

Version: v25.02

Table of Contents

1. Introduction

1.1. What can a Brain Builder AI Model Do?

1.2. Installation Packages

1.3. Plans for Brain Builder for AITRIOS

1.4. Operation Flow of Brain Builder for AITRIOS

2. Installing Brain Builder for AITRIOS

2.1. Install

2.2. Activate Your License

2.2.1. Download and Install CodeMeter™

2.2.2. Activate your License

2.3. Upgrade

2.4. Uninstall

2.5. Initial Setup

3. Capturing Images for Training

3.1. Camera Setup

3.2. Prepare Your Objects for Capture

3.3. Dataset Image Requirement

3.4. Tips for Each Model Type

3.5. Capture Images with Camera

4. Creating Your Project and Datasets

4.1. Create a Project

4.2. Create a Dataset

4.3. Training Using Brain Builder

5. Classifier

5.1. Introduction

5.2. Upload Datasets

5.2.1. Dynamic Split

Brain Builder for AITRIOS™ : Getting Started

1. Introduction

Brain Builder for AITRIOS simplifies the creation of deep learning models. Prepare a dataset, create and train a deep learning model, evaluate the inference results, and deploy the model to an AI camera - all without needing to be an AI expert.

1.1. What can a Brain Builder AI Model Do?

Brain Builder supports three AI tasks: Classification, Detector, and Anomaly Recognizer. Here are the key features of the each AI model:

Classifier

The Classifier model efficiently classifies images into predefined classes, providing labels ranked by confidence (for example, "apple: 80%", "peach 40%").

It operates on the image without pinpointing specific objects, resulting in faster and more accurate results when categorizing precise tasks.

Use cases and examples

- Sorting product images into categories, like fruits (apple, orange, banana)
- Ranking product quality based on placement, packaging, assembly, and other measurements
- Monitoring the status of an environment, such as door open/close, machine status reading, etc.
- Indicating the status of cargo occupancy in a truck or on a shelf
- Identifying products from a mix of items on a conveyor belt

Detector

The Detector model type can identify and locate multiple objects within an image. It outputs bounding boxes around the detected items and labels them with a confidence score (for example, "car: 90%" at (x,y)).

This is great for detecting or counting objects as it effectively identifies what objects are and where they are.

Use cases and examples

- Detecting vehicles and pedestrians in street scenes
- Counting items on a conveyor belt ("3 apples, 2 oranges")
- Identifying human attributes of a customer
- Detecting protective equipment wear
- Detecting specified shapes or letters in an image.
- Detecting the presence/absence of specific object (for example, new cars or people arriving in the scene)
- **Detecting cars to improve the parking experience**
By detecting cars, the Detector model can achieve the following:
 - Visualize the parking lot status
 - Display available spaces
 - Assist drivers in finding empty spots
- **Detecting products to support customer purchases in retail stores**
By monitoring product shelves, the Detector model can achieve the following:
 - Identify empty shelves or low stock and notify for restocking
 - Display dynamic targeted advertisements to promote underperforming products

Anomaly Recognizer

The Anomaly Recognizer model type identifies deviations from normal patterns and determines whether an image is "normal" or "anomaly". It can also localize areas of possible anomalies with a heatmap (Hi-Fi).

Unlike the Classifier model, it uses a few anomaly images for validation, not for training, making it suitable when anomalies are rare or their patterns are varied.

Usecases and examples

Brain Builder for AITRIOS can detect anomalies such as scratches, dents, irregularities, or other defects.

- Inspecting manufacturing parts for scratches, dents, irregularities, or other defects
- Detecting irregularities in medical scans or quality control images
- Detecting foreign object contamination
- Identifying and localizing the anomaly areas for later analysis



For details on Brain Builder, You can also consult the Official Neurala Documentation:
<https://support.neurala.com/docs/via-for-sony-aitrios>

1.2. Installation Packages

- Brain Builder is available for GPU-equipped machines and CPU-only machines.
- Consult the following table and choose the appropriate installer based on your PC environment.
- If you have a compatible GPU, we recommend using the GPU-Installer, but otherwise use the CPU-Installer.

Table 1. System Requirement

	GPU-Installer	CPU-Installer
Operating System	Windows 10 or Windows 11 Ubuntu 20.04 or Ubuntu 22.04	
CPU	Minimum: Intel i5 Coffee Lake Recommended: Intel i7 or better	
GPU	Minimum: Nvidia GTX 10xx (8+GB VRAM) Recommended: Nvidia RTX 20xx, 30xx, 40xx (10+GB VRAM)	Not Required
Memory	Minimum: 16GB Recommended: 32GB or more	
Storage	32GB or more space available	



Local Edition is limited to the Windows CPU-Installer

1.3. Plans for Brain Builder for AITRIOS

There are three service plans for "Brain Builder for AITRIOS" each offering different features.

- **Basic Plan:** Allows you to create Classifier and Detector models and create up to 10 datasets.
- **Standard Plan:** Allows you to create Classifier and Detector models with unlimited dataset creation.
- **Pro Plan:** Provides access to all features, including Classifier, Detector, and Anomaly Recognizer, with unlimited dataset creation.

Table 2. Here is the summary of the features included in each plan:

	Basic	Standard	Pro
Classifier	✓	✓	✓
Detector	✓	✓	✓
Anomaly Recognizer			✓
Dataset Creation	Up to 10	Unlimited	Unlimited

To purchase a license for Brain Builder, follow the instructions on the [Local Studio product page](#) and contact your distributor/reseller for details.





Link to Local Studio product page:
<https://developer.aitrios.sony-semicon.com/en/studio/brain-builder>



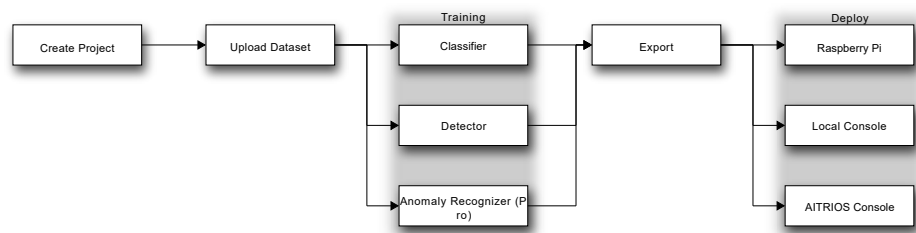
Anomaly Recognizer model Type is **NOT** supported by Local Console

1.4. Operation Flow of Brain Builder for AITRIOS

Here is the operation process flow for using Brain Builder for AITRIOS.



Basic Plan and Standard Plan do not support creating and training Anomaly Recognizer AI models. Find out more about [Plans for Brain Builder for AITRIOS](#).



2. Installing Brain Builder for AITRIOS



Before installing Brain Builder for AITRIOS, make sure that you are running on a compatible system and that you have all of the necessary files. Follow this checklist:

- 1. Your PC meets the minimum requirements, see [System Requirements](#).
- 2. You downloaded the installation package that fits your environment.
- 3. You have purchased the license file for Brain Builder for AITRIOS, see [Plans for Brain Builder for AITRIOS](#).

2.1. Install

1. Download the Brain Builder for AITRIOS installer package for your system. Follow the instructions on [the Brain Builder for AITRIOS product page](#).

2. Install the software.

Windows

- Unzip the installer.
- Run the installer using administration privileges.

Ubuntu

- Unzip the installer.
- Run the installer from a terminal window with the following commands:



Swap {filename} in `Neurala_BrainBuilderForAITRIOS_{filename}.run` depending on the installer file name.

Copy

```
$ chmod +x Neurala_BrainBuilderForAITRIOS_{filename}.run
$ ./Neurala_BrainBuilderForAITRIOS_{filename}.run
```

3. Brain Builder for AITRIOS will be installed on your system.



If you are unable to install or start Brain Builder, please check [Troubleshooting](#) for possible solutions.

2.2. Activate Your License

Before you can use Brain Builder for AITRIOS, you must have a valid license. There are different license plans for Brain Builder for AITRIOS with different features and functionality.

The process is:

1. Obtain your trial or paid license. This will be either a license file (filename ends with `.wibuCmRaU`) or an activation URL (which usually starts with `https://lc.codemeter.com/`).
2. Add the license to the CodeMeter application

Follow the instructions on [the Local Studio product page](#). You may need to contact your distributor/reseller for specific details.

Notes on License

Checking the License Expiration Date

Click the gear icon at the bottom left corner to open the "Account" page where you can find the license expiration date.

Importing a New License (Recommended Method)

We recommend activating a new license after the existing license expires.

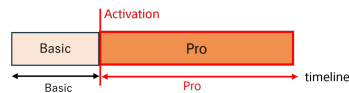


Figure 1. Import New License after Previous One Expires

Behavior When Importing Multiple Licenses

Brain Builder will prioritize by rearranging the order and applying the license with the most features first.

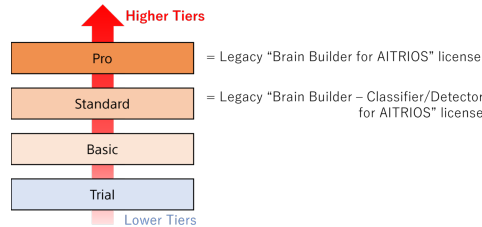


Figure 2. Rearranging the License Order

If you wish to import a new license while the previous one is still active, please ensure that you fully understand the license behavior below.



If you import licenses in the sequence of **Paid > Trial**, or **Trial #1 > Trial #2**, wait until your existing license expires before importing the new one. Importing before expiration may cause an error.

Case 1. Upgrading License (Lower Tier to Higher Tier)

You may import a new license at any time. The new license will be activated immediately, and Brain Builder will recognize the new license shortly. However, please note that the remaining validity period of the existing license will be disregarded.

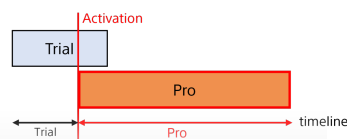


Figure 3. Upgrading License

Figure 3. Upgrading License

Case 2. Extending License (Same Tier)

You may import a new license at any time. The remaining period of the existing license will be combined with the new license period.

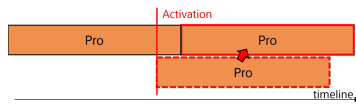


Figure 4. Extending License

Case 3. Downgrading License (Higher Tier to Lower Tier)

Please wait until the existing license expires before importing a new license. If you do not, the new license will be activated immediately while Brain Builder still recognizes the existing license, which may result in a loss of the validity period for the new license.

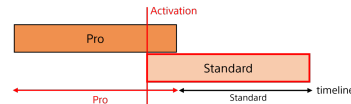


Figure 5. Downgrading License

Using Brain Builder on Multiple PCs

You need a separate license for each PC. If you wish to use Brain Builder on multiple PCs, you need to purchase a license for each one.



Figure 6. Activating License on Each PC

2.2.1. Download and Install CodeMeter™

We use CodeMeter licensing software to handle licenses.

1. [Download the CodeMeter User Runtime](#) for your environment.
2. Install the CodeMeter runtime on the system where you will install BrainBuilder for AITRIOS.

2.2.2. Activate your License**2.2.2.1. Activate Using a License File (Trial License)**

This is a file with the extension `.WibuCmRaU`.

Windows

1. Double click the file (Windows) or start CodeMeter and select **File / Import License** and then select the license file.

The **CodeMeter Control Center** will open and activate the license on the PC.

2. Confirm your license and expiration date on the CodeMeter Control Center.

Ubuntu

1. Open a terminal window and run the following command. Swap {license_file} for the actual license file name:

```
$ cmu -i -f {license_file}
```

Copy

2. Open **CodeMeter Control Center** and confirm your license and expiration date on the CodeMeter Control Center.

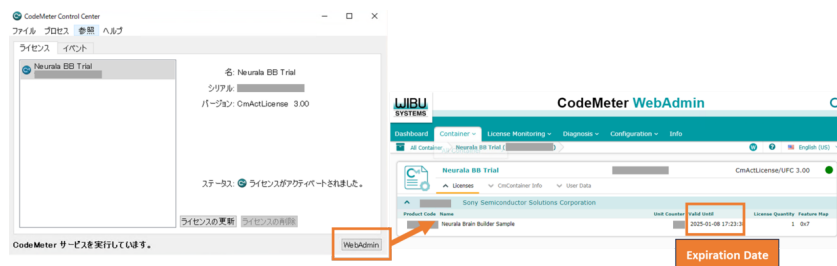


Figure 7. CodeMeter Control Center

3. You are now ready to install Brain Builder for AITRIOS. See [Install Brain Builder](#). If you already installed Brain Builder, you should restart your PC to reflect systems.



Please activate the license file before the expiration date indicated in the file name. For example, the expiration date of `BBTrialLicense251231f0x0000003WLd10.WibuCmRau` is 2025-12-31 (December 31, 2025).

2.2.2.2. Activate Using an Activation URL (Paid License)

1. Contact your distributor to purchase a license.
2. After purchase, you will receive an email containing an activation URL. This usually starts with <https://lc.codemeter.com/>.
3. Open the license activation URL in a browser.
4. On the web page, select **Activate Selected License Now** to activate and import the license to your computer.

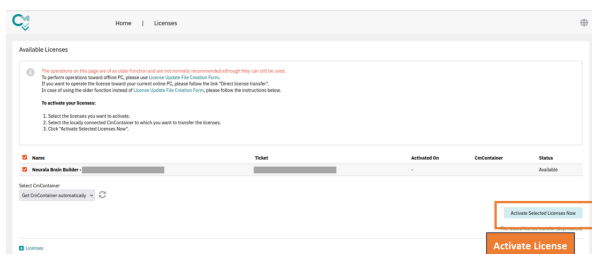


Figure 8. URL based activation

5. You are now ready to install Brain Builder for AITRIOS. See [Install Brain Builder](#). If you already installed Brain Builder, you should restart your PC to reflect changes.

2.3. Upgrade

To upgrade to the latest version of Brain Builder for AITRIOS, you need to obtain the newest Brain Builder installer. Then [run the installer](#) to upgrade the software.



If Brain Builder fails to start, and gives errors related to SDK services, consult the first item in [Troubleshooting](#) and restart Neurala related services.

2.4. Uninstall

Windows

Open **Add/Remove Programs** from Settings or Control Panel. Find NeuralaVIA in the list, and select **Uninstall**.



This process will leave your project files and settings on the computer. To permanently remove these files, delete the folders `C:\ProgramData\Neurala` and `C:\Program Files\Neurala`

Ubuntu

Open a terminal window and run the installer with the uninstall option:

```
$ ./NeuralaVIA-{filename}.run -- --uninstall
```

This process will leave your project files and setting on the computer. To permanently remove these files, delete the following folders by running the command:

```
$ sudo rm -rf /var/Neurala /var/log/Neurala /opt/Neurala
```

2.5. Initial Setup

The first time you use the application, you need to create an account.

1. Start Brain Builder.
2. Select the **Create Account** link under the **Log in** button.
3. Fill in the information for the account and then select **Create Account**.

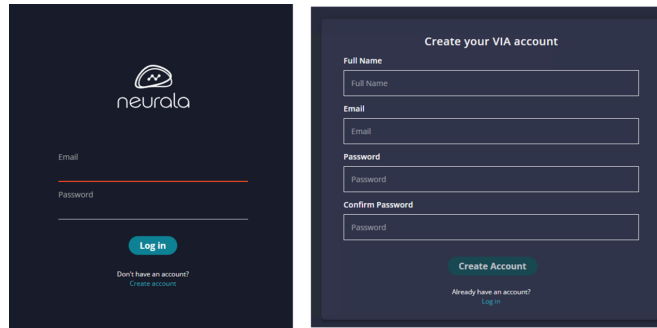


Figure 9. Create an account



Your account details will be saved locally. Please store your email and password somewhere safe as you may need to use them in future. See [Troubleshooting](#), if you forget your password.

3. Capturing Images for Training

Brain Builder's AI models are trained on open datasets and are generalized to recognize a wide range of objects. However, to create a model optimized for your deployment environment, it is most effective to use images captured by the actual camera that will be used, featuring the real-world objects it will encounter.

3.1. Camera Setup

Let's set up the camera in a fixed position and prepare proper lighting and the object(s) to be captured. We recommend the following guidelines for the setup:

- **Camera:** Use a camera equipped with an IMX500 sensor.
- **Camera position:** Position the camera as close to the target as possible. If available, mount it on a camera arm or tripod for stability.
- **Visibility:** Make sure that the object is clearly visible, occupies a large portion of the image, and appears sharp with detailed features.
- **Lighting:** Minimize reflections and ensure adequate lighting. Additionally, if possible, include images taken under various lighting conditions.
- **Background:** Keep the background as simple as possible and minimize variations.

3.2. Prepare Your Objects for Capture

Prepare the objects you wish to capture. The installer includes several sample datasets to provide you with examples of suitable objects for capture. See [Using the Provided Datasets in Brain Builder for AITRIOS](#) for more detail on these datasets.

- Classifier:
 - Printed shapes (folder name: `PrintedShapes_classifier`)
- Detector:
 - Printed shapes (folder name: `PrintedShapes_Detector`)
 - Japanese coins (folder name: `JapaneseCoins_Detector`)
- Anomaly Recognizer:
 - Printed shapes (folder name: `PrintedShapes_AnomalyRecognizer`)
 - Coffee beans (folder name: `CoffeeBeans_AnomalyRecognizer`)

3.3. Dataset Image Requirement

- **Resolution:** You can use images with a resolution larger than the input tensor (e.g., 512x512). In such cases, the image will be rescaled to the size of the input tensor, which may result in the loss of fine details contained in the image.
- **Number of images:** We recommend starting with the minimum requirements and gradually building up the datasets. The minimum requirements vary for each model type. Read the following [tips](#) for more details.
- **Formats:** Supported image formats are JPEG (.jpeg, .png), PNG (.png), and bitmap (.bmp).

3.4. Tips for Each Model Type

Classifier

Description	Each image should contain only one object or product.
Input Tensor	256x256px
Minimum Detection Size	Larger than 10x10px in a 256x256px image, and maintaining this ratio for larger images.
Number of Images	Provide 50+ images per class.

Detector

Description	Include all variations of the objects. The image can include multiple instances of the same object as well as different objects.
Input Tensor	320x320px
Minimum Detection Size	Larger than 12x12px in a 320x320px image, and maintaining this ratio for larger images.
Number of Images	Provide 50+ images. Each class with 150+ tagged objects.

Anomaly Recognizer/Hi-Fi

- **Description:** Each image should contain only one object or product.
- **Input Tensor and Minimum detection size:** Varies according to the defect sizing setting:

Defect Sizing	Input Tensor	Minimum Detection Size
Coarse	256x256px	16x16px
Regular	256x256px	8x8px
Fine	512x512px	8x8px
Ultra-fine	512x512px	4x4px

- **Number of Images:** Provide 50+ images of "Normal" and 5-10 images of anomalies. Additional images

may be needed depending on the use case.

3.5. Capture Images with Camera

Follow the specific guide for your camera on image capture:

- [Raspberry Pi AI Camera](#)
- [Sony AIH-IVRW2/Rayprus CSV26](#)

4. Creating Your Project and Datasets

Once you have your datasets on hand, you are ready to build your first Brain Builder project.

4.1. Create a Project

1. Select **Create project** in the upper-right corner of the screen.
2. Enter a project name.
3. Select **Create**.



Within each project, you can create multiple datasets and different AI models.

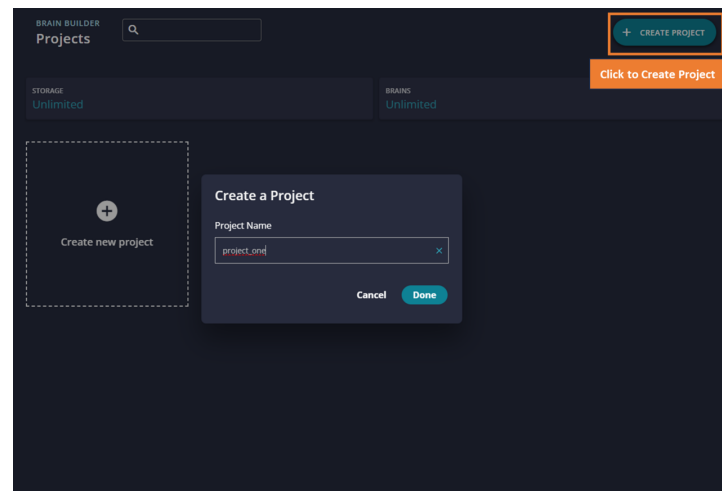
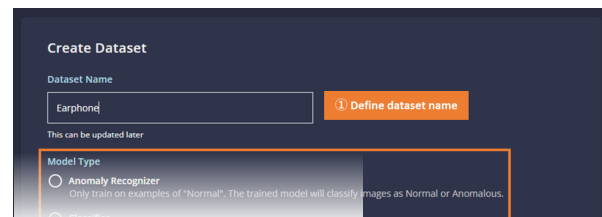


Figure 10. Create project

4.2. Create a Dataset

1. Select **Create dataset** in the upper-right corner.
2. Select a model type from the following options:
 - Classifier
 - Detector
 - Anomaly Recognizer

Tip: Review the descriptions in [What can a Brain Builder AI Model Do?](#) to help you choose the most appropriate model type.



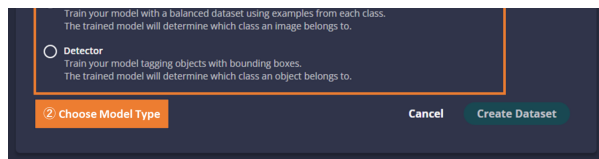


Figure 11. Create dataset and select model type selection

4.3. Training Using Brain Builder

Select the model type you will be training for the next steps:



Training a Detector model on a CPU-only machine may require significant time to complete.

- [Classifier](#)
- [Detector](#)
- [Anomaly Recognizer](#)

5. Classifier

5.1. Introduction

What is Classifier?

The Classifier model efficiently classifies images into predefined classes, providing labels ranked by confidence (for example, "apple: 80%", "peach 40%").

It operates on the image without pinpointing specific objects, resulting in faster and more accurate results when categorizing precise tasks.

Classifier types in Brain Builder

- **Learning Classifier:** Performs automatic learning during data upload
- **Static Classifier:** Performs required training after the data upload

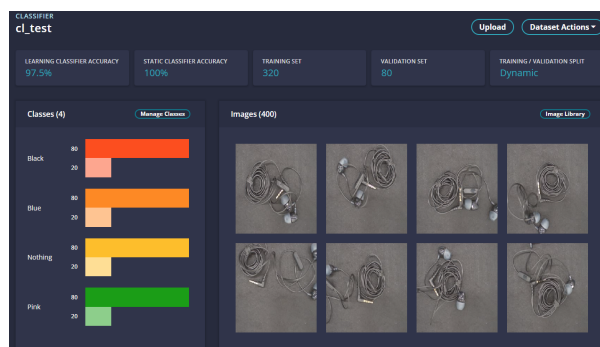


Figure 12. Operation flow for classifier

5.2. Upload Datasets

1. Go to the Dashboard page and select **Upload data**.
2. Choose either Dynamic Split or Static Split:

- **Dynamic Split:** Brain Builder automatically divides your images into training and validation sets. Note that ground images may change each time you upload the same data.

grouped images may change each time you upload the same data.

- **Static Split:** Uses your predefined dataset divisions. Choose this option when you need to train your model with identical training and validation datasets.



The upload process might take some time to complete as Brain Builder evaluates your images using a learning classifier in the background.

5.2.1. Dynamic Split

You can use either file-based or folder-based dynamic split.



Tips for class label names:

- Use only numbers (0-9), letters (A-Z, a-z), dashes (-), and underscores (_)
- Don't include spaces

File-based

Use this option to choose images and label classes manually: 1. Select **File-based** 2. Select **Browse** 3. Enter a class label in the text box 4. Select **Start upload**

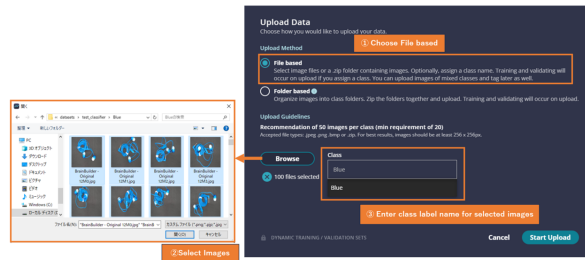


Figure 13. File Based upload process

Folder based

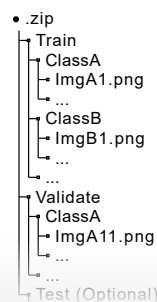
Upload a ZIP file with folders named by label and with images inside each folder. Brain Builder uses the folder names as class labels.



Figure 14. Template folder structure for the ZIP file

5.2.2. Static Split

Upload a ZIP file containing folders named "train", "validate", and optionally "test", with labeled subfolders containing images. Brain Builder uses this structure during training and assigns class labels based on the subfolder names.



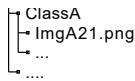


Figure 15. Template folder structure for the ZIP file



If you have trouble uploading datasets, see [Troubleshooting](#).

5.3. Evaluate Datasets with Learning Classifier

After upload completes, the Learning Classifier results appear. The Brain Score displays on the right side as one of these values:

- Great
- Good
- Low
- Still Learning

Use this score to determine if your dataset is suitable for training. To improve your score to "Great," select **Upload** in the upper right corner and add more images.

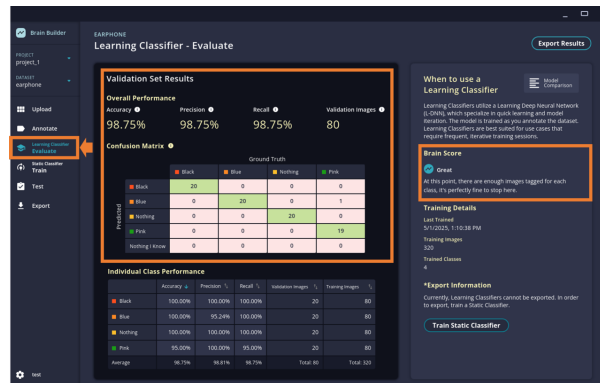


Figure 16. Learning Classifier result shows "Great"

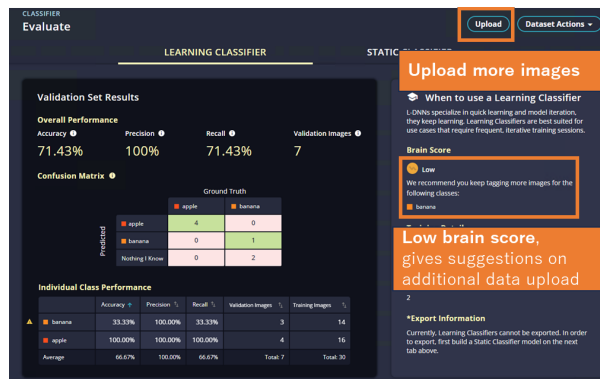


Figure 17. Learning Classifier result shows "Low" - upload more images directly

5.4. Train with Static Classifier

The Static Classifier lets you adjust processing time and false positive (FP) tolerance based on your requirements. Select **Build static classifier** to start training.

Table 3. Duration (optimization comprehensiveness)

Duration	Processing Detail

Quick	Use the predefined AI model for training
Balanced	Tests with various AI models and trains with the optimal model
Thorough	Tests with all AI models and trains with the optimal model



Brain Builder tests multiple AI models and augmentations with your input images. Longer processing durations produce more optimized models for your datasets.

Table 4. Performance (false positive occurrence)

Performance	Meaning
Highest average accuracy	Balances accuracy and false detection during inference
Highest accuracy	Improves accuracy by using relaxed criteria for determining correct answers
Lowest occurrence rate of false positives	Reduces false detection by using strict criteria for determining correct answers

5.5. Analyze Training Results

The Static Classifier produces the following inference results:

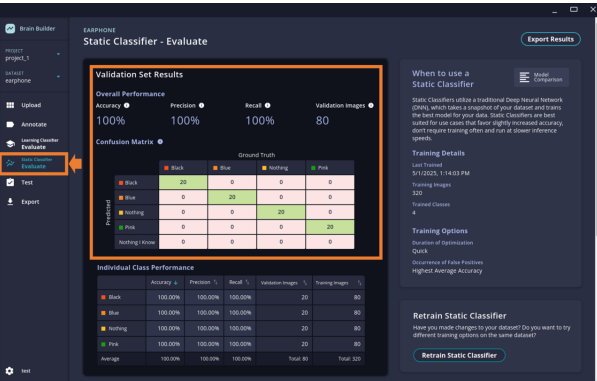


Figure 18. Static Classifier result

Table 5. Definition of TP,FP,FN,TN

	Predicted Negative	Predicted Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

Accuracy

The percentage of brain predictions that is correct - either for a single class or averaged for all classes.

Accuracy is calculated as: **Count of correct predictions / Count of all predictions**

Precision

Precision measures how many of the Brain’s predictions are correct for a given class (or averaged for all classes). For example, if the Brain makes 10 predictions of Class A, and 8 of those predictions are correct, then the Brain has 80% precision for that class.

Precision is calculated as: **TP / (TP + FP)**

Recall

Recall measures the percentage of a class that the Brain correctly identifies.

For example, if there are 10 images of Class A and the Brain correctly identifies 7 of them, then recall for that class is 70%.

Recall is calculated as: $TP / (TP + FN)$



These results may vary slightly from the Learning Classifier type since it uses a different architecture.

5.6. Test the Model (Optional)

Testing is optional.

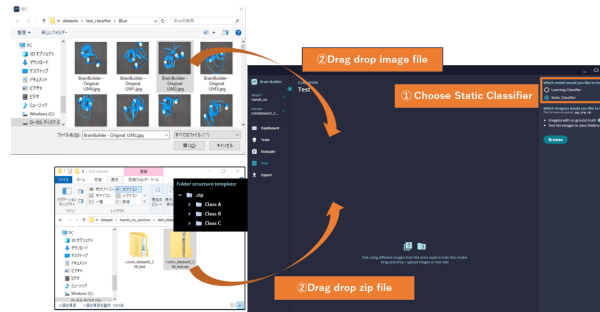


Figure 19. Testing the model

You can:

- Drag images directly onto the screen
- Upload a ZIP file containing images grouped by class label names (recommended for better accuracy verification)



Use different images for testing than those used for training.

5.7. Export Your Model

1. Select **Download** on the model you want to export
2. Specify the save location
3. Output the AI model



This process might take some time as Brain Builder converts and quantizes the model to ensure compatibility with the IMX500.

6. Detector

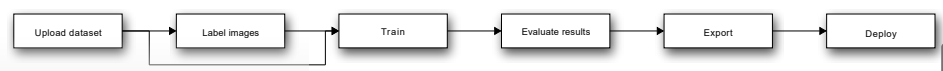
6.1. Introduction

What is Detector

The Detector model type can identify and locate multiple objects within an image.

It outputs bounding boxes around the detected items and labels them with a confidence score (for example, "car: 90%" at (x,y)).

This is great for detecting or counting objects as it effectively identifies what objects are and where they are.



6.2. Upload Dataset and Label Images

The detector supports labeling inside Brain Builder or loading pre-labeled datasets. Select **Upload data** to begin.

Image upload and labeling

1. Select and upload all images for training.

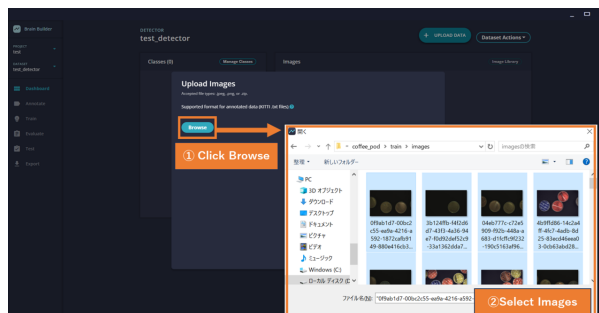


Figure 20. Load images to Brain Builder

2. Create a new class and type its name in the text box.

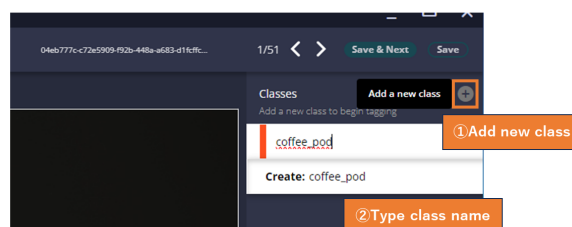


Figure 21. Create a new class

3. Select the label in the upper right corner and start selecting target objects in the images.

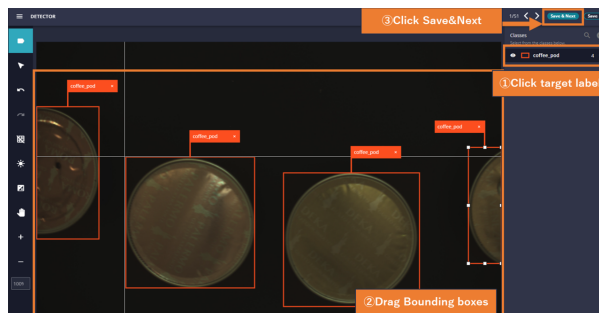


Figure 22. Apply the class and insert bounding boxes

4. Repeat this process for the remaining images.



Tips for class label names:

- Use only numbers (0-9), letters (A-Z, a-z), dashes (-), and underscores (_)
- Don't include spaces

Uploading Labeled Dataset

Brain Builder also supports direct annotated data (KITTI format) upload. In KITTI format:

- Each corresponding image and label should have the same filename
- Both dynamic and static folder architectures are supported (as in Classifier and Anomaly Recognizer)

KITTI Format for Object Detection

In the KITTI format, the class name is in the 1st element, and bounding box coordinates begin at the 5th element.

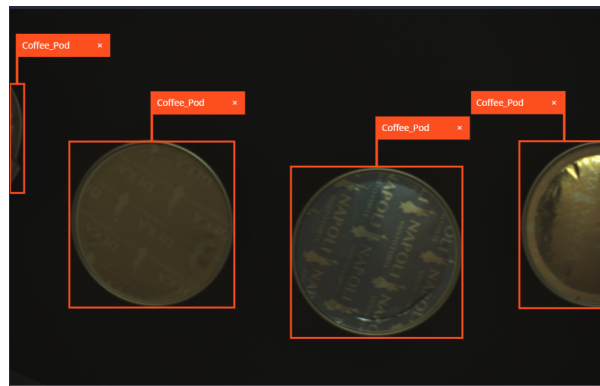


Figure 23. Example image with bounding boxes

For example, the bounding boxes for the above image are written as:

Copy

```
Coffee_Pod 0 0 0 0 242 46 594 0 0 0 0 0 0 0
Coffee_Pod 0 0 0 192 428 725 964 0 0 0 0 0 0 0
Coffee_Pod 0 0 0 908 509 1462 1062 0 0 0 0 0 0 0
Coffee_Pod 0 0 0 1646 427 1936 966 0 0 0 0 0 0 0
```

These files are saved as text(.txt) format and stored in the `labels` directory.

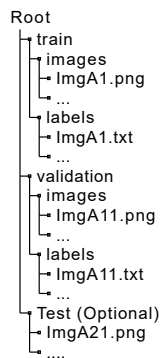


Figure 24. File structure of KITTI format



If you have trouble uploading datasets, see [Troubleshooting](#).

6.3. Start Training

For detector models, you can customize your training duration. Select **Train** to begin.

Table 6. Duration (training comprehensiveness)

Duration	Processing Detail
Quick	Tests with some AI models and trains with the optimal model
Balanced	Tests with most AI models and trains with the optimal model
Thorough	Tests with all AI models and trains with the optimal model





Brain Builder tests multiple AI models and augmentations with your input images. Longer processing durations produce more optimized models for your datasets.

6.4. Evaluate Training Results and Improve Your Model

After training completes, the AI model is validated against your validation dataset, providing quantitative evaluation metrics for precision and recall.

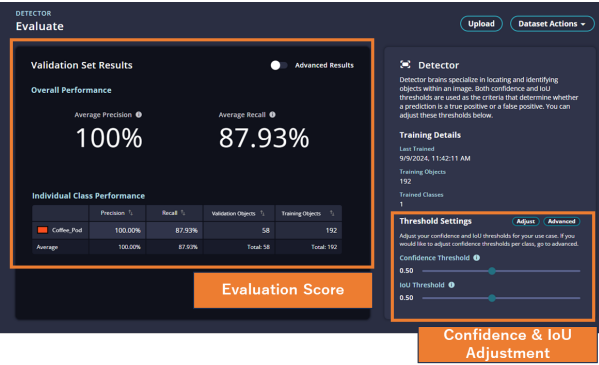


Figure 25. Training result

Precision

The proportion of correct predictions (true positives, TP) out of all predictions made by the AI as positive (TP + FP). Also known as false detection rate: a high value indicates few false detections, while a low value suggests many false detections.

Precision is calculated as: $TP / (TP + FP)$

Recall

The proportion of correct predictions (TP) out of all actual positive cases (TP + FN). A high value indicates fewer missed true positives. A low value indicates more missed true positives. Recall is calculated as: $TP / (TP + FN)$

Definitions of TP, FP, and FN in Object Detection

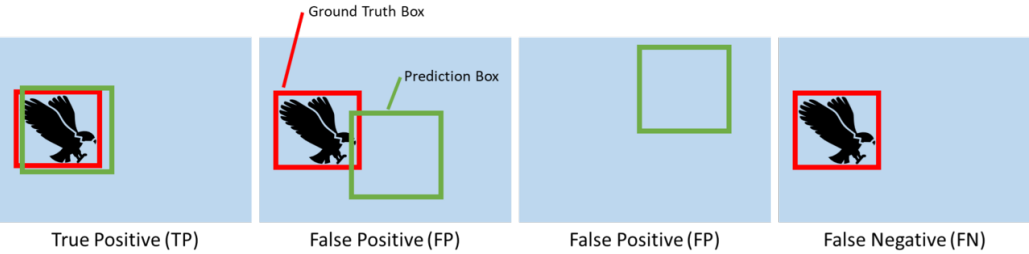


Figure 26. Object detection evaluation metrics illustrated

True Positive (TP)

When the target object and detection box have an IoU overlap higher than the specified threshold.

False Positive (FP) [false detection] Left

When there is a target object in the image, but the detection box has an IoU overlap lower than the specified threshold or doesn't overlap.

False Positive (FP) [false detection] Right

When there is no target object in the image, but an inference box appears.

False Negative (FN) [missed detection]

When there is a target object in the image, but no detection box appears.



6.4.1. Improve Your Model (Optional)

You can adjust these two parameters to fine-tune your model:

Confidence Threshold

The AI model calculates a confidence level for each detection box within a range of 0 to 1. The confidence threshold determines which detections are considered valid and can be adjusted in the UI. The default is 0.5, but you should adjust it based on your dataset and objectives.

- Lower confidence threshold: Classifies uncertain cases as positive, increasing recall score
- Higher confidence threshold: Classifies uncertain cases as negative, increasing precision score

IoU (Intersection over Union) Threshold

The IoU value represents the overlap between the detection box output by the AI model and the correct area. The IoU threshold determines whether a detection box is counted as a true positive (TP).

- Lower IoU threshold: Allows more detections, even with lower overlaps
- Higher IoU threshold: Requires the AI model to return only high-overlap predictions

6.5. Test the Model (Optional)

1. Select **Test** on the left side
2. Select **Browse** on the right side and select an image



Use different images for testing than those used for training.

As shown below, the AI model performs inference on new images and displays detection box overlays.

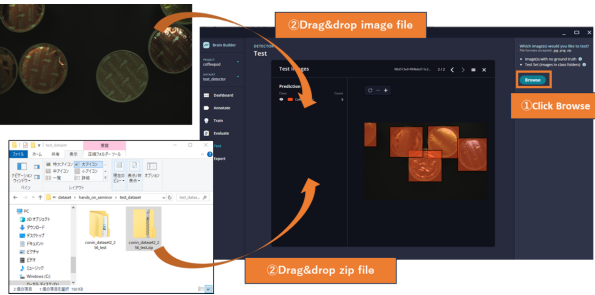


Figure 27. Test results with detection boxes

6.6. Export Your Model

1. Select **Download** on the model you want to export
2. Specify the save location
3. Output the AI model



This process might take some time as Brain Builder converts and quantizes the model to ensure compatibility with the IMX500.

7. Anomaly Recognizer

7.1. Introduction

What is Anomaly Recognizer?

The Anomaly Recognizer model type identifies deviations from normal patterns and determines whether an image is "normal" or "anomaly". It can also localize areas of possible anomalies with a heatmap (Hi-Fi). Unlike the Classifier model, it uses a few anomaly images for validation, not for training, making it suitable

when anomalies are rare or their patterns are varied.



Anomaly Recognizer model Type is **NOT** supported by Local Console

Anomaly Recognizer types in Brain Builder

- **Anomaly Recognizer:** Provides a binary label: "normal" or "anomaly". Performs automatic learning during data upload
- **Anomaly HiFi:** Generates a heatmap which highlights specific areas with anomalies. Performs required training after the data upload



Figure 28. Operation flow for Anomaly Recognizer

7.2. Upload Datasets

1. Go to the Dashboard page and select **Upload data**.
2. Choose either Dynamic Split or Static Split:
 - **Dynamic Split:** Brain Builder automatically divides your images into training and validation sets. Note that grouped images may change each time you upload the same data.
 - **Static Split:** Uses your predefined dataset divisions. Choose this option when you need to train your model with identical training and validation datasets.



The upload process might take some time to complete as Brain Builder evaluates your images using the Anomaly Recognizer model in the background.

7.2.1. Dynamic Split

You can use either file-based or folder-based dynamic split.



Tips for class label names:

- Use only numbers (0-9), letters (A-Z, a-z), dashes (-), and underscores (_)
- Don't include spaces

File-based

Choose images and label classes manually:

1. For normal images:
 - Select **File-based**
 - Select **Browse**
 - Choose **Normal** class
 - Select **Start upload**
2. For anomaly images:
 - Select **File-based**
 - Select **Browse**
 - Choose **Anomaly** class
 - Select **Start upload**

Figure: File-based upload process image::anomaly-1.png[width=500,align=center]

Folder based

Upload a ZIP file containing folders named by label, with images inside each folder. Brain Builder uses the folder

Neural AI, LLC, Exhibit 2043

Page 2043 - 20

IPR2025-00609, NVIDIA Corporation v. Neural AI, LLC

7/28/2025, 4:12 PM

names as class labels.

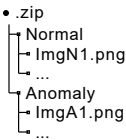


Figure 29. Template folder structure for the ZIP file

7.2.2. Static Split

Upload a ZIP file containing folders named "train", "validate", and optionally "test", with labeled subfolders containing images. Brain Builder uses this structure during training and assigns class labels based on the subfolder names.

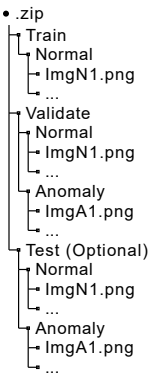


Figure 30. Template folder structure for the ZIP file



Do not include Anomaly class images in the `Train` folder. Training only uses Normal class images.



If you're having trouble uploading datasets, please refer to [Troubleshooting](#).

7.3. Evaluating the Datasets

After upload completes, Anomaly Recognizer results appear. Review these results to determine if the datasets are suitable for training. You'll later use Anomaly Hi-Fi to create and train the brain for IMX500 devices.

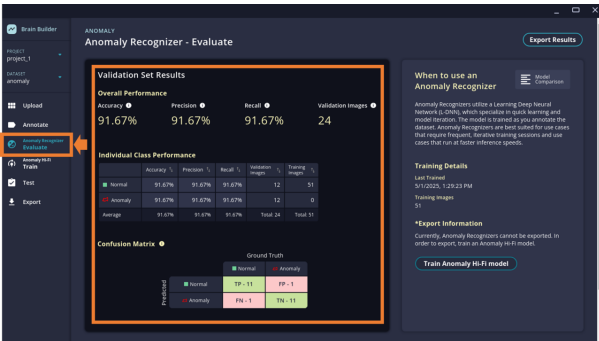


Figure 31. Evaluating datasets using Anomaly Recognizer

Optionally, you can filter incorrect results to identify what types of images the brain needs more of. Then, add n images by selecting `Upload` in the top right.

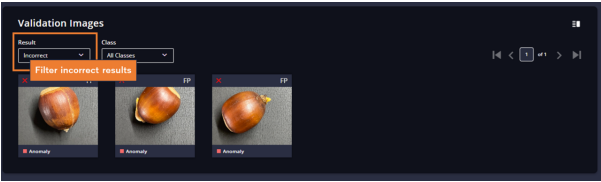


Figure 32. Filter incorrect results

7.4. Training with Anomaly Hi-Fi

Anomaly Hi-Fi lets you adjust the processing time and defect size. Customize these settings based on your needs. Select **Build Anomaly Hi-Fi** to start training.

Duration setting

Brain Builder tests multiple AI models and augmentations with your input images. Longer processing durations produce more optimized models for your datasets.

Table 7. Duration (optimization comprehensiveness)

Duration	Processing Detail
Quick	Use the predefined AI model for training
Balanced	Tests with various AI models and trains with the optimal model
Thorough	Tests with all AI models and trains with the optimal model

Defect size setting

Defect size choice are linked to the input tensor and its minimum defect detection size. Please select your setting based on your datasets. Input tensor is the image resolution that AI model uses for training and inference.

Table 8. defect size setting:

Defect Sizing	Input Tensor	Minimum Detection Size
Ultra-fine	512x512px	4x4px
Fine	512x512px	8x8px
Regular	256x256px	8x8px
Coarse	256x256px	16x16px

7.4.1. Evaluate Training Results

Anomaly Hi-Fi outputs the anomaly degree of each region in an image as a heatmap in a two-dimensional matrix. By setting a threshold on this heatmap, an anomaly region map is created.

The definitions of related terms are shown below.

Table 9. Definitions of TP, FP, FN, TN

	Ground Truth Anomaly	Ground Truth Normal
Predicted Anomaly	True Positive (TP)	False Positive (FP)
Predicted Normal	False Negative (FN)	True Negative (TN)

Accuracy

The ratio of both normal and abnormal predictions that the AI model correctly inferred for all images.

Accuracy is calculated as follows: $(TP + TN) / (TP + TN + FP + FN)$

Precision

The ratio of correct predictions among all anomaly predictions made by the AI model. It is also the probability of false detection.

Precision is calculated as follows: $TP/(TP + FP)$

Recall

The ratio of correctly inferred anomaly images among the ground truth anomaly images. It is also the probability of not missing anomalies.

Recall is calculated as follows: $TP / (TP + FN)$

7.4.2. Tune Anomaly Hi-Fi Heatmap (Optional)

You can adjust the degree of anomaly using the sensitivity threshold in the Anomaly Hi-Fi heatmap to create a map with anomaly region. This value is exported as `pixelThreshold` in the `neurala_hi-fi_pp1_parameters.json` file.

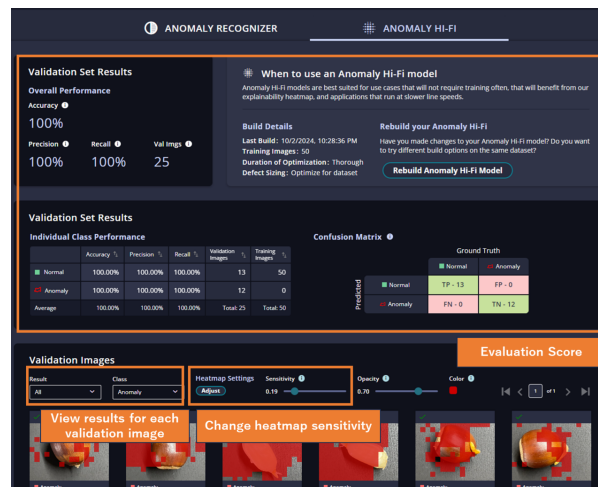


Figure 33. Anomaly Hi-Fi heatmap interface with sensitivity threshold

7.5. Test the Model

In Test, you can evaluate AI model's accuracy for images other than the training images.

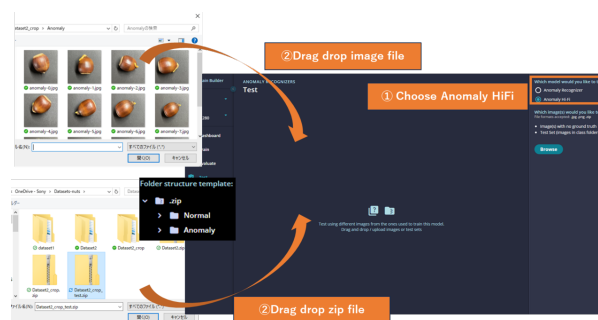


Figure 34. Testing the model

You can:

- Drag images directly onto the screen
- Upload a ZIP file containing images grouped by class label names (recommended for better accuracy verification)

7.6. Export Your Model

1. Select **Download** on the model you want to export
2. Specify the save location

3. Output the AI model



This process might take some time as Brain Builder converts and quantizes the model to ensure compatibility with the IMX500.

8. Deploy on AITRIOS Camera

Follow the specific guide for your camera:

- [Raspberry Pi AI Camera](#)
- [Sony AIH-IVRW2/Rayprus CSV26](#)
- [LUCID Vision Labs Triton Smart](#)

9. Troubleshooting

9.1. Unable to Install Brain Builder

- Try reinstalling Brain Builder by first uninstalling the existing software
- If the issue persists, retrieve the log and contact your distributor:
 - **Windows:** Enter "%temp%" in the Windows search box. Look for a folder starting with "Neurala VIA" that contains log files.
 - **Ubuntu:** When starting the Brain Builder installation (.run file) via command console, the console displays the log.

9.2. Unable to Start Brain Builder

9.2.1. "SDK Service" error

- **Windows:**
 1. Type "Services" in the search bar
 2. Find "Neurala SDKService," right-click it, and select "Restart"
- **Ubuntu:**
 1. Open Terminal and restart SDKService:

```
sudo systemctl restart neurala_sdkservice.service
```

[Copy](#)

9.2.2. "Brain Bank" and other service-related errors

- **Windows:**
 1. Type "Services" in the search bar
 2. Select "Run as administrator"
 3. Find services beginning with "Neurala," right-click each one, and select "Restart"
- **Ubuntu:**
 1. Open Terminal and restart services:

```
sudo systemctl restart neurala-brainbank-database.service neurala-brainbank.service  
neurala_detectiontrainingservice.service neurala_sdkservice.service
```

Alternatively, restart your PC to restart all services.

9.2.3. Windows blocking the application

Always select "allow" if Windows blocks the application



9.3. Create Datasets

- If Brain Builder shows an error when importing a dataset, make sure your ZIP file follows the correct folder structure and labels don't contain spaces. Each image must have a unique name.
- To find the cause of upload errors, select **Show Upload Details** in the upload images dialog, go to **Failed Images**, and hover over the "i" icon.



Figure 35. Upload fails

9.4. Training Related

- **GPU memory unavailable after training a detection model:** Restart the Neurala Detection Training Service to release GPU memory.
- **Incorrect imageThreshold value from Anomaly Hi-Fi model:** Don't use anomalyScore and imageThreshold. Use heatmap and pixelThreshold with Brain Builder v25.05 and earlier.

1. The image and pixel threshold in Anomaly HiFi does not work correctly for inference results on the device.

- The two thresholds are supposed to set whether the image or a pixel is normal or anomaly.
- However, since the model is sensitive to environmental changes (e.g., lighting and angle changes), the threshold outputted by Brain Builder should be considered as an estimation on a PC.
- Therefore, we recommend adjusting the threshold accordingly during device inference.

2. Slow performance on evaluation results page when using very large datasets for training

- Brain Builder attempts to process all of the validation images which is taking a long time. Consider using smaller datasets to speed up the process.

9.5. Login Related

• Forgot login password:

- **Windows:** Open PowerShell, go to `C:\Program Files\Neurala\BrainBank\bin`, and run: `.\db-tool.exe reset-password <email> <newPassword>`
- **Ubuntu:** Open Terminal, go to `/opt/Neurala/BrainBank/bin`, and run: `./neurala-brainbank-db-tool reset-password <email> <newPassword>`

9.6. Other Issues

Check the known issues in the latest release notes and Neurala's Official Knowledge Base (<<https://support.neurala.com/docs/via-for-sony-aitrios>>).

If issues persist, retrieve logs and contact your distributor

To export logs, go to application settings. On the "About" tab, select **Generate Logs** and save to your desired location.

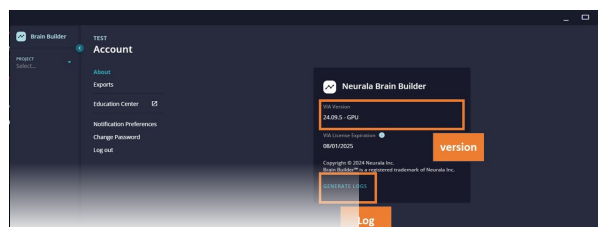




Figure 36. Retrieve logs

Appendix A: Using the Provided Datasets in Brain Builder for AITRIOS

The installer includes sample datasets to provide ideas for suitable objects. To import into Brain Builder:

- For Classifier and Anomaly Recognizer: Choose **Dynamic training and validation sets** and **Folder-based** in the upload dialog, then upload the ZIP file
- For Detector: Directly select the ZIP file for upload

Classifier

- **Printed shapes** (folder name: `PrintedShapes_classifier`)
 - Three-class classification on circle, square, and triangle shapes
 - Size: approximately 42 MB
 - Number of images: 516

Detector

- **Printed shapes** (folder name: `PrintedShapes_Detector`)
 - Three-class detection on circle, square, and triangle shapes
 - Size: 69 MB
 - Number of images: 516
- **Japanese coins** (folder name: `JapaneseCoins_Detector`)
 - Six-class detection dataset using 1, 5, 10, 50, 100, and 500 yen coins
 - Size: 36 MB
 - Number of images: 300

Anomaly Recognizer

- **Printed shapes** (folder name: `PrintedShapes_AnomalyRecognizer`)
 - Shape change detection: Circle shape as normal, square and triangle as anomaly
 - Size: 39 MB
 - Number of images: 110
- **Coffee beans** (folder name: `CoffeeBeans_AnomalyRecognizer`)
 - Foreign objects inside coffee beans
 - Size: 46 MB
 - Number of images: 100



The original printed shapes datasets are available at <https://github.com/SonySemiconductorSolutions/aitrios-rpi-dataset-sample>. You can also print the shapes yourself to create your own datasets.

Appendix B: Uninstall Brain Builder



Your project files and settings will remain on the computer.



B.1. Uninstall in Windows

Open **Add/Remove Programs** from the system page, find "Neurala VIA" and select **Uninstall**.

B.2. Uninstall in Ubuntu

Open a terminal window and run the installer with the uninstall option:

```
$ ./NeuralaVIA-{filename}.run -- --uninstall
```

[Copy](#)

10. Trademark Acknowledgements

All product names, company names, logos, and trademarks mentioned in this document are the property of their respective owners.

AITRIOS™ and **AITRIOS logos** are the registered trademarks or trademarks of Sony Group Corporation or its affiliated companies.

Brain Builder and Neurala logo are the trademarks of Neurala, Inc.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Windows and **PowerShell** are the trademarks of Microsoft Corporation.

Ubuntu is a trademark of Canonical Ltd.

CodeMeter is a trademark of WIBU SYSTEMS AG.

Ethernet is a trademark of XEROX CORPORATION.

ArenaView, **Triton Edge**, and **LUCID logo** are the trademarks of Lucid Vision Labs, Inc.

GigE Vision, is a trademark of Automated Imaging Association.

Modbus is a trademark of Schneider Automation Inc.

EtherNet/IP is a trademark of ODVA, INC.

*AITRIOS and AITRIOS logos are the registered trademarks or trademarks of Sony Group Corporation or its affiliated companies.

[Sony Group Portal](#)[Privacy Policy](#)[Manage Cookies](#)[Terms of Use](#)

© 2025 Sony Semiconductor Solutions Corporation



