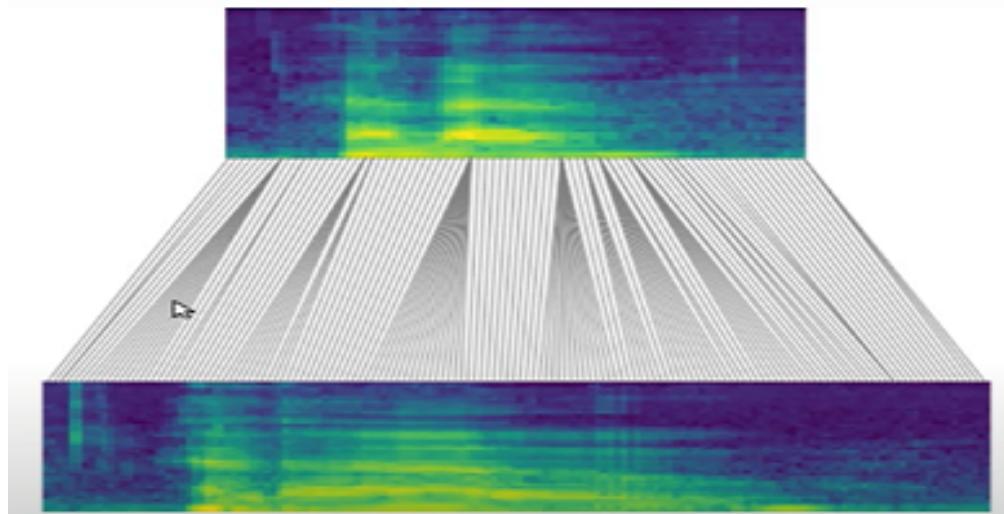
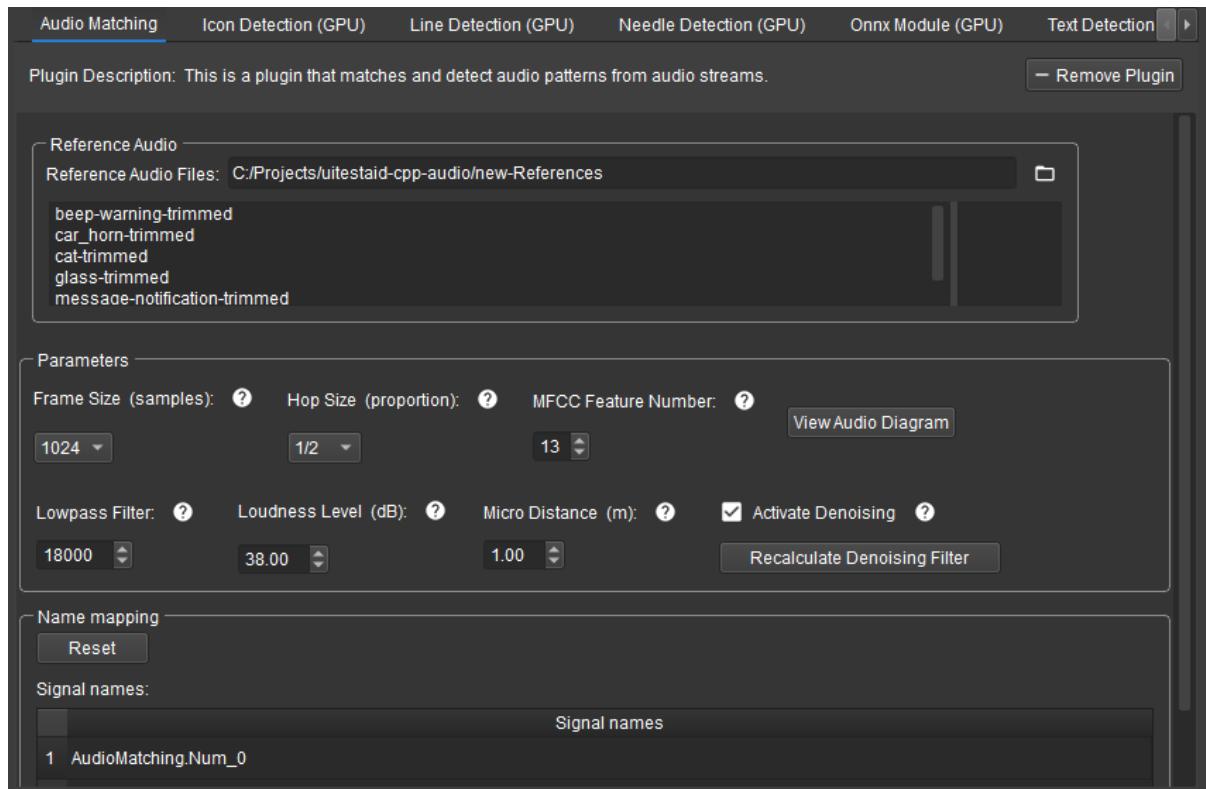


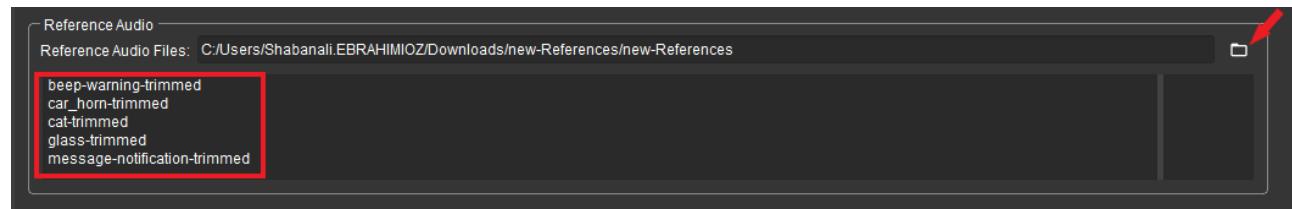
Figure 1. Dynamic Time Warping

### 5.8.1. Audio Matching Settings

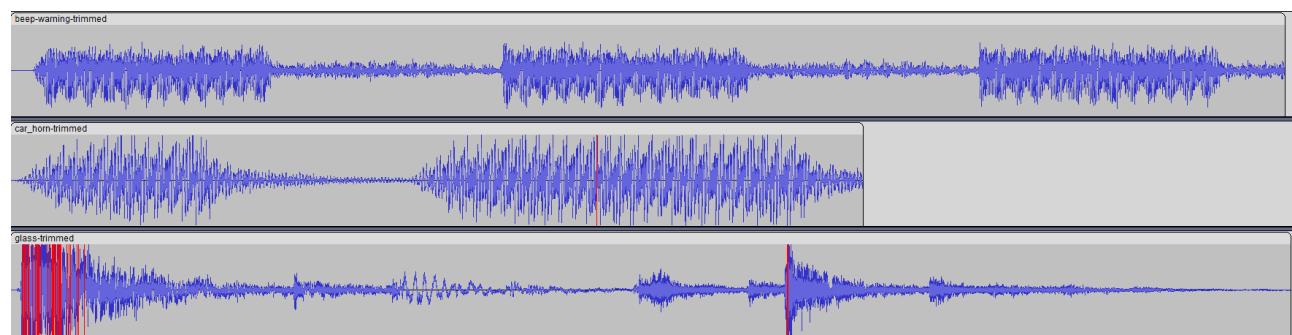


- **Importing the reference audio files:**

To import your folder of reference audio files, you need to click the folder icon, which is shown with the red arrow on the above picture. After importing the folder, you can see the names of audio files in the panel, as shown in the red box in the above picture.



The reference audio files need to be in ".wav" format. For optimal results, they need to be trimmed with third-party audio tool, for example "Audacity".



- **Frame size:**

To process an audio signal stream, we need to divide the whole signal into several chunks, which are called frames, of the same size. A frame is a chunk of the whole audio signal to be processed. Frame size is the number of samples in each frame. For some mathematical reason, the frame size needs to be a power of 2, for example, a frame with 2048 samples.

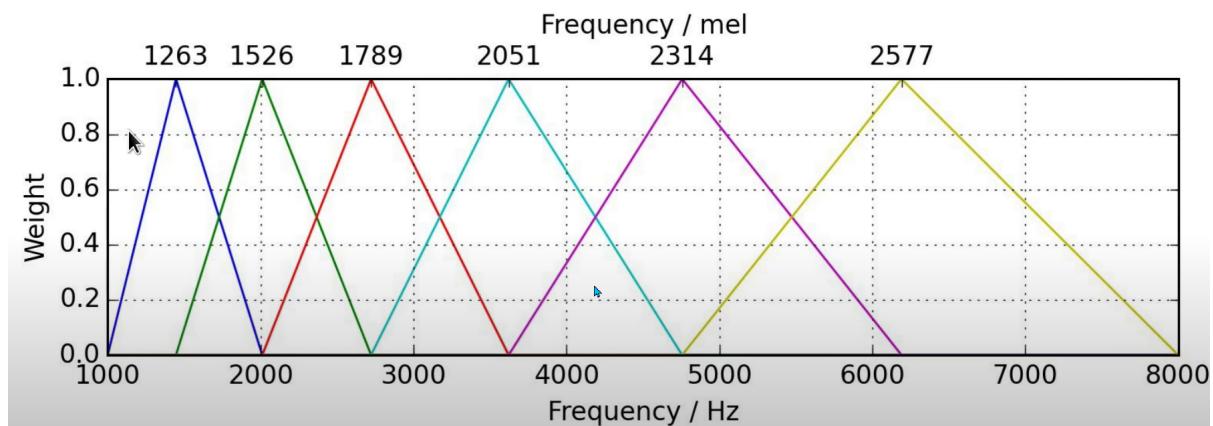
- **Hop size:**

In DPS for calculating the spectrum of an audio signal, frames need to be overlapped with the length of the hop size in order to solve the problem of spectral leakage. Each new frame starts at the hop size of the last frame, as shown in the following picture.



- **MFCC Feature Number:**

MFCC is a technique that allows you to extract audio features from the raw audio signal. Our ears have better resolution at a lower frequency than at a higher, for example, we hear better sounds at 200 Hz than at 2000 Hz. In MFCC, we have a mel-scale to map the actual frequency to the frequency that human beings will perceive.



Mfcc (mel-frequency cepstrum) is derived from cepstrum. Cepstrum is made of the number of mel-bands that will be used to make a mel filter bank, which is used to reduce the number of computed frequency points to mel frequency points through the mel scale, which is a dimensionality reduction technique for the spectrum.

On the other hand, MFCCs are coefficients similar to mel bands. The differences between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum with typical value of 13. The MFCC feature number parameter is the number of coefficients in range of [8, 13].

- **Low-pass Filter:**

Many unwanted noises live in high-frequency. The low-pass filter passes only signals below its cut-off frequency so that it removes high-frequency sounds from an audio signal. You can choose your cut-off frequency based on human hearing range, which is from 20 Hz to 20 kHz.

- **Loudness Level (dB):**

Audio matching plugin matches reference signals with active signals from the auditory stream. To extract the active parts, AI-Core measures the average level of audio signal amplitudes and compares it with a loudness threshold. If the input signal is higher than a specific loudness level, it will treat the subsequent signals as active until the audio becomes silent again.

Based on the threshold of human hearing, the power level and quality of the Microphone, and even the quality of the target sounds, which will be received with the microphone, you may need to change this parameter to have optimal results.

The value range is in [0, 999.9], 0 means it needs to capture and process all incoming sounds. The higher the threshold is, the more low-level signals will be ignored. The default value 30 dB is usually the lowest hearable volume level.

**Notice:** To obtain the optimal result, you should keep the environment as silent as possible and increase the test signal loudness.

- **Microphone Distance:**

To calculate the intensity level feature from the signal, it is required to set the microphone distance. It will scale the signal based on the distance between the microphone and the audio source.

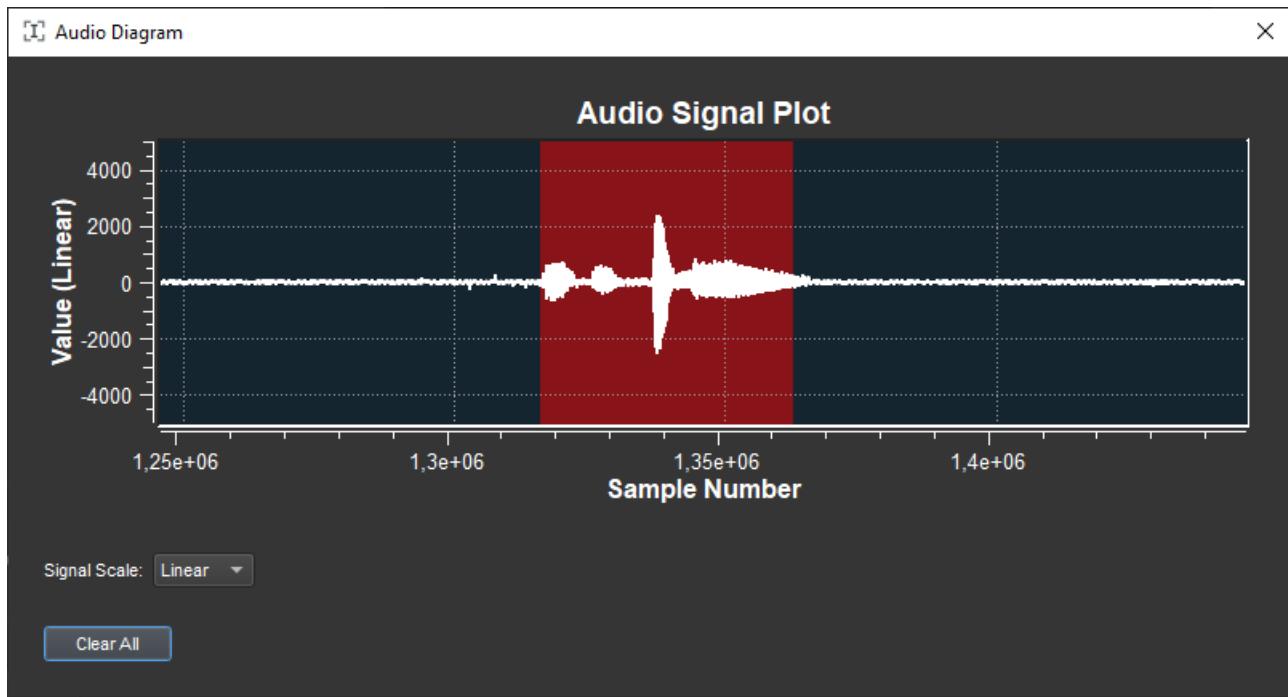
- **Denoising**

In case of inevitable noise in the test environment, AI-Core provides a denoising function. If you activate this function, every time when a test starts, AI-Core first records a section of environment sound and analyze the noise frequency range. Those frequencies which are higher than the loudness level threshold will be recorded. During the test, AI-Core filters out these noise frequencies temporarily and then estimates the active sections.

Through pressing the button "Recalculate Denoising Filter", you can manually reset the noise frequencies.

- **Audio Diagram:**

After starting the audio matching plug-in, you can go to Plugin Configuration and click on this button to see a real-time visualization of audio matching processes. The diagram plots the microphone audio signal sample values (Linear/dB) on the y-axis and sample number on the x-axis. The detected active section will be labeled in a red region.



## 5.8.2. Test Result Schema

After AI-Core detects a match between the auditory stream and one of the reference audio files, the results will be shown in the test result table. In the following, all concrete components of the detection results are listed:

MONITORING PLUGIN CONFIGURATION VIDEO CONFIGURATION TEMPLATES APP CONFIGURATION						
Audio Matching		ID	Result Content			
Frame	ID					
0	0	glass-trimmed (ClassId: 3): mfccDist: 101158802.878, matchedPropToTest: 0.520, matchedPropToRef: 0.987, weightedMfccDist: 197129395.987, LenDiff: 2.00, RmsDiff: 4.73				

- **Audio Reference Name:** It is the name of a reference audio file that the detected sound matched to.
- **mfccDist:** The normalized calculated distance by DTW of the detected source sound and the target reference audio sound.
- **matchedPropToTest:** The normalized calculated distance by DTW, based on source Mfcc size, of the detected source sound and the target reference audio sound.
- **matchedPropToRef:** The normalized calculated distance by DTW, based on reference Mfcc size, of the detected source sound and the target reference audio sound.
- **weightedMfccDist:** The normalized calculated distance by DTW, based on both source and reference Mfcc size, of the detected source sound and the target reference audio sound.
- **LenDiff:** The difference between the number of frames of the detected source sound and the target reference audio sound
- **RmsDiff:** The difference between the RMS value of the detected source sound and the target reference audio sound.

## 5.9. Screen Analysis

Since 2023, AI-Core supports screen analysis in test mode. Screen analysis is a plugin that helps find abnormal screens from the input. In this plugin, three types of abnormal screens, such as blank, blurred, and frozen, are under analysis.

- **Blank screen detection:**

A blank screen is one that has no visual content inside it, for example, a pure black image. This kind of screen can appear during test and can be caused by a wrong camera setting or focus area, as well as a blank image inside an image folder.

To detect a blank screen, AI-Core conducts two steps.

1. First, it calculates the maximal brightness standard deviation among three color channels. If the maximum deviation value is lower than the user-defined threshold, the screen will be detected as a candidate to be blank.
2. To avoid ignoring small objects which has weak influence on the deviation. AI-Core will then check the contours of objects on the screen. If the screen has no object inside, the screen will be categorized as blank.

- **Frozen screen detection:**

Sometimes, the camera could keep frozen, such as when the camera's software needs to be updated or there is possibly an issue with a malfunctioning camera. In these cases, AI-Core receives the same screen repeatedly. Frozen screen detection helps to find such a case, which returns the probability if the current screen and the previous one are identical. AI-Core computes the identically between the current screen and the previous one. If the two screens are identical for a user-defined amount of time in milliseconds, it will be detected as a frozen screen.

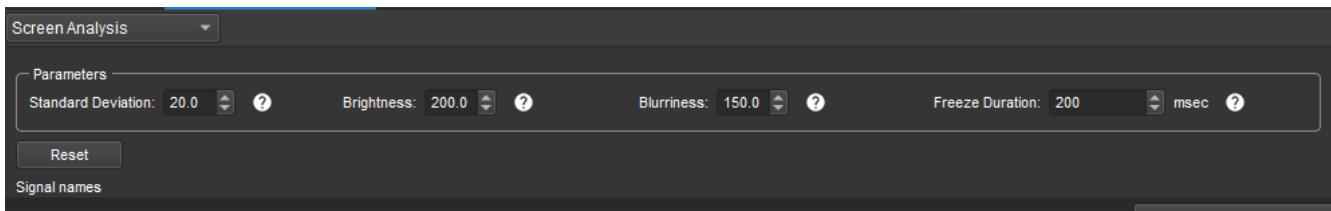
- **Blurred screen detection:**

A blurred screen is a low-contrast screen, that has fewer details than a normal screen, which isn't proper for analysis with AI-Core. This kind of screen can be caused by the focus change, the movement of the camera, or improper settings.

To detect such screens, AI-Core examines the distribution in frequency domain. If there is a low amount of high frequencies, then the image can be considered blurry. In more details, AI-Core adopts the Laplacian function to measure the second derivative of an image. The Laplacian highlights regions of an image containing rapid intensity changes, corresponding to details in the image. By taking the variance (standard deviation squared) of the response, if the variance falls below a pre-defined threshold, then the image is considered blurry.

### 5.9.1. Screen Analysis Settings

After creating a test project, you can configure some parameters of the screen analysis plugin as follows:



- **Standard derivation threshold:** The threshold for the maximum standard deviation of the brightness among three channels, below which the screen will be detected as blank, if no contour is found from the screen further. The default value is 20.0.
- **Brightness threshold:** The threshold for the maximum deviation from the mean value of the value part of the HSV color of the screen (three channels), below which the screen will be detected as black color and above that will be white. The default value is 200.0.
- **Blurriness threshold:** The threshold, below which the screen will be detected as blurred, after the calculation of the variance of the Laplacian of the screen. The default value is 150.0.
- **Freeze Duration:** The threshold in milliseconds, above which the screen will be detected as frozen, after consecutive amount of time in milliseconds, when the current screen and the previous one were always identical.

## 5.9.2. Test Result Schema

After AI-Core detects one of the abnormal screens, the results will be shown in the test result table. In the following, all the concrete components of the detection results are listed. For more information, please read the documentation above:

MONITORING		PLUGIN CONFIGURATION		VIDEO CONFIGURATION		TEMPLATES		APP CONFIGURATION	
Screen Analysis									
Frame	ID	Result Content							
7	0	[screenType:none, maxStd:25.1, Brightness:255.0, changeType:frozen, MatchProb:1.0, Blurriness:1000.0, FrozenCount:1]							
6	0	[screenType:none, maxStd:25.4, Brightness:255.0, changeType:blurred, MatchProb:0.3, Blurriness:67.7, FrozenCount:0]							
5	0	[screenType:white, maxStd:0.0, Brightness:230.0, changeType:none, MatchProb:0.0, Blurriness:1000.0, FrozenCount:0]							

- **ScreenType:** The name of the detected screen type, which is one of the items in the list [none, white, black], which means the blank screen is black. If the calculated value is less than the user-given brightness threshold, it is white above the threshold.
- **MaxStd:** A value in the range [0.0, 255.0]. Based on the user given threshold, below this value, the screen is a candidate to be blank.
- **Brightness:** A value in range [0.0, 255.0]. Based on the user given threshold, below this value the screen will be detected as black, above that it is white.
- **Blurriness:** A value in range [0.0, 1000.0]. Based on the user given threshold, below this value the screen will be detected as blurred.
- **Freeze Duration:** Amount of time in milliseconds. Based on the user-given threshold, above this value, the screen will be detected as frozen after a consecutive amount of time in milliseconds if the current screen and the previous one were always identical.
- **ChangeType:** The name of the detected screen type, which is one of the items in the list [none,

frozen, blurring].

- **MatchProb:** In the process of detecting a frozen screen, this value is the probability that the current screen and the previous one are identical.

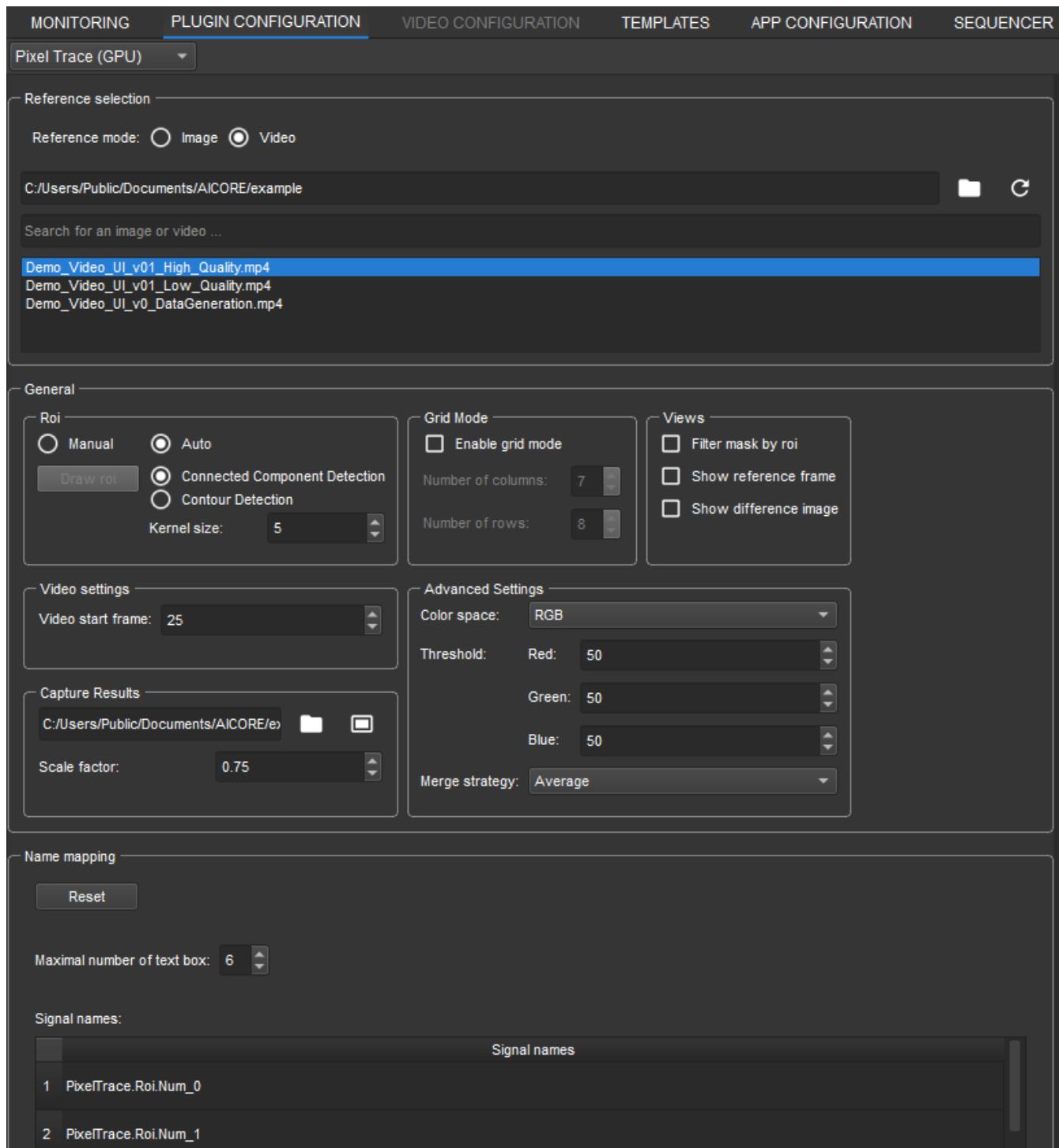
# 5.10. Pixel Trace

Pixel Trace is a plugin to compare images. You can use it, to find the difference of source and reference frame.

It calculates the difference between a frame from the source and a reference frame. The result is shown as mask above the source frame.

## 5.10.1. Pixel Trace Settings

You have control over multiple settings for Pixel Trace.



- **Reference selection:** Select the reference file, which should be compared to the current source

frame.

- **Reference mode:** Select between image and video mode. In image mode, you compare each frame of the source with the same reference image. If you use the video mode, the source and reference video are compared frame by frame.
- **Reference file directory:** Select the directory, where your reference files are located by writing the path in the line edit. Or you use the folder button to navigate to the desired directory with a file dialog.
- **Reference file:** After selecting the reference directory, the available references are listed in the user interface. The references are listed by type depending on video or image mode. You can search for references by using the search bar. Select the reference file by clicking.
- **General:** In this section different Pixel Trace Settings can be set to fine tune the comparison.
  - **Roi:** For Pixel Trace you can select between a **Manual** and an **Auto** roi mode to get more detailed results.
    - **Manual mode:** In manual mode the "Draw roi" button is enabled. After clicking this button, you can draw Pixel Trace bounding boxes on the player.
    - **Auto mode:** The auto mode can use two different algorithms to group different pixels and draw a bounding box around them.
  - **Grid Mode:** The grid mode divides the full player in multiple smaller elements and calculates the match and mismatch of these.
    - **Enable grid mode:** This check box enables the grid mode. If it's enabled, the player, the reference frame and the difference frame, will show the grid.
    - **Number of columns:** Select the number of columns in the grid.
    - **Number of rows:** Select the number of rows in the grid.
  - **Views:** In views, you can filter the result mask and show or hide extra widgets.
    - **Filter mask by roi:** With this check box, you can enable or disable a filter for the output mask. If filter is activated, the result difference mask will only show differences inside the "Pixel Trace" regions of interest.
    - **Show reference frame:** Here you can show or hide the current reference frame the source frame is compared to. The frame is displayed inside dock widget and can be placed all over the screen.
    - **Show difference frame:** You can show or hide the a dock widget, containing the by Pixel Trace calculated difference image.
  - **Advanced Settings:** In the advanced settings you can customize the difference calculation algorithm.
    - **Color Space:** Select the color space to perform the calculations. You can choose between None, Gray Scale, RGB or HSV. If selecting "None" the original input type is used and no threshold filter will be attached. Otherwise you can define thresholds for each channel to fit your requirements.
    - **Merge strategy:** The merge strategy can be used on multichannel frames. Then the three channels are merged into one channel by different methods. You can merge by minimum, average or maximum pixel value. If you select "None", no channel merging is

performed.

- **Video settings:** In video settings additional options for the video mode can be set. You can define the offset between source and reference video. The reference video starts at the frame of the source video, which is defined by the offset.
- **Capture Results:** The capture result gives you the ability to take a screenshot containing three images ordered vertically. The first one is the player with mask overlay. The second one is the reference frame and the last one is the raw difference mask. To take a screenshot define the capture directory by typing the path in the line edit and confirm with "Return" key or by clicking the folder button and selecting the directory. With the most right button you take a screenshot, which is named the following pattern: "PixelTrace\_YYYYMMDD\_hhmmss\_zzz.png". The scale factor resizes the screenshot before saving it to the requested directory.

From PROVEtech:TA, you can send commands to change the configurations of icon detection plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.10.2. Test Result Schema

After AI-Core compared the frames, the results will be shown in the test result table. In the following, all concrete components of detection results are listed:

- Match: The percentage of matching pixels between source and reference.
- Mismatch: The percentage of matching pixels between source and reference.

If bounding boxes are drawn, additionally the coordinates of the boundary are given:

- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.

For grid mode related results, the grid element details are also given:

- C: Column of the grid element.
- R: Row of the grid element.

## 5.11. Template Matching

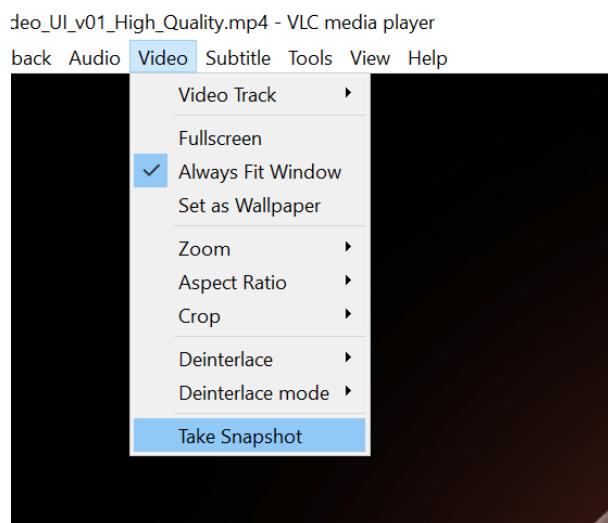
Template Matching is a plugin similar to Icon Detection, but it does not require prior training of a model. In this context, you will provide a list of images, referred to as templates, which contain a single icon or object that you want to detect in your input source. The plugin will then identify the matching icons in the provided input.

### 5.11.1. Recording Templates

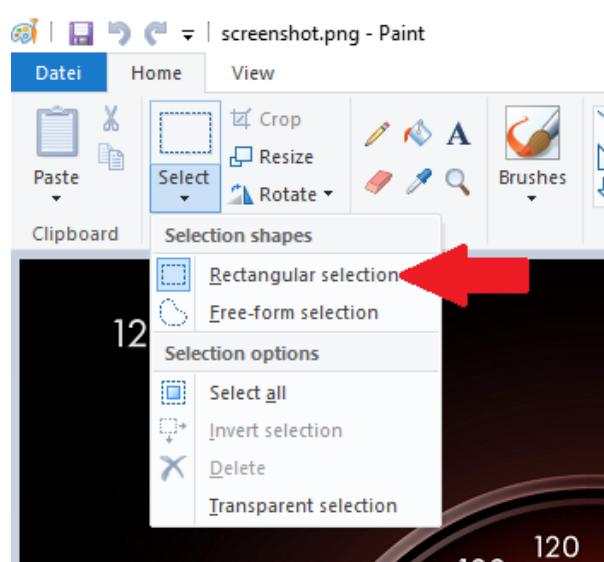
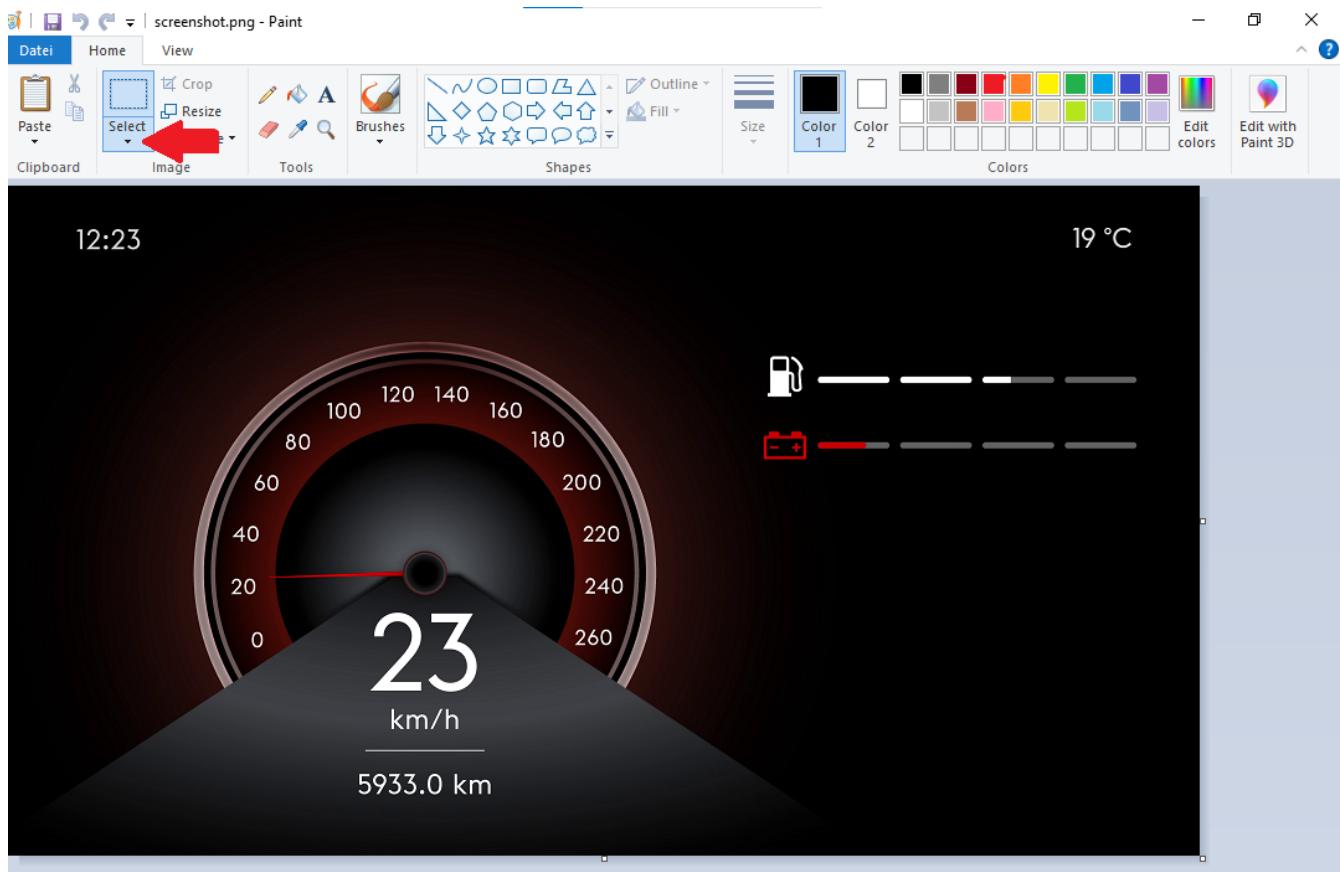
The plugin requires you to specify a folder path that contains a collection of images in commonly used formats, such as PNG and JPEG.

If you are using Windows and want to create the template images from a video, please do not create these images by taking screenshots of your screen, because the system embedded screenshot will change the original resolution. However, template matching plugin is sensitive to the resolution.

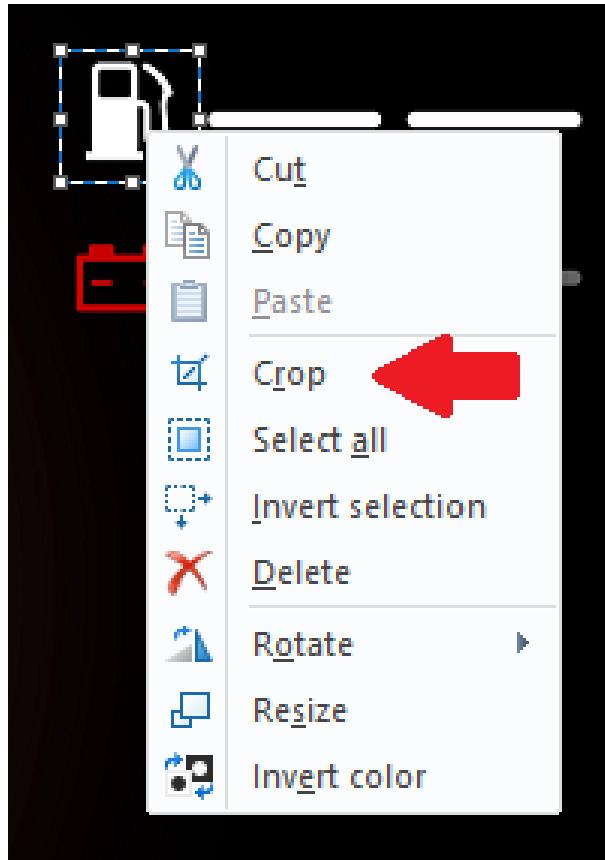
Instead, you should create the screenshot by VLC video player (Menubar/Video/Take Snapshot) to ensure that they are of the highest quality.



For cropping the screenshot to single icons, please use Microsoft Paint. Start by opening the application and loading an image that contains all the icons you want to use. Next, click on the "Select" button in the top toolbar and choose "Rectangular selection" from the dropdown menu.



After that, click and drag to draw a box around the icon you wish to keep. Once you have your selection, right-click inside the selected area and choose "Crop" from the context menu.



You should now see only the selected icon. Finally, save this image in your desired template folder. Linux users can use a similar program called GIMP to perform the same tasks.

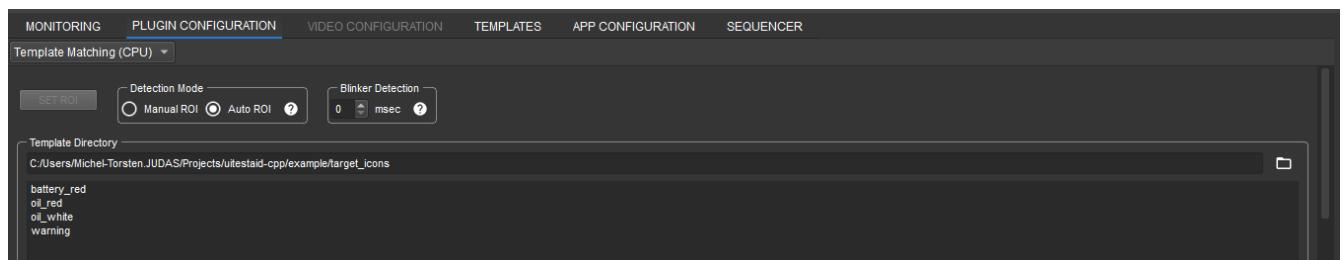
Example Template:



Notice: We strongly recommend using Microsoft Paint or GIMP instead of relying on screenshots taken with Windows. This is because screenshots may not capture the images at the desired resolution or quality, which can lead to poor results when using them in the plugin.

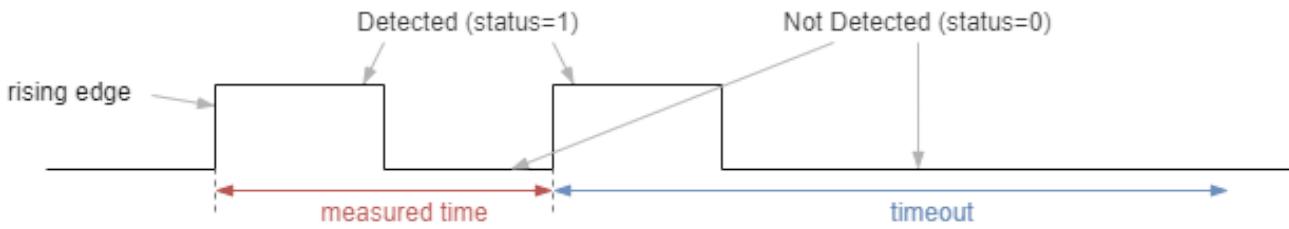
### 5.11.2. Template Matching Settings

You have control over multiple settings for template matching.

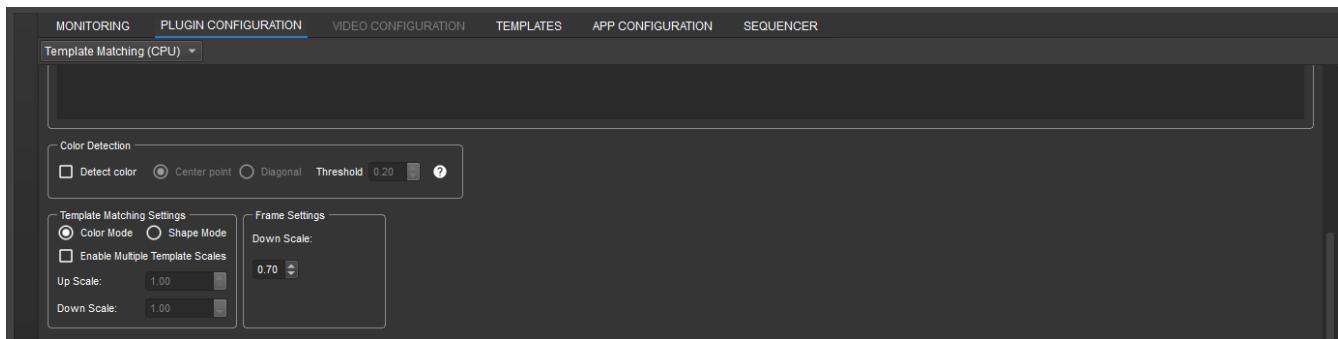


- **Auto/Manual ROI:** Switch between auto and manual template detection mode.
- **Blinker timeout (msec):** In many test case, the occurrence frequency of some blinking objects, such as LED indicator, is very important and needed to be detected. In manual mode, AI-Core

can also detect the blinking frequency. Blinker timeout defines the longest waiting time for the calculation of blinking frequency. If the region does not change for such a long time, the blinking frequency will be set to 0.



- **Template Directory:** Before the Template Matching plugin can begin detecting objects, you need to specify a template folder that contains all the template images you wish to use. You can do this by either typing the folder path directly into the field located beneath "Template Directory" or by clicking the folder icon, which will open a new window. From there, you can navigate to and select your desired folder. Once you have selected a folder, the names of the included template images will be displayed directly below the folder path



- **Detect color:** This option controls whether AI-Core should detect icon color. AI-Core detects color in HSV color space like the color detection plugin. For more details please refer to [color detection](#).
- **Template Matching Settings:** In this section, you will find a variety of adjustable values, allowing you to choose between Color Mode and Shape Mode. Color Mode is recommended if you have the same icon or object in different colors, as the plugin can distinguish between colors and accurately detect the correct icon that matches the specified color. On the other hand, Shape Mode is suitable if you do not have multiple templates of the same icon in varying colors. However, it is important to note that Shape Mode does not detect colors, which means that if you provide the same icon in different colors, the plugin may detect it multiple times. Additionally, by enabling "Multiple Template Scales," you can assist the plugin in locating your templates within the provided source. This feature generates both downscaled and upscaled versions of your templates for the algorithm to use, which is useful because sometimes the provided templates may be too large or too small for the plugin to detect effectively. You can adjust the scaling by modifying the values next to "Up Scale" and "Down Scale".
- **Frame Settings:** To further assist the plugin in locating your templates, you can also decrease the internal frame size. This adjustment is useful because there are times when the frame size does not match the size of the templates. By reducing the frame size, you provide an alternative option to using multiple template scales.
- **Name mapping:** The name map defines how the detection is shown in the signal tree in

PROVEtech:TA. After connected with PROVEtech:TA, the signal tree will be unchangeable. Therefore, you need to predefine the signal name mapping before the connection.

- **Maximal object number:** defines the maximal number of objects from each class which can be shown in PROVEtech:TA. If AI-Core detects more objects from the same class than the maximum, the detection result will be discarded. If you want to customize your own signal names, you can define them in this configuration group. Especially, when you want to switch among different class sets or projects, you need to set the name map to "Customized", because the class names will be changed. The old signal names cannot be reused, which will lead connection failure between AI-Core and PROVEtech:TA.

From PROVEtech:TA, you can send commands to change the configurations of template matching plugin remotely. The concrete commands are listed in [supported commands](#).

### 5.11.3. Test Result Schema

After AI-Core detects the templates, the results will be shown in the test result table. In the following, all concrete components of detection results are listed:

- L: Left boundary position of the box in pixel.
- T: Top boundary position of the box in pixel.
- R: Right boundary position of the box in pixel.
- B: Bottom boundary position of the box in pixel.
- Content: The detected template name will be shown behind the coordinates. In the manual mode, if there is no object detected in the drawn ROI, it will show "Not Found".
- Color: In the brackets at the end, AI-Core will show the detected color, if you activated the color detection.

Following results are only available in the test automation tool:

- ColorId: The ID of detected color. 0,1,2 are white, gray, black respectively. Depending on the color definition, the ID increases with the color number. See the details from the following [color definition table](#).
- ColorHue, ColorSaturation, ColorValue: The hsv component values of the detected color.

# Chapter 6. Sequencer

The sequencer feature is like a lightweight version of an integrated test automation tool. It enables the ability to use detection results in AI-Core to trigger different commands.

The screenshot shows the 'SEQUENCER' tab selected in a top navigation bar. Below it, there's a 'Configurations' section with a 'Sampling interval' input set to 5 ms. The main area is titled 'Sequencer definition' and shows a file path 'C:/Projects/uitestaid-cpp/libs/sequencer/example/example.seq'. A table below lists triggers with their states and actions:

Trigger	State	Action
<input checked="" type="checkbox"/> t_line	Active	line_detect

The sequencer can be configured in test mode in the sequencer plugin. Here a sampling interval can be set in milliseconds. This interval defines how often the detection results are polled and the statement is estimated.

Below you can choose a definition file by setting the path. If you click on the folder button, a file selection menu pops up, where you can navigate to the definition file. If the file is valid, the table will show the current sequencer definition. If the file is changed, you can reload the file by pressing the reload button. Each trigger is listed with its current state and the connected action. By clicking the checkboxes, you can enable or disable a trigger. The state shows, if the trigger is currently active, inactive, deactivated or invalid.

## 6.1. Sequencer Definition

A sequencer is defined in a sequencer definition file. These are marked with the ".seq" extension and follow the JSON syntax. The file is separated in two main keys: triggers and actions.

### 6.1.1. Triggers

A trigger defines the conditions, which must be fulfilled to initialize an action. The following example shows, how to implement a trigger.

```
{  
    "triggers": [ // some comment  
        {  
            "id": "t_line",  
            "conditions": {  
                "c1": "Icon.warning.Num_0.Probability > 0.9",  
                "c2": "Icon.warning.Num_0.Status == 1"  
            },  
            "statement": "(c1 && c2) || (c1 ^ ( ! c2))",  
            "action-id": "line_detect",  
            "repeat": true,  
            "results": [  
                {
```

```

        "name": "var_1",
        "type": "signal",
        "value": "Icon.warning.Num_0.BoxLeft"
    },
    {
        "name": "var_2",
        "type": "signal",
        "value": "Icon.warning.Num_0.BoxTop"
    },
    {
        "name": "var_3",
        "type": "signal",
        "value": "Icon.warning.Num_0.BoxRight"
    },
    {
        "name": "var_4",
        "type": "signal",
        "value": "Icon.warning.Num_0.BoxBottom"
    },
    {
        "name": "var_5",
        "type": "string",
        "value": "'C:/myPath'"
    },
    {
        "name": "var_6",
        "type": "num",
        "value": 99
    },
    {
        "name": "var_7",
        "type": "bool",
        "value": true
    }
]
}
],
"actions": [...]
}

```

Each trigger is an object in a list. The object must contain an identifier (id). The statement contains a string, which connects the conditions with boolean operators like AND (&&), OR (||), XOR (^) and NOT (!). To connect a trigger to an action, you have to define the action-id, which contains the name of the action. The optional “repeat” key sets, if the trigger behavior. If it’s set to true, the trigger can activate the action continuously, else it only triggers one time and then goes inactive. On default, the trigger only activates the action one time. To use detection results, you have to define them inside the results key. It’s defined as an array of objects, which define a variable.

There are 4 types of variables, which can be defined. Variables of type “signal” refer to an internal plugin signal of AI-Core. In this way, detection results can be used. When the trigger activates an

event, a snapshot of the signal value is stored and send to the action. Other possible types are “bool”, “num” and “string”. Strings are special in definition. To match the JSON and AI-Core syntax they are embraced in double and single quotes.

### 6.1.2. Actions

An action describes which commands has to be executed. The possible operations are listed in [Supported Commands](#).

```
{
  "triggers": [...],
  "actions": [
    {
      "id": "line_detect",
      "preprocessing": {
        "input_1": "(var_1 + var_3)/2 - 200",
        "input_2": "(var_2 + var_4)/2 - 200"
      },
      "steps": [
        "LineDetection roiwithid 100 {input_1} {input_2} {var_3} {var_4}"
      ]
    },
    {
      "id": "action_2",
      "steps": [
        "JumpTo 10"
      ]
    }
  ]
}
```

The actions are like the triggers defined as objects inside a list. They also must contain an id, which the trigger uses to connect to the action. The “preprocessing” is optional. Here you can define new variables, which combine detection results. At the moment, only basic math operations are supported. The different commands, which are executed sequentially, are strings in a list. The commands are the same like for test automation tools. You can define the command arguments directly in the string or use placeholders to use detection or preprocessing results. A placeholder is defined by a braces around the variable id. The detection result must then be defined inside the connected trigger or in the preprocessing.

### 6.1.3. Summary

*Table 1. Sequencer Definition Options*

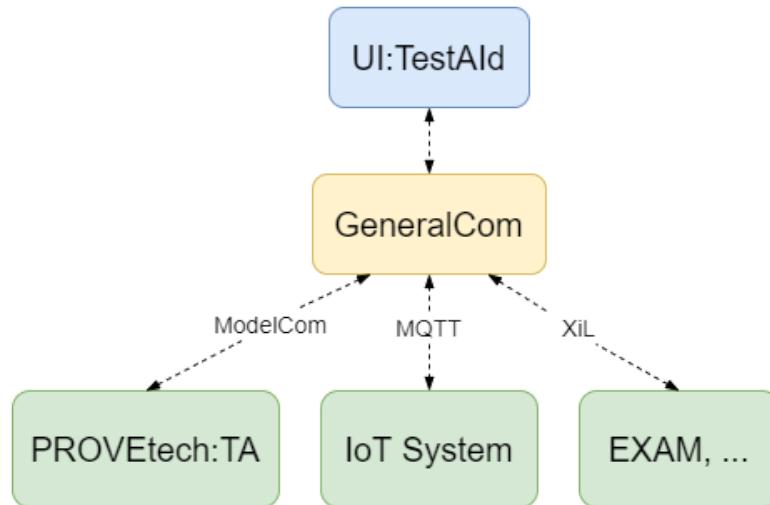
Group	Key	Type	Require d	Description
Trigger	id	string	Yes	Name of the trigger.
	conditions	list	Yes	Conditions to trigger.

<b>Group</b>	<b>Key</b>	<b>Type</b>	<b>Required</b>	<b>Description</b>
	statement	string	Yes	Connect conditions with boolean operators.
	action-id	string	Yes	Name of the action to connect the trigger to.
	repeat	bool	No	Trigger behavior. Default: One time trigger (false)
	results	list	No	Detection results, the action should use to execute a command.
Action	id	string	Yes	Name of the action.
	preprocessing	list	No	Processing of detection results, before executing commands.
	steps	list	Yes	List of the commands to execute, when triggered.

# Chapter 7. Communication

AI-Core can not only work on test users' systems standalone, but also cooperate with other test automation platform such as PROVEtech:TA. The plugin "GeneralCom" is responsible for the communication between AI-Core and other systems.

AI-Core supports different communication protocols. They includes [ModelCom](#) (with PROVEtech:TA), [MQTT](#), and ASAM [XIL](#).



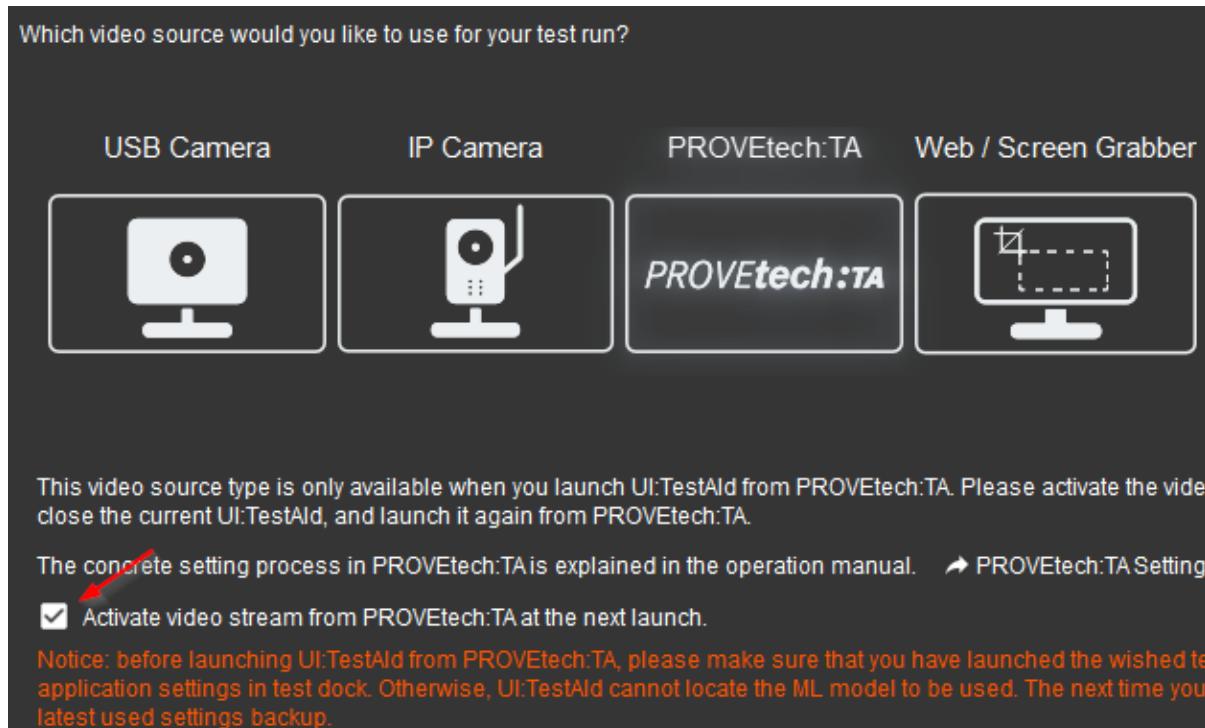
Before testing, the first step is Configuration of AI-Core.

## 7.1. Configuration of AI-Core

Before building the communication with a testbench, you always need to confirm the configuration of AI-Core:

### 1. Get test project ready

Make sure that the video source is working. The input video stream of AI-Core can not only come from video files, camera capture, but also be delivered from PROVEtech:TA. If you want to use the transmitted video stream from PROVEtech:TA, you should activate the PROVEtech:TA option.



Since the video stream can only be transmitted when PROVEtech:TA launches AI-Core, to continue the test setting, you should temporarily select another video source. It is recommended to use the training video.

Select a machine learning model which you first want to use for the testing. Later in the test, you can still send command to switch other models.

### 2. Set communication configuration

In all testbench, signals are organized in form of a signal tree. The signal tree always starts with a root node. To distinguish between multiple AI-Core instances, you should first define an unique root node name of signal tree for each instance. You can configure the root node name in Settings > Test Mode > Communication > Root Node Name.



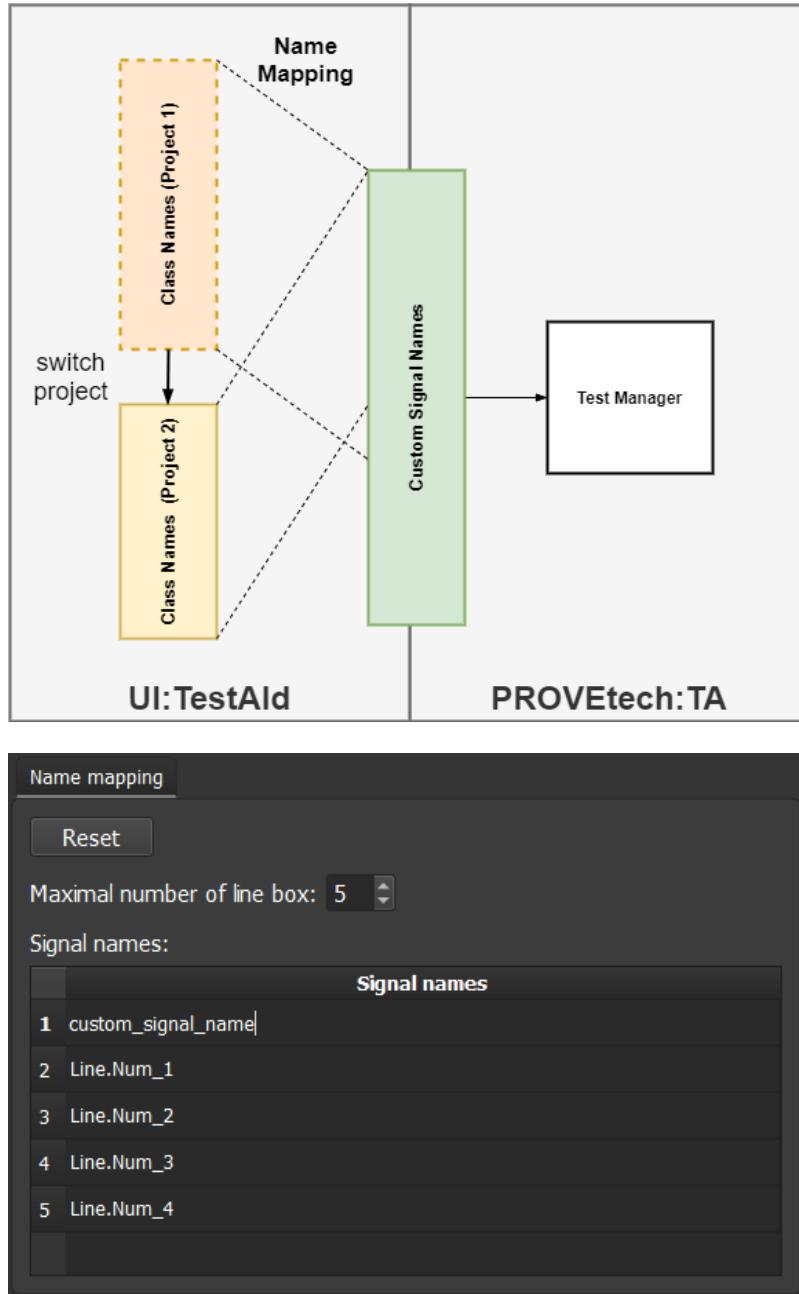
Under the root node, different plugins have their own signal branches. AI-Core uses symbol '.' to separate levels in the tree. For example, icon detection plugin will generate signals, such as

AICORE.Icon.class\_name.Num\_0.Status.

### 3. Check name mapping (optional)

The control and detection results are transferred between AI-Core and PROVEtech:TA via signals. All signals must be registered in PROVEtech:TA during the connection building. During the test, it is not allowed to change the signals anymore, unless you reload the model in PROVEtech:TA again.

By default, the signal names are automatically generated by each plugin. If you want to rename the signals or reconstruct the signal tree, you can customize it in each plugin setting with help of **name mapping**.



You need to first set the maximal number of reserved signals for the detection type. The number of custom signals should be sufficient so that no output will be missing.

In the table, you can preview the signal names. By default, for example in line detection plugin

settings, the signal name is in form of "Line.Num\_XX". You can also edit your own preferred names in the table. Press "Reset" button, then the default signal name will be recovered. In PROVEtech:TA, the final complete signal names will be attached to the root node, such as "AICORE.Line.Num\_XX".

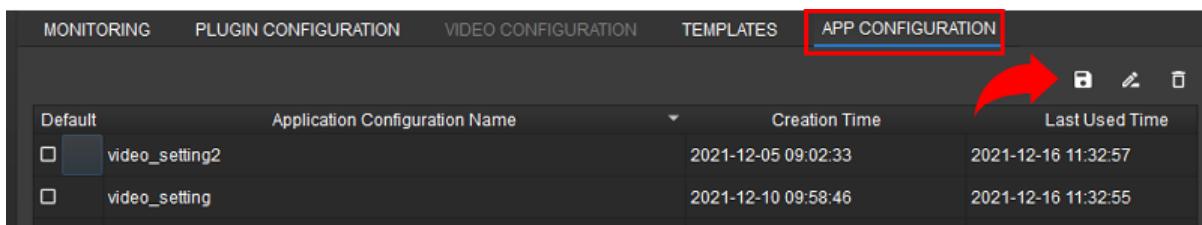
#### 4. Backup application configuration

The testing often works with a specific conditions saved in form of application configuration file. For example, if the video stream comes from a camera. The light condition, camera parameters should to the best be kept the same. If the testing needs to detect texts, you might want to assign a specific detection languages. AI-Core allows you to take a snapshot of the whole application configurations.

It includes:

- generic configurations: language, theme, ...
- configurations of each video sources: camera parameters, web renderer url, ...
- plugin configurations
- communication configurations: address and port of ASAM XIL, sampling rate, ...
- signal name mapping and root node name

After you setup all the configurations above, to "APP CONFIGURATION" in the test tool widget, click the "Save" button, and give a name for it.

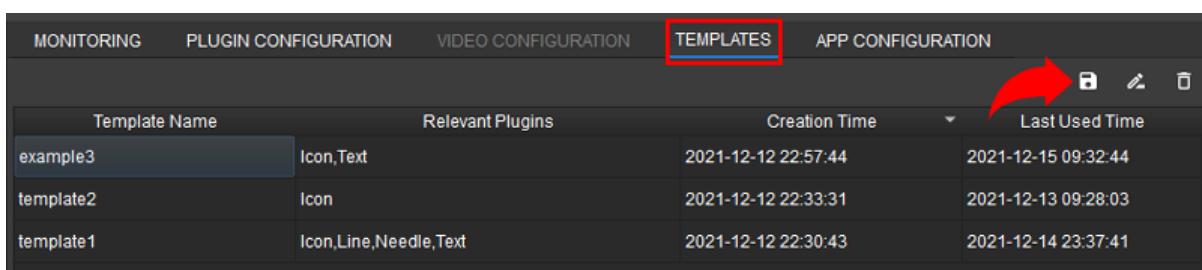


Default	Application Configuration Name	Creation Time	Last Used Time
<input type="checkbox"/>	video_setting2	2021-12-05 09:02:33	2021-12-16 11:32:57
<input type="checkbox"/>	video_setting	2021-12-10 09:58:46	2021-12-16 11:32:55

AI-Core will then copy the current application configuration file (.appcfg) to test\_project\_directory/AppConfigs/backup\_name.appcfg. Later, you can recover these saved configurations through double clicking the item in the table or sending commands. The concrete operations can be found in section [Supported Commands](#).

#### 5. Save templates

For some specific projects, you may want to save specific interesting regions that needs to be detected in the test. In this case, you can save these regions to a template by clicking the "Save" button in "Templates". Later, you can recover these detection boxes at the next time by loading the template.



Template Name	Relevant Plugins	Creation Time	Last Used Time
example3	Icon,Text	2021-12-12 22:57:44	2021-12-15 09:32:44
template2	Icon	2021-12-12 22:33:31	2021-12-13 09:28:03
template1	Icon,Line,Needle,Text	2021-12-12 22:30:43	2021-12-14 23:37:41

## **6. Close AI-Core and open testbench**

After all configurations, you should always close AI-Core, since the communication is initiated by the testbench. Then, you can open your testbench, e.g. PROVEtech:TA, and launch AI-Core with required parameters. The launching steps vary greatly depending on the individual testbench. In the following section, you can read more details.

## **7. Continue setting testbench**

- [PROVEtech:TA](#)
- [MQTT](#)
- [ASAM XIL](#)

## 7.2. PROVEtech:TA

### 7.2.1. Configuration in TA

PROVEtech:TA is a powerful software application developed by Akkodis Germany Consulting GmbH for the control and automation of test systems. It is tailored to the complex industry requirements and able to join the testing of Hardware/Software/Model-in-the-Loop.

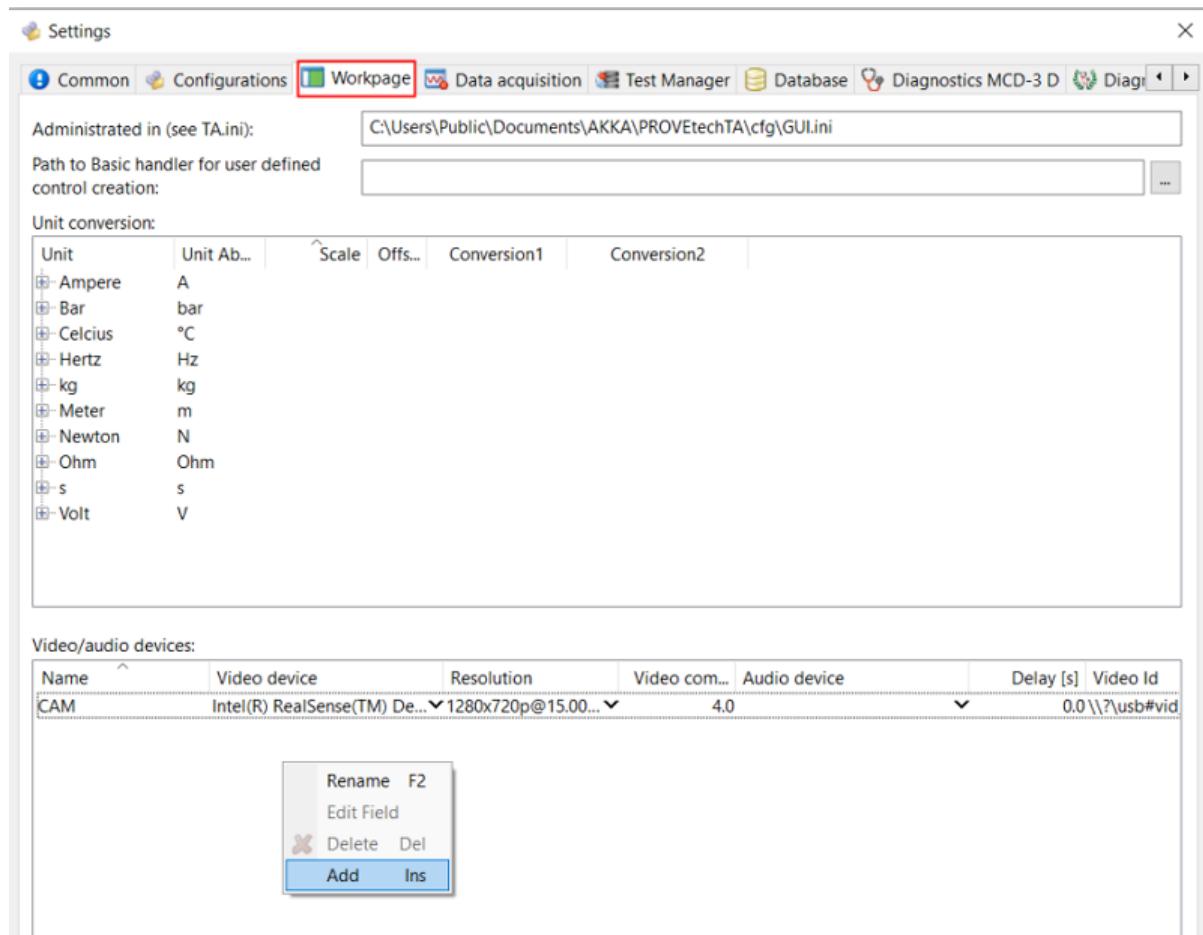
As an extension, AI-Core can cooperate with PROVEtech:TA to extract test information from raw video stream. With the help of AI-Core, the testing procedure will be completely automatized even with complex graphical user interface.

In PROVEtech:TA, please first open settings dialog by clicking the button .

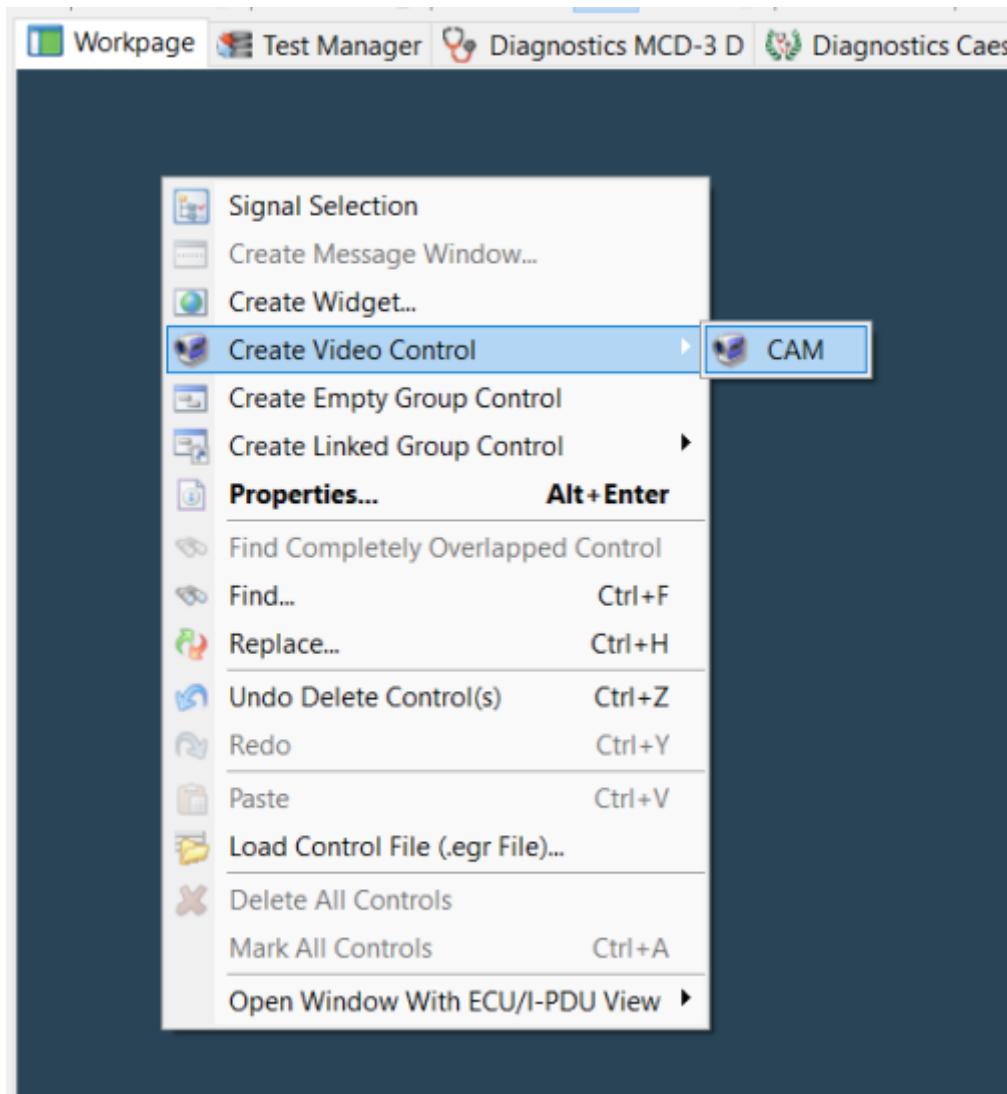
#### 1. Set Video Control

If you want to read the test video stream from PROVEtech:TA intermediately and transfer it to AI-Core, you should first connect PROVEtech:TA with the video device.

To setup a new video device, you should go to Settings > Workpage > Video > audio devices (shown in the following image). Right click the empty space and select "Add" option. A new row will appear in the table. Then, rename the device, and select the wanted video device and resolution. After that, save and close the dialog.



Optionally, if you want to check the camera signal, right click the empty space of the workpage, select option "Create Video Control", then the video viewer will be shown on the workpage.



## 2. Enable communication to the model and automatic model loading

Enable communication to the model

Load and start model automatically (recommended)

Make sure that you have enabled the communication to the model and let PROVEtech:TA launch models automatically.

## 3. Set the executable file path

Set the executable file path (by default: C:/Program Files/AICORE/AICORE.exe) in setting option "Path to RE.exe/AICORE.exe on target computer".

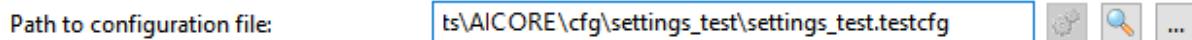
Path to RE.exe/AICORE.exe on target computer:

## 4. Set path to configuration file

Add the configuration file path of the test project to the field

**Notice:** instead of settings.ini in previous versions, AI-Core (UI:TestAId) 2022 starts using test configuration file to manage and save machine learning models, testing setting backups and

templates.



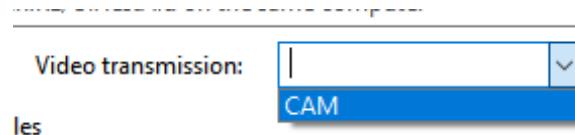
## 5. Set the connection timeout

Sometimes, AI-Core will take a while to launch the test project, such as building connection, loading models. To prevent PROVEtech:TA from terminating the connection, please set the connection timeout to a larger number. For example, 10000 milliseconds.

Connection timeout:  ms

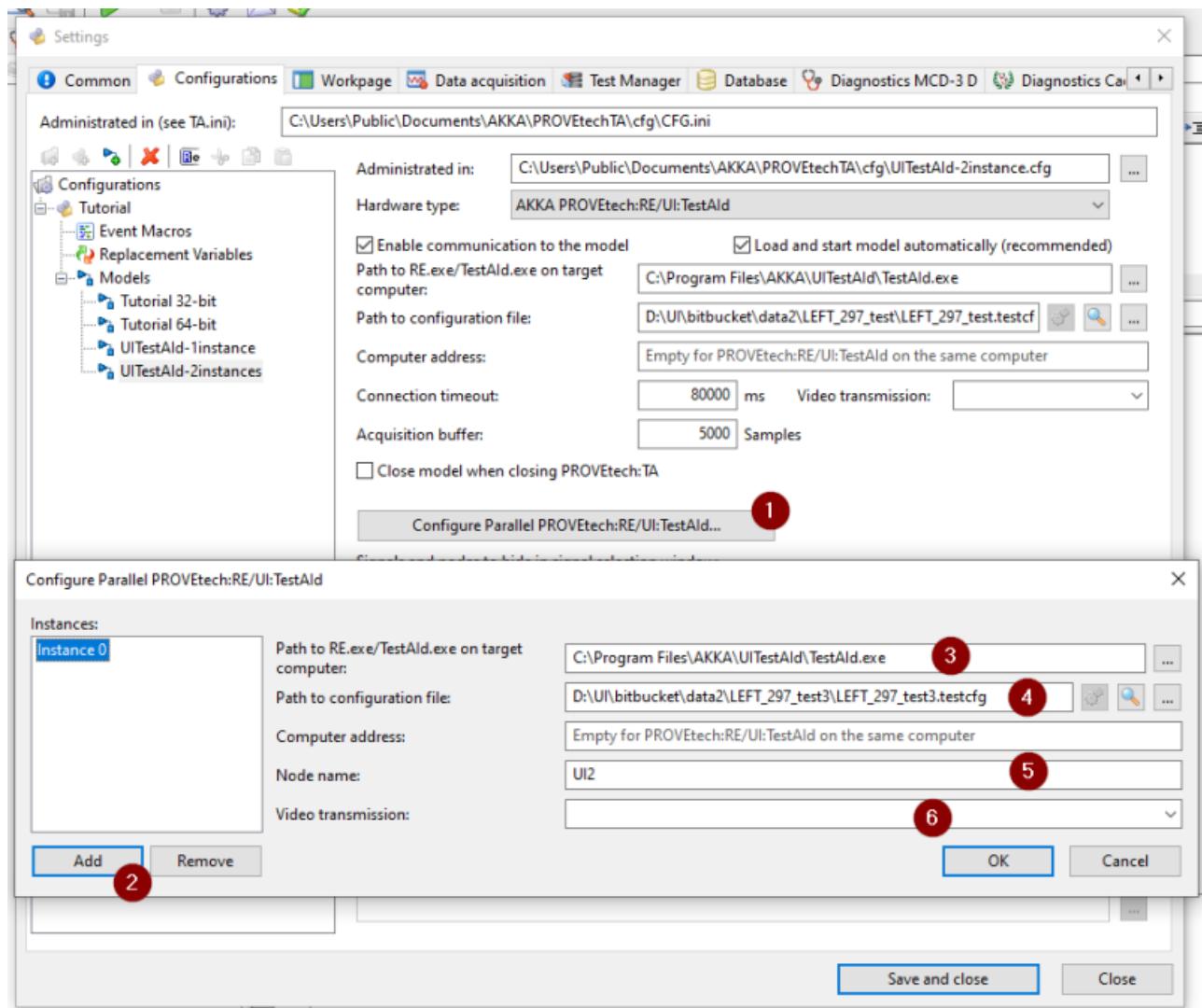
## 6. Set video transmission

Since PROVEtech:TA supports multiple video source, you need to select one of them that you want to transmit AI-Core.



## 7. Set parallel AI-Core

If you have already more than one parallel test hardware, for example another AI-Core instance, RE.exe application, or Veristand, which is required to be loaded for the testing and occupy the settings, you can configure them as parallel models by clicking button "Configure Parallel PROVEtech:RE/AI-Core...".

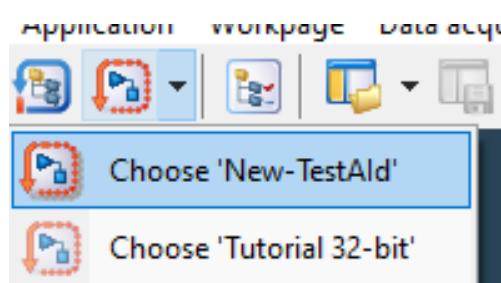


In the dialog, you should create wished number of instances. Then, configure the rest parameters as previously mentioned. If you want to launch two AI-Core instances, you should assign an individual root node name for each of them. If you have multiple AI-Core instances. Otherwise, the signal will share the same signal root node and overlap each other. The assigned root node name should be the same as you defined in AI-Core.

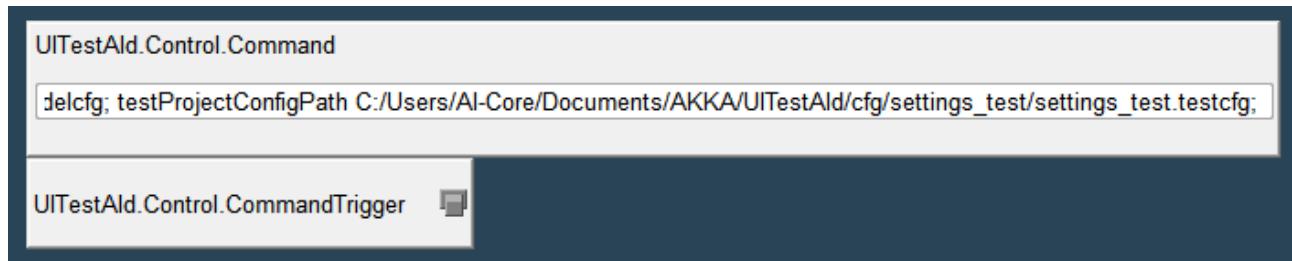
## 7.2.2. Test

### 1. Start Testing

The entry point of testing is always from PROVEtech:TA. After the configuration, you should click button  , and select the wished model and start it. PROVEtech:TA will launch AI-Core with the settings file. If the connection is successful, AI-Core will load the specified detection model and start testing.

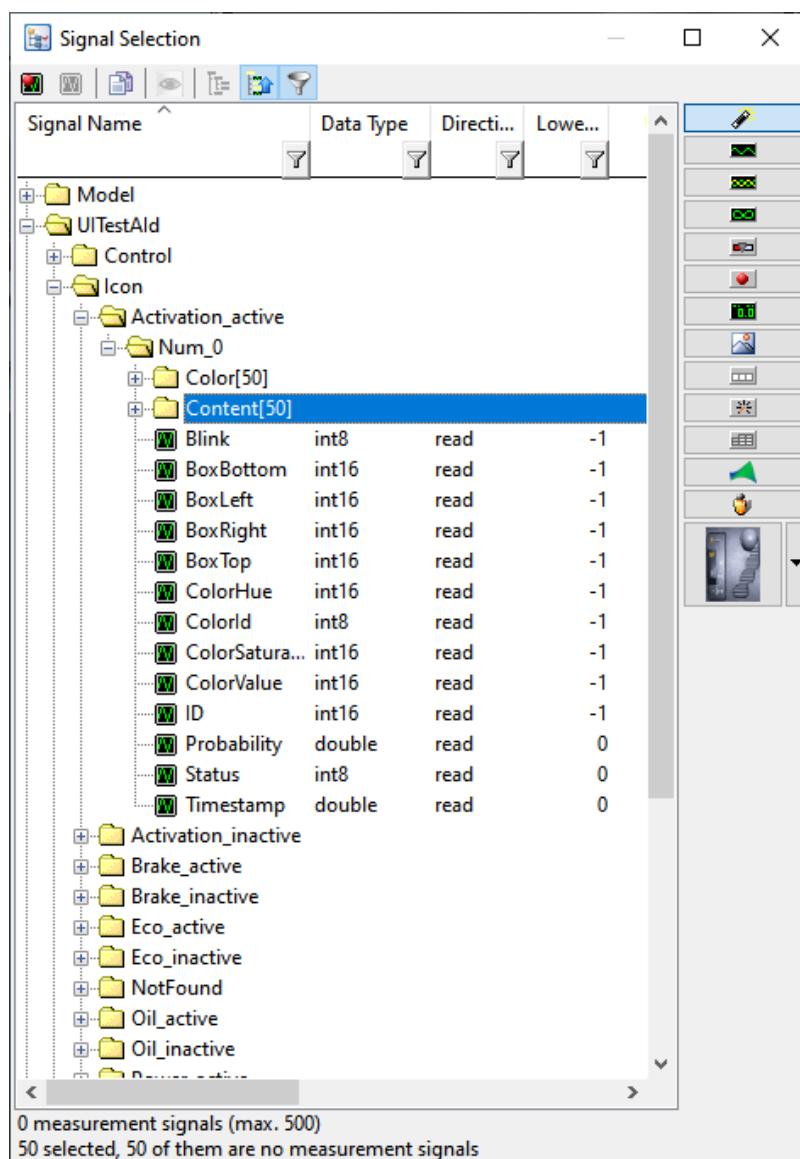


If AI-Core is successfully loaded, you will check the loading information in the command signal. It includes the loaded machine learning models, test project configuration file and so on.



## 2. Check signals

Then, click button , the signal selection dialog will pop up. You will see the the signal tree that you defined in AI-Core previously.



Drag one or more signal nodes onto the workpage. The signals will be visualized according to the signal data type.

## 3. Switch Detection Models

From AI-Core 2024 SE, you are allowed to switch models dynamically without reloading all PROVEtech:TA models.

In your test script, you can call the model switching in the following way:

```
System.ModifyModelNodeConfig("Model", "SwitchModel IconDetection  
'{model_directory}/model.modelcfg'", 45)
```

System.ModifyModelNodeConfig is a function newly supported since PROVEtech:TA 2023 SP1. It allows the user only reload one single test model and refresh the signal tree during the test.

- The first parameter "Model" indicates the root node name of AI-Core instance. If you launch AI-Core as the primary model, the node name is always "Model". If you run AI-Core as a parallel model, you should change it to your customized node name.
- The second parameter is a command string, which specifies the modelcfg path to be switched. The command signal "UITestAId.Control.Command" is no longer responsible for switching AI-Core models.
- The third parameter defines the timeout in seconds. PROVEtech:TA will wait for AI-Core until the model is reloaded or timeout exceeds.

You are not limited to using only one model. If your system capability allows it, multiple models can be loaded by extending the command above:

```
SwitchModel IconDetection '{model_directory}/model.modelcfg' IconDetection  
'{model_directory}/model.modelcfg'
```

Additionally, you can give each model config file a name by using the following command instead:

```
SwitchModel IconDetection 'modelName' '{model_directory}/model.modelcfg'
```

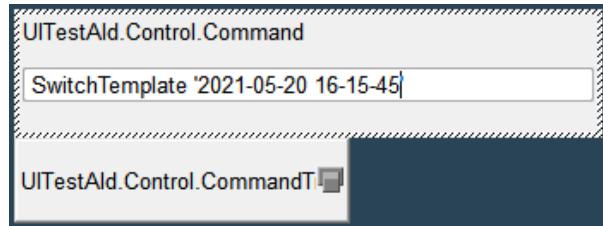
Providing a model name is optional and can be neglected.

When PROVEtech:TA detects the disconnection from AI-Core, please reload the model again to update the signal tree.

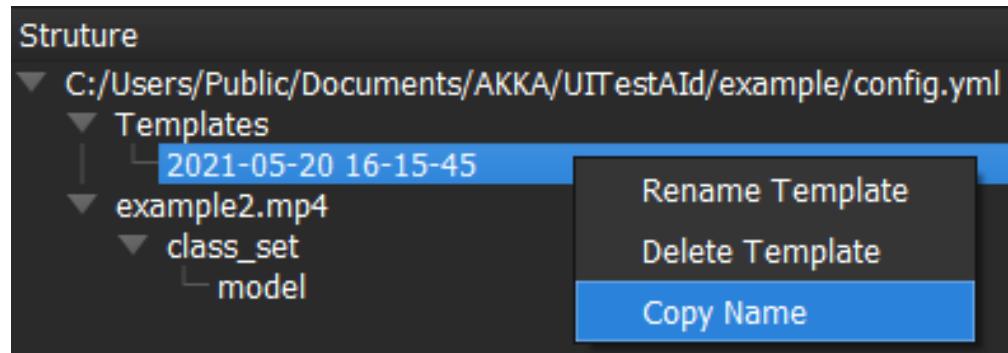
#### 4. Switch Templates

Similar to model switching, you are also allowed to send commands in PROVEtech:TA to let UI:Test load saved templates. You need to set the command code and then activate the trigger. Then, AI-Core will search the template and load it.

```
SwitchTemplate 'template_name'
```



Small trick: You can right click the template name in AI-Core and select "Copy name" to access the name quickly.



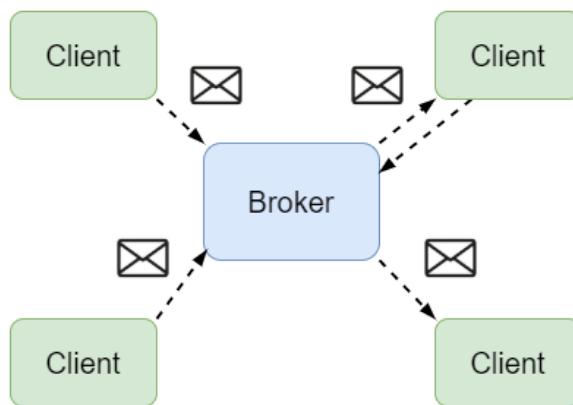
## 5. More supported commands

Besides of the commands mentioned above, there are more commands to control the camera parameters, play, and pause the test. Details are listed in section [Supported Commands](#).

## 7.3. MQTT

Message Queuing Telemetry Transport (MQTT) is an open OASIS and ISO standard (ISO/IEC 20922) lightweight, publish-subscribe network protocol that transports messages between devices. The protocol runs over TCP/IP or any MQTT-supported network protocol that provides ordered, lossless, bi-directional connections.

In a MQTT network, there are two entities: a message broker (obligatory) and a number of clients (optional). The MQTT broker is a server that receives all messages from the clients and then routes the messages to the appropriate destination clients. MQTT clients can connect the broker and communicate with each other by publishing and subscribing messages. AI-Core supports publishing detection results and subscribing commands via MQTT messages.



### 1. Install MQTT broker

To build a test based on MQTT communication, you first need to download and install a MQTT broker. A recommended version is Eclipse Mosquitto, which is an open source message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. You can download the binary package in link. Follow the installation steps, you will have a folder called "mosquitto". In this folder, you will find several files such as "mosquitto.exe", "mosquitto\_pub.exe", and "mosquitto\_sub.exe".

### 2. Modify MQTT configuration file

Configure the configuration file with the name "mosquitto.conf". It defines the bind address, communication port, password, and so on. In default, the broker uses port 1883. It is typically for uncertificated connection. Alternatively, you can also enable SSL/TLS support by using port 8883. More details are described in the configuration files.

### 3. Start MQTT broker

Once you finish the configuration, you can now start the broker by running the following code in your console.

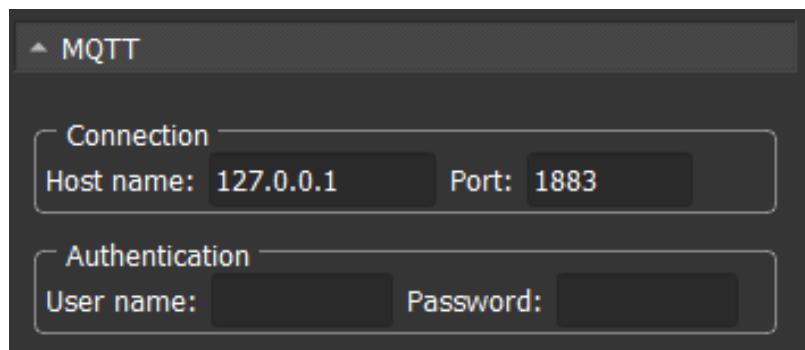
```
./mosquitto -v -c mosquitto.conf
```

If everything is fine, you should see the following information

```
1603733173: mosquitto version 1.6.11 starting
1603733173: Config loaded from mosquitto.conf
1603733173: Opening ipv6 listen socket on port 1883.
1603733173: Opening ipv4 listen socket on port 1883.
1603733173: mosquitto version 1.6.11 running
```

#### 4. Set AI-Core

Then, go to Settings > Test Mode > Protocol Type > MQTT. Check MQTT connection configuration, you should make sure that the parameters: host name, port, user name, and password are identical to the settings in mosquitto.conf.



#### 5. Launch test

Start a command line, and type

```
{AICORE_installation_directory}/AICORE.exe --conn MQTT --port 1883
{test_project_directory}/{test_project_name}.testcfg
```

#### 6. Subscribe signals

For each detection type, AI-Core will create different topics. You can subscribe specific topic according to your own requirement.

For example, you can open another console. Type following command,

```
./mosquitto_sub.exe -t 'AICORE/Text'
```

The messages of text detection results will be subscribed and displayed. Messages are packaged in JSON format. Different detection types have their own attributes. A result example of text detection are shown below:

```
{
  "boxBottom": 345,
  "boxLeft": 616,
  "boxRight": 796,
  "boxTop": 216,
  "content": "Hello World",
```

```
"id": 0,  
"timestamp": 67  
}
```

All topics are list here:

Detection Type	Topic
Icon Detection	AICORE/Icon
Text Detection	AICORE/Text
Needle Detection	AICORE/Needle
Line Detection	AICORE/Line

To subscribe all detection topics, you can use the following command:

```
./mosquitto_sub.exe -t '#'
```

## 7.4. XIL

### 7.4.1. Overview of XIL

The [XIL standard](#) is developed by ASAM to standardize the communication between test automation software and X-in-the-loop testbenches. It supports testbenches at all stages in the software development process – such as MIL1, SIL2, HIL3.

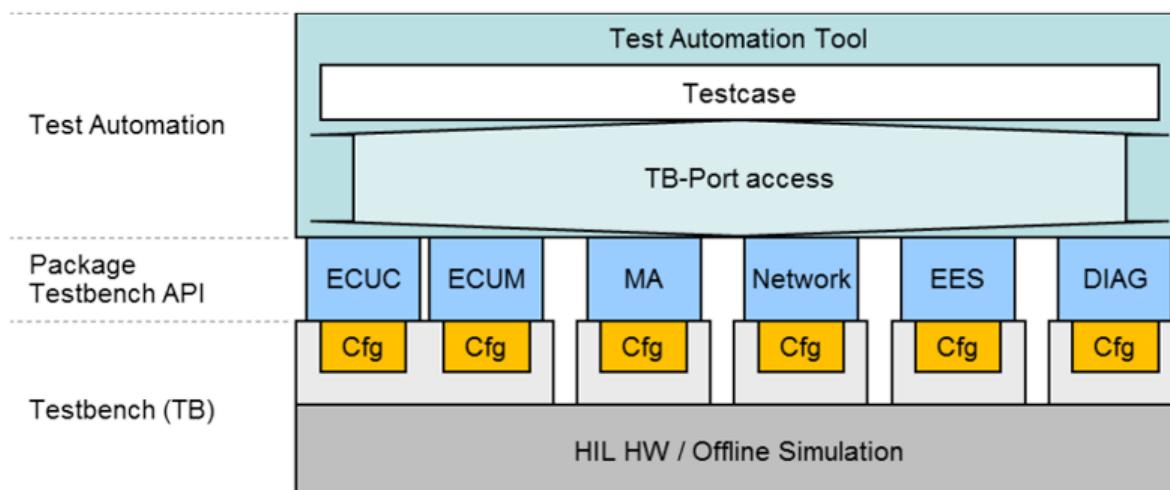
The application of the XIL standard allows users to combine the software and hardware tools independent of the vendor, thus enabling the reuse of the existing programmed tests for subsequent development stages (e.g. reuse of test cases for function models in software tests at a HIL simulator). This results in reduced costs for test programming and reduced training costs for testers.

ASAM XIL was developed to allow to exchange combinations of test automation software and test hardware. The standard consists of two layers:

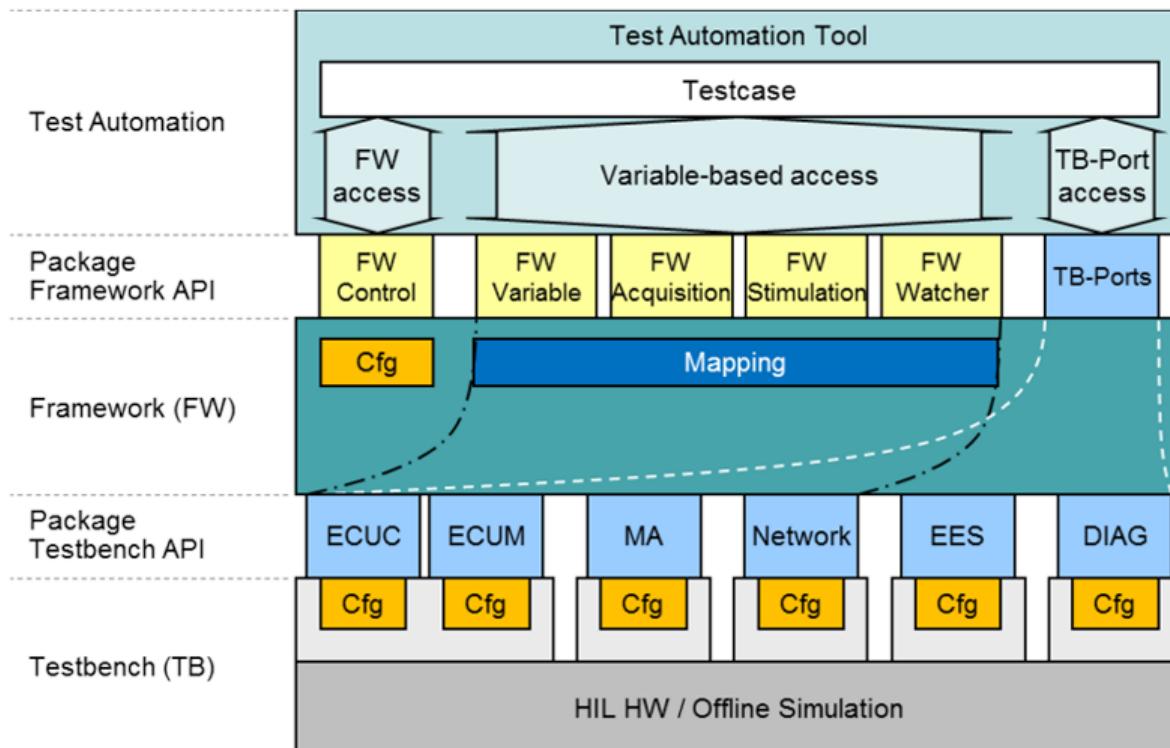
- Testbench API: This API separates the test hardware from the test software and allow a standardized access to the Hardware.
- Framework API: This API allows the exchange of the test automation software by a standardized access point for test cases, so a decoupling of test cases from real and virtual test systems takes place.

A typical XIL testbench consists of a system under test, which is a coupled system of ECUs, hardware components, and simulated parts of the system, depicted by the dark grey box at the bottom. The interaction between these parts is managed by different tools of different vendors, shown in the light grey boxes. These tools serve different purposes, such as accessing ECUs, or offer capabilities to interface bus systems, like CAN. The simulated parts might be executed by simulation tools or on the basis of compiled code. These tools have their proprietary means for configuration (orange blocks).

ASAM XIL has standardized interfaces to different types of tools by means of specific Testbench ports in a Testbench API. Based on these testbench port interfaces, a test automation tool could utilize a standardized access to variables and signals on these ports without dealing with proprietary details of different vendors and their tools.



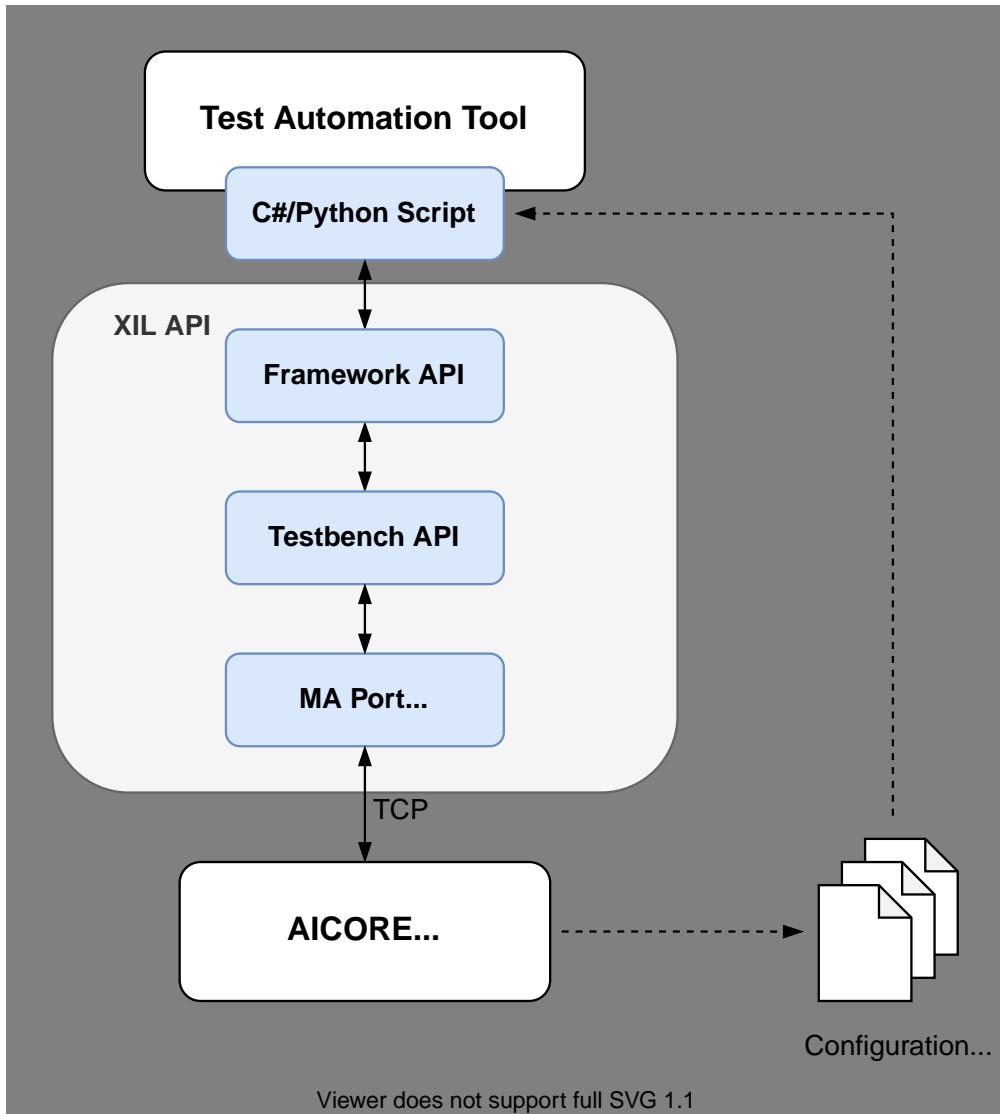
From XIL API 2.0, Framework API has been introduced to solve the problem that test developers have to deal with port-specific addresses and data types. Framework API offers an abstract object-oriented access to variables on Testbench ports with the help of variable mapping. According to some testbench specific configuration within the framework, different ports can be started and shutdown in a specified order.



You can either use Testbench or Framework API to access AI-Core. However, it is recommended to use Framework API to perform the signal reading and application controlling since it is much more abstract and easier to use. Before starting test with AI-Core and its XIL API, you should first [download](#) and install ASAM XIL Standard Assemblies in your computer.

#### 7.4.2. AI-Core XIL Implementation

AI-Core is a visual analysis tool, which is responsible for extracting information from raw video stream and send measured data to the test automation tool. Hence, it can be treated as a simulated hardware sensor under the concept of ASAM XIL. To enable these XIL-supported test automation tools to access AI-Core, AI-Core developed an XIL-based application interface. It includes implemented Testbench, Framework APIs, and a Model Access Port (MA Port). The MA Port provides access to the AI-Core read and write parameters, capture and generate signals.



Viewer does not support full SVG 1.1

The ASAM XIL applies a set of configuration files to define items such as test quantities, units, testbench ports etc. The following table lists all configuration files used for AI-Core XIL API.

- **AICORE\_ImplementationManifest.imf:**

provides information for locating and instance creation of the vendor specific Testbench and/or Framework class. This file is always located in the folder `\%PROGRAMDATA%\%ASAM\XIL\Implementation`.

- **AICOREFrameworkConfiguration.xml:**

The framework configuration file is the central configuration file. It references a set of mapping files and a set of port configuration files.

- **AICOREFrameworkMapping.xml:**

The mapping files particularly contain information on the available framework variables, e.g. data types, unit definitions, associations to specific testbench port labels and conversions. Through this file, the framework, testbench, and MA Port can create required signal variables and then the detected result can be read and written during the test via them.

- **AICOREMAPortConfiguration.xml:**

The port configuration files encapsulate all the port specific configuration settings required to setup the testbench ports, such as the address and port number of the TCP connection between MA Port and AI-Core. In the use case of multiple instances, the testbench will create one

individual MA Port for each AI-Core instance. In order to distinguish them, the port configuration file also defines the port ID in form of AI-Core\_MAPort\_{instance index}.

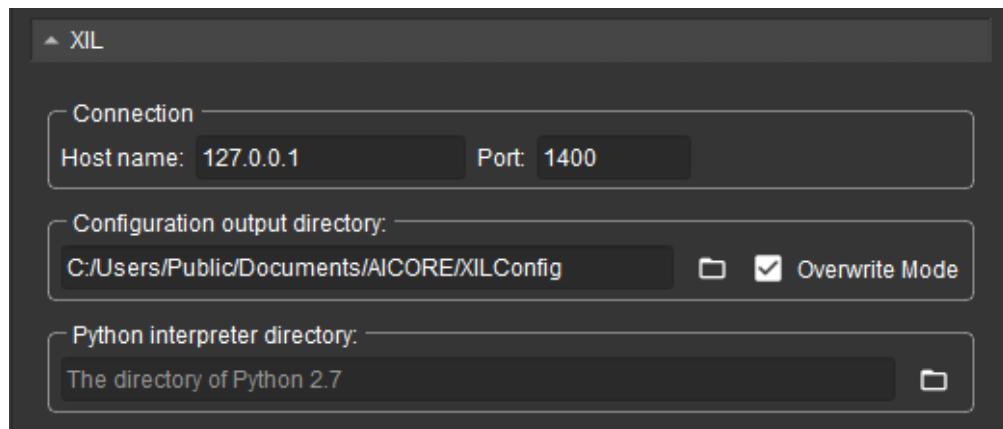
The implementation of Framework and Testbench API is contained in two C# dynamic link library (DLL) files, AICORE.XIL.Framework.dll and AICORE.XIL.Testbench.dll. The both files are located in the installation directory of AI-Core. To inform the test automation tool where it can find DLLs, AI-Core should always check whether AICORE\_ImplementationManifest.imf references the correct location.

Besides, since the variable names and number depend on the loaded project and settings in AI-Core, thus AI-Core should also be responsible for generating all these four configuration files to specify the connection method and the variables needed to be created in XIL.

To launch the test, you should accomplish following steps:

## 1. Set AI-Core

Follow the steps mentioned in section [Configuration of AI-Core](#). Additionally, you need to check the connection configuration in Settings > Test Mode > Protocol Type > XIL.



- **Host name and port:** The host name and port of TCP server for connecting with MA Port. If you want to launch multiple instances of AI-Core, please assign different ports for them.
- **Configuration output directory:** You should define the output folder of the XIL configuration files. When you load framework configuration in the test script, you will use the path.
- **Python interpreter directory:** ASAM XIL also supports the references to C# implementations in Python. To achieve this, you need to first build a Python environment and install Python.NET package,

```
pip install pythonnet
```

which gives Python programmers nearly seamless integration with the .NET Common Language Runtime (CLR) and provides a powerful application scripting tool for .NET developers. Then, you should register the path of your Python interpreter (python.exe) in the configuration of XIL. It will be written in AICORE\ImplementationManifest.imf.

## 2. Launch AI-Core with test script or a command line

```
{AICORE_installation_directory}/AICORE.exe --conn XIL --port 1400  
{test_project_directory}/{test_project_name}.testcfg
```

The port number should be identical to the settings in AI-Core. According to the trained model, AI-Core automatically generates all XIL configuration files and exposes the TCP port waiting for the connection with MA Port.

### 3. Create an AKKODIS-specific framework

In the test script, first creates an AKKODIS-specific framework via ASAM XIL API.

```
IFrameworkFactory myFrameworkFactory = new FrameworkFactory();  
IFramework myFramework =  
myFrameworkFactory.CreateVendorSpecificFramework("AKKODIS", "2.0");
```

The API will retrieve the implementation manifest file in the path

```
\%PROGRAMDATA%\ASAM\XIL\Implementation
```

and load the Framework and Testbench assemblies.

### 4. Load configuration in framework

Let Framework load the configuration file. Accordin to the file, it will then create Testbench and MA Port(s) automatically.

```
IFrameworkConfig myFrameworkConfig =  
myFramework.LoadConfiguration("{path_to_configuration_files}/AICOREFrameworkConfigur  
ation.xml");  
myFramework.Init(myFrameworkConfig);
```

### 5. Get signal variables

The MA Port, as a TCP client, then will build its connection with AI-Core (TCP server). AI-Core detects objects from a video stream, then send messages to the MA Port. The MA Port receives the messages and parses them to readable variables through Framework API in the test script.

```
IStringVariable textColorContent =  
(IStringVariable)myFramework.CreateVariable("AICORE.Text.Num_0.Content");  
IStringQuantity stringQuantity = textColorContent.Read();  
Console.WriteLine("AICORE.Text.Num_0.Content: {0}", stringQuantity.Value);
```

### 6. Send command to AI-Core

Vice versa, when the test script writes a value to a Framework variable (e.g. send a command to

change icon detection mode), the MA Port will send a message via TCP to AI-Core. AI-Core parses and reacts the message.

```
IStringVariable commandContent =  
(IStringVariable)myFramework.CreateVariable("AICORE.Control.Command");  
IQuantityFactory quantityFactory = myFramework.QuantityFactory;  
IStringQuantity commandQuantity = quantityFactory.CreateStringQuantity("play");  
commandContent.Write(commandQuantity);
```

Additionally, you can also write your test script in Python.

```
import clr  
import time  
import sys
```

```
sys.path.append (r"C:\Program Files (x86)\ASAM e.V\ASAM XIL Standard Assemblies  
2.2.0\bin")  
clr.AddReference("ASAM.XIL.Interfaces")  
clr.AddReference("ASAM.XIL.Implementation.Framework")  
clr.AddReference("ASAM.XIL.Implementation.FrameworkFactory")  
clr.AddReference("ASAM.XIL.Implementation.ManifestReader")
```

```
from ASAM.XIL.Implementation.FrameworkFactory import FrameworkFactory  
from ASAM.XIL.Interface.Framework.Variables import IStringVariable
```

```
if __name__ == '__main__':  
    com = ASAM.XIL.Interfaces.Testbench.Common  
    myFrameworkFactory = FrameworkFactory()  
    myFramework = myFrameworkFactory.CreateVendorSpecificFramework("AKKODIS", "2.0")  
    myFrameworkConfig =  
    myFramework.LoadConfiguration("{path_to_configuration_files}/AICOREFrameworkConfigu  
ration.xml")  
    myFramework.Init(myFrameworkConfig)
```

### 7.4.3. AI-Core Client-Server XIL Setup

The AI-Core XIL Framework allows for the establishment of a Client-Server setup. This configuration enables control of AI-Core from multiple PCs within the network.

#### Prerequisites

Before proceeding, ensure that you have followed the previous instructions on the Host PC to enable local control of AI-Core using the XIL Framework. It is also recommended to use Python or C# for writing the test scripts.

## Finding the Host IP Address

To determine the Host IP Address on your Host PC, please follow these steps:

### On Windows:

1. Open the Command Prompt.
2. Type the command `ipconfig /all` and press Enter. This will display the configuration details for each network interface.
3. Identify the network interface through which each Client will communicate with the Host PC. In a typical scenario, each Client can connect to the Host via the default Wi-Fi connection.
4. Look for the entry labeled "IPv4 Address" under each used interface. This address will be required in the next step.

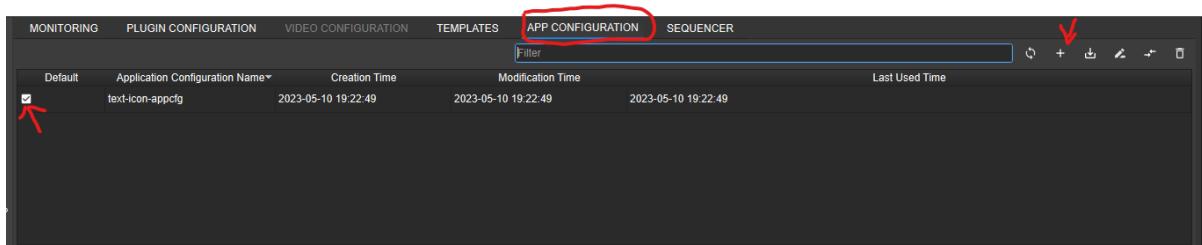
### On Linux:

1. Open a terminal window.
2. Type the command `ip addr show` or `ifconfig` (if `ifconfig` is available) and press Enter. This will display the configuration details for each network interface.
3. Identify the network interface through which each Client will communicate with the Host PC. In most cases, this will be the interface connected to your local network (e.g., `wlan0` for Wi-Fi or `eth0` for Ethernet).
4. Look for the line that starts with `inet` followed by the IP address. This is the "IPv4 Address" for the interface. Note that if you are using `ip addr show`, the IPv4 address will be listed after the `inet` keyword. This address will be required in the next step.

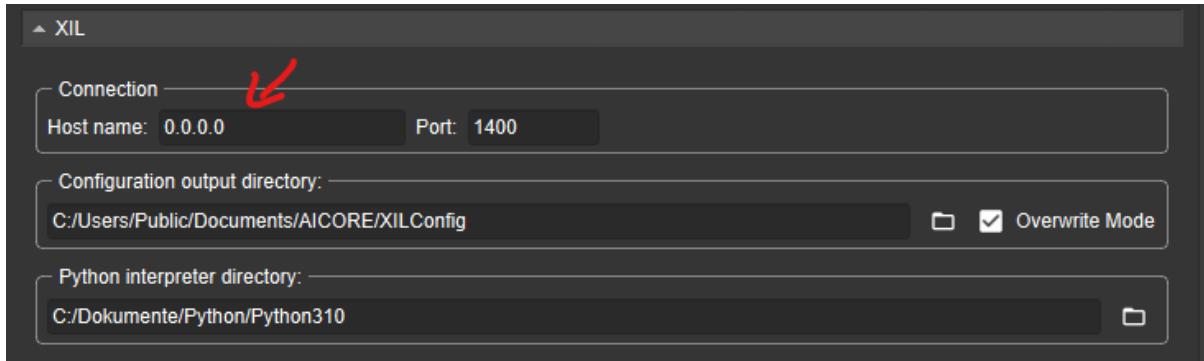
## Reconfigure AI-Core

To reconfigure AI-Core, follow these steps:

1. Open AI-Core and start a test session.
2. In the test session, create a new AppConfiguration.
3. Once the new AppConfiguration is created, activate it by clicking the checkbox on the left side.



1. Navigate to the settings and open the XIL configuration.
2. Replace the local IP address `127.0.0.1` with the IP address obtained in the previous step. You can keep the port number set to `1400`.



- Close AI-Core and launch it from the command line using the following command:  
`{AICORE_installation_directory}/AICORE.exe --conn XIL --port 1400 {test_project_directory}/{test_project_name}.testcfg`

Now, the XML files have been updated to reflect the new IP address.

## Copying Required Files to the Client PCs

The following files need to be copied from the Host PC to the Client PCs:

- Host: `{AICORE_installation_directory}/AICORE.XIL.Framework.dll` → Client: `{Wherever you want}/AICORE.XIL.Framework.dll`
- Host: `{AICORE_installation_directory}/AICORE.XIL.Testbench.dll` → Client: `{Wherever you want}/AICORE.XIL.Testbench.dll`
- Host: `C:/User/Public/Documents/AICORE/XILConfig` → Client: `C:/User/Public/Documents/AICORE/XILConfig`

Additionally, you need to copy and edit the following file:

- Host: `C:/ProgramData/ASAM/XIL/Implementation/ImplementationManifest` → Client: `C:/ProgramData/ASAM/XIL/Implementation/ImplementationManifest`

In the **ImplementationManifest** on the Client PC, update the paths to reflect the current location of the files on the Client PC.

```
<?xml version="1.0" encoding="utf-8"?>
<ImplementationManifest xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.asam.net/XIL/ImplementationManifest/2.1 ImplementationManifest.xsd" xmlns="http://www.asam.net/XIL/ImplementationManifest/2.1">
  <NetTestbenchImplementation ProductName="AICORE" ProductVersion="2.0" SortableProductVersion="2021.1.1" XILVersion="2.2.0" TechnologyVersion="2" VendorName="AKKODIS">
    <AssemblyName>AICORE.XIL.Testbench</AssemblyName>
    <AssemblyVersion>2.0.4</AssemblyVersion>
    <AssemblyKeyToken></AssemblyKeyToken>
    <AssemblyCulture>neutral</AssemblyCulture>
    <AssemblyPath>{Path to AICORE.XIL.Testbench}</AssemblyPath>
    <TestbenchClassName>AICORE.XIL.Testbench.Testbench</TestbenchClassName>
  </NetTestbenchImplementation>
  <NetFrameworkImplementation ProductVersion="2.0" SortableProductVersion="2021.1.1" XILVersion="2.2.0" TechnologyVersion="2" VendorName="AKKODIS">
    <AssemblyName>AICORE.XIL.Framework</AssemblyName>
    <AssemblyVersion>2.0.4</AssemblyVersion>
    <AssemblyKeyToken></AssemblyKeyToken>
    <AssemblyCulture>neutral</AssemblyCulture>
    <AssemblyPath>{Path to AICORE.XIL.Implementation}</AssemblyPath>
    <FrameworkClassName>AICORE.XIL.Implementation.Framework</FrameworkClassName>
  </NetFrameworkImplementation>
  <PyTestbenchImplementation ProductName="AI-Core" ProductVersion="2.0" SortableProductVersion="2021.1.1" XILVersion="2.2.0" TechnologyVersion="3.10" VendorName="AKKODIS">
    <TestbenchClassName>AICORE.XIL.Testbench.Testbench</TestbenchClassName>
    <TestbenchClassModuleName>PyTestbench</TestbenchClassModuleName>
  </PyTestbenchImplementation>
  <PyFrameworkImplementation ProductVersion="2.0" SortableProductVersion="2021.1.1" XILVersion="2.2.0" TechnologyVersion="3.10" VendorName="AKKODIS">
    <LibPaths>
      <LibPath>{Path to python installation}</LibPath>
    </LibPaths>
    <FrameworkClassName>AICORE.XIL.Framework.Framework</FrameworkClassName>
    <FrameworkClassModuleName>PyFramework</FrameworkClassName>
  </PyFrameworkImplementation>
</ImplementationManifest>
```

## Starting the Test Script

Once everything is configured, you can execute your Python or C# test script. Ensure that you have started AI-Core on the Host PC using the following command:

```
{AICORE_installation_directory}/AICORE.exe      --conn      XIL      --port      1400
{test_project_directory}/{test_project_name}.testcfg
```

## Troubleshooting

If your test script cannot connect to AI-Core, please try the following steps:

1. Check if you can ping the Host PC by executing the command: `ping {Host IP}` If you receive a "packets received" message, please continue with step 2. If all packets are lost, proceed to step 4.
2. Open the file `C:/User/Public/Documents/AICORE/XILConfig/AICOREMAPortConfiguration_0.xml` and verify that the IP address is indeed the Host IP. If it is correct, continue with the next step.

```
<?xml version="1.0" encoding="utf-8"?>
<MAPortConfigMapping xmlns="http://www.asam.net/XIL/Mapping/2.2.0">
    <MappingFile>C:/Users/Public/Documents/AICORE/XILConfig/AICOREFrameworkMapping.xml</MappingFile>
    <PortId>AICORE_MAPort_0</PortId>
    <TcpAddress>0.0.0.0</TcpAddress>
    <TcpPort>1400</TcpPort>
    <TaskList>
        <String>Task10</String>
        <String>Task50</String>
        <String>Task100</String>
        <String>Task1000</String>
    </TaskList>
</MAPortConfigMapping>
```

1. On the Host PC, check if the XIL configuration of AI-Core is correct by opening the settings within AI-Core. Navigate to the XIL settings and verify the IP address. The IP address should match the Host IP. If it does not, change it and create a new AppConfigration.
2. Ensure that both PCs are on the same network. Verify that the firewall on the Host PC is configured to allow communication with other PCs on the same network.

## 7.5. REST

### 7.5.1. Overview of REST

The REST API (Representational State Transfer Application Programming Interface) enables communication between computers, whether they are within the same network or connected over the internet. It provides a standardized way for devices or applications to exchange data and interact using HTTP methods such as GET, POST, PUT, and DELETE. By leveraging REST, systems can request resources, send data, or trigger actions on connected devices in a lightweight and efficient manner. Since REST APIs are stateless, each request contains all the necessary information, eliminating the need for additional context from previous interactions. This simplicity, scalability, and ease of integration make REST APIs particularly valuable in both local network environments and distributed systems. Data exchanged through REST APIs is typically formatted in JSON or XML, ensuring seamless communication across different platforms and programming languages.

### 7.5.2. AI-Core REST Implementation

Since AI-Core is a visual analysis tool designed to extract information from raw video streams, you can use the REST API to control it via a test script, which may or may not reside on the same system. To enable communication through the REST API, you need to start AI-Core either with a test script or via the command line using the following command:

```
{AICORE_installation_directory}/AICORE.exe --conn REST  
{test_project_directory}/{test_project_name}.testcfg
```

To control AI-Core via the REST API, you will need a test script. We recommend using Python for this purpose due to its simplicity and versatility, but any other programming language capable of making HTTP requests should work equally well. To make the setup as simple as possible we provide multiple example scripts in [C:/User/Public/Documents/AICORE/example\\_test/RESTScripts](C:/User/Public/Documents/AICORE/example_test/RESTScripts)

### 7.5.3. AI-Core using REST and XIL

To simplify the AI-Core XIL setup on remote systems, we provide multiple REST API endpoints that transmit the content of the following files:

- **AICORE\_ImplementationManifest.imf:**  
provides information for locating and instance creation of the vendor specific Testbench and/or Framework class. This file is always located in the folder `\%PROGRAMDATA%\%ASAM/XIL/Implementation`.
- **AICOREFrameworkConfiguration.xml:**  
The framework configuration file is the central configuration file. It references a set of mapping files and a set of port configuration files.
- **AICOREFrameworkMapping.xml:**  
The mapping files particularly contain information on the available framework variables, e.g. data types, unit definitions, associations to specific testbench port labels and conversions. Through this file, the framework, testbench, and MA Port can create required signal variables

and then the detected result can be read and written during the test via them.

- **AICOREMAPortConfiguration.xml:**

The port configuration files encapsulate all the port specific configuration settings required to setup the testbench ports, such as the address and port number of the TCP connection between MA Port and AI-Core. In the use case of multiple instances, the testbench will create one individual MA Port for each AI-Core instance. In order to distinguish them, the port configuration file also defines the port ID in form of AI-Core\_MAPort\_{instance index}.

To use this setup, AI-Core must be started either from a test script or via the command line using the following command:

```
{AICORE_installation_directory}/AICORE.exe --conn REST_XIL --port 1400  
{test_project_directory}/{test_project_name}.testcfg
```

For guidance, please refer to the example scripts located in the [C:/User/Public/Documents/AICORE/example\\_test/RESTScripts](C:/User/Public/Documents/AICORE/example_test/RESTScripts) folder, as well as the API Documentation provided in the same directory.

#### 7.5.4. AI-Core RemoteLauncher

When AI-Core is started for the first time, the AICORERemoteLauncher.exe will be executed and will run in the background. It will also automatically start after each system restart. The purpose of AICORERemoteLauncher.exe is to remotely start AICORE.exe and monitor whether AICORE.exe is running. For communication, we use the REST API, which requires the IP address of the host PC and a port number. By default, the port number is:

Port Number: 10043

We recommend using a Python script to establish the connection. This script should start the application and periodically check if it is still running. As usual, we provide an example script located at: [C:/User/Public/Documents/AICORE/example\\_test/RESTScripts](C:/User/Public/Documents/AICORE/example_test/RESTScripts)

# Appendix A: Supported NVIDIA GPUs

<https://en.wikipedia.org/wiki/CUDA>

Architecture/Compute Capability	GeForce	Quadro, NVS	Tesla/Datacenter GPU
Maxwell 5.0	GeForce GTX 750 Ti, GeForce GTX 750, GeForce GTX 960M, GeForce GTX 950M, GeForce 940M, GeForce 930M, GeForce GTX 860M, GeForce GTX 850M, GeForce 845M, GeForce 840M, GeForce 830M	Quadro K1200, Quadro K2200, Quadro K620, Quadro M2000M, Quadro M1000M, Quadro M600M, Quadro K620M, NVS 810	Tesla M10
Maxwell 5.2	GeForce GTX Titan X, GeForce GTX 980 Ti, GeForce GTX 980, GeForce GTX 970, GeForce GTX 960, GeForce GTX 950, GeForce GTX 750 SE, GeForce GTX 980M, GeForce GTX 970M, GeForce GTX 965M	Quadro M6000 24GB, Quadro M6000, Quadro M5000, Quadro M4000, Quadro M2000, Quadro M5500, Quadro M5000M, Quadro M4000M, Quadro M3000M	Tesla M4, Tesla M40, Tesla M6, Tesla M60
Pascal 6.0		Quadro GP100	Tesla P100
Pascal 6.1	Nvidia TITAN Xp, Titan X, GeForce GTX 1080 Ti, GTX 1080, GTX 1070 Ti, GTX 1070, GTX 1060, GTX 1050 Ti, GTX 1050, GT 1030, GT 1010, MX350, MX330, MX250, MX230, MX150, MX130, MX110	Quadro P6000, Quadro P5000, Quadro P4000, Quadro P2200, Quadro P2000, Quadro P1000, Quadro P400, Quadro P500, Quadro P520, Quadro P600, Quadro P5000(Mobile), Quadro P4000(Mobile), Quadro P3000(Mobile)	Tesla P40, Tesla P6, Tesla P4
Volta 7.0	NVIDIA TITAN V	Quadro GV100	Tesla V100, Tesla V100S
Turing 7.5	NVIDIA TITAN RTX, GeForce RTX 2080 Ti, RTX 2080 Super, RTX 2080, RTX 2070 Super, RTX 2070, RTX 2060 Super, RTX 2060, GeForce GTX 1660 Ti, GTX 1660 Super, GTX 1660, GTX 1650 Super, GTX 1650, MX450	Quadro RTX 8000, Quadro RTX 6000, Quadro RTX 5000, Quadro RTX 4000, T1000, T600, T400 T1200(mobile), T600(mobile), T500(mobile), Quadro T2000(mobile), Quadro T1000(mobile)	Tesla T4
Ampere 8.0			A100 80GB, A100 40GB, A30

<b>Architecture/Compute Capability</b>	<b>GeForce</b>	<b>Quadro, NVS</b>	<b>Tesla/Datacenter GPU</b>
Ampere 8.6	GeForce RTX 3090, RTX 3080 Ti, RTX 3080, RTX 3070 Ti, RTX 3070, RTX 3060 Ti, RTX 3060, RTX 3050 Ti, RTX 3050	RTX A6000, RTX A5000, RTX A4000, RTX A2000, RTX A5000(mobile), RTX A4000(mobile), RTX A3000(mobile), RTX A2000(mobile)	A40, A16, A10
Ampere 8.9	GeForce RTX 4090, RTX 4080 Super, RTX 4080, RTX 4070 Ti Super, RTX 4070 Ti, RTX 4070 Super, RTX 4070, RTX 4060 Ti, RTX 4060	RTX 6000 Ada, RTX 5880 Ada, RTX 5000 Ada, RTX 4500 Ada, RTX 4000 Ada, RTX 4000 SFF	L40S, L40, L20, L4, L2

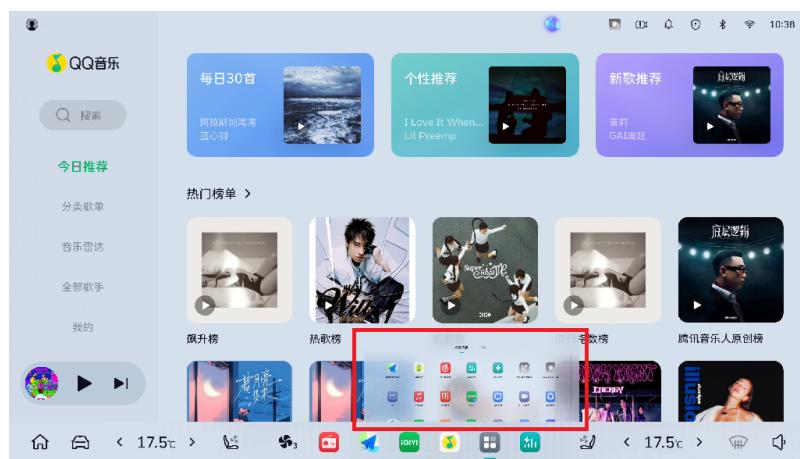
# Appendix B: Troubleshoot

## B.1. Labeling & Training Issues

### 1. Must I label each frame?

No. In the following cases, you can / should skip the frames without labeling:

- If the video is high-quality, you can skip those repeated frames without any changes. The same frames just need 10 - 50 times labeling.
- Don't label transition animations when the object is not clear or occluded.
- Objects may keep changing size e.g. during transition animations. If you are not interested in them, you can ignore them. In contrast, if you want to detect these scaling objects accurately, you should label them carefully.



### 2. Why is my training result so bad?

#### ◦ Wrong labels: Often wrong labels include

- Wrong bounding box position, size, assigned class.
- Partial labeling: You have labeled on frame (at least one bbox in the frame), but you have not labeled all objects belonging to each defined class.
- One object has been assigned to multiple different classes.
- Sometimes, one class can be a subset of another, for example: and . Such kind of icons should be avoided or need more training samples.

#### ◦ Inappropriate training presets:

- The input size should be close to the original resolution. Too small input size will lead bad detection quality.
- If the objects differ in color or darkness, you should disable data augmentation.

### 3. When can I stop my training?

A successful training should meet following metrics:

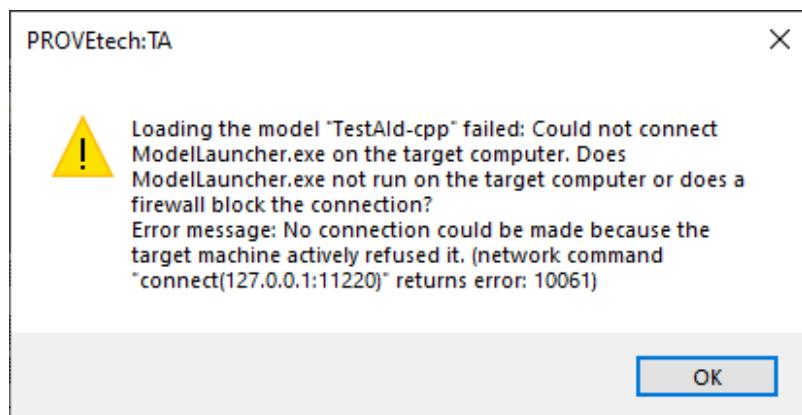
- The loss value (white curve) should be lower than 0.2. If you have many classes (e.g. 10-50), the value should be lower (e.g. 0.1), because frequently occurring classes will make the average score low, and we need to make the rare classes also have relatively low loss values.
- If you activate "Calculate mAP" or "Enable validation", you will also see the mAP value (red curve). It indicates how accurate your model detects the objects. This value should be stable at around 100% (at least four times reach 98% - 100%). If you use data augmentation, mAP could fluctuate but it is normal.

If the two metrics are met, you can already manually stop the training. Alternatively, you can activate the loss or validation trigger to stop the training at an optimal moment.



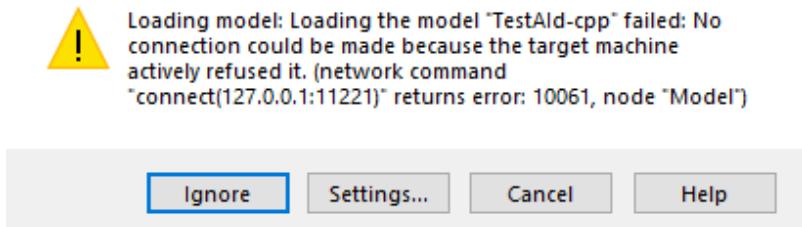
## B.2. Communication Issues

### 1. Cannot Connect ModelLauncher



TestAIdModelLauncher.exe is a process running in the background. PROVEtech:TA uses it to launch models as well including AI-Core. The appearance of this warning shows that TestAIdModelLauncher.exe has not been successfully launched or it has not been launched with correct parameters (-l -p:11220). TestAIdModelLauncher.exe is a part of AI-Core (in the directory of AI-Core) and AI-Core installer adds it to the auto start of the Windows user. Please make sure that TestAIdModelLauncher.exe is running correctly.

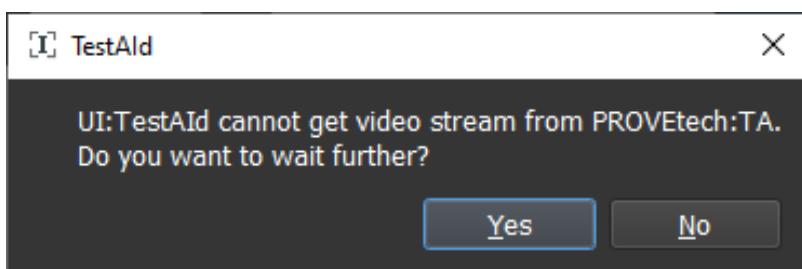
### 2. Fail to build connection between AI-Core and PROVEtech:TA



This warning appears when PROVEtech:TA has already launched AI-Core via TestAIdModelLauncher. However, the network communication failed. This failure could be caused by many reasons:

- Your AI-Core version is not compatible with the current PROVEtech:TA or RE.
- The settings file of AI-Core has not been set in the field of command line parameters. Or the file does not exist.
- In the settings file, the last launched project cannot be found. Or the project is invalid. In this case, please launch AI-Core standalone, launch the corresponding project, and check whether the test is conducted successfully.
- When the video source is selected as camera directly captured by AI-Core, the camera however is not connected to the computer.
- You did not activate the GeneralCom plugin in AI-Core.
- When you are working on a workstation on which there are other account has launched AI-Core through PROVEtech:TA, however you are not working as the local administrator, the connection between your AI-Core and PROVEtech:TA could be blocked by system or firewall. In this case, you can solve the problem by restarting your system and optionally run PROVEtech:TA as local administrator.

### 3. Cannot get video stream from PROVEtech:TA



This warning appears when you have select PROVEtech:TA as the video source via shared memory, however AI-Core cannot receive the shared video stream from PROVEtech:TA.

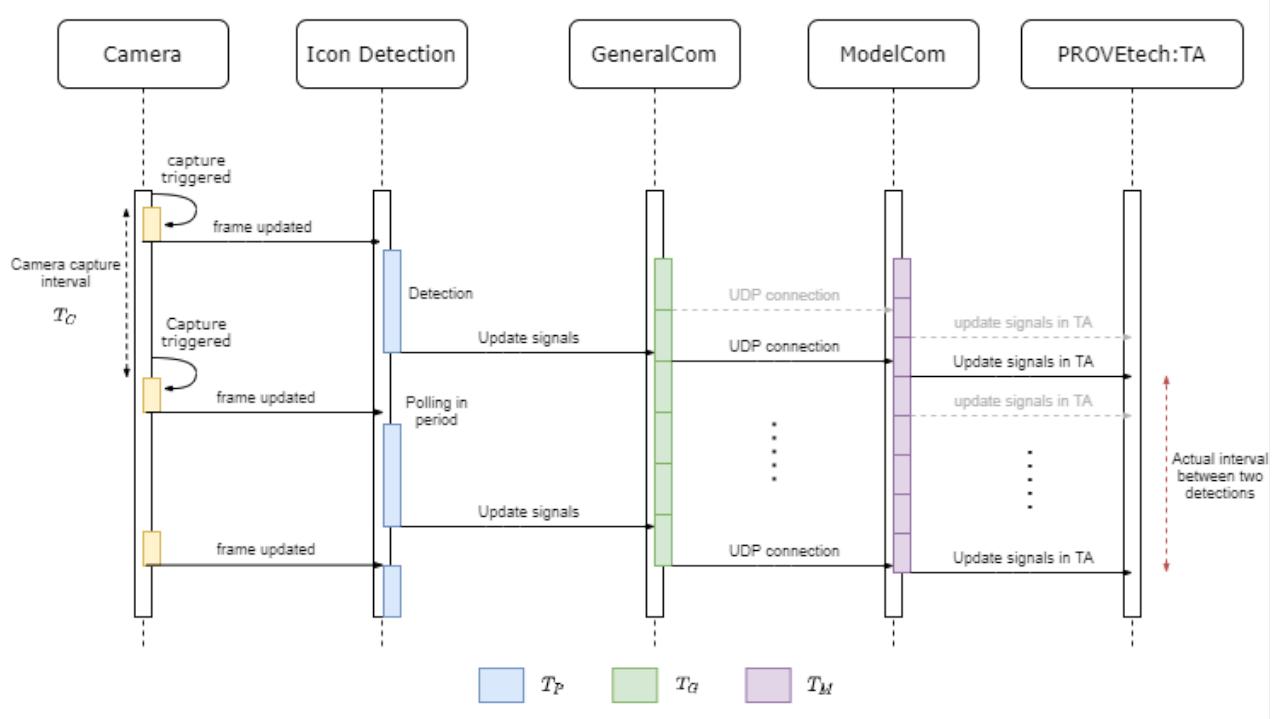
- It usually happens when you have not created video device capture in the settings dialog of PROVEtech:TA.
- In Windows, one video or audio device is only allowed to be occupied by one process at the same time. When you create multiple video captures on the same device inside of PROVEtech:TA or in other software. The video stream sharing will also fail.
- One rare reason for this failure is that your current NVIDIA GPU driver is not compatible for

video stream sharing. In this case, please roll back your NVIDIA GPU driver, then restart your computer, and check whether the problem can be solved.

#### 4. The actual detection speed cannot reach the capture frame rate

The detection speed is influenced by many factors:

- Camera often cannot reach the configured frame rate. The actual capture speed could be influenced by the cable type, USB version, CPU occupy ratio, and so on. Thus, the average frame rate might be lower than it announced.
- The plugin will take time  $T_P$  to perform the image analysis algorithms. For example, the icon detection will take at least 10 ms to finish the process. The more objects are detected, the more time will be taken.
- The capturing and detection are working in parallel. The final duration is the larger one of them. For example, if the detection lasts longer than camera capture interval, then the final detection interval depends on the detection duration.
- Color detection in icon detection plugin is not accelerated by GPU. Thus, it will also increase the detection time.
- When detection results are obtained, each plugin will update the values in their own signal container. The GeneralCom plugin will poll all signals and send the values to PROVEtech:TA via ModelCom in a certain period  $T_G$ . You can change this period in GeneralCom's configuration panel.
- ModelCom will then process the received signals and update the values in PROVEtech:TA every  $T_M$ .



In conclusion, the "worst" detection duration can be calculated by

$$T_{\{result\}} = \max(T_C, T_P) + T_G + T_M$$

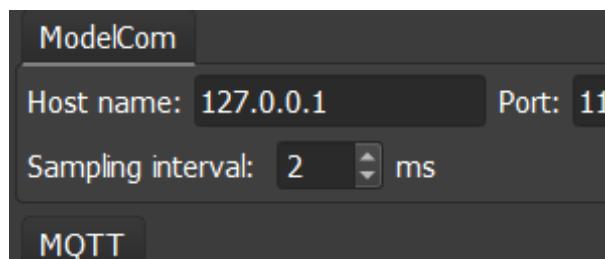
In conclusion, the "worst" detection duration can be calculated by

$$T_{\{result\}}(ms) = \max(19, 13) + 2 + 2 = 23$$

The capturing samples every 19 ms (52 Hz). The detection takes 13 ms (77 Hz). The two communication steps need 4 ms in total. Thus, the resulting detection interval is 23 ms (43 Hz).

To increase the actual detection speed, you can adjust following parameters, but it will increase the CPU occupation correspondingly.

- Use faster CPU and GPU.
- Use a high-performance camera with high capture speed and stable performance.
- Unload unnecessary plugins. Even if some plugins do not detect anything, it will still increase the workload of the system slightly.
- Avoid using color detection.
- Reduce the period of GeneralCom and ModelCom.



## 5. Monitor GPU Performance

If you find that the test is randomly interrupted without any dump creation in <C:\Users\Public\Documents\AICORE\dump>, you need to monitor the GPU performance and check whether it works at a high temperature ( $\geq 80^{\circ}\text{C}$ ).

You can create an environment variable "CHECK\_CUDA" with the value of "true". Then, relaunch your test. AI-Core will then regularly print the status of GPU in the logging file. Based on the performance information, you can check whether the GPU is overheated and needs additional cooling or pause the testing regularly.

# Appendix C: Supported Commands

In this chapter, you can have an overview of all commands used for controlling AI-Core.

Each part in one command is separated by spaces from each other.

## C.1. General Controls

Explain	Example
Resume/Run Test	Play
Pause Test	Pause
Exit the Program	Exit
Jump To A Frame	JumpTo 0
Set Local Video FPS	SetFps 30
Activate Frame by Frame Processing	FrameByFrame true
Activate/lock Aruco markers	You can enable/disable using on/off options and also can lock aruco markers position using Aruco lock command: Aruco on Aruco off Aruco lock
Capture All Screens	The optional parameter is a scale factor (float). To reduce the storage usage, users can downsample the captured screen by the scale factor: Capture Capture 0.5
Capture Frame	The optional parameter is a scale factor (float). To reduce the storage usage, users can downsample the captured screen by the scale factor: CaptureFrame CaptureFrame 0.5
Start Recording Video	There are three parameters: > bit rate (video compression): -1 = missing: lossless, 664000: 664kbit/s > video file path > audio file path > return: finally used video and audio file path  If parameters are not given, AI-Core will use default values. Examples: StartRecording 664000 StartRecording 664000 C:/Users/Public/Documents/example.avi StartRecording C:/Users/Public/Documents/example.wav StartRecording C:/Users/Public/Documents/example.avi C:/Users/Public/Documents/example.wav

Explain	Example
Start Recording with All Detection Results	<p>There are three parameters:</p> <ul style="list-style-type: none"> <li>&gt; bit rate (video compression): -1 = missing: lossless, 664000: 664kbit/s</li> <li>&gt; video file path</li> <li>&gt; audio file path</li> <li>&gt; return: finally used video and audio file path</li> </ul> <p>If parameters are not given, AI-Core will use default values. Examples:</p> <pre>StartRecordingWithResults 664000 StartRecordingWithResults 664000 C:/Users/Public/Documents/example.avi StartRecordingWithResults C:/Users/Public/Documents/example.wav StartRecordingWithResults C:/Users/Public/Documents/example.avi C:/Users/Public/Documents/example.wav</pre>
Enable test data backup	EnableResultBackup true
Stop Recording Video	StopRecording
Save Test Results	SaveResult 'path\to\json.json' or 'path\to\xml.xml'
Clean the detection results	CleanResult
Switch Test Project	SwitchProject 'D:/DemoTest/settings_test.testcfg'
Switch Model (not influence next relaunch)	SwitchModel IconDetection 'D:/DemoTrain/Models/model1/model.modelcfg'
Switch Model (not influence next relaunch) (loads multiple models)	SwitchModel IconDetection 'D:/DemoTrain/Models/model1/model.modelcfg' IconDetection 'D:/DemoTrain/Models/model2/model.modelcfg' ...
Switch Model (not influence next relaunch) (loads multiple models) (each model receives a name)	SwitchModel IconDetection 'model name' 'D:/DemoTrain/Models/model1/model.modelcfg' IconDetection 'model name' 'D:/DemoTrain/Models/model2/model.modelcfg' ...
Switch Model (will be recovered at next relaunch)	SwitchModel save IconDetection 'D:/DemoTrain/Models/model1/model.modelcfg'
Switch Model (will be recovered at next relaunch) (loads multiple models)	SwitchModel save IconDetection 'D:/DemoTrain/Models/model1/model.modelcfg' IconDetection 'D:/DemoTrain/Models/model2/model.modelcfg' ...

Explain	Example
Switch Model (will be recovered at next relaunch) (loads multiple models) (each model receives a name)	SwitchModel save IconDetection 'model name' 'D:/DemoTrain/Models/model1/model.modelcfg' IconDetection 'model name' 'D:/DemoTrain/Models/model2/model.modelcfg' ...
Switch Template	SwitchTemplate 'template1'
Clear Template	ClearTemplate
Switch Application Configuration	SwitchAppConfig 'demoBackup'
Load A New Image Through Path for Test	LoadNewImage 'relative path/absolute path'
Show the Path of the Current TestCfg	TestConfigPath
Add ROI to be detected	AddROI x1 y1 x2 y2
Clear all ROIs	ClearROI

If there are spaces in the path or template name, please add ' on both sides of the texts

## C.2. Video Controls

You can switch the video source via the following commands. The video source can be a camera, a video file, website, or screen.

**Notice:** Please do not call SwitchVideo through Control.Command but ModifyModelNodeConfig instead! For example,

```
System.ModifyModelNodeConfig("Model", "SwitchIPCamera  
rtsp://aicore:password@192.168.0.3:8553/", 45)
```

Explain	Example
Switch Video File	SwitchVideo 'D:/DemoTest/demo_video.mp4'
Switch to IP Camera	SwitchIPCamera 'rtsp://192.168.0.3:554/'
Switch to web renderer	SwitchWebRenderer 'http://www.google.com'
Switch to screen	SwitchScreenCapturer '\\.\DISPLAY1'

## C.3. Camera Setting

The concrete value ranges depend on the used camera. If you set an invalid value, the camera will not react it.

	Value Range	Interval	Example
SwitchDevice			Camera SwitchDevice "Integrated Camera"
CurrentPreset			Camera CurrentPreset "Available Preset"
Brightness	0~255	1	Camera Brightness 128
Contrast	0~255	1	Camera Contrast 113
Saturation	0~255	1	Camera Saturation 128
Gain	0~255	1	Camera Gain 200
White Balance	2000~7500	1	Camera WhiteBalance 4000
White Balance (R)	0.3~16.0	0.1	Camera WhiteBalance(R) 1.8
White Balance (G)	0.3~16.0	0.1	Camera WhiteBalance(G) 2.0
White Balance (B)	0.3~16.0	0.1	Camera WhiteBalance(B) 5.2
Auto Exposure	true, false		Camera AutoExposure true
Exposure	-11~2	1	Camera Exposure -5
Auto Focus	true, false		Camera AutoFocus true
Focus	0~255	1	Camera Focus 10
Zoom	100~1000	1	Camera Zoom 120
Pan	-10~10	1	Camera Pan 1
Tilt	-10~10	1	Camera Tilt 0
Rotate	true, false		Camera Rotate true
Translate X	-0.2~0.2	0.001	Camera TranslateX 0.1
Translate Y	-0.2~0.2	0.001	Camera TranslateY -0.1

## C.4. Icon Detection

	Value Range	Interval	Unit	Example
Detection Mode	auto, manual			IconDetection Mode auto
Reliability Threshold	0.01~0.99	0.01	100%	IconDetection Reliability 0.5
Color Detection	true, false			IconDetection Color true
Color Extraction Mode	center, diagonal			IconDetection ColorExtractionMode center

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Color Threshold	0.0~1.0	0.01	100%	IconDetection ColorThreshold 0.30
Blink Timeout	0~30000	10	milliseconds	IconDetection Blink 100
Set ROI (left, top, right, bottom)	0~frame width/height	1	pixel	IconDetection Roi 20 50 60 80
Get Model Path (*.weights)				IconDetection ModelPath
Get Model Path (*.weights) for Specific Model Name				IconDetection ModelPath ModelName
Get Modelcfg Path				IconDetection ModelCfgPath

## C.5. Color Detection

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Color Extraction Mode	center, diagonal, contour			ColorDetection ColorExtractionMode center
Point Visualization	true, false			ColorDetection PointVisualization true
Diagonal Visualization	true, false			ColorDetection DiagonalVisualization false
Color Threshold	0.0~1.0	0.01	100%	ColorDetection ColorThreshold 0.30
Set ROI (left, top, right, bottom)	0~frame width/height	1	pixel	ColorDetection Roi 20 50 60 80
Set ROI (left, top, right, bottom, color extraction mode)	see above			ColorDetection roi 977 807 1087 892 center
Set ROI (left, top, right, bottom, color extraction mode, threshold)	see above			ColorDetection roi 977 807 1087 892 diagonal 0.4

## C.6. Text Detection

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Detection Language	eng, deu, fra+chi_sim, ...			TextDetection Language eng

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Page Segmentation Mode	Auto OSD, Auto only, Auto, Single column, Single block (V), Single block (H), Single line, Single word, Circle word, Single char, Sparse text, Sparse text OSD			TextDetection Psm 'Single Block (H)'
White List	any			TextDetection Whitelist dcst!3
Black List	any			TextDetection Blacklist aeoi
Number Only	true, false			TextDetection NumberOnly true
Preprocessing	Erode, Dilate, Off			TextDetection Preprocessing erode
Highlight Detection	true, false			TextDetection highlightdetection true
Background	dark, light			TextDetection background dark
Highlight Threshold	0 ~ 255	1		TextDetection highlightthreshold 253
Set ROI (left, top, right, bottom)	0~frame width/height	1	pixel	TextDetection roi 20 50 60 80

## C.7. Needle Detection

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Angle Interval	0.00~45.00	0.01	degree	NeedleDetection angleinterval 0.10
Set full needle ROI (EX,EY,CX,CY)	0~frame width/height	0.01	pixel	NeedleDetection roi full 446.7 713.6 671 623
Set half needle ROI (P1X,P1Y,P2X,P2Y,P3X,P3Y)	0~frame width/height	0.01	pixel	NeedleDetection roi half 443.8 714.5 516.8 685.3 896.1 534.2

## C.8. Line Detection

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
1. Edge Detection Threshold	0.00~999.99	0.01		LineDetection 1.threshold 200
2. Edge Detection Threshold	0.00~999.99	0.01		LineDetection 2.threshold 400

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Dilation Size	0.00~99.99	0.01	pixel	LineDetection dilationsize 2
Erosion Size	0.00~99.99	0.01	pixel	LineDetection erosionsize 3
Angle Resolution	0.01~90.00	0.01	degree	LineDetection angleresolution 0.1
Distance Resolution	0.01~99.99	0.01	pixel	LineDetection distanceresolution 1
Reliable Key Point Number	1~99	1		LineDetection reliability 8
Set ROI (left, top, right, bottom)	0~frame width/height	1	pixel	LineDetection roi 464 535 616 600
Set ROI (left, top, right, bottom, 1.threshold, 2.threshold, angle res, distance res, dilation, erosion, reliability)	see above			LineDetection roi 977 807 1087 892 20 30 0.1 1 1.15 1.45 8

## C.9. Audio Matching

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Frame Size	512, 1024, 2048, 4096, 8192	2^n	samples	AudioMatching frameSize 2048
Hop Size	0.0625, 0.125, 0.25, 0.5, 0.75		proportion	AudioMatching hopSize 0.8
MFCC Feature Number	5~40	1		AudioMatching mfccFeatureNumber 13
Lowpass Filter	1~999999	1	Hz	AudioMatching lowpassFilter 20000
Loudness Level	0.00~999.99	0.01	dB	AudioMatching loudnessLevel 35
Microphone Distance	0.00~99.99	0.01	meter	AudioMatching microDistance 1
Activate Denoising	true, false			AudioMatching activateDenoising true

## C.10. Screen Analysis

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
maxStd	0~255	1		ScreenAnalysis maxStd 23
brightness	0~255	1		ScreenAnalysis brightness 150
blurriness	10~500	1		ScreenAnalysis blurriness 100

## C.11. Pixel Trace

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Set Reference Video				PixelTrace setReferenceVideo 'C:/Users/Public/Documents/reference.mp4'
Set Reference Image				PixelTrace setReferenceImage 'C:/Users/Public/Documents/reference.png'
Set Roi Mode	'manual', 'auto'			PixelTrace setRoiMode auto
Set Auto Roi Mode	'component', 'contour'			PixelTrace setAutoRoiMode component
Set Kernel Size	1~10	1	px	PixelTrace setKernelSize 3
Enable Grid Mode	true, false			PixelTrace enableGridMode false
Show Reference Frame	true, false			PixelTrace showReferenceFrame false
Show Difference Frame	true, false			PixelTrace showDifferenceFrame false
Set Color Space	None, Gray Scale, RGB, HSV 0~255	1		PixelTrace setColorSpace None PixelTrace setColorSpace Gray 150 PixelTrace setColorSpace RGB 150 150 150 PixelTrace setColorSpace HSV 150 150 150
Set Channel Merge Strategy	None, Minimum, Average, Maximum			PixelTrace setChannelMergeStrategy None PixelTrace setChannelMergeStrategy Minimum PixelTrace setChannelMergeStrategy Average PixelTrace setChannelMergeStrategy Maximum

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Set Reference Video Frame Offset	-99999~99999	1	frames	PixelTrace setReferenceVideoFrameOffset 50
Capture Results				PixelTrace captureResults 'C:/Users/Public/Documents/scre enshot.png'
Capture Results with Scaling	> 0			PixelTrace captureResults 0.5 'C:/Users/Public/Documents/scre enshot.png'

## C.12. Template Matching

	<b>Value Range</b>	<b>Interval</b>	<b>Unit</b>	<b>Example</b>
Detection Mode	auto, manual			TemplateMatching Mode auto
Color Detection	true, false			TemplateMatching Color true
Color Extraction Mode	center, diagonal			TemplateMatching ColorExtractionMode center
Color Threshold	0.0~1.0	0.01	100%	TemplateMatching ColorThreshold 0.30
Blink Timeout	0~30000	10	milliseconds	TemplateMatching Blink 100
Template Directory				TemplateMatching TemplateDirectory TemplateDirectoryPath
TemplateMatching Mode	color, shape			TemplateMatching TemplateMatchingMode color
Multiple Template Scales	true, false			TemplateMatching MultipleTemplateScales true
Template Up Scale Value	1.0~4.0	0.05	100%	TemplateMatching TemplateUpScaleValue 3.45
Template Down Scale Value	0.0~1.0	0.05	100%	TemplateMatching TemplateDownScaleValue 0.45
Frame Down Scale Value	0.2~1.0	0.1	100%	TemplateMatching FrameDownScaleValue 0.45
Set ROI (left, top, right, bottom)	0~frame width/height	1	pixel	TemplateMatching Roi 20 50 60 80
Set ROI With Id (id, left, top, right, bottom)	0~frame width/height	1	pixel	TemplateMatching RoiWithId 0 20 50 60 80

# Appendix D: Dependencies



AI-Core uses Qt version 5.15.14.

Qt is a C++ toolkit for cross-platform application development. Qt provides single-source portability across all major desktop operating systems. It is also available for embedded Linux and other embedded and mobile operating systems.

Qt is available under multiple licensing options designed to accommodate the needs of our various users. Qt licensed under our commercial license agreement is appropriate for development of proprietary/commercial software where you do not want to share any source code with third parties or otherwise cannot comply with the terms of GNU (L)GPL. Qt licensed under GNU (L)GPL is appropriate for the development of Qt applications provided you can comply with the terms and conditions of the respective licenses.

Please see [qt.io/licensing](https://qt.io/licensing) for an overview of Qt licensing.

Copyright © 2020 The Qt Company Ltd and other contributors. Qt and the Qt logo are trademarks of The Qt Company Ltd.

Qt is The Qt Company Ltd product developed as an open source project. See [qt.io](https://qt.io) for more information.



**Darknet**

AI-Core developed multiple object detection algorithms owned by Akkodis. A part of the algorithms uses an optimized version of Darknet library. Even with the modifications, the Darknet library continues to further support open-source Darknet YOLO models, to strengthen the open-source movement.

Darknet is an open source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation.

Darknet is licensed under [YOLO License](#), version 2. According to the license, Darknet is public domain. Users can do whatever they want with it.



AI-Core uses OpenCV version 4.8.0.

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel).

OpenCV 4.4.0 and lower versions, including OpenCV 3.x, OpenCV 2.x, and OpenCV 1.x, are licensed under the [3-clause BSD license](#).

Copyright © 2000-2020, Intel Corporation, all rights reserved.

Copyright © 2009-2011, Willow Garage Inc., all rights reserved.

Copyright © 2009-2016, NVIDIA Corporation, all rights reserved.

Copyright © 2010-2013, Advanced Micro Devices, Inc., all rights reserved.

Copyright © 2015-2016, OpenCV Foundation, all rights reserved.

Copyright © 2015-2016, Itseez Inc., all rights reserved.

Copyright © 2019-2020, Xperience AI, all rights reserved.

Third party copyrights are property of their respective owners.

---



AI-Core uses Tesseract 5.3.3 for only basic optical character recognition. More advanced text recognition models are available and owned by Akkodis additionally.

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, version 2.0.

Tesseract was originally developed by Hewlett-Packard as proprietary software in the 1980s, it was released as open source in 2005 and development has been sponsored by Google since 2006.

Tesseract uses Leptonica library which essentially uses a [BSD 2-clause license](#). Please see [tesseract-ocr/licensing](#) for details of Tesseract OCR licensing.

---



AI-Core uses Intel RealSense SDK 2.0.

Intel® RealSense™ SDK 2.0 is a cross-platform library for Intel® RealSense™ depth cameras (D400 series and the SR300) and the T265 tracking camera.

The SDK allows depth and color streaming, and provides intrinsic and extrinsic calibration information. The library also offers synthetic streams (pointcloud, depth aligned to color and vice-versa), and a built-in support for record and playback of streaming sessions.

This project is licensed under the [Apache License, Version 2.0](#). Copyright 2018 Intel Corporation. The RealSense logo is the trademark of Intel Corporation.

---



AI-Core uses Eclipse Paho MQTT C++ Client Library.

The Eclipse Paho project provides reliable open-source implementations of open and standard messaging protocols aimed at new, existing, and emerging applications for Machine-to-Machine (M2M) and Internet of Things (IoT). Eclipse Paho MQTT C Client Library enables C11 applications to connect to an MQTT broker, publish messages to the broker, and to subscribe to topics and receive published messages.

All rights reserved by Eclipse Foundation. Eclipse Paho MQTT C++ Client Library and the accompanying materials are made available under the terms of the Eclipse Public License v1.0 and Eclipse Distribution License v1.0 which accompany this distribution.

Copyright (c) 2007, Eclipse Foundation, Inc. and its licensors. The Eclipse Distribution License is an OSI Approved Open Source License by means of the [New BSD License](#).

The Eclipse Public License is available at <http://www.eclipse.org/legal/epl-v10.html> and the Eclipse Distribution License is available at <http://www.eclipse.org/org/documents/edl-v10.php>.

---

## yaml-cpp

AI-Core uses yaml-cpp Library.

yaml-cpp is a YAML parser and emitter in C++ matching the [YAML 1.2 spec](#).  
yaml-cpp uses CMake to support cross-platform building.

Copyright (c) 2008-2015 Jesse Beder.

yaml-cpp is licensed under [MIT](#).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software

---

## Protobuf

AI-Core uses Protobuf Library.

Protocol Buffers (a.k.a., protobuf) are Google's language-neutral, platform-neutral, extensible mechanism for serializing structured data.

You can find [protobuf's documentation on the Google Developers site](#).

Copyright 2008 Google Inc. All rights reserved.

The details of Protobuf License can be found in [link](#).

---

## **Qwt**

AI-Core uses Qwt.

Qwt is distributed under the terms of the [Qwt License, Version 1.0](#).

The Qwt library and included programs are provided under the terms of the GNU LESSER GENERAL PUBLIC LICENSE (LGPL) with the following exceptions:

- Widgets that are subclassed from Qwt widgets do not constitute a derivative work.
- Static linking of applications and widgets to the Qwt library does not constitute a derivative work and does not require the author to provide source code for the application or widget, use the shared Qwt libraries, or link their applications or widgets against a user-supplied version of Qwt.  
If you link the application or widget to a modified version of Qwt, then the changes to Qwt must be provided under the terms of the LGPL in sections 1, 2, and 4.
- You do not have to provide a copy of the Qwt license with programs that are linked to the Qwt library, nor do you have to identify the Qwt license in your program or documentation as required by section 6 of the LGPL.

Copyright © 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

---

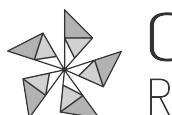
## **OpenH264**

AI-Core uses OpenH264.

openh264 is licensed under the BSD 2-Clause "Simplified" License ([link](#))

Copyright (c) 2013, Cisco Systems All rights reserved.

---



**ONNX Runtime**

AI-Core uses ONNX Runtime.

ONNX Runtime is a cross-platform inference and training machine-learning accelerator.

ONNX Runtime inference can enable faster customer experiences and lower costs, supporting models from deep learning frameworks such as PyTorch and TensorFlow/Keras as well as classical machine learning libraries such as scikit-learn, LightGBM, XGBoost, etc. ONNX Runtime is compatible with different hardware, drivers, and operating systems, and provides optimal performance by leveraging hardware accelerators where applicable alongside graph optimizations and transforms.

ONNX Runtime is licensed under MIT.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

# Appendix E: Contact

Introduction and leaflet of AI-Core

<https://www.provetech.de/index.php/products/ui-testaid>

Download page of AI-Core

<https://www.provetech.de/index.php/downloads/ui-testaid>

PROVEtech:TA Technical Support

[provetech@akkodis.com](mailto:provetech@akkodis.com)

# Appendix F: Version Migration

## F.1. Migration from UI:TestAId 2022 to AI-Core 2023+

Since version 2023, there are two major changes in the application.

- In the past, UI:TestAId plays a role of visual perception tool specialized in GUI testing. However, with the development of techniques, it is now able to perform more visual and auditory detection tasks. People can use it to locate workpieces in a product line or count visitors at an entrance.

Therefore, we rename our product **AI-Core** for the purpose of unifying the brand and updating the product positioning.

- Following Adecco's acquisition of AKKA, AKKA has been merged with Modis. Adecco's engineering division is thus given a new name **Akkodis**. All stale naming in the application and default installation and working directory paths are updated.

In detail,

- The common license file is stored in:

**C:/Users/Public/Documents/AKKA/License/License.key**

**C:/Users/Public/Documents/PROVETechLicense/License.key**

- The default installation directory is:

**C:/Program Files/AKKA/UITestAId**

**C:/Program Files/AICORE**

- The default executable file is:

**C:/Program Files/AKKA/UITestAId/TestAId.exe**

**C:/Program Files/AICORE/AICORE.exe**

**Notice:** If you previously used UI:TestAId in PROVETech:TA, you need to update "Path to RE.exe/AICORE.exe on target computer:" to the value.

- The default config file is

**C:/Users/Public/Documents/AKKA/UITestAId/cfg/settings.appcfg**

**C:/Users/Public/Documents/AICORE/cfg/settings.appcfg**

- The default log directory is

**C:/Users/Public/Documents/AKKA/UITestAId/Log**

**C:/Users/Public/Documents/AICORE/Log**

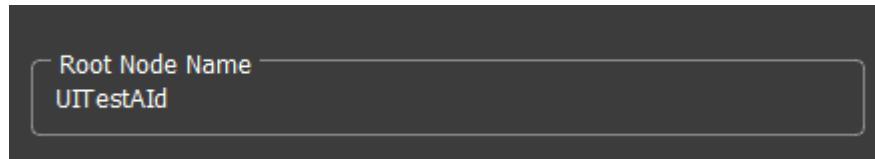
- The default root node name of the new version in PROVETech:TA is

**UITestAId**

**AICORE**

For example, the signal path of a text detection result will become **AICORE.Text.Num\_0.Content**.

**Notice:** If you previously used UI:TestAId in PROVEtech:TA, You should either update the codes in your TA test scripts to make sure the signals are all correct, or you can change the root name manually in AICORE like the following image and overwrite the appcfg to avoid the code changes.



## F.2. Migration from UI:TestAId 2021 to AI-Core 2023+

In the previous version of AI-Core (e.g. UI:TestAId 2021), the application and projects are organized by following files:

In each training project, the configuration and project structure information are saved in the config.yml under the project directory.

- **Application settings file (settings.ini)**

The application settings file (settings.ini) is saved in the path C:/Users/{username}/Documents/AKKA/UITestAId/cfg. It saves all application settings, such as camera, plugin configurations.

- **Training project configuration file (config.yml)**

In each training project, the configuration and project structure information are saved in config.yml directly under the project directory. The project name, videos, class sets, classes, and models are all saved here.

- **Labeling data file (labels.yml)**

The concrete labeling data, such as bounding box positions, categories, object IDs, are saved in labels.yml as well as under the project directory.

Compared to the previous version, UI:TestAId 2022 has improved the organization of training and test project.

- **Application configuration file (\*.appcfg)**

The application configuration file corresponds to the old settings.ini. It saves the current status of the application, including system language, video source, camera and plugin configurations. Different from the previous version, new \*.appcfg will not save the last loaded trained model in the test. Instead, the default trained models used in the test will be saved in the test project.

- **Test project configuration file (\*.testcfg)**

To better organize the test configurations, one test project is stored in an individual folder in UI:TestAId 2022. The test configuration file has extension with \*.testcfg. The name of the configuration file is the same as the project name by default. The test configuration file saves the history of loaded machine learning models, templates, and backup application configurations, with which users can recover specific test conditions.

- **Training project configuration file (\*.traincfg)**

The training project configuration file still uses similar schema as its previous \*.yml file. However, the templates used during test will be saved in \*.testcfg instead of here. Additionally, the concrete details of training will be moved to \*.modelcfg.

- **Training model configuration file (\*.modelcfg)**

This configuration file is newly adopted to save the details of training results. It includes the preprocessing setup, training parameters, duration and loss status at the end of the training. With these information, you can check the model status and easily recover a training process.

Here is the correspondence between the documents in the old and new versions:

UI:TestAId 2021	UI:TestAId 2022 +	Location
settings.ini	*.appcfg	C:/Users/Public/Documents/AICORE/cfg
config.yml	*.traincfg	{training_project_directory}
labels.yml	labels.yml	{training_project_directory}
-	*.modelcfg	{training_project_directory}/Models/{model_name}
-	*.testcfg	{test_project_directory}

If you are the user of the previous version, please do not worry about the migration. The process will be mostly done by AI-Core itself. Assumed that in the old version you have a training project and multiple .ini files for the test configuration initialization, you just need to do following steps:

1. Start AI-Core standalone
2. Upgrade all training projects through import config.yml in the home page. config.yml will then be converted to config.traincfg. The old config.yml will still be kept.
3. Import one of the .ini file in the home page. AI-Core will create a test project. The application configuration will be upgraded and saved as a backup.
4. Select a video file as the video source.
5. Select one trained model to start a test.
6. Import the rest of .ini files in "APP CONFIGURATION".
7. Choose one application configuration as the default one.
8. The templates which were saved in the training project cannot be imported. Please resave them in the test project.

# Appendix G: ArUco Markers

