EdgeBench Documentation



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Table of Contents

- User Guide
  - User Guide
  - About
    - About EdgeBench
  - Getting Started
    - Getting Started with EdgeBench
  - Key Concepts
    - Key Concepts
  - Workspace Navigation
    - Navigating the Workspace
  - Workspace Management
    - Workspace Tasks
  - Exploring Model Zoo
    - Exploring the ADI Model Zoo
  - Exporting Builds for CFS
    - CodeFusion Studio Builds
  - Prepare for BYOM
    - Preparing for BYOM
    - Data Processing Scripts
      - Data Processing
  - Build Process
    - Build Process Overview
    - Device Selection
      - Device Selection
    - Configure and Compile
      - Configure and compile
    - Upload Input Data
      - Upload Input Data

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Add a Pre-processing Script

Pre-Processing Script

Add a Post-processing Script

Post-Processing Script

Deploy Inference

Deploy and Inference

Check Inference Results

Inference Results

Frequently Asked Questions

Frequently Asked Questions

Support

Support Resources

Terms of Use

Terms of Use

Release Notes

Release Notes

1.2.0

1.1.0

1.0.0

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

User Guide

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

User Guide

EdgeBench is a cloud-based platform designed to streamline the evaluation and benchmarking of AI/ML models on Analog Devices (ADI) embedded hardware. It eliminates the complexities traditionally associated with embedded AI development such as hardware setup, tools setup, and model deployment by offering a browser-accessible interface that enables remote compilation, deployment, and performance analysis.

Watch

EdgeBench Demo Video Using a Model from the Model Zoo



- About
- Logging In
- Application Overview
- Navigating the Workspace
- Workspace Tasks
- Exploring Model Zoo
- Preparing for BYOM
- Exporting Builds for Codefusion Studio
- Build Process Overview
- Build Step 1 - Device Selection
- Build Step 2 - Configure and Compile
- Build Step 3 - Upload Input Data
- Build Step 4 - Pre-Processing Script
- Build Step 5 - Post-Processing Script
- Build Step 6 - Deploy and Inference
- Build Step 7 - Inference Results

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Note

You can toggle between light and dark mode using the sun and moon icons on the top right of the page.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

About

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

About EdgeBench

EdgeBench is a cloud-based platform designed to streamline the evaluation and benchmarking of AI/ML models on Analog Devices (ADI) embedded hardware. It eliminates the complexities traditionally associated with embedded AI development such as hardware setup, tools setup, and model deployment by offering a browser-accessible interface that enables remote compilation, deployment, and performance analysis.

At the heart of the platform is ADI's Remote Board Farm, a scalable and resilient infrastructure that provides secure access to a variety of ADI evaluation boards. With consistent lab conditions and 24/7 secure access, teams reduce evaluation delays and derisk Edge AI development.

EdgeBench allows users to run inference on actual hardware, gaining real-world insights into model performance. Users get hardware-accurate latency, memory, and behavior metrics, enabling confident device selection and faster iteration.

Key features

- **Remote Hardware Access**: Test models from anywhere—no manual setup or shipping required.
- **Zero-setup, zero-touch workflow**: No firmware flashing, SDK installation, or physical bring-up required.
- **ADI Model Zoo Integration**: Access a curated library of pre-trained models or upload your own AI/ML models for testing.
- **Bring Your Own Model**: Seamlessly upload supported models for remote evaluation and benchmarking.
- **Browser-Based Workflow**: Compile, deploy, and test models directly through a web interface—no local installation needed.
- **Real Hardware, Real Results**: No simulations—observe actual latency, memory usage, and inference behavior.

Use cases

- **AI Workload Validation**: Assess whether ADI solutions meet your specific AI performance needs.
- **Model Benchmarking**: Compare multiple models under consistent, real-world conditions.
- **Early Stage Exploration**: Begin development and testing without the need to ship physical boards.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Supported devices

EdgeBench enables seamless model deployment across a variety of ADI digital devices. The ADI Remote Board Farm hosts and integrates evaluation boards with the client application.

Device specifications

Device Name	Description	Key Specifications
MAX78002	The MAX78002 is an advanced system-on-chip featuring an Arm Cortex-M4 with FPU CPU for efficient system control and with hardware-based convolutional neural network (CNN) accelerator optimized for deep learning tasks.	• Arm Cortex-M4 processor (up to 120 MHz) and a RISC-V coprocessor (up to 60 MHz) • 2.5 MB Flash, 384 KB SRAM, and 1.3 MB CNN data memory • Support for 1-, 2-, 4-, and 8-bit weights with up to 2 MB weight storage • Interfaces for camera (MIPI CSI-2, 12-bit parallel), audio (I^2S), and SD/eMMC storage • Built-in CNN Accelerator
MAX32690	The MAX32690 features an Arm Cortex-M4 with a floating-point unit (FPU) Microcontroller and Bluetooth LE 5, suitable for industrial sensors and health/medical wearables applications.	• 120 MHz Arm Cortex-M4 Processor with FPU • Ultra-Low-Power, 32-Bit RISC-V (RV32) Coprocessor • 3.25 MB Flash and 1 MB SRAM, with external memory expansion support

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Device Name	Description	Key Specifications
		• Integrated Bluetooth 5.2 Low Energy (LE) radio
ADSP-SC835	The ADSP-SC835 is a high-performance DSP, part of the SHARC-FX family, designed for demanding signal processing applications such as professional audio, automotive, and industrial systems.	• SHARC-FX DSP core running up to 1 GHz • Integrated Arm Cortex-M33 connectivity core at 400 MHz • 256-bit SIMD vector architecture • Up to 2 MB of on-chip L2 memory • Built-in FIR and IIR accelerators

Supported AI models

EdgeBench supports two primary model sources:

1. **ADI Model Zoo:** A curated repository of pre-trained models developed by Analog Devices, optimized for supported evaluation boards.
2. **Bring Your Own Model (BYOM):** Upload and deploy custom AI pre-trained models using PyTorch, TensorFlow, and ONNX.

Recommended browsers

- MacOS Sequoia 15.6 – Safari Version 18.6
- Windows 11 – Chrome Version 139.0.7258.127 (64-bit)
- Windows 11 – Edge Version 139.0.3405.86 (64-bit)

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- Ubuntu 22.04 – Mozilla Firefox Version 143.0

Limitations

- **Single Active Project:** Users can work on only one active project at a time
- **Model Compatibility:** Not all model architectures may be compatible with target hardware

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Getting Started

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Getting Started with EdgeBench

To begin using the EdgeBench application, you'll need a **MyAnalog** account. If you don't have one yet, you can create your account either from scratch or by using a personal login method. The system supports account creation and login through:

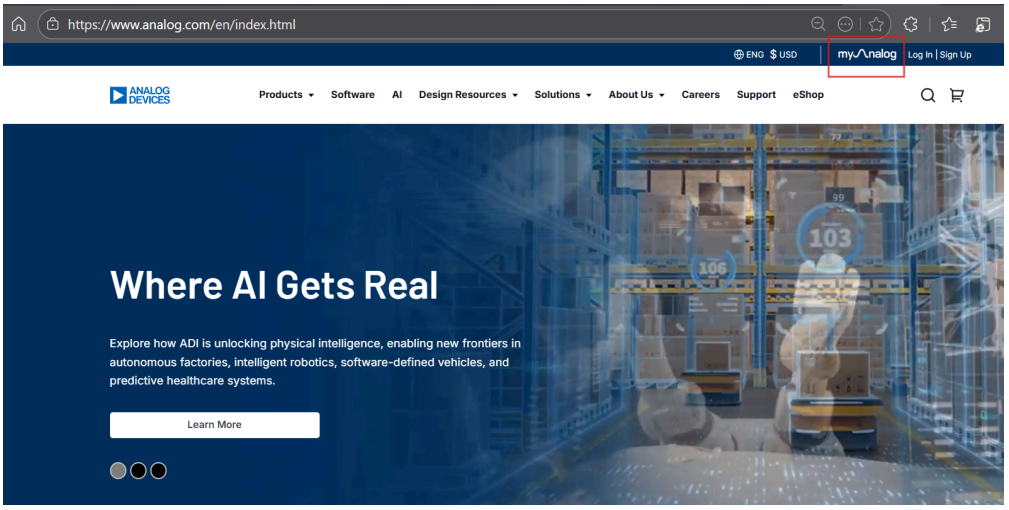
- Option A - Google Account
- Option B - LinkedIn Account
- Option C - Analog Devices Account

Choose the method that works best for you and follow the steps below to complete your registration and access the EdgeBench platform.

Create a MyAnalog account from scratch

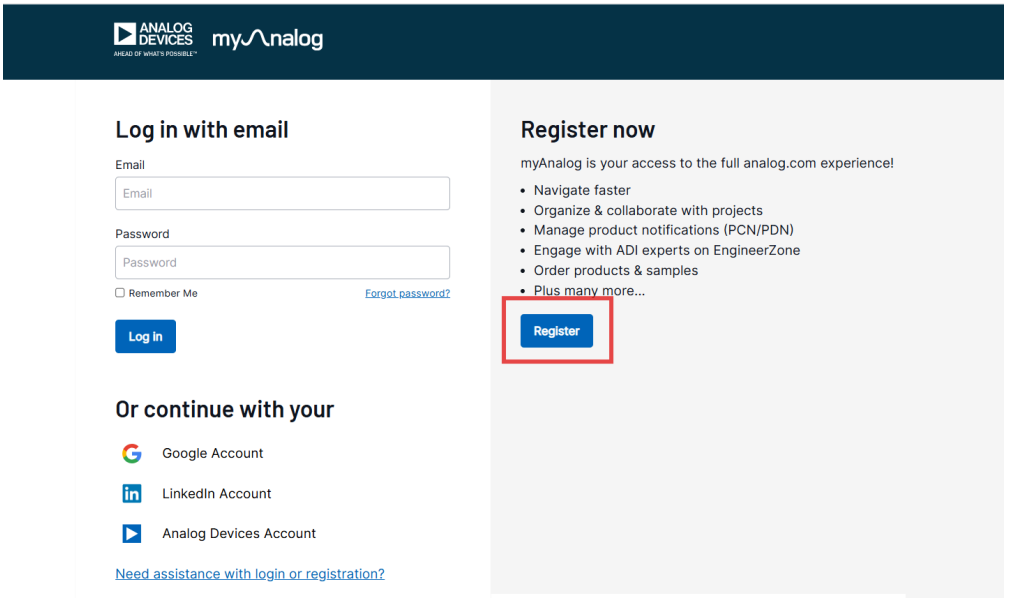
If you're new to **MyAnalog**, follow these steps to register:

1. Go to analog.com.
2. Click the MyAnalog icon located in the site's top header.



3. When the MyAnalog sign-up page opens, click **Register**.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



4. Fill out the registration form. Enter all required information marked with an asterisk (*), such as:
   - **Full name**
   - **Email address**
   - **Company name**
   - **Country**
   - **Password**
5. Read the **Privacy Policy**. Review the terms to understand how your data will be used.

Once all required fields are completed:

1. Click **Register** to submit the form.
2. Check your inbox for a verification email.
3. Click the verification link to activate your account.

Create a MyAnalog account via Single Sign-on (SSO)

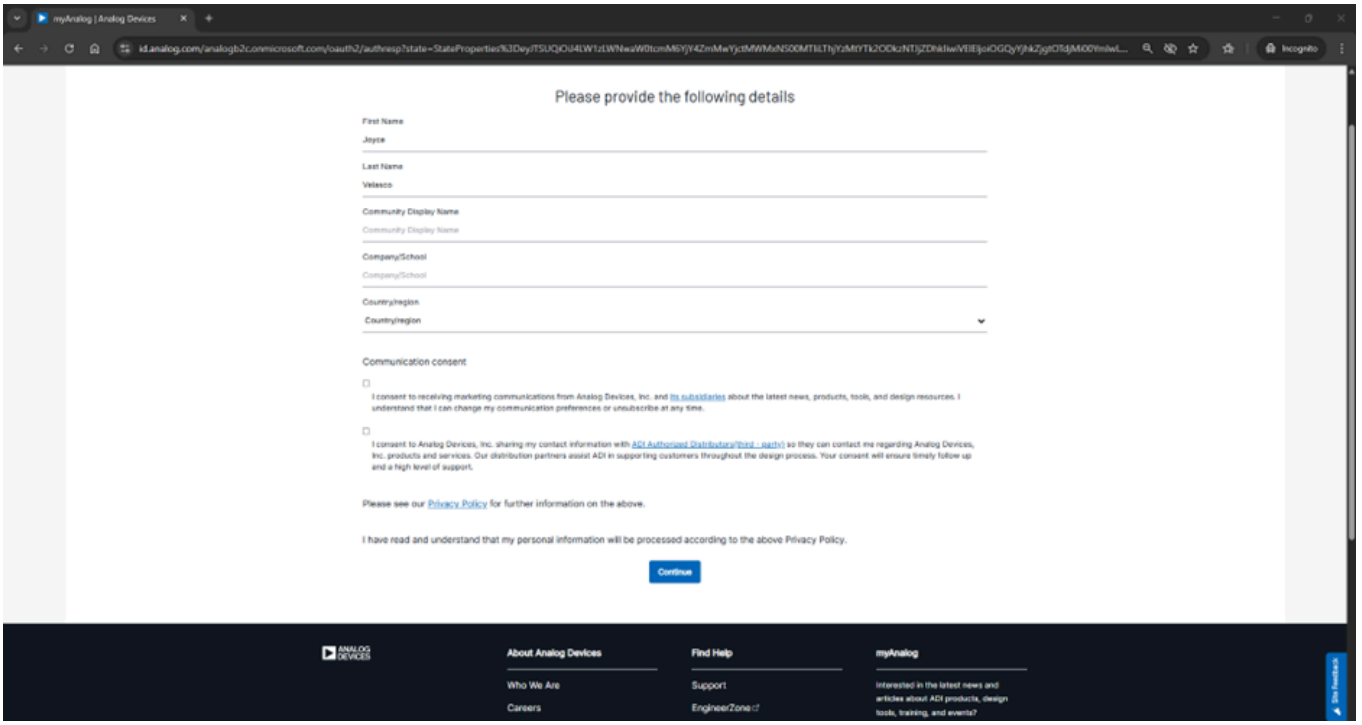
The EdgeBench login page supports creating accounts through **SSO** options:

Option A - Google Account

1. Click the **Google Account** option.
2. Enter your **Google** credentials.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

3. Complete authentication via:
   - **Verification email**, or
   - **Authentication app**
4. Fill out the **MyAnalog** profile form:
   - **Name**
   - **Company/School**
   - **Country/Region**



5. Review and accept the **Consent Agreement** and **Privacy Policy**.
6. Click **Continue** to access EdgeBench.

> **Tip**
>
> The system stores login credentials. Always sign out after using public computers.

Option B – LinkedIn Account

1. Click the **LinkedIn Account** login option.
2. Enter your **LinkedIn** credentials.
3. Fill out the **MyAnalog** profile form:
   - **Name**

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Company/School**
- **Country/Region**

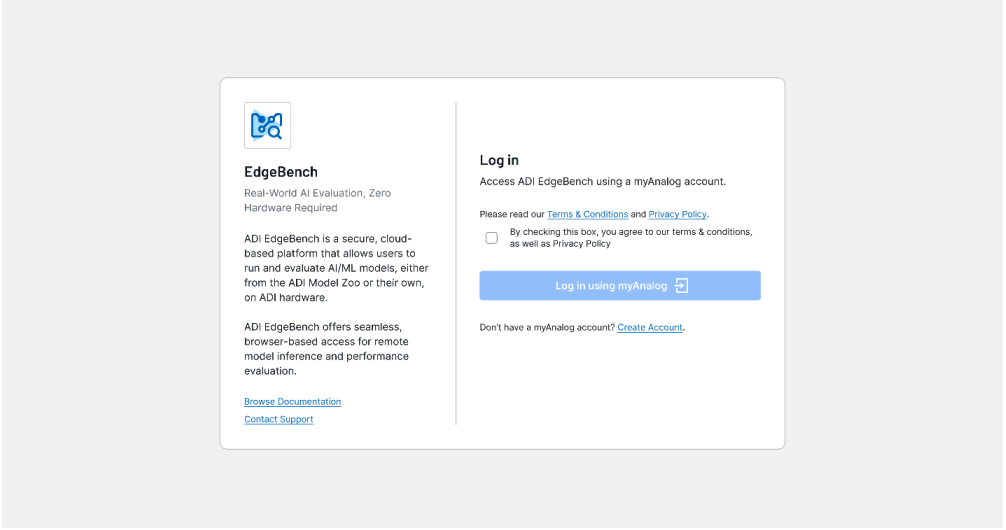
4. Review and accept the **Consent Agreement** and **Privacy Policy**.
5. Click **Continue** to access EdgeBench.

Option C – Analog Devices Account

Available for ADI employees only

1. Click **Analog Devices account**
2. Log in using your **ADI NT** credentials

Log in to EdgeBench app



For existing **MyAnalog** account holders:

1. Review **Terms and Conditions** and **Privacy Policy**.
2. Accept the terms by checking the checkbox.
3. Click **Log in using MyAnalog**.
4. Enter your **MyAnalog** credentials.

Returning users

If previously logged in via **SSO** (**Google**, **LinkedIn**, or **ADI**):

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- Click your preferred login method on the login page.
- Use stored credentials for quick access.

Logging out

1. Click the **Profile** dropdown menu (top-right corner).
2. Select **Logout**.

> **Important**
>
> Always log out when using shared devices. Note that logging out of EdgeBench doesn't log you out of **MyAnalog**.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Key Concepts

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Key Concepts

Welcome to the EdgeBench Application Overview. This section is designed to help users understand the core concepts and features of the EdgeBench web application. It provides contextual information to support efficient navigation and utilization of the platform.

User

The User is the central actor in the EdgeBench, responsible for initiating and managing the entire workflow. Users create or import projects, upload models and input data, configure builds, and trigger deployment & inference requests.

Workspace

The Workspace functions as a central hub and dedicated personal storage area within the platform, allowing users to manage their AI/ML projects and models. Each account is provided with 10 GB of cloud storage, which can be used to store model files, project configurations, project builds, and related resources.

For procedures on how to access or add projects and models, go to Navigating the Workspace section.

Project

A Project is the foundational unit within the EdgeBench application, acting as a structured container that organizes and encapsulates both Model and Build. It defines the scope and context of the build, enabling streamlined management and deployment of an AI model.

Each Project includes key attributes such as:

- **Name** – A unique identifier for the project.
- **Framework** – Specifies the underlying model training framework (TensorFlow, PyTorch, ONNX).
- **Task Type** – Indicates the nature of the AI task (e.g., image classification, object detection).

Supported Upload Formats:

- ZIP
- TAR.GZ

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Model

A Model is the core element used in builds and inference. It includes the following key attributes:

- **Framework** – The AI framework used
- **Name** – Model identifier
- **Task Type** – Defines the model's purpose

Models can be selected from the model zoo or uploaded via BYOM.

Model zoo

The model zoo is a curated catalog of pre-trained AI models optimized for edge deployment on ADI hardware. Covering domains such as audio, sensor, and vision, these models feature compact, efficient architecture designed for real-time, low-power applications. Each entry includes pre-trained files, sample scripts, benchmark data, and quantized versions, enabling streamlined evaluation and integration into edge AI deployment workflows.

To learn how to access and use these models, see the Exploring the ADI Model Zoo section.

BYOM

The Bring Your Own Model (BYOM) feature allows users to upload and deploy their own pre-trained AI models. Supported frameworks and formats include:

Framework	Required Files	Remarks
PyTorch	- Single Model File: `.pt` or `.pth` (complete model with architecture) - `input_shape.txt` - (Optional) `model.py` (model definition)	Input shape file must be in one of the formats: `tuple:(1,3,224,224)`, `array: [1,3,224,224]`, or `csv: 1,3,224,224`
TensorFlow	`.h5`, `.pb`, or `SAVED_MODEL`	Complete model with architecture
ONNX	`.onnx` file	Complete model with architecture

See Workspace Tasks > Uploading a Model (BYOM) for complete details on Supported Formats.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Models uploaded through BYOM are private and accessible only to the individual who uploaded them. They are not visible or available to other users on the platform. For publicly accessible models, users are encouraged to explore ADI's Model Zoo.

Disclaimer: Please note that not all BYOM models may successfully run on the target device. Models that are too large or contain unsupported operations may fail during compilation or during runtime.

Device

EdgeBench supports seamless model deployment across a range of remote devices. ADI evaluation boards such as MAX78002EVKIT, MAX32690EVKIT, and ADSPSC835W-EV-SOM are currently supported. These boards are all hosted within the ADI Remote Board Farm — a secure, web-connected infrastructure that enables real-time access to physical hardware through a browser-based interface. This setup eliminates the need for manual configuration or on-site devices, streamlining development and accelerating validation.

EdgeBench users benefit from 24/7 access to shared hardware via the Remote Board Farm, enabling continuous development and testing without the need for physical board shipments or setup.

Build

A Build represents a deployable or executable unit within a project. It is generated by compiling an AI model for a specific target device. A build offers several configuration options, allowing a user to tweak and experiment. Options include:

- Device settings
- Compilation optimization flags
- Quantization
- Kernel options

Build version

A Build Version is a uniquely identifiable snapshot of a build that captures the AI model, the target device, configuration, and compilation settings at the time of an inference run. Each build version includes metadata such as:

- **Build Status** - Represents the current progress of a build. It indicates whether the build has been completed successfully, encountered an error, or is still ongoing. This status may also show if the build is still in an intermediate step, such as compilation.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Last Updated Timestamp** - Shows the most recent time the build was modified or executed.
- **File Size** - Displays the total size consumed by the build.
- **Compilation Settings** - Lists the parameters used during model compilation.

Data input

Data input refers to the data used for inference. Supported formats include image, audio, and time-series files.

Data processing script

Data Processing Scripts are Python codes used to transform raw data into tensor data, and vice versa. If using the BYOM workflow, the user is required to provide these scripts into the platform. For ADI Models, this is already handled by EdgeBench.

The platform expects two kinds of data processing scripts:

- **Pre-Processing Script** – Transforms the end-user's uploaded data (image, audio, csv) into tensor data that aligns with the model's expected input tensor shape (required for BYOM).
- **Post-Processing Script** – Transforms tensor data from the model output back into raw data format (image, audio, csv) that can be easily consumed by the end-user (required for BYOM).

Result

Results are the outputs generated from inference runs, and are presented with the following details:

- **Timestamp** – Datetime when the inference was executed
- **Batch** – Input dataset that was sent as group for inference execution
- **Hardware Performance Metrics**
- Inference latency
- Memory usage
- Power consumption

Inference history

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The Inference History logs the outcomes of model evaluations, enabling users to monitor and analyze performance over time.

It includes:

- **Build Version** – Identifies which compiled model was evaluated
- **Inference Run Status** – Indicates success, failure, or in-progress state
- **Timestamp** – When the inference was last executed or updated
- **Device Target** – Specifies the ADI hardware used for the run
- **Compilation Settings** – Parameters used during model build and deployment

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Workspace Navigation

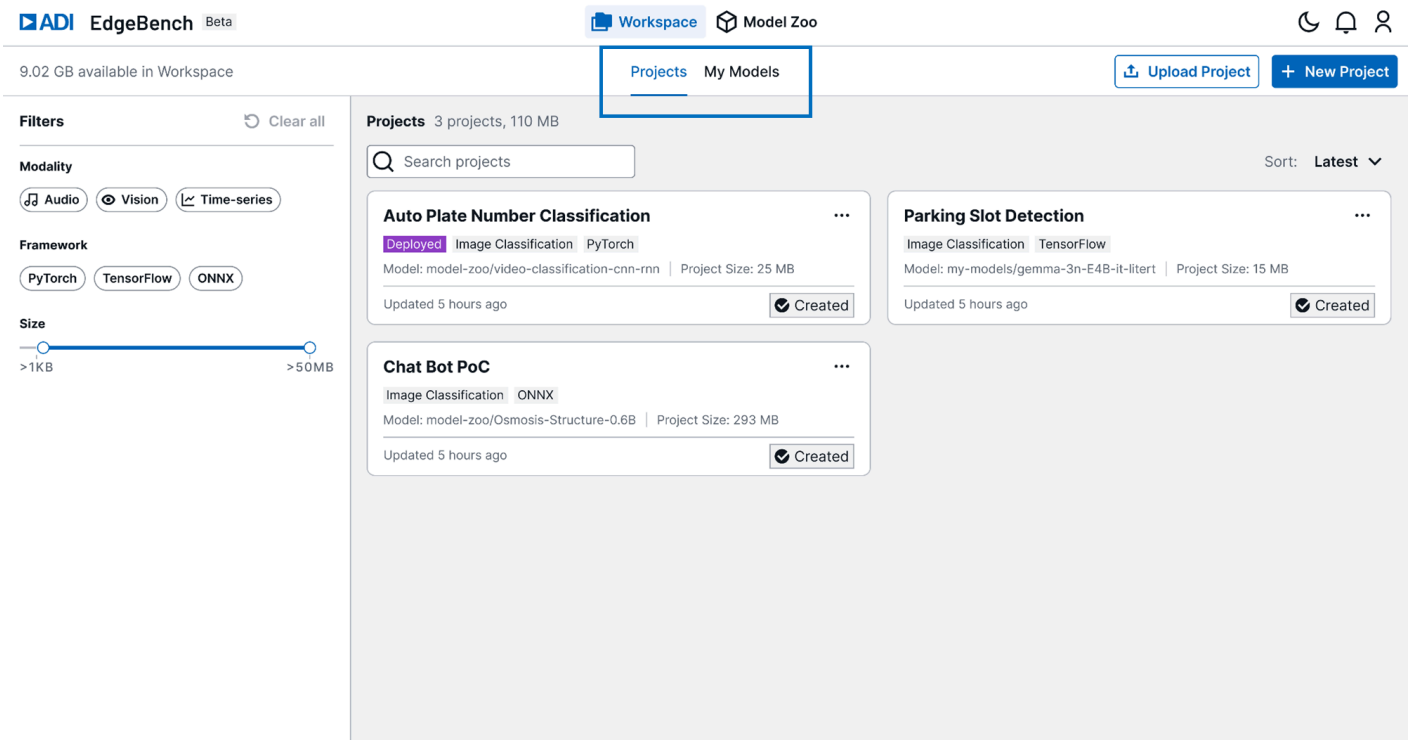
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Navigating the Workspace

The **EdgeBench Workspace** serves as a central hub and personal storage area where users can manage their projects and models. It includes two main tabs:

- **Projects** – dedicated to viewing and managing project builds for inference tasks.
- **My Models** – provides access to user-uploaded models via the **BYOM** feature.

Users can also explore pre-trained models via the **Model Zoo** tab.

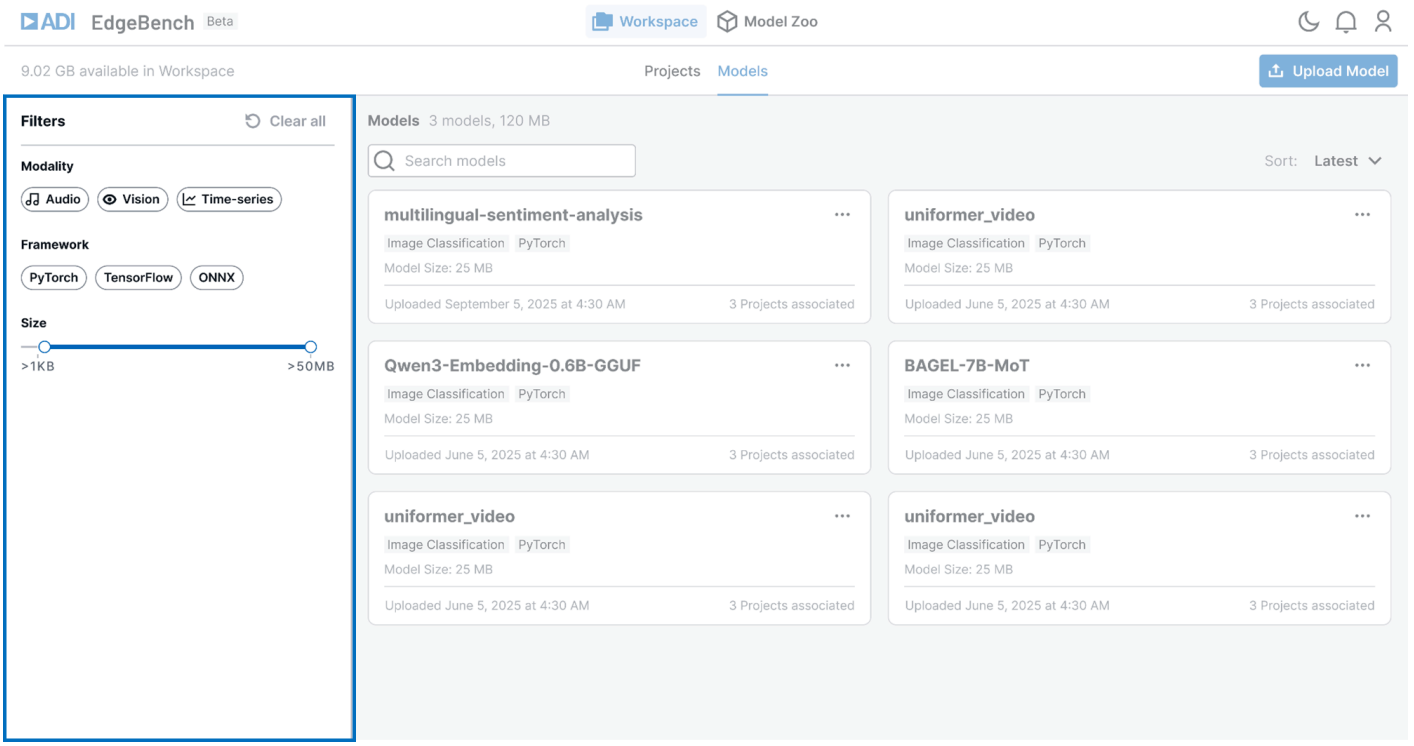


Common user controls

These controls are consistently available across both **Projects** and **Models** sections in the **Workspace**, as well as in the **model zoo**. They help users manage lists and individual items efficiently.

Filter panel

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Located on the left side of the interface, the **Filter Panel** allows users to narrow down lists using the following criteria:

Modality

- Audio
- Vision
- Time-series

Framework

- Filter by AI training framework (e.g., **TensorFlow**, **PyTorch**, **ONNX**)

Size

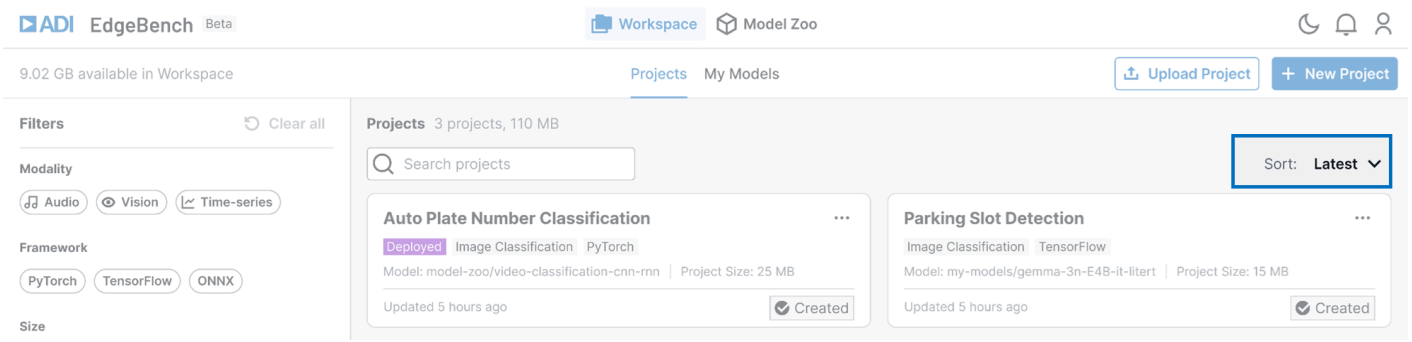
- Use the horizontal slider to filter models by file size, ranging from >1 KB to >50 MB.

Clear All

- Click this button to remove all selected filters

Sort options

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Accessible via dropdown menus, sorting helps organize entries based on:

Date Last Modified

- Latest First (Descending)
- Oldest First (Ascending)

File Size

- Largest to Smallest
- Smallest to Largest

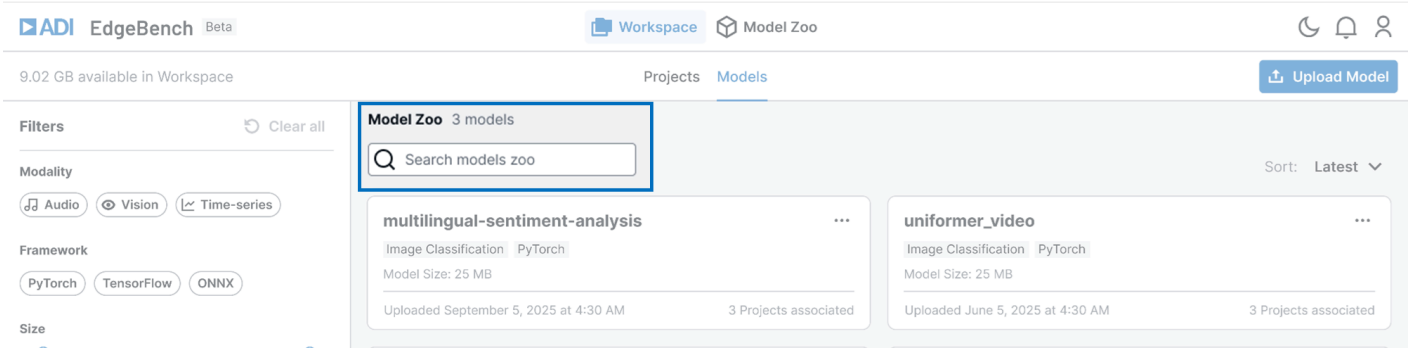
File Name

- A to Z (Ascending)
- Z to A (Descending)

Upload Date (for models only)

- Newest First
- Oldest First

Search bar



A keyword-based search tool that helps users locate specific projects or models by name or function. Available in:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Workspace → Projects**
- **Workspace → My Models**
- **Model Zoo**

Card actions

Each card (project or model) includes an ellipsis menu offering quick actions:

- **Rename**
- Edit the name of the item.
- **Delete**
- Permanently remove the item from the list.

> **Note**
>
> For **BYOM** models, deletion is only allowed if the model is not linked to any projects.

Top bar overview



No.	Control	Function
1	**Workspace**	Workspace provides each user with 10 GB of personal cloud storage to store AI projects and models, including **BYOM** uploads. It allows users to view and manage projects, builds, and related resources in a centralized location.
2	**Model Zoo**	Contains ADI-verified and pre-trained models available for deployment and evaluation.
3	**Theme Toggle**	Switch between light and dark mode using the sun and moon icons.

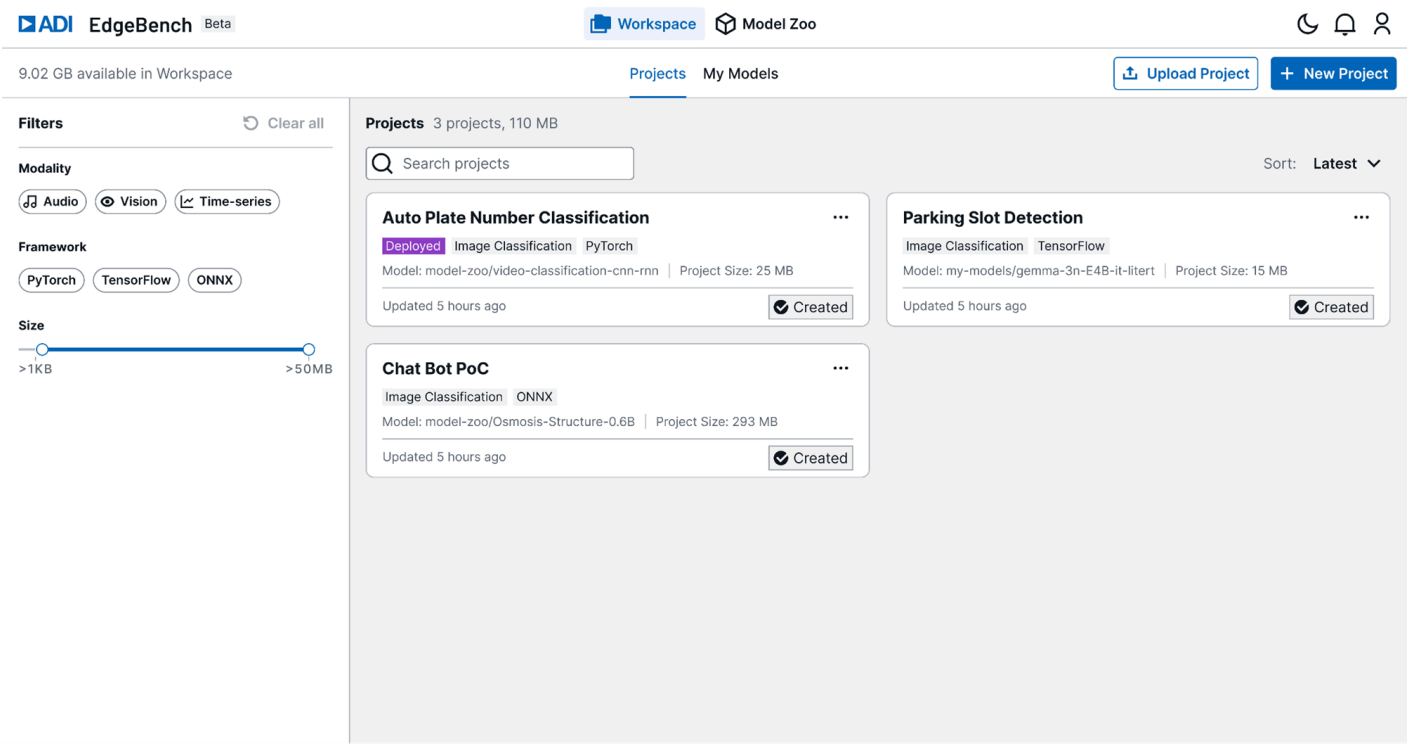
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

No.	Control	Function
4	**Notification Bell**	Available to keep users informed about ongoing and completed activities. Whenever a compilation or inference process is initiated, a corresponding notification is generated and displayed.
5	**Profile Avatar**	The Profile menu includes the following options: **My Profile** – Allows users to view and edit their personal profile details. **Online Help** – Directs users to support resources and documentation for guidance on using the application. **Send Feedback** – Enables users to share feedback, report issues, or suggest improvements. **EngineerZone** – Provides access to a community and knowledge-sharing platform for technical discussions and support. **Log Out** – Allows users to securely sign out of the application.

Projects tab

The **Projects Tab** is the central dashboard for managing AI projects within the **Workspace**. It presents all user-created projects in a visually organized card layout, making it easy to browse, access, and manage work at a glance.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Project list management

Refer to **Common User Controls** for:

- **Filter Panel**
- **Sort Options**
- **Search Bar**
- **Rename**
- **Delete**

Project card info

The **Projects Tab** displays all user-created projects as cards. Each card displays the following information:

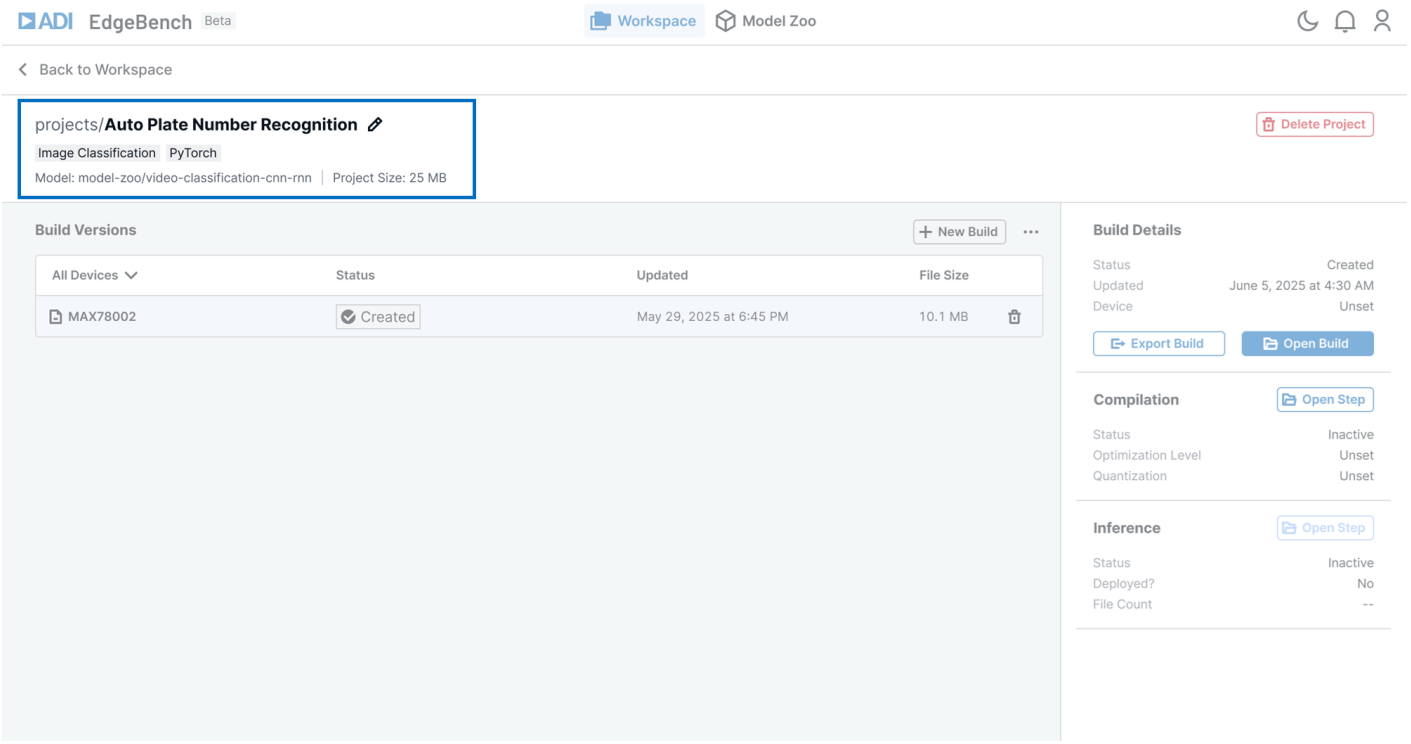
Displayed Info	Description
Project Title	Name of the project
Task Type	Model task type (e.g., classification, detection)

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Displayed Info	Description
Framework	AI training framework used (e.g., TensorFlow, PyTorch)
Model Name	Associated model name
Project Size	Total storage used by the project
Last Update	Timestamp of the most recent modification

Project details page

When a user clicks on a project card in the **Workspace** → **Projects** tab, they are taken to the **Project Detail** Page, which is divided into the following key sections:



This section provides high-level information about the selected project:

Field	Description
Project Name	Editable field showing the name of the project.
Task Type	Indicates the model's task (e.g., classification, detection).

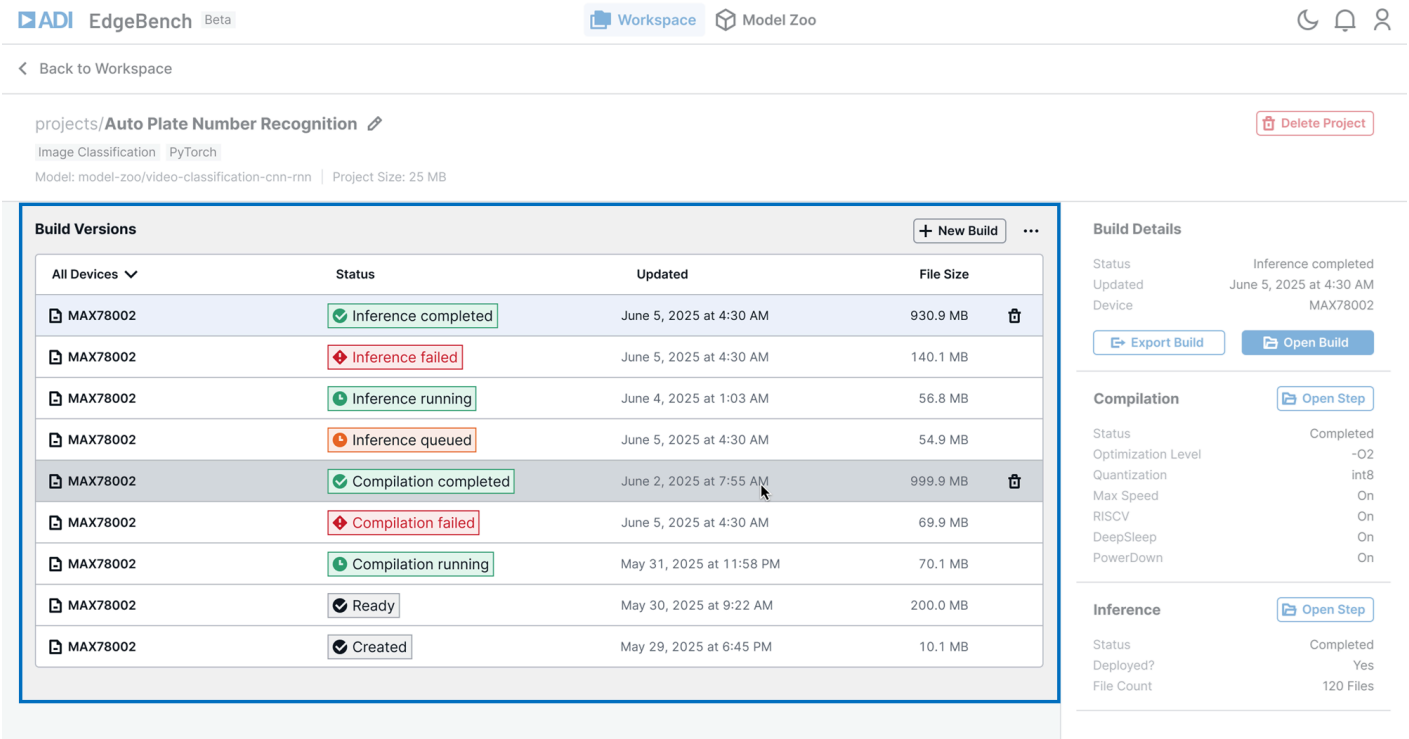
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Field	Description
Framework	Shows the training framework used (e.g., TensorFlow, PyTorch).
Model Name	Displays the model associated with the project.
Project Size	Total storage used by the project files.

Clicking on a **Project card** opens the **Build Version** listing.

Build version listing

This dashboard provides a comprehensive log of all model evaluation runs performed on each ADI hardware under the selected project. This view enables users to trace, audit, and compare inference performance across different build versions and configurations.



Each row represents a build created for a specific device and includes the following information:

Field	Description
Status	Displays the current build state such as Compilation Completed, In Progress, or Failed.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Field	Description
Last Updated Timestamp	Shows when the build was last modified.
File Size	Size of the generated build artifact.
Build Actions	Options to Export Build, Open Build, or Delete Build.

Clicking on a build entry opens the **Build Details Panel** on the right.

Status tags overview

Status tags provide real-time visibility into the progress and outcome of both the build and inference workflows. Each step in the process generates a corresponding status indicator, allowing users to quickly understand what the system is doing, whether an action is ongoing, completed, or requires attention.

Note that these tags are also visible during the build process, shown at the upper right portion of the window.

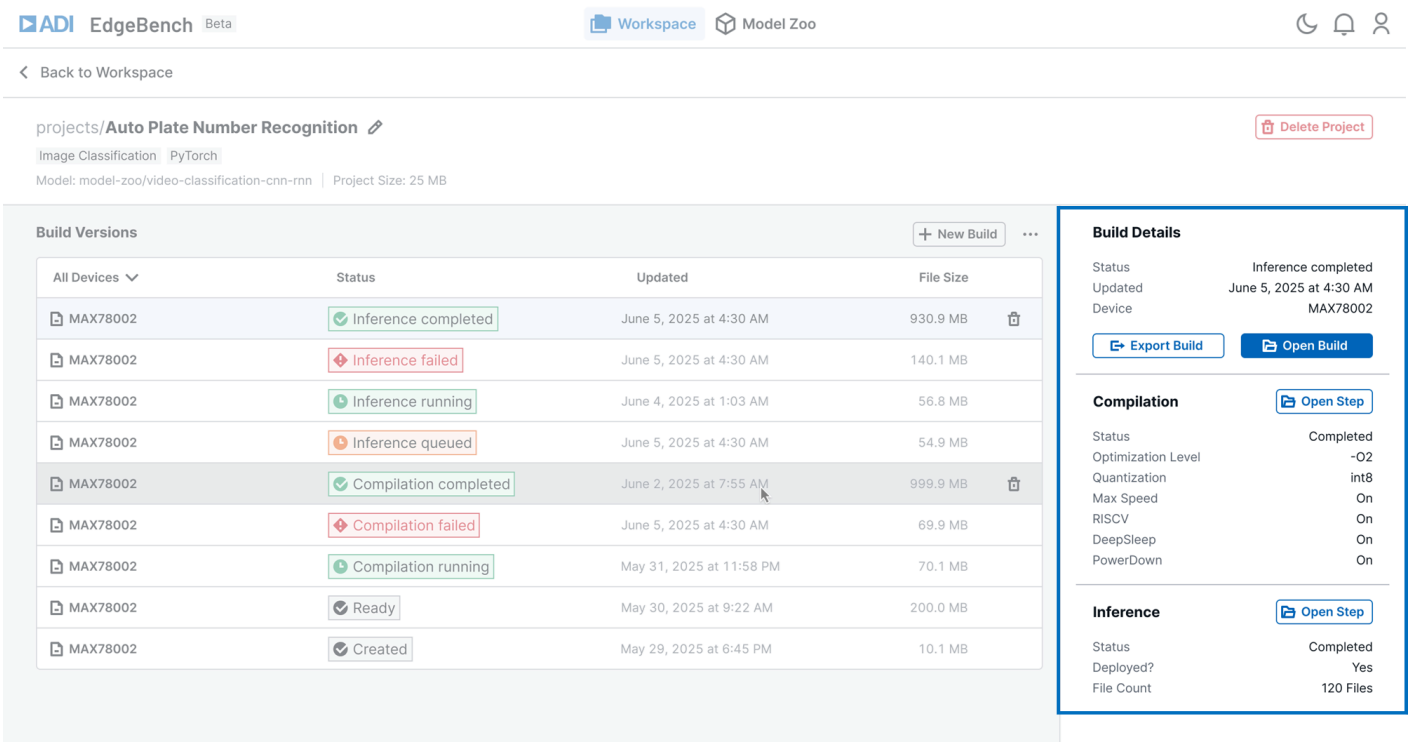
Status Tag	Description
Created	Indicates the beginning of the build process. The project already exists, and the user can proceed with device selection.
Ready	States that device, model, and configuration settings have been fully set.
Compilation Queued	Compilation request has been initiated; waiting to begin processing
Compilation Failed	Errors were encountered during compilation. User needs to check the logs for more details and address the errors for the compilation to continue.
Compilation Running	Compilation is currently in progress.
Compilation Successful	The compilation completed with no errors.
Compilation Aborted	Triggered when the user aborts the “Start Compilation”, this action stops the compilation process.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Status Tag	Description
Compilation Completed	All configurations for the selected device and model have been successfully compiled and applied. This unlocks the next step in the build process. This status is reflected through Steps 3 (Uploading Input Data), 4 (Pre-processing Script), and 5 (Post-processing Script).
Inference Queued	This status shows after clicking the “Start Inference” button; the request for deployment to the selected device has begun.
Inference Failed	Indicates an unsuccessful inference attempt. Logs should be checked for details.
Inference Aborted	Inference started but was stopped (user-initiated stop).
Inference Running	The inference operation is currently being executed on the selected device.
Inference Completed	Inference ran successfully on the selected device. User can now view the results.

Build details side panel

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



This panel provides detailed information about the selected build version:

Section	Description
Build Details	Displays the current status of the build. This section also provides two key actions: • **Export Build** – Allows the user to export the compiled build artifact to their local machine; this enables to download the build for deployment or manually share with another user. • **Open Build** – Directs the user to the active build step for further configuration or review.
Compilation Parameters	Displays flags and settings used during model compilation.
Open Step (Compile)	Allows users to go back to **Step 2 - Compilation** and modify compilation settings.
Inference Details	Shows deployment status, file count, and inference result status.
Open Step (Inference)	Enables users to open **Step 6 - Deploy and Inference** and modify inference settings.

Note: The **Open Step** buttons are disabled for builds with inactive status.

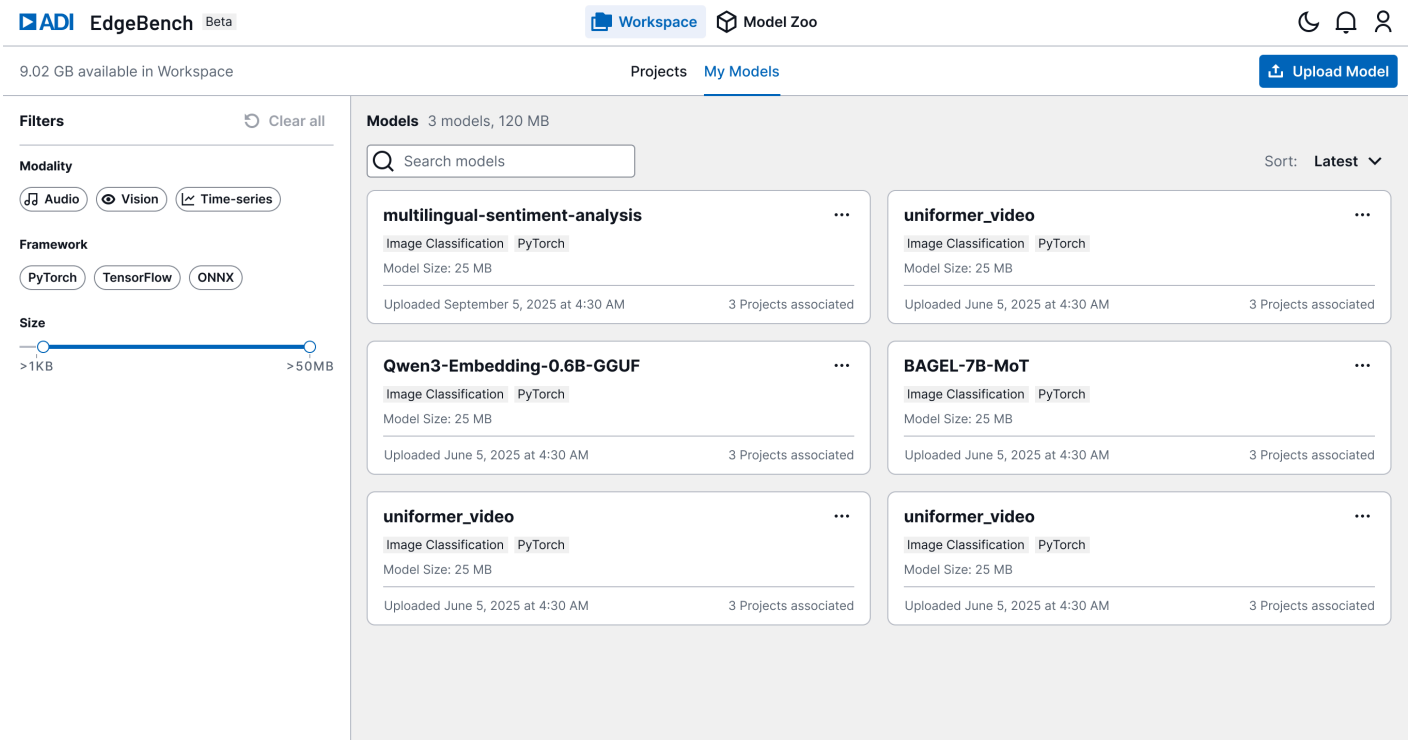
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Available controls

Control	Function
Rename Project	Click the pencil icon next to the project name to edit.
Delete Build	Click the trash icon next to a build entry to remove it.
Delete Project	Use the top-right button to delete the entire project.

Models tab

The **Models Tab** in **Workspace** displays all user-uploaded models. Each model is presented as a card with key information for quick reference and management.



Model list management

Refer to **Common User Controls** for:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Filter Panel**
- **Sort Options**
- **Search Bar**
- **Rename**
- **Delete**

> **Note**
>
> **BYOM** models can only be deleted if not linked to any projects.

Model card info

Each model card includes the following fields:

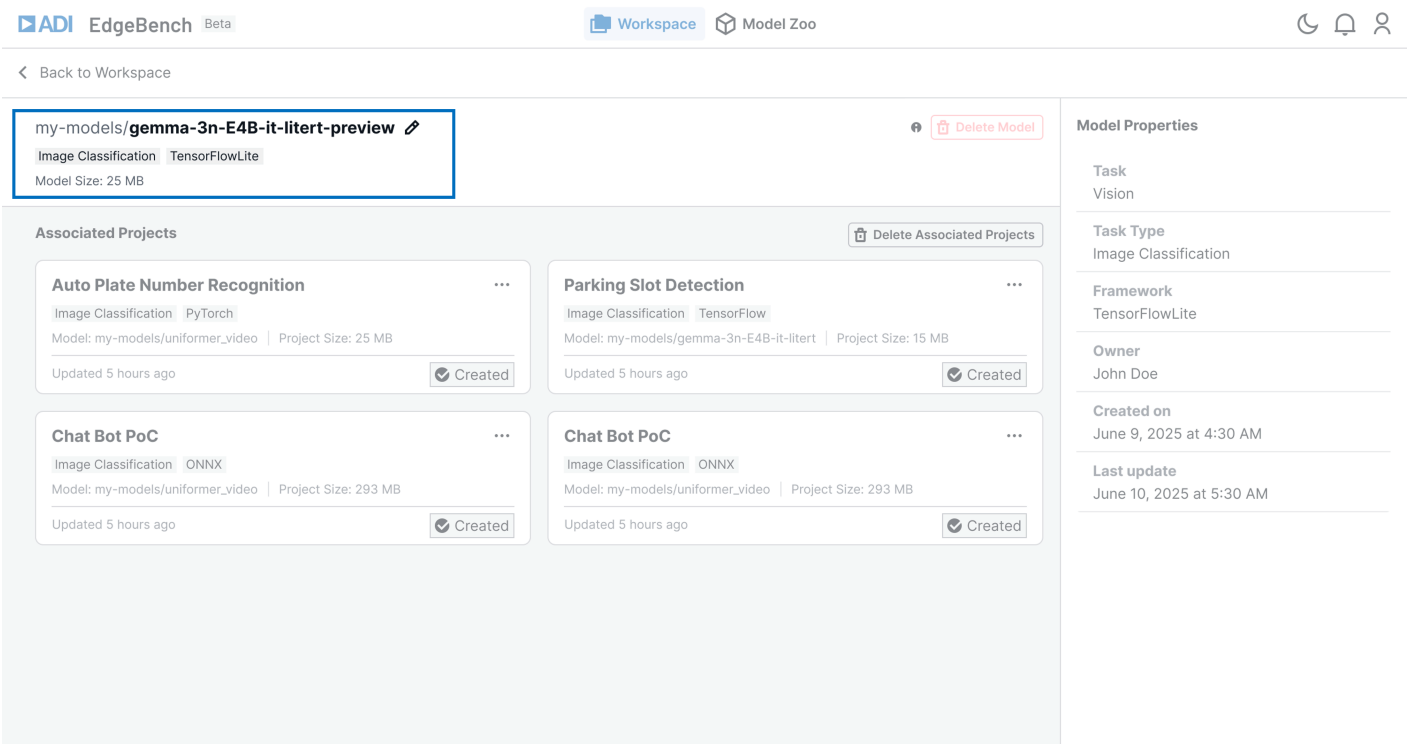
Field	Description
Model Name	Name of the uploaded model
Task Type	Model task type (e.g., classification, detection)
Framework	Model training framework used (e.g., TensorFlow, PyTorch)
Model Size	File size of the model
Upload Date	Timestamp of when the model was uploaded
Associated Projects	Number of projects linked to this model

Model detail page

Clicking a model card opens the **Model Detail Page**, which is divided into the following components:

Model overview

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



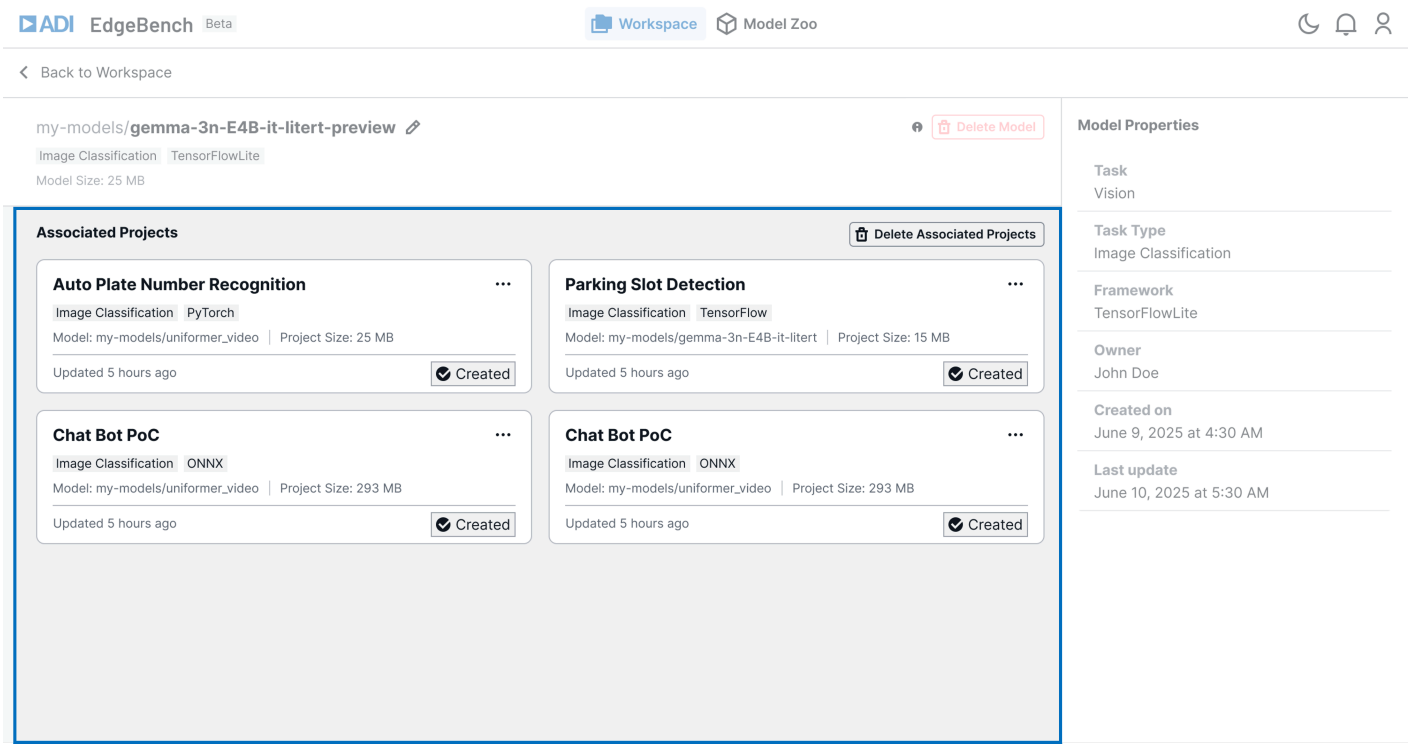
This section provides high-level information about the selected model:

Element	Description
Model Name	Displayed prominently; editable if needed
Task Type	Indicates the AI task type
Framework	Shows the AI training framework used
Model Size	File size of the model

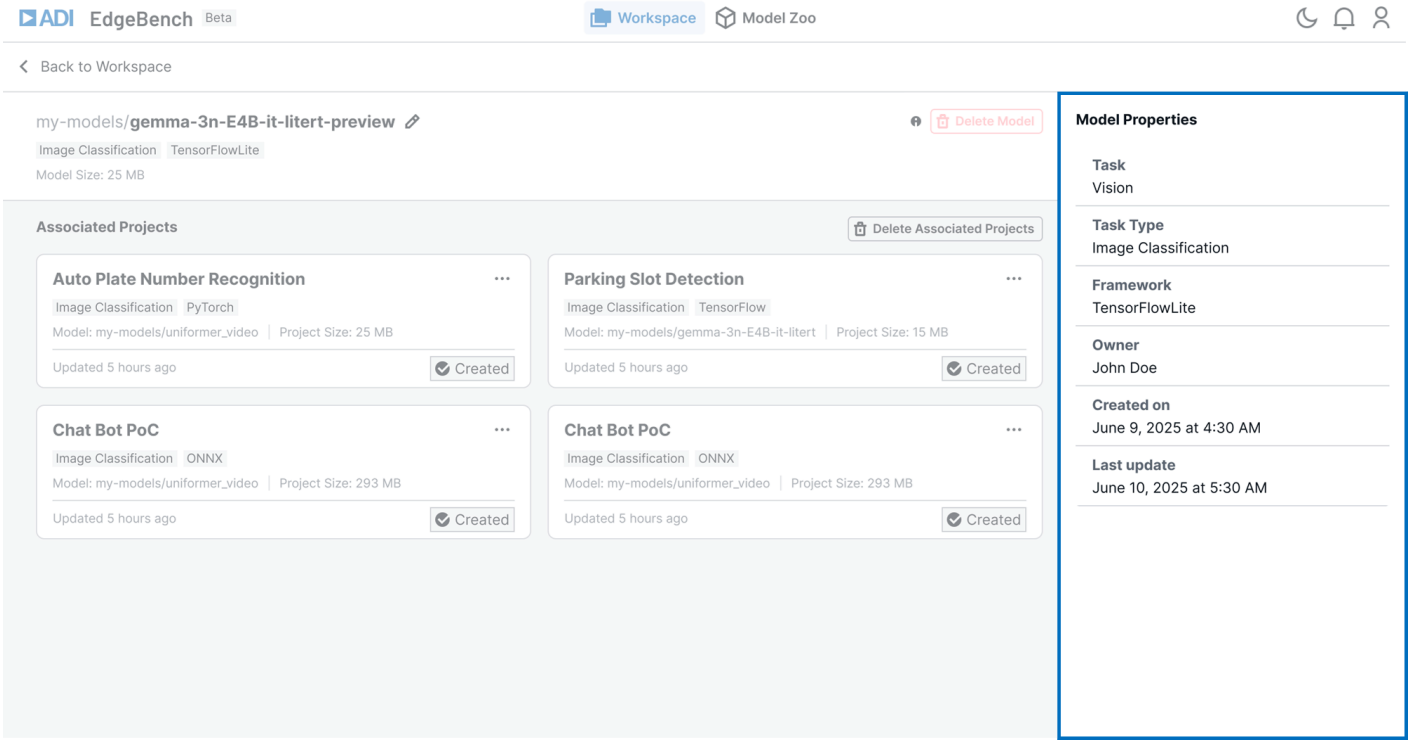
Associated projects listing

The **Associated Projects** section provides a quick overview of all projects linked to a specific model.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Model properties side panel



This panel displays basic metadata about the model:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Property	Description
Task	General task category (e.g., vision, audio)
Task Type	Specific model task (e.g., object detection)
Framework	Model training framework used
Owner Name	User who uploaded the model
Upload Date	When the model was added to the Workspace
Last Updated	Most recent modification timestamp

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Workspace Management

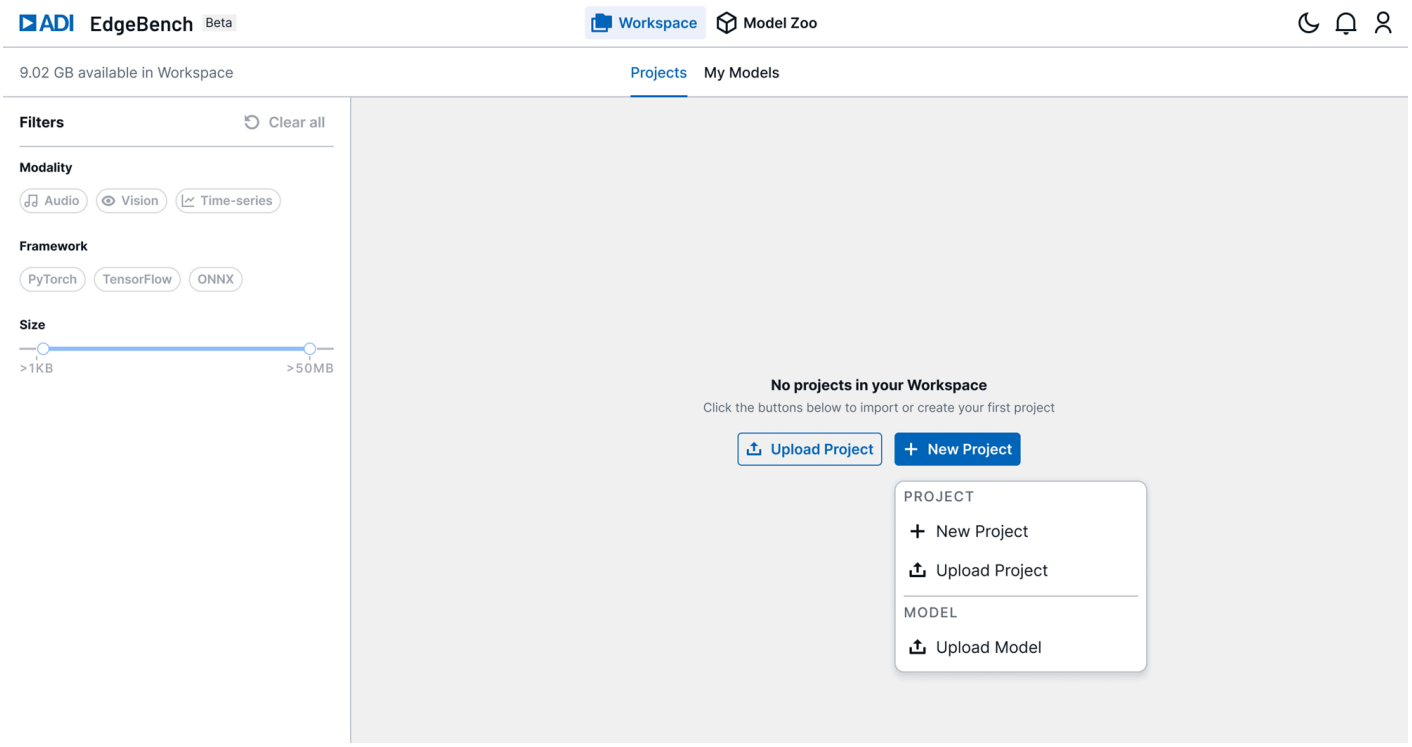
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Workspace Tasks

This section outlines the core tasks users can perform within the Workspace to manage projects and models effectively. Whether you're starting a new project or continuing to work on an existing one, the workspace provides tools to streamline your workflow.

Adding new items

To begin adding content to the Workspace, click the **New** button. This opens options for creating a new project or uploading a model.

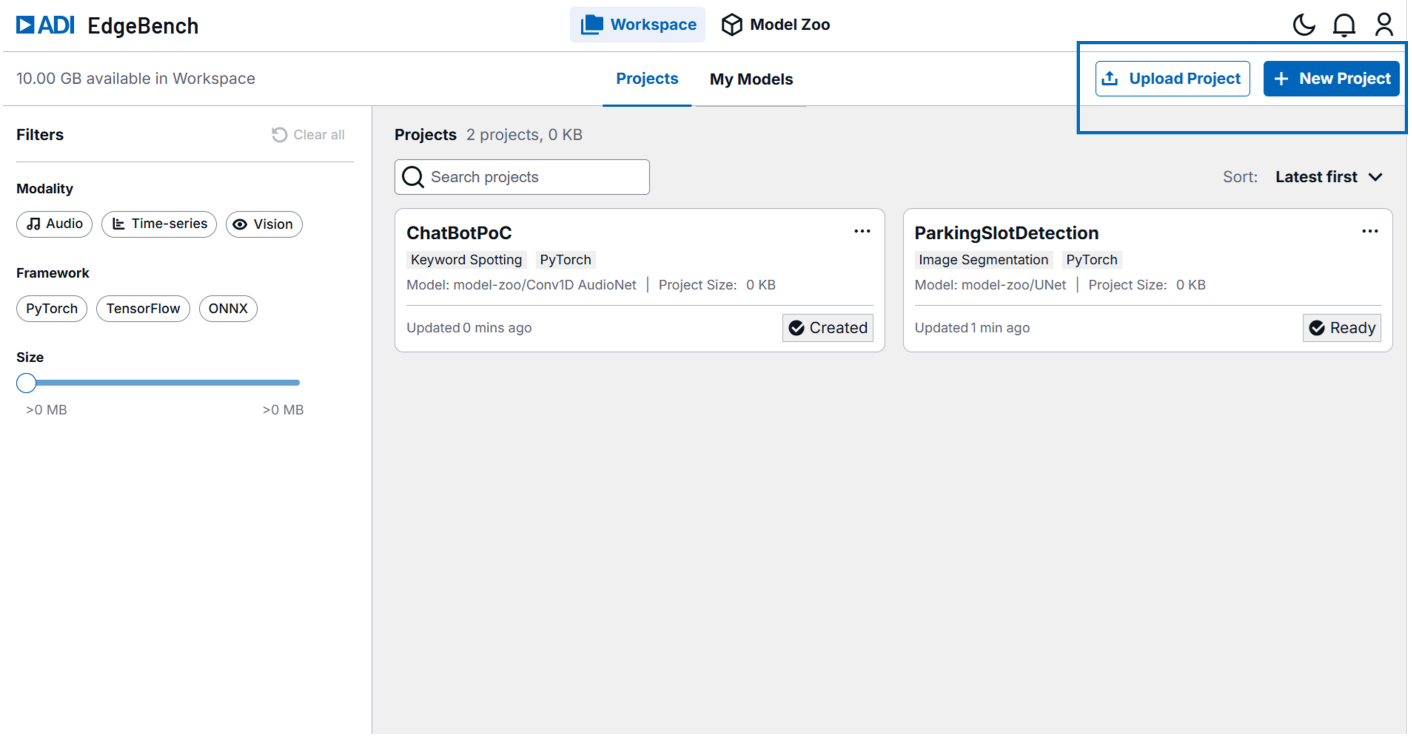


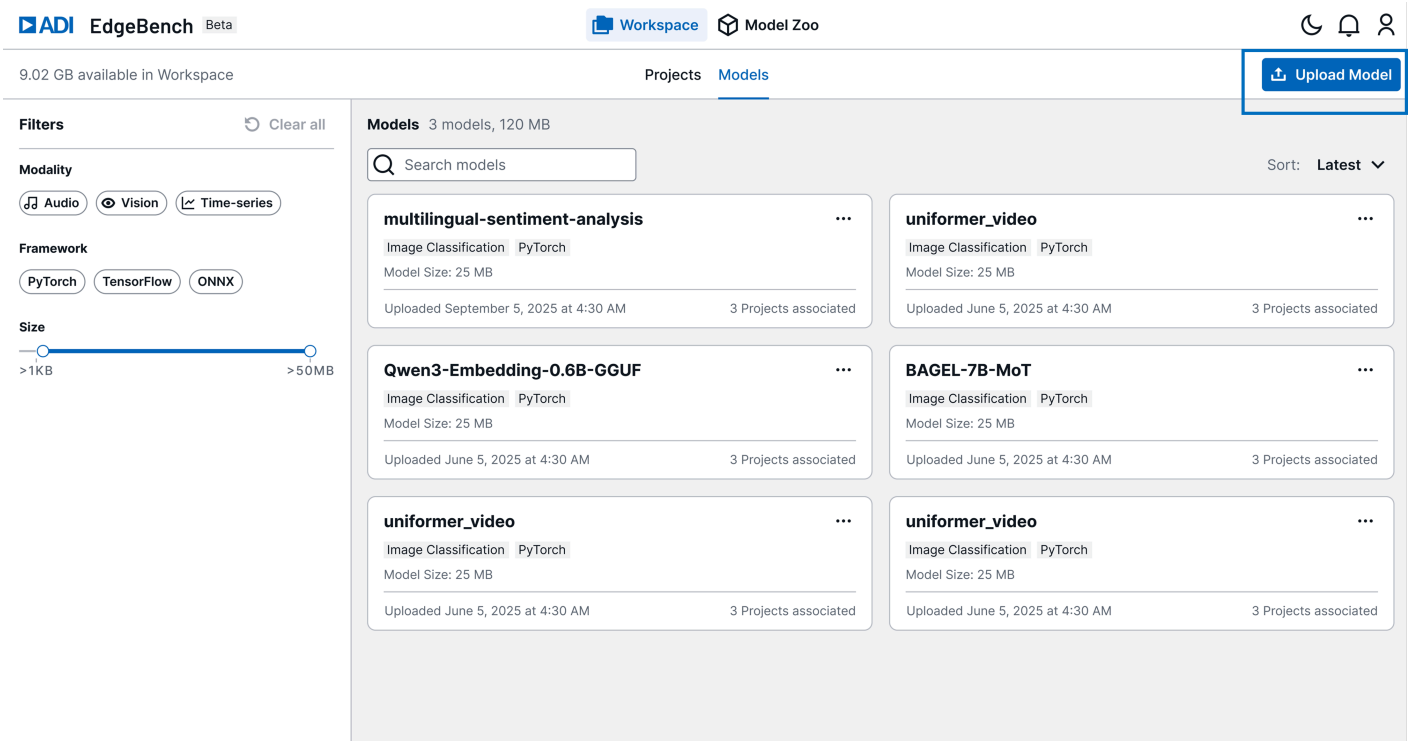
Note

This Empty State view is only shown on first use of the application.

If the Workspace already contains projects and models, use the **Upload Project** and **Upload Model** buttons located in the upper-right corner of the application window.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

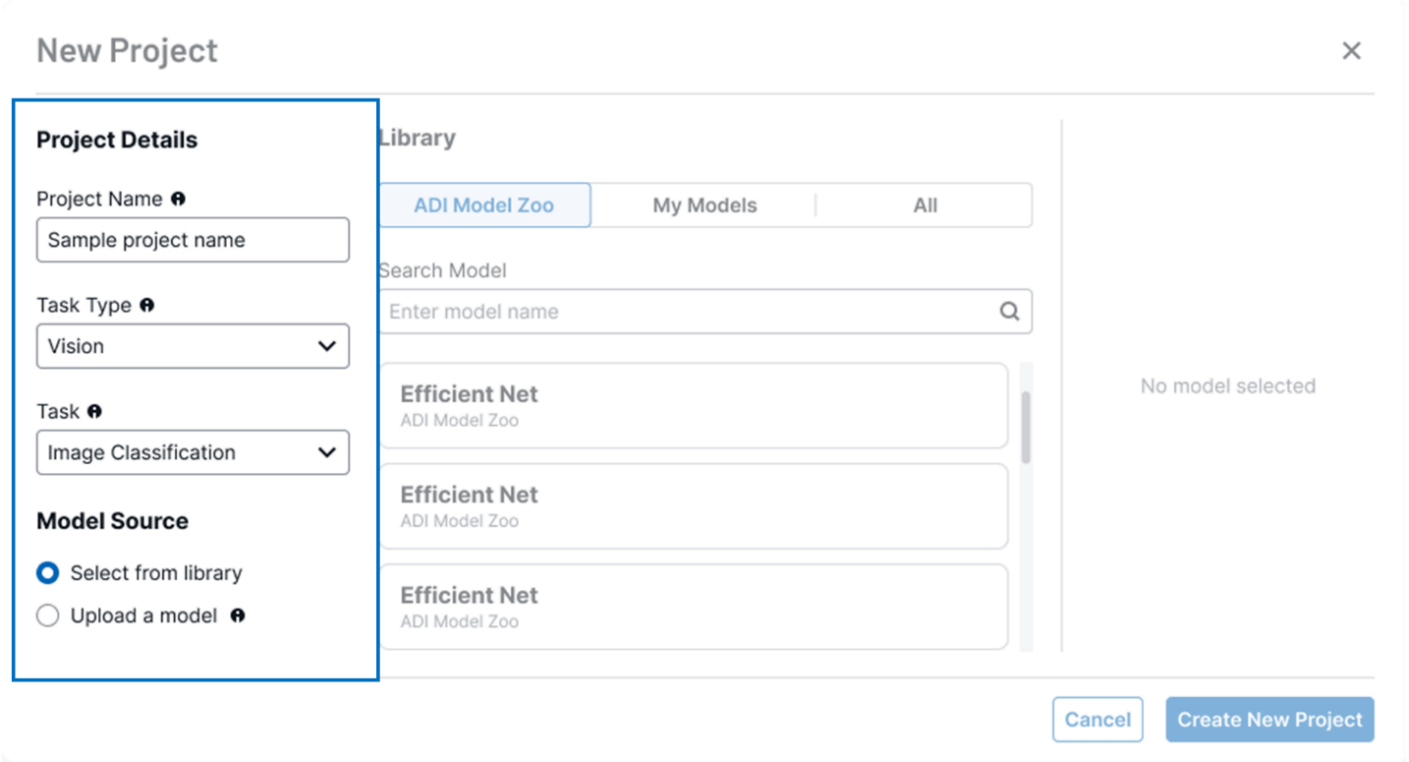




Adding a new project

The Create Project feature allows users to initialize a new workspace for model development and evaluation.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

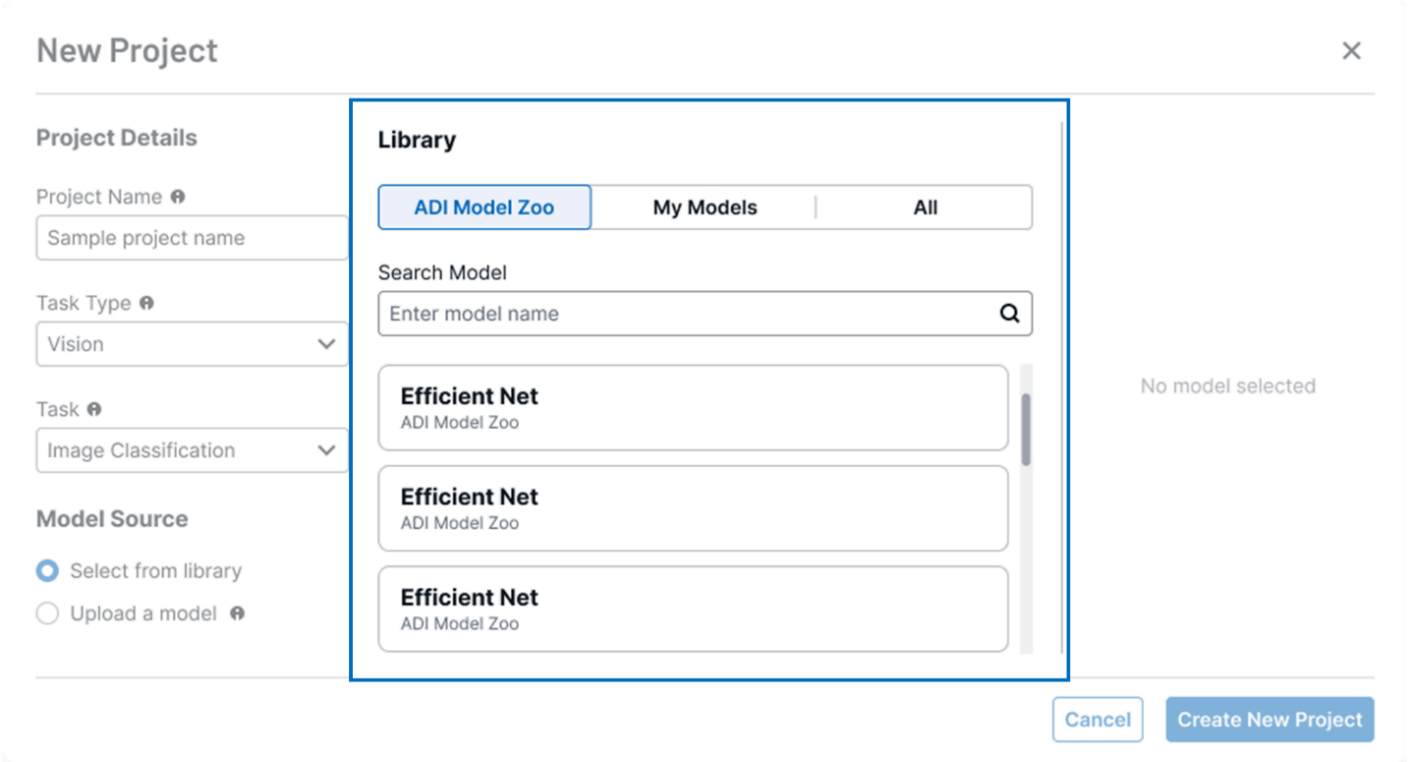


To start a project from scratch:

1. Within the Workspace, click on the **New Project** button to begin. This action will open the **Create Project** window.
2. Enter the desired **Project Name** in the input field.
3. Select the appropriate **Task** and **Task Type** from the dropdown list.
4. Choose a **Model Source** using one of the following options:

Option 1: Select a Model from the Library

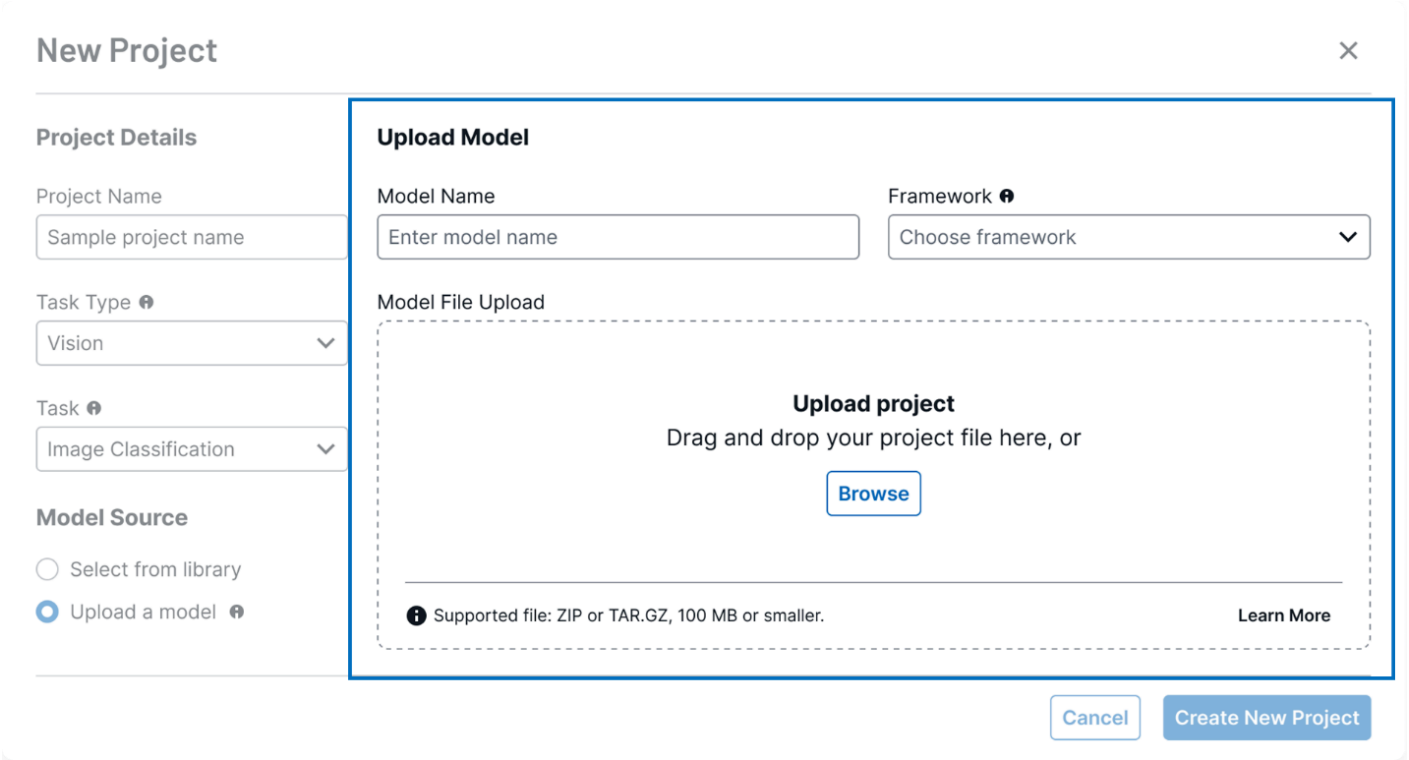
Copyright © 2026 Analog Devices Inc. All Rights Reserved.



1. Choose **Select from Library**.
2. The center pane will display available models organized into three categories:
   a. **ADI Model Zoo** – Curated models provided by ADI.
   b. **My Models** – Models previously uploaded by the user. By default, this section is empty if you haven't uploaded any custom models.
   c. **All** – Combined view of ADI and user-uploaded models.
3. Browse the list and select a model to use in your project.

Option 2: Upload Your Own Model

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



1. Choose **Upload a Model**.
2. Enter the **Model Name** in the input field.
3. Upload the model file:
   a. Drag and drop the file into the designated area, or
   b. Click **Browse** to select the file manually.
4. After you upload the model, the system automatically detects the framework used and displays it.
5. Review all entered information to ensure accuracy. Click the **Import Project** to save your project to the Workspace.

Additional Controls

- **Cancel button**

  Clicking the `Cancel` button redirects the user back to the Workspace without creating the project.
- **Close button**

  Clicking the `Close (X)` icon in the top-right corner closes the window without creating a project.

Uploading a project

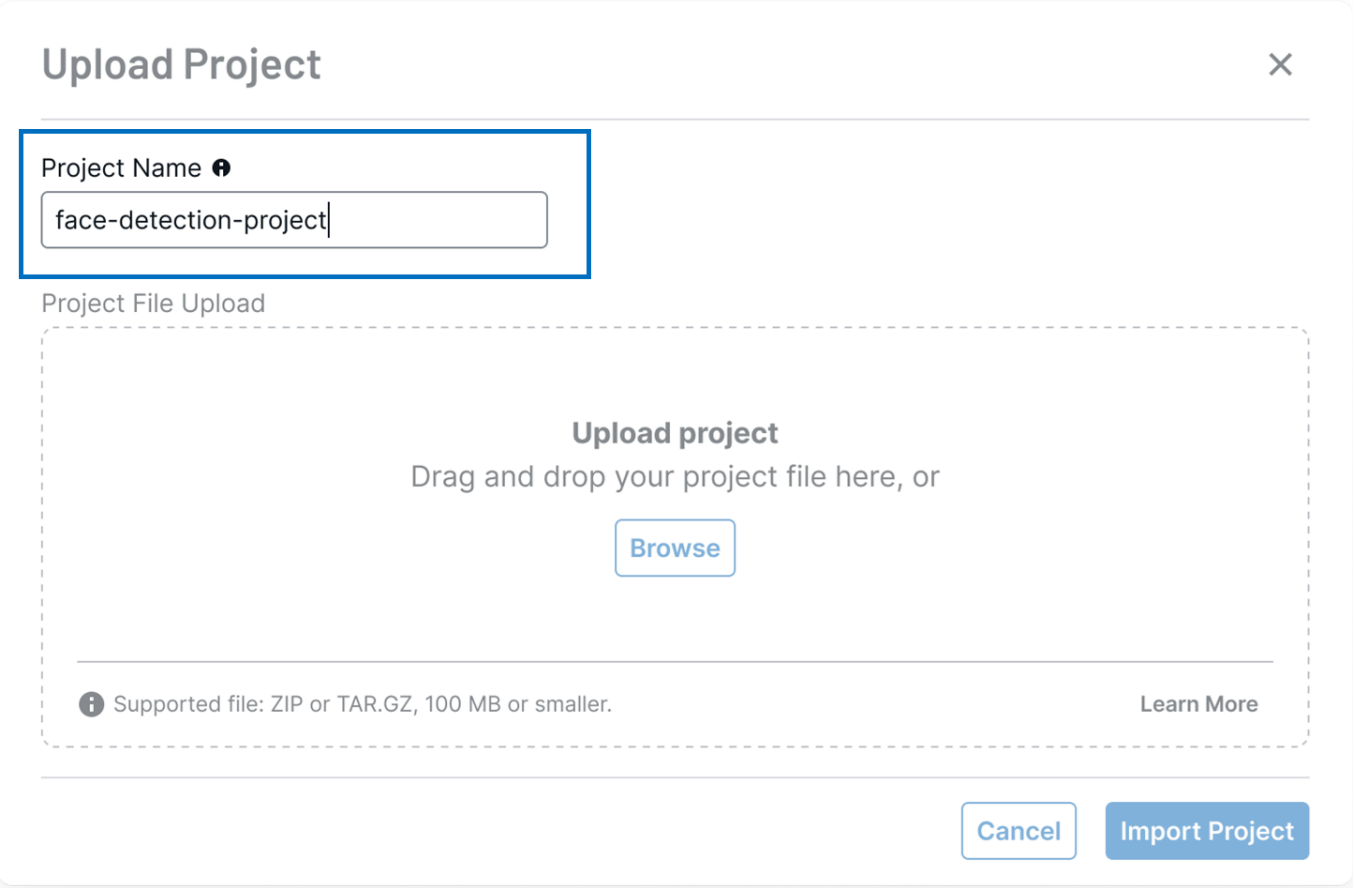
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The Upload Project feature allows users to upload a previously created and exported project.

After you create a project, the **Export Project** option becomes available. The exported project file can later be uploaded using the Upload Project option.

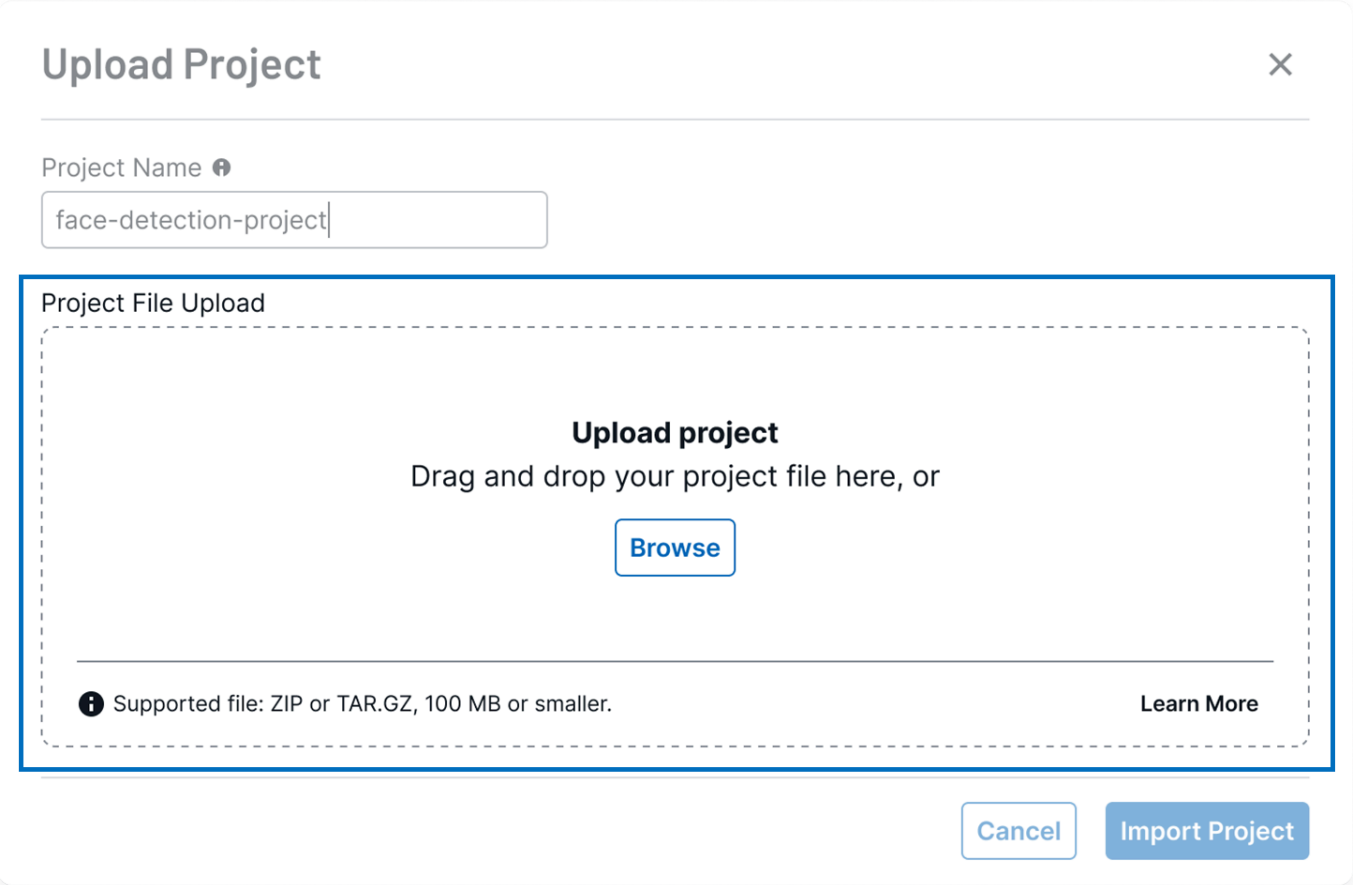
To import a project from a local computer:

1. Click **Upload Project** to import an existing project from your computer.
2. Enter a descriptive **Project Name** for the project.



3. Drag and drop the project file into the designated area or click **Browse** to select the file manually.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



File Requirements

- Supported format: ZIP and TAR.GZ
- Maximum file size: 100 MB

After selecting a file, a progress bar will indicate upload status. During and after upload, the following buttons are available for managing the selected file:

- **Cancel Upload**: Stops file upload midway and discards the selected file.
- **Replace File**: Discards upload of the currently selected file and allows choosing of another project file from PC.

4. Click **Import Project** to confirm project file upload.

Uploading a model (BYOM)

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

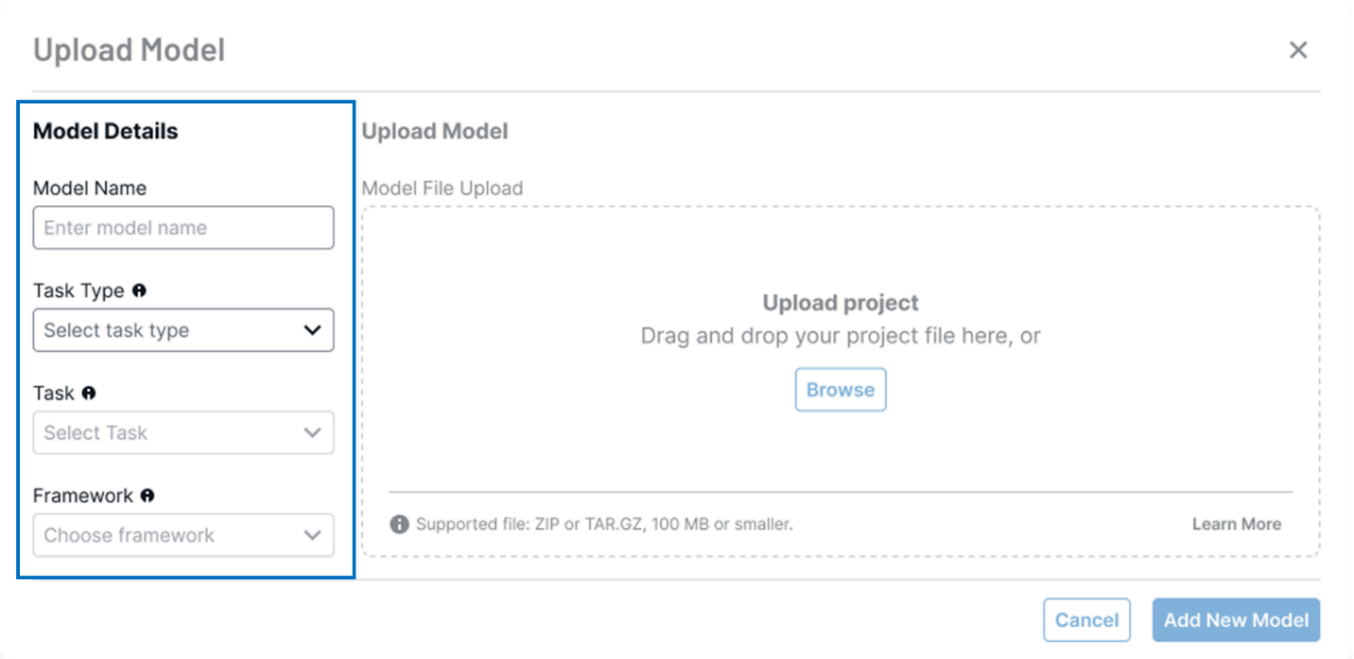
The Bring Your Own Model (BYOM) feature allows users to upload and deploy their own trained AI models to the EdgeBench platform. This is ideal for users who want to integrate custom architectures or domain-specific models that are not available in the ADI Model Zoo. BYOM supports flexible deployment on supported hardware, enabling tailored AI solutions.

Why Use BYOM?
BYOM empowers developers to leverage their own models for specialized tasks, ensuring compatibility with unique data and workflows. This feature also allows users to validate whether their models can be successfully compiled and deployed on target ADI hardware platforms.

Step 1: Prepare Your Model
Refer to the BYOM Models for EdgeBench for instructions on how to prepare and package your custom model.

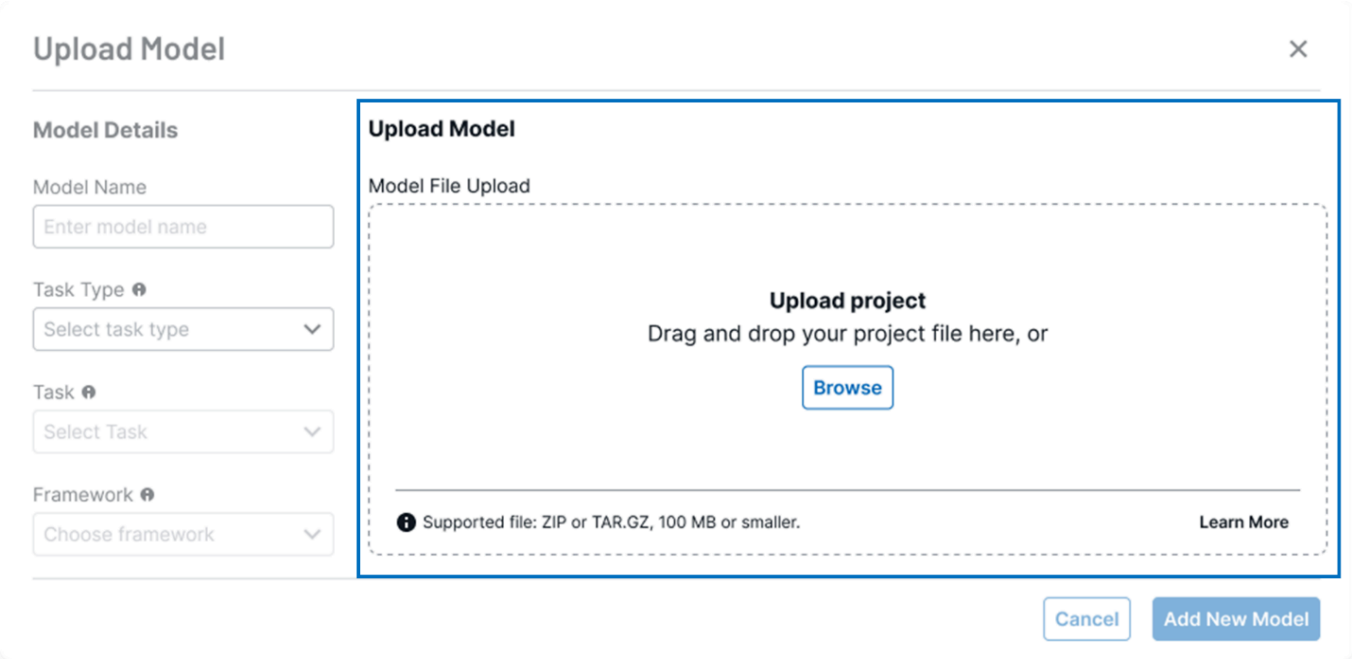
Step 2: Upload Your Model
Once your model package (ZIP, TAR) is prepared, you can upload it via Workspace.

1. Navigate to **Workspace → Models**.
2. Click **Upload Model**.
3. Fill in the required model details:
   a. **Model Name**: A descriptive name for your model.
   b. **Task Type**: Select the appropriate task type.
   c. **Task**: Choose appropriate task from the Task dropdown



4. Upload your model file:
   a. Drag and drop the archive into the upload area or click **Browse** to select it manually.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



5. After you upload the model file, the system automatically identifies the **Framework**. No need to select this manually.
6. Monitor the upload progress via the progress bar.

   Upload Controls

   - **Cancel Upload**: Stops the upload and discards the selected file.
   - **Replace File**: Allows you to discard the current file and select a new one.
7. Click **Add New Model** to complete the upload.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Exploring Model Zoo

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Exploring the ADI Model Zoo

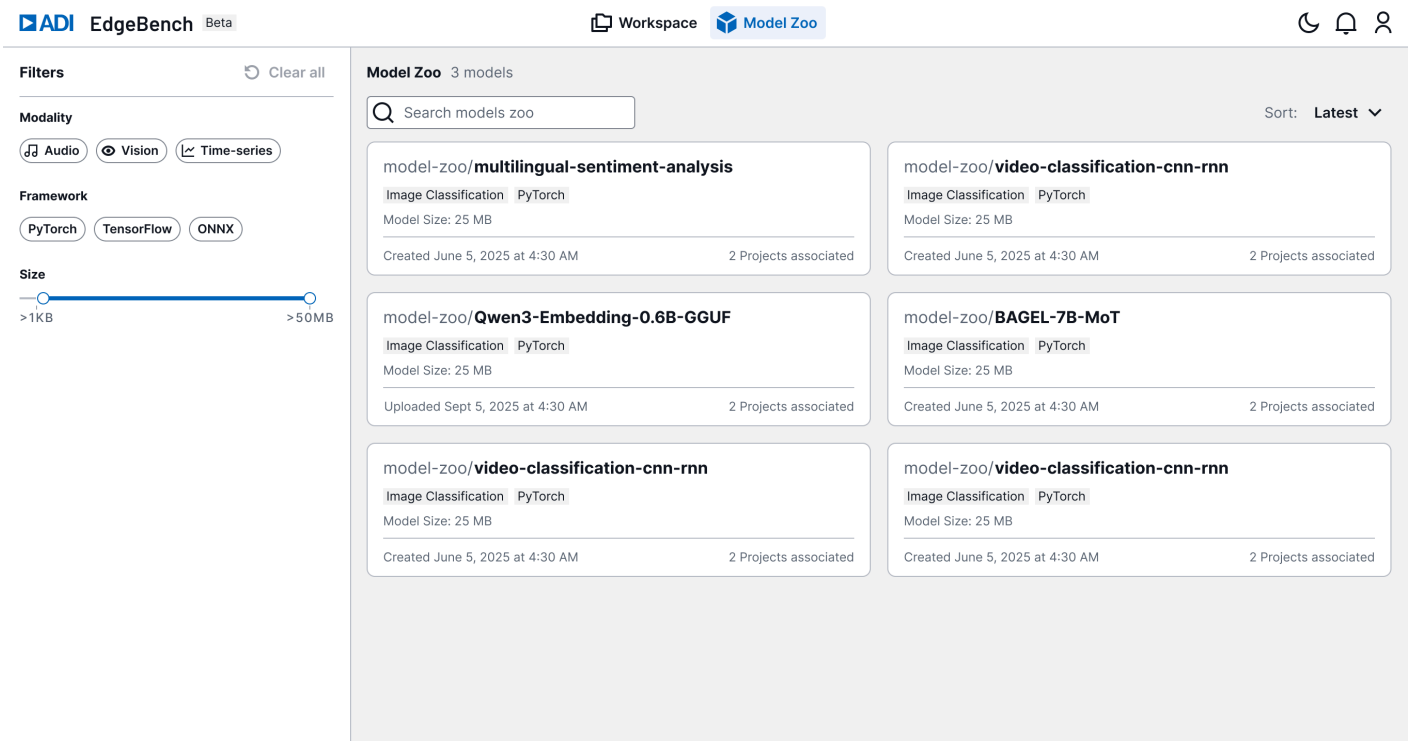
The **ADI Model Zoo** is a centralized repository of machine learning models provided by ADI. It offers a curated collection of **pre-trained models** that users can browse, search, and filter. Models available in EdgeBench are also accessible through Analog Devices' public GitHub repository: ADI Model Zoo

Key characteristics

- **Read-Only Access**: Users can view and utilize models from the model zoo, but these models do not count against their personal storage allocation.
- **Reusable Resources**: Models in the zoo can be reused across multiple projects, promoting consistency and efficiency.

Model list

Models in the zoo are displayed as **cards**, each providing essential information for quick reference.



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Model list management

Refer to Common User Controls for:

- Filter Panel
- Sort Options
- Search Bar
- Model Card Info

Model card info

Each card includes the following details:

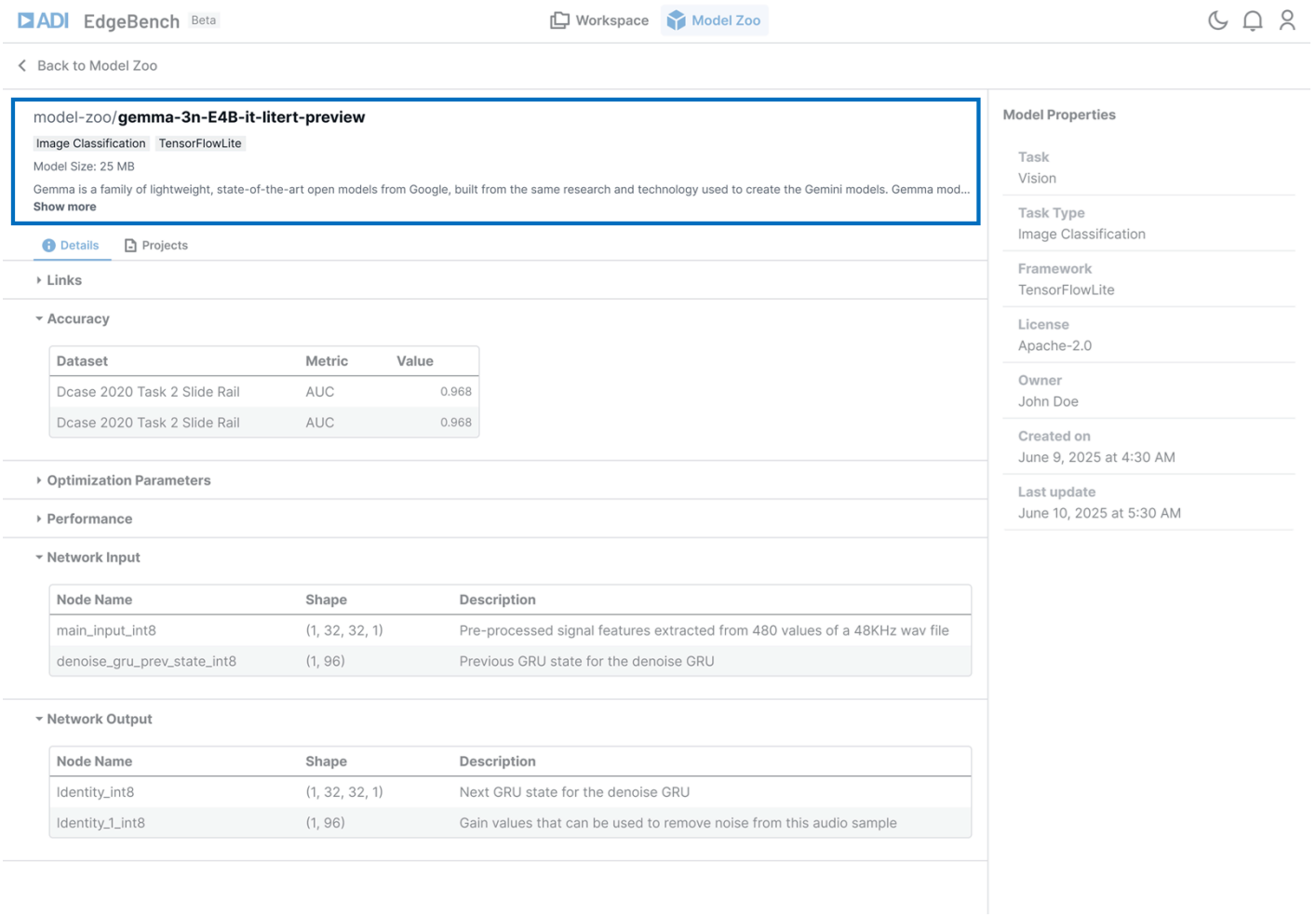
Field	Description
Model Name	The name assigned to the model by the user
Task Type	The specific task the model is designed to perform
Framework	The machine learning framework used to build the model
Model Size	The file size of the model, useful for assessing resource requirements
Creation Date and Time	Indicates when the model was uploaded or created
Number of Projects Associated	Indicates how many projects currently use this model

Model details page

The **Model Details Page** provides a comprehensive view of a selected model. It is divided into three main sections, each offering specific insights and controls.

Model information

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

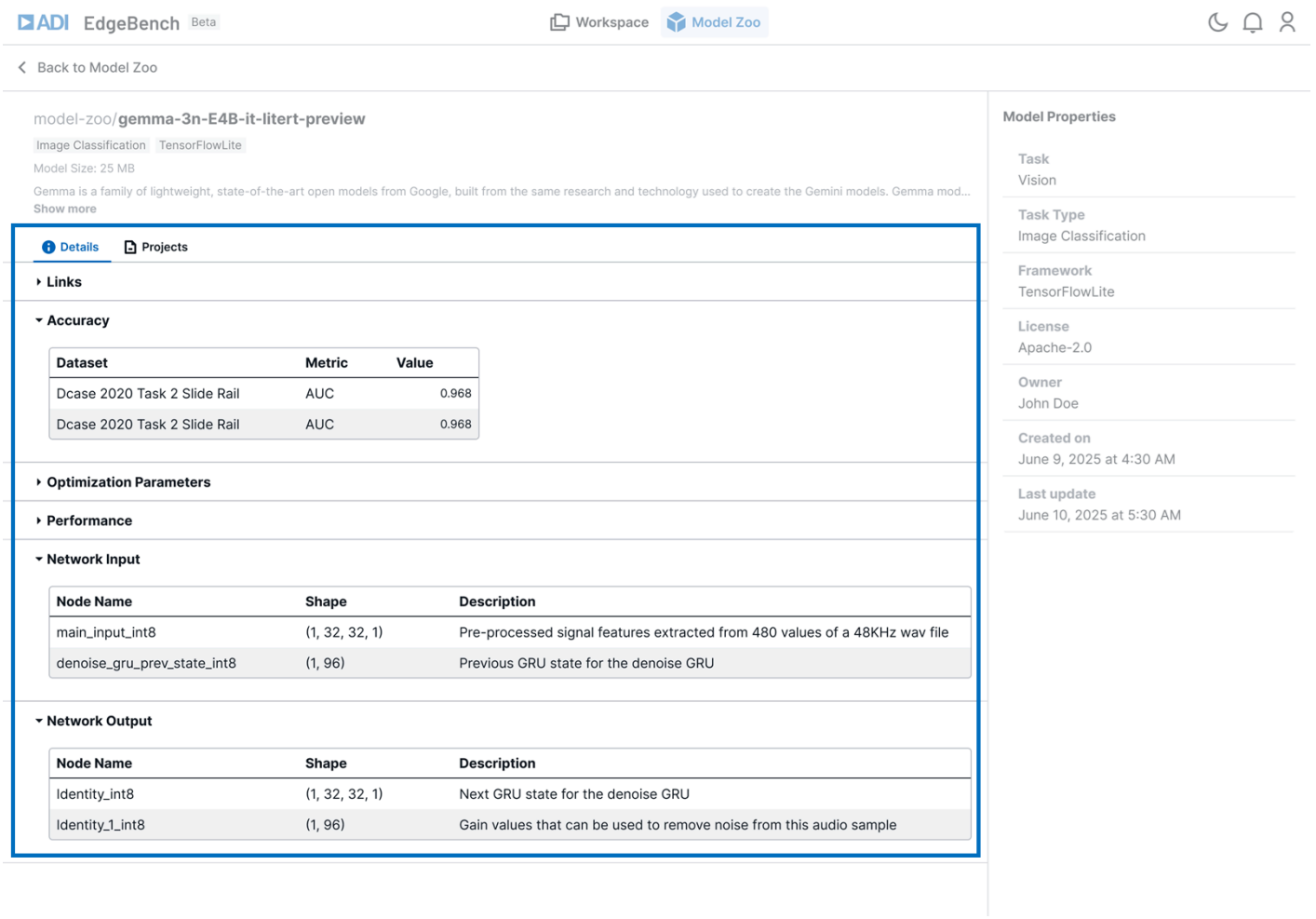


This section presents high-level identification and metadata about the model.

Field	Description
Model Title	The complete model name; helps users identify the model quickly.
Framework	The training framework used to develop the model.
Model Size	The size of the model.
Model Description	A brief description of the model

Technical details

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



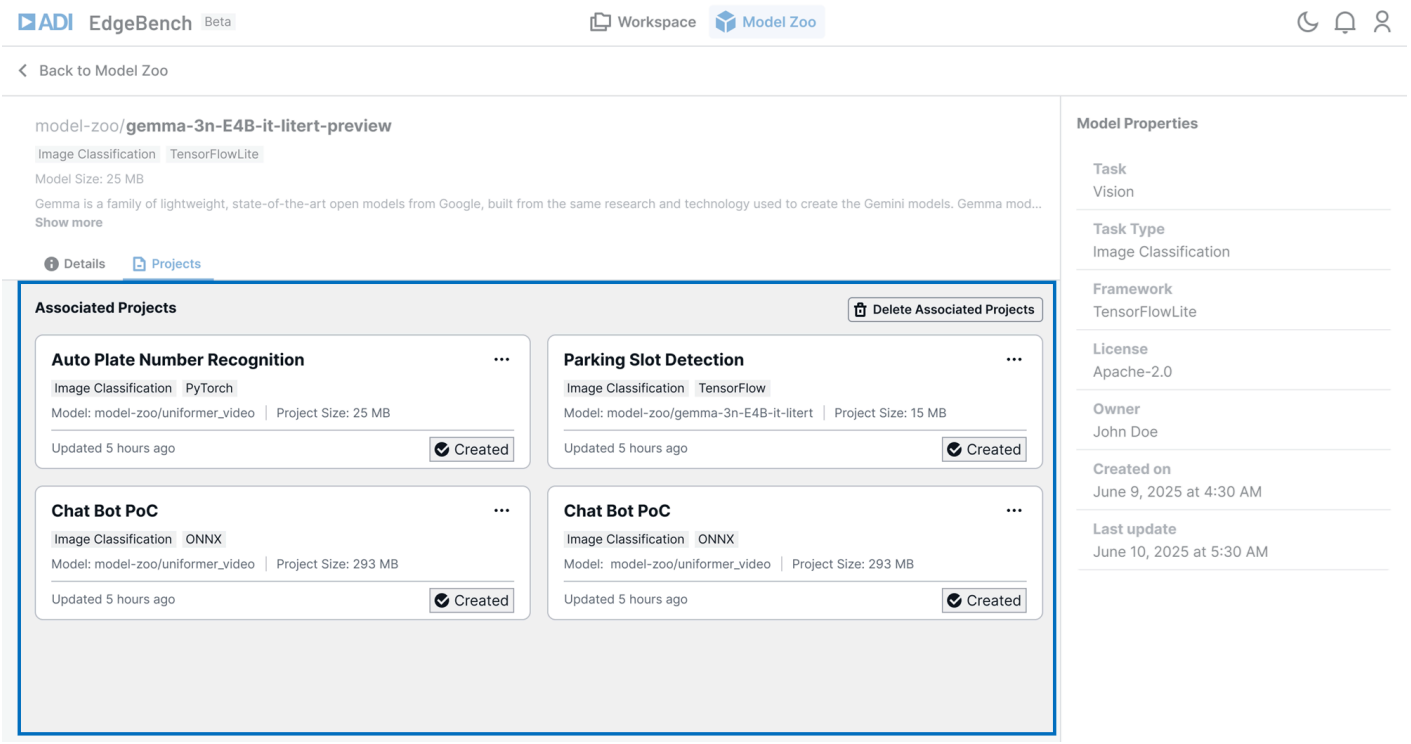
This section provides in-depth technical specifications and performance data for the selected model.

Section	Description
Resource Links	Direct access to documentation, repositories, or related assets
Accuracy	Performance metrics relevant to the model's task
Optimization Parameters	Configuration settings used to fine-tune the model for performance and efficiency
Network Input/Output	Specifications for data formats and expected input/output

Associated projects

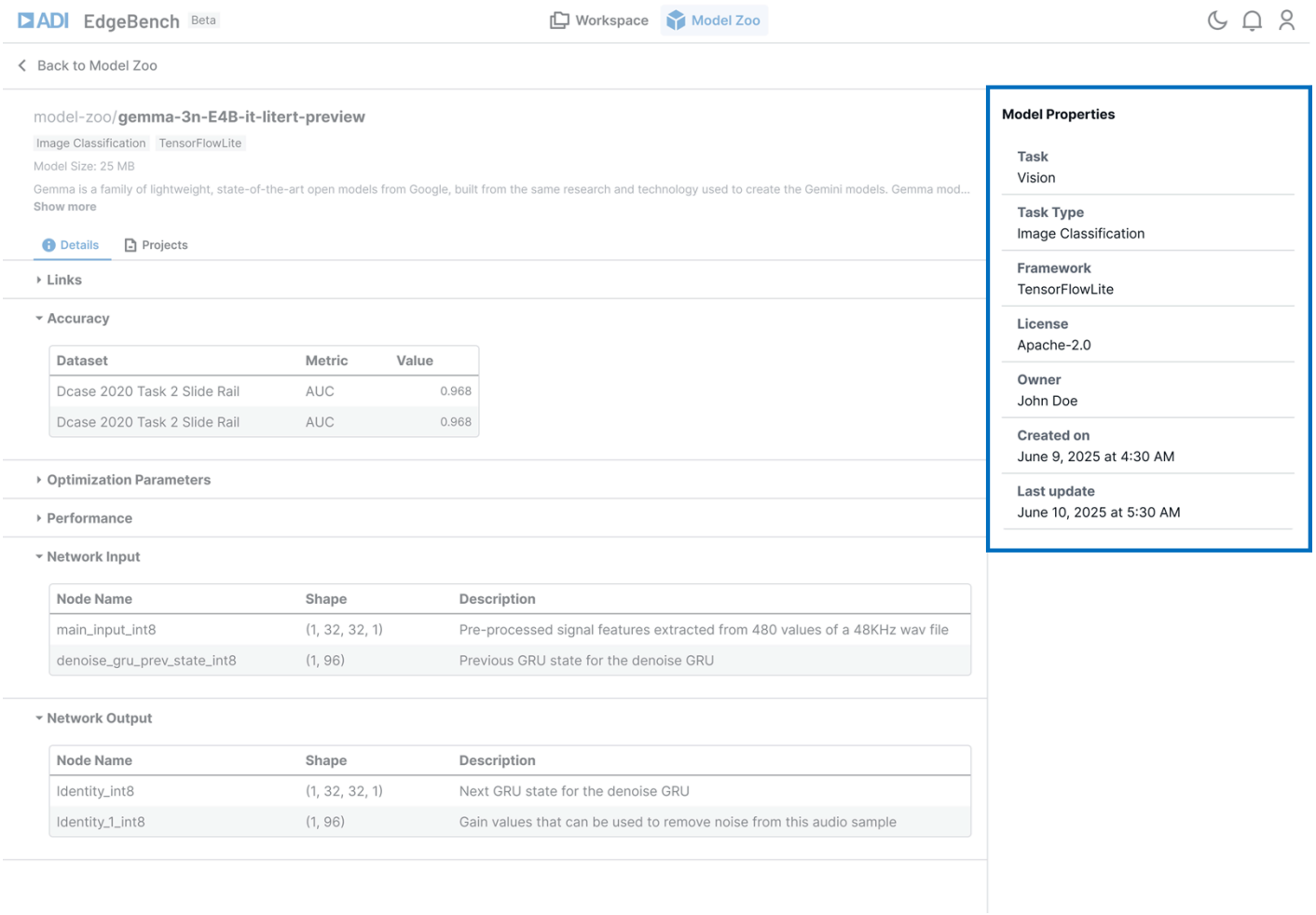
Clicking on the **Projects** tab reveals all projects associated with the selected model.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Model properties side panel

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



The right-hand panel displays the model's basic properties in a structured format. This includes essential metadata such as:

Field	Description
Model Name	The name assigned to the model
Task Type	The task the model is designed to perform (e.g., classification, detection)
Framework	The model training framework used to build the model
Owner Name	The user who uploaded or created the model
Creation Date	Timestamp of when the model was first uploaded or created
Last Updated	Timestamp of the most recent modification

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Exporting Builds for CFS

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

CodeFusion Studio Builds

The **EdgeBench** platform enables exporting EdgeBench builds into a CodeFusion Studio (CFS) project and allows a natural transition into production-grade development, integration, and deployment.

Pre-requisites

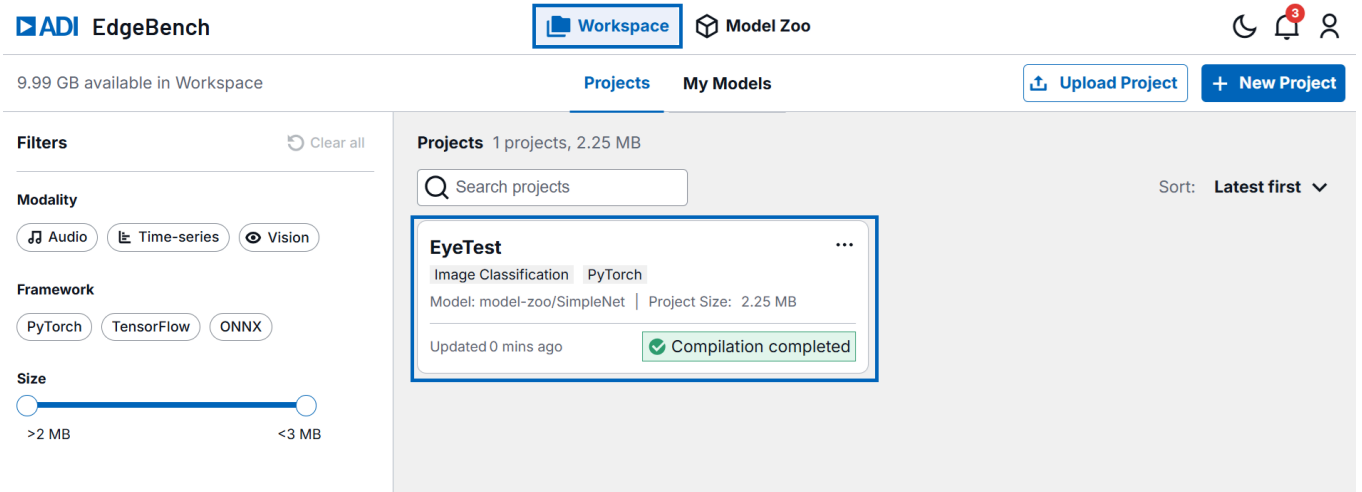
- Visual Studio Code
- CodeFusion Studio Plugin Version 2.2.0 or newer
- Compiled EdgeBench build

> **Important**
>
> You can only export a build if a project has sucessfully passed the **Compilation completed** state. Please see the Configure and compile page for more information.

Exporting builds

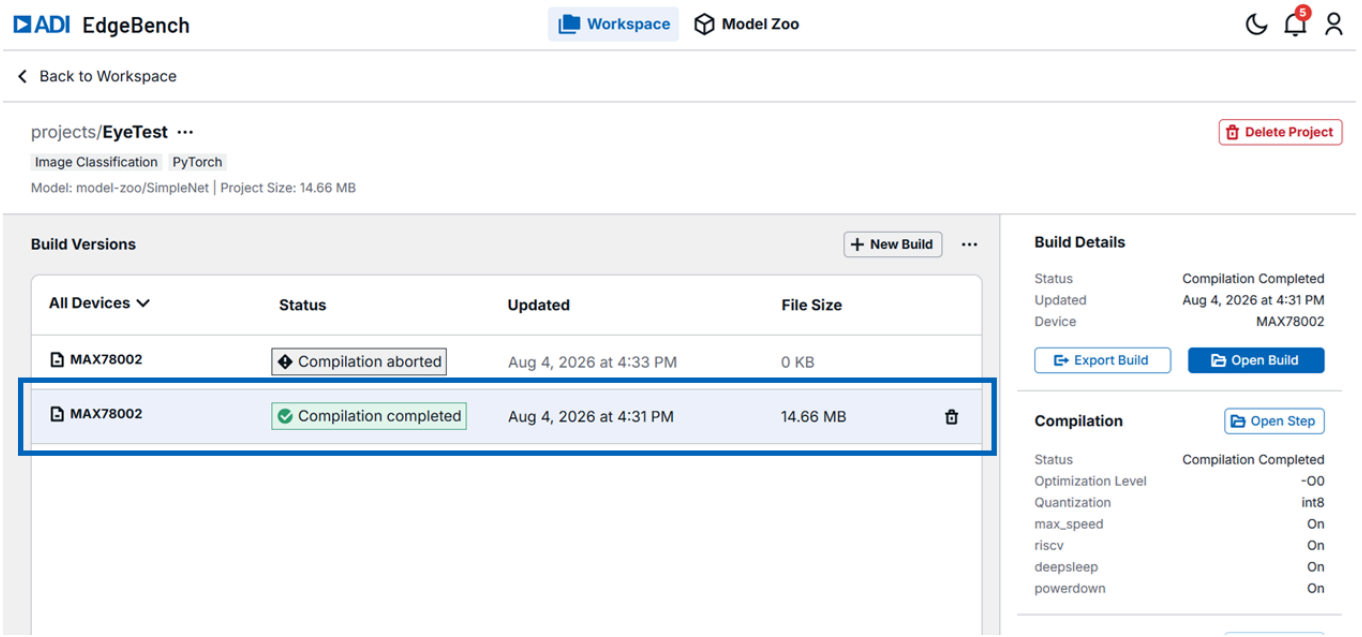
This section walks you through the exporting process, broken down into these key steps:

1. On the EdgeBench app, go to your **Workspace** and click the project with the build you want to export.

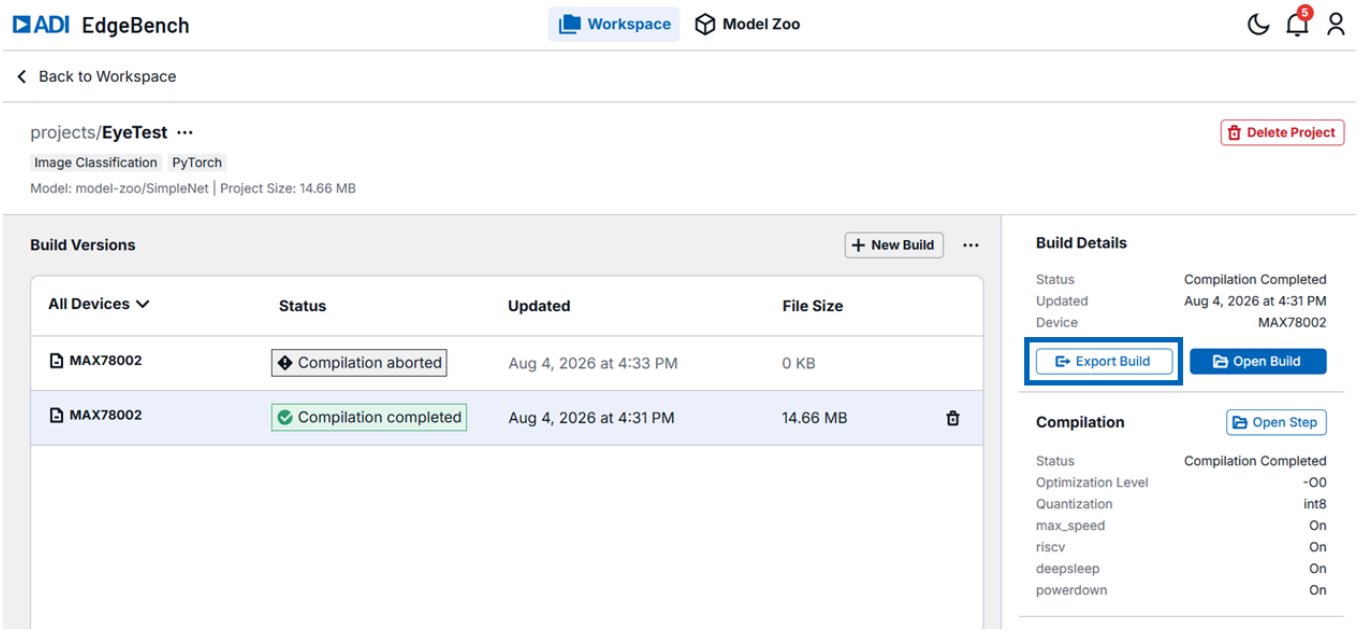


2. Click the specific build you want to export.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



3. Under Build Details, click **Export Build**.



> **Note**
>
> If the Export Build button is greyed out, the project has not yet reached the **Compilation completed** state.

4. EdgeBench automatically generates and downloads the file. Extract the folder.

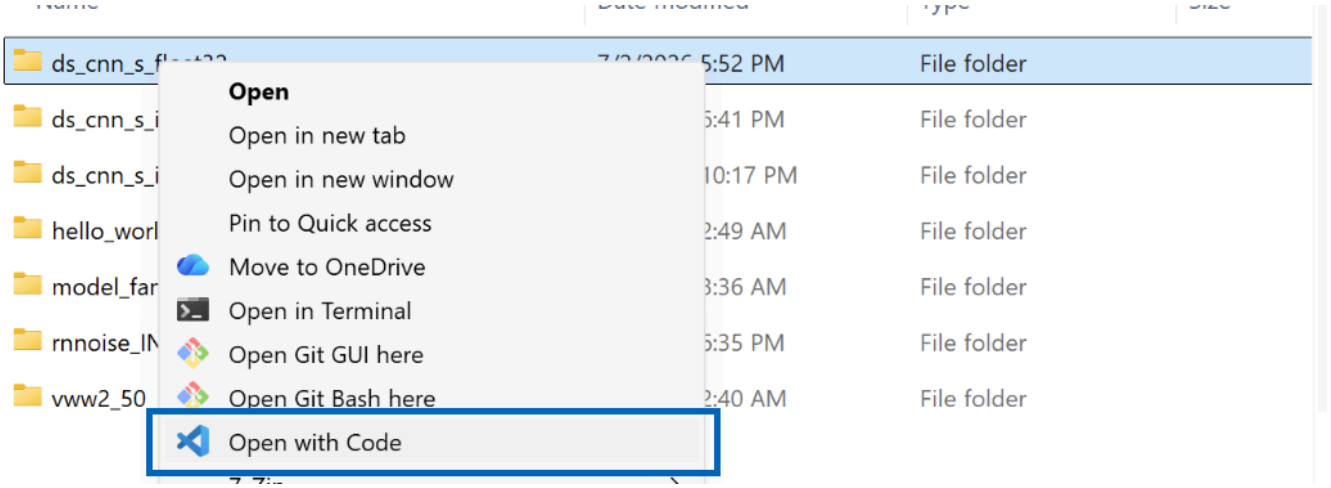
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Note

The folder will include the following necessary files:

```
.
├── CMakeLists.txt                          # CMakeLists document
├── aimodels.cmake                          # AI model's CMake source file
├── boards
│   ├── max32690evkit_max32690_m4.conf      # Board's configuration file
│   └── max32690evkit_max32690_m4.overlay   # Board's overlay file
├── ds_cnn_s_int8.tflite                    # Model's TFLITE file
├── prj.conf                                # Project's configuration file
└── src
    ├── adi_tflm
    │   ├── ds_cnn_s_int8.cpp               # Model's C++ source file
    │   └── ds_cnn_s_int8.hpp               # Model's C++ header file
    ├── input.h                             # Input source file
    └── main.cpp                            # Main source file
```

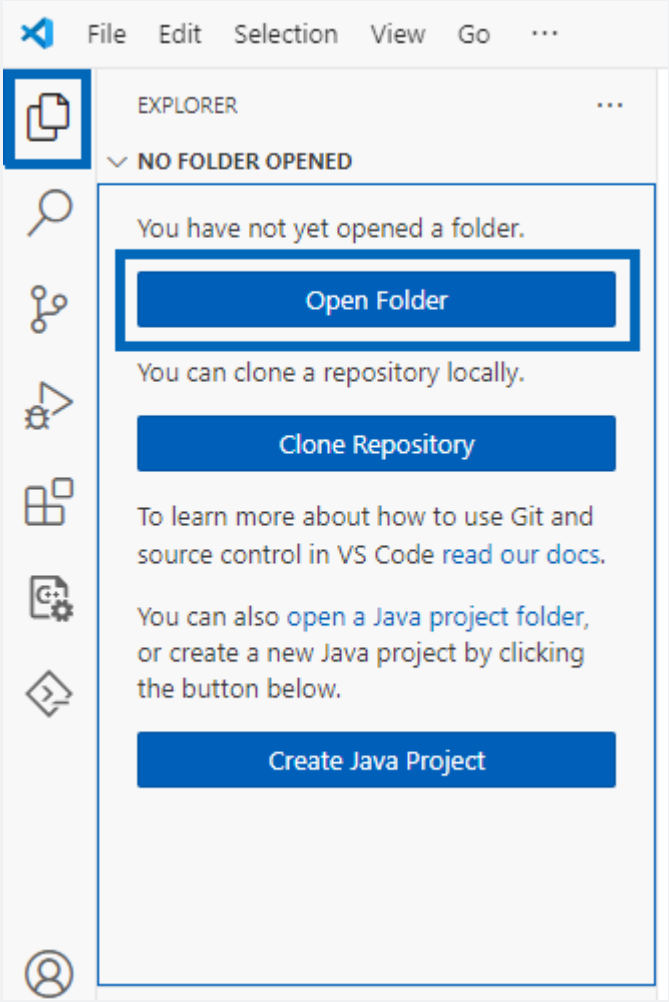
5. Right-click the extracted folder and click **Open in Code**.



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Tip

Alternatively, you can click on the **Explorer icon** in the VS Code activity bar and then click **Open Folder** or click **File** > **Open Folder** from the VS Code menu.



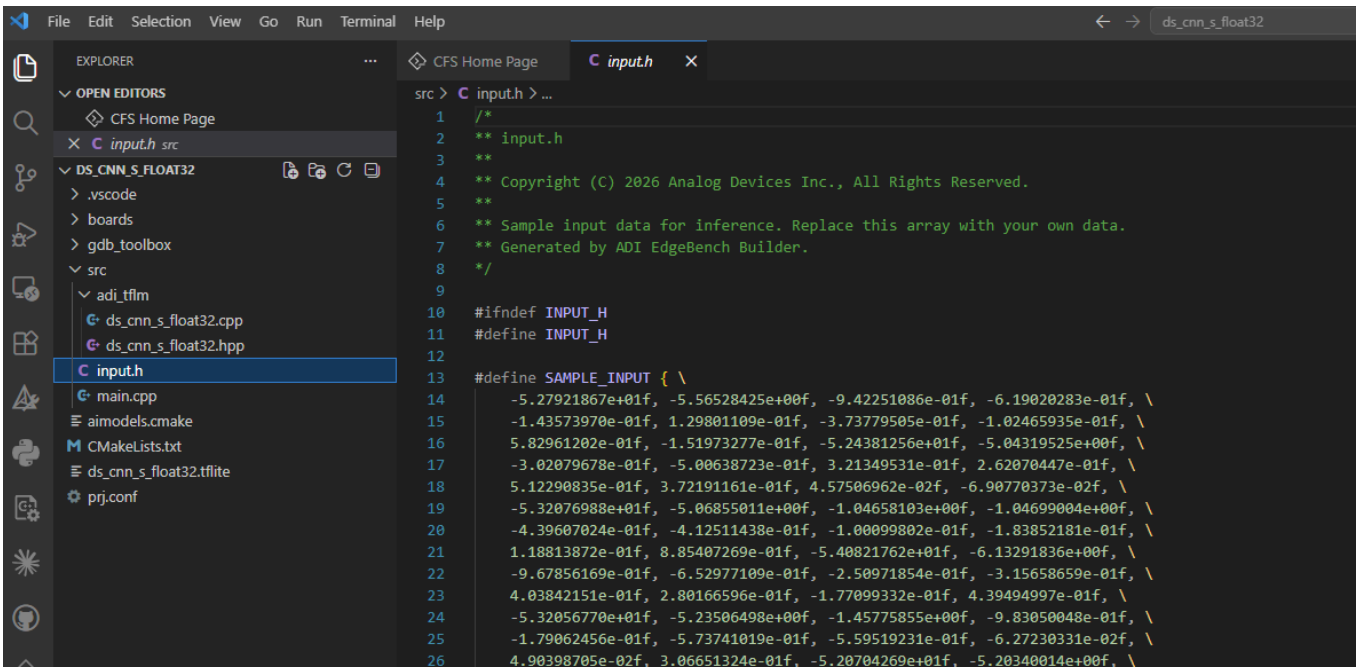
6. If a pop-up window at the bottom right appears that says **CFS: Configure workspace as a CodeFusion Studio Project**, click **Yes**.

Note

If you receive the notification **Do you trust the authors of the files in this workspace?**, click **Yes, I trust the authors** to continue.

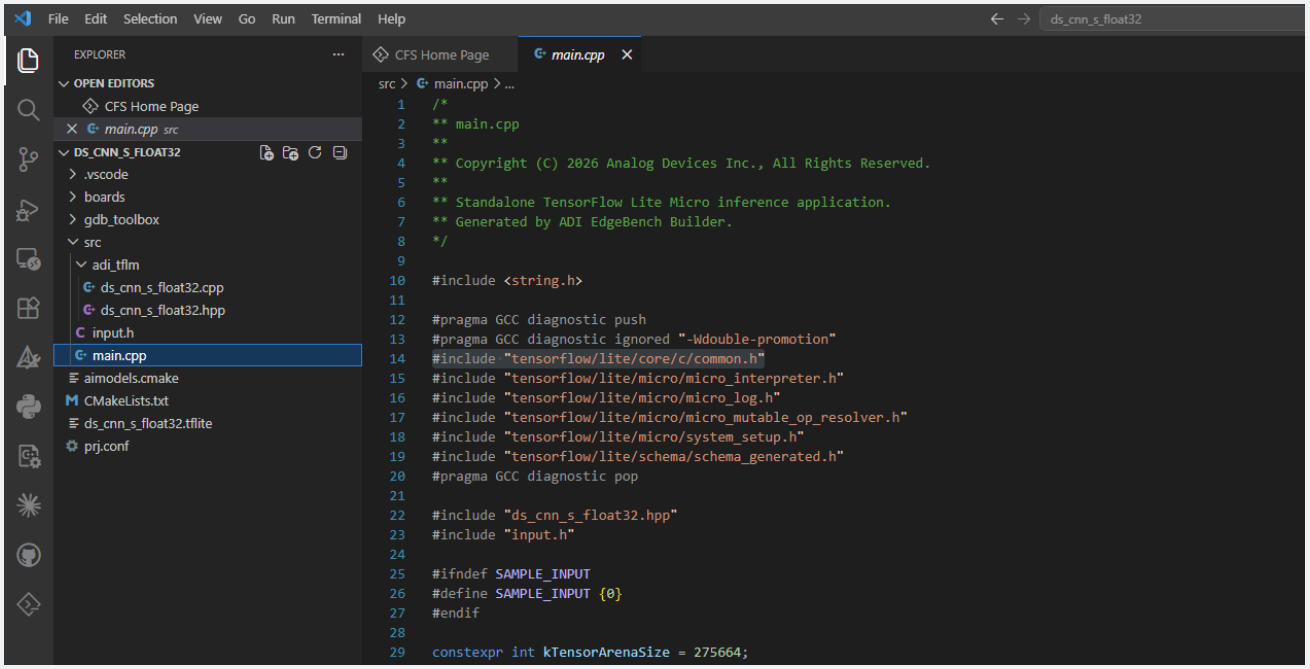
7. Edit the sample input (`input.h`) with your own data.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Note

Default sample data is 0. You can also edit the main source code (`main.c`) as needed.



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Important

For the MAX78002 only: The build uses `sampledata.h` and `sampleoutput.h`, which generate during the synthesis step of ai8x-synthesis.

Insert the following code into the `main.c` file:

```
// TODO: Replace with actual input data loading for your model.
// Each AI8X model has a specific memory layout for input data.
// See the model documentation for load_input() implementation.
void load_input(void)
{
    // Placeholder: load sample input data into CNN data memory
}
```

8. Your model is now a full CFS project. From here, you can build, flash, and debug on your target board. To learn how to use CFS, see CFS' User Guide and Documentation.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Prepare for BYOM

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Preparing for BYOM

The Bring Your Own Model (BYOM) feature allows users to upload and deploy their own trained AI models to the EdgeBench platform. This is ideal for users who want to integrate custom architectures or domain-specific models that are not available in the model zoo.

This guide provides guidelines on training, exporting, and packaging custom BYOM models for deployment with ADI EdgeBench to ensure successful compilation and flashing.

Training your model

The remote boards used in EdgeBench are edge devices with limited flash memory. As a result, large models may not fit within the available memory and can fail during compilation. Design your training pipeline with embedded constraints in mind such as avoiding high-resolution inputs or keeping input resolutions small

Supported dependency versions

To get best results, train and export your model using the following package versions.

Package	Version
Python	python>=3.1.0
Tensorflow	tensorflow>=2.12.0
PyTorch	torch>=2.1.2
Open Neural Network Exchange (ONNX)	onnx<=1.13.0

Model input and output size constraints

Additionally, the target platform uses nanopb (Protocol Buffers for embedded systems) to transmit input/output tensors. Nanopb supports messages up to `64 KB` by default. If a model's input or output tensor exceeds this limit, firmware compilation will fail even if the model is otherwise valid.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

As an example, for an input tensor with shape [224 × 224 × 3] and `float32` datatype (4 bytes) will result to 588KB. This exceeds nanopb's 64 KB limit by ~9×, causing a build failure during firmware compilation. `int8` quantization is strongly preferred.

Exporting your model

Ensure that your exported model includes both the network architecture and the trained weights. These may be packaged together in a single file or provided separately, depending on the framework. Refer to the guidelines below for framework-specific instructions.

PyTorch

Case	Required Files	Remarks
Case 1: Single Model File	- `PT` or `*PTH` (complete model architecture + trained weights) - `input_shape.txt`	`input_shape.txt` file must only contain a valid input shape in one of the accepted formats below: Tuple: `(1, 3, 224, 224)` Array: `[1, 3, 224, 224]` CSV: `1, 3, 224, 224`
Case 2: Model Definition + Weights	- `MODEL.PY` (full model definition) - `*PT` or `*PTH` (weights) - `input_shape.txt`	- Must include `MODEL.PY` containing the full architecture definition - `MODEL.PY` must contain the complete architecture class definition. The architecture class name must match the one used during training. Ensure no syntax errors and that file loads properly. - The **class name** in `MODEL.PY` must match the trained model - `input_shape.txt` uses the same accepted formats: Tuple: `(1, 3, 224, 224)` Array: `[1, 3, 224, 224]` CSV: `1, 3, 224, 224`

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Important

- If you upload weights only (PT/PTH), you must also include model.py with the complete architecture class definition.
- The architecture class name in model.py must match the name used during training.
- Ensure that:
- The MODEL.PY has no syntax errors.
- You can load the model without runtime errors in PyTorch.
- The input shape must match the model's expected dimensions.

TensorFlow

Case	Required Files	Remarks
Case 1: Model in H5 format	- `H5`	Complete model (architecture + weights).
Case 2: Model in PB format	- `PB`	Complete model with architecture. Must represent a **frozen graph**.
Case 3: Model in `SAVED_MODEL` directory format	- `SAVED_MODEL/` directory	Directory must contain the **complete exported model** (do not modify or delete internal files).

Important

- Ensure the model is properly exported from TensorFlow.
- Do not upload checkpoint-only or partial models.
- The model must be loadable using standard TensorFlow methods:
- H5 for full models,
- PB for frozen graphs,
- SAVED_MODEL/ for full SavedModel exports (keep internal structure intact)

ONNX

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Case	Required Files	Remarks
Case 1: Model in ONNX format	- `ONNX` - `ONNX.DATA`	Must contain the complete model graph and all trained parameters. Also include an *.ONNX.DATA file containing the large tensors (weights), particularly when they are too large to be embedded directly in the main .ONNX file.

Warning

- ONNX models must use opset version 13 or lower

Important

- Export with all required operators included (no missing ops). Check the list below.
- Ensure ONNX version compatibility with your target runtime.
- Verify the file is not corrupted or partially exported.

List of ONNX operators convertible to TF by opset

- Abs
- Acosh
- Add
- AveragePool
- Concat
- Conv
- ConvTranspose
- Div
- Dropout
- GRU
- Input
- InstanceNormalization
- LSTM
- LayerNormalization

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- Matmul
- MaxPool
- Mod
- Mul
- RNN
- Reshape
- Shape
- Slice
- Softmax
- Sub

Creating model metadata files

Some tasks require metadata files so that the EdgeBench model and data-processing pipeline can correctly interpret outputs. Missing metadata may result in errors or inaccurate results. Metadata filenames are case-sensitive and must match exactly.

Class labels metadata

The `classes.txt` metadata file is a text file that contains the labels for classification models. Tasks requiring `classes.txt` are as follows:

- Keyword Spotting
- Genre Identification
- Image Classification
- Image Segmentation
- Object Detection
- Visual Wake Word

When create the class labels metadata file, keep in mind the following.

Rule	Description
One per line	Write **one class label per line**

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Rule	Description
No extras	Don’t include commas, brackets, extra spaces, or numbers (unless part of the label)
No empty lines	The file must not contain blank lines
Order matters	Class order must **match the model’s output order**
Count must match	The **number of labels** must equal the model’s **output dimension**

Click here to download an example `classes.txt` file.

Sampling rate metadata

The `sampling_rate.txt` metadata file contains a single numeric value representing the sampling rate (in Hz). Tasks requiring `sampling_rate.txt` are as follows:

- Denoiser

When create the sampling rate metadata file, keep in mind the following:

Rule	Description
First line only	The first line must contain **one integer**. The integer value is implicitly in Hz; no need to write the unit in the line.
No units	Do **not** include “Hz” or any text.
No extra characters	No symbols or trailing spaces
Must match training	Value must match the **training sampling rate**.

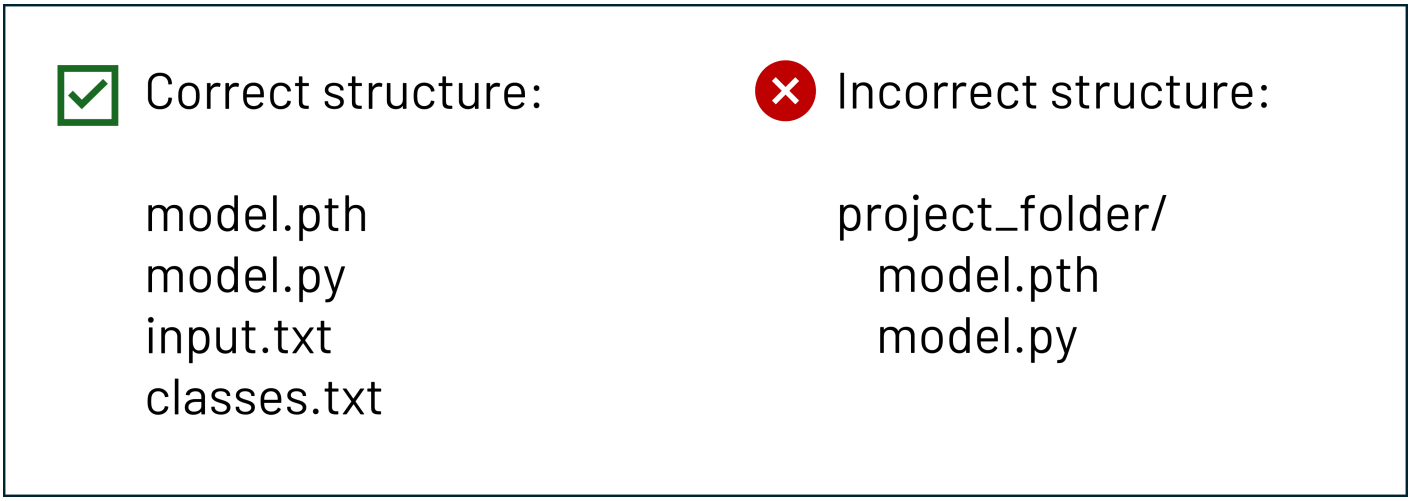
Click here to download an example `sampling_rate.txt` file.

Packaging your model

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The EdgeBench platform requires you to create a compressed archive file containing all of the model and metadata files.

- Accepted archive formats: ZIP or TAR.GZ
- Maximum file size: 100 MB
- Folder structure: Place model files at the top-level directory inside the archive.
- Input Shape File: Required only for PyTorch models in .txt format with accepted formats as listed above.



Once you prepare the model package (ZIP, TAR), you can upload it through the Workspace. See Workspace Tasks page for detailed instructions.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Data Processing Scripts

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Data Processing

The **Bring Your Own Model (BYOM)** feature requires data processing scripts to ensure that model inputs and outputs are compatible with the model architecture and the target task.

Data processing consists of two script types:

- **Pre-processing scripts** transform raw input files (for example, `.JPG`, `.PNG`, `.WAV`) into the input shape and data format expected by the model.
- **Post-processing scripts** transform raw model outputs into human-readable and task-appropriate results for presentation and analysis.

This guide provides guidelines on creating custom data processing scripts, which are **mandatory** for all BYOM workflows.

Script Editor Environment

The script editor operates within a Jupyter-like Python environment, these three imports are mandatory for data processing script files:

- argparse
- numpy
- pathlib

These are the only supported libraries in the runtime environment:

- torch (2.1.2+cpu)
- torchvision (0.16.2+cpu)
- tensorflow-cpu (2.15.0)
- requests (2.32.5)
- pyyaml (6.0.3)
- pandas (2.3.3)
- scipy (1.15.3)
- matplotlib (3.10.8)
- opencv-python (4.11.0.86)
- librosa (0.9.1)

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Develop your script outside EdgeBench before uploading it for validation. This environment allows users to test custom post-processing logic directly within the platform.

Creating a Pre-processing Script

When entering the pre-processing step, the script editor displays a template similar to the following:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```python
import argparse               # Standard library for parsing command-line arguments
from pathlib import Path      # Cross-platform, safer file path handling
import numpy as np            # NumPy for numerical operations and .npy files
                              # These three imports are mandatory for all postprocessing script
files

def preprocess(file_path: str, quant_type: str) -> np.ndarray:
    """
    Main preprocess function which applies the preprocessing logic
    on the input files

    This function should:
    1. Load the raw input data from file path
    2. Apply domain-specific preprocessing
    3. Return a NumPy array ready to send to model for inference

    Parameters
    ----------
    1. file_path : str
        Absolute Path to the input file.
        This may be an audio file, image, or time-series file.
        Example: "data/sample.wav" or "images/input.jpg"

    2. quant_type: str
        Quantization type to use while processing the input files
        Example: "int8"

    Returns
    -------
    np.ndarray
        Preprocessed data formatted to pass to model for inference.

    Notes
    -----
    Implement task-specific preprocessing logic here, such as:
    - Audio: loading waveform, resampling, Mel spectrograms
    - Vision: resizing, normalization, channel reordering
    - Time series: windowing, smoothing, feature extraction
    """
    # TODO: Load file and implement real preprocessing logic
    # Example:
    # data = load_audio(file_path)
    # features = compute_mel_spectrogram(data)
    # return features

    return  # Placeholder return


if __name__ == "__main__":
    """
    Script entry point.

    Example command-line usage:
    ---------------------------
    python preprocess.py \
        --source data/input.wav \
        --output_file_path outputs/preprocessed.npy \
        --quant_type float32
```

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```
    """

    # Create an argument parser that defines expected CLI inputs
    parser = argparse.ArgumentParser(
        description="Preprocess an input file and save the result as a NumPy (.npy) array"
    )

    # Required argument: path to the raw input file
    parser.add_argument(
        "--source",
        required=True,
        help=(
            "Absolute Path to the input file to be preprocessed."
        )
    )

    # Required argument: output .npy file path (including filename)
    parser.add_argument(
        "--output_file_path",
        required=True,
        help=(
            "Absolute Path where the preprocessed .npy file will be saved"
        )
    )

    # Required argument: quantization type
    parser.add_argument(
        "--quant_type",
        required=True,
        help=(
            "Quantization type: int8, int16, float32"
        )
    )

    # Parse command-line arguments into a namespace object
    args = parser.parse_args()

    # Extract parsed arguments
    source_path = args.source
    output_file_path = Path(args.output_file_path)
    quant_type = args.quant_type

    # Run preprocessing on the input file
    x = preprocess(source_path, quant_type)

    # Save the resulting NumPy array to disk
    np.save(output_file_path, x)

    # Print helpful confirmation messages
    print(f"Processed data shape: {x.shape}")
    print(f"Saved preprocessed data to: {output_file_path}")
```

The core of this template is the `preprocess()` function, which accepts the following parameters:

1. **file_path: str** - Absolute path to the input file. Supported inputs include image, audio, and time-series files. An example is `"images/input.jpg"`.
2. **quant_type: str** - Quantization type applied to the processed data. An example is `"int8"`.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The pre-processing script **must** output a NumPy array and write it to a `.NPY` file for inference.

Pre-processing scripts typically perform the following steps:

1. Load raw data from the input file
2. Extract task-specific features
3. Transform data to match the model's input shape
4. Apply quantization (if required)

The following discussion uses an image classification pre-processing script as an example.

Loading Raw Data

Load the raw data from the `file_path` argument using commonly supported libraries:

- **Images (`.PNG`, `.JPG`):** `PIL.Image.open(file_path)`
- **Audio (`.WAV`, `.MP3`):** `librosa.load(file_path)`
- **Time series (`.CSV`):** `pandas.read_csv(file_path)`

Example for loading an image:

```
from PIL import Image

p = Path(file_path)
if not p.exists():
    raise FileNotFoundError(f"Input file not found: {file_path}")

img = Image.open(p)
```

Feature Extraction

Extract the desired features from the raw data. Implementation here depends on the task and model training pipeline.

Common operations include:

- For **image tasks**, resizing, normalization, channel reordering
- For **audio tasks**, resampling, conversion to mel spectrograms
- For **time-series tasks**, windowing, smoothing, feature extraction

Example for image classification:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```
import numpy as np

img = img.convert('L')
img = img.resize((28, 28), Image.BILINEAR)
arr = np.asarray(img).astype(np.float32) / 255.0
```

Transforming the Data

Processed data must match the exact input shape expected by the model.

Example for an image classification model expecting (1, 1, 28, 28) input shape:

```
arr = arr[np.newaxis, np.newaxis, :, :]
```

Quantizing Inputs

If required, quantize data based on `quant_type` parameter:

```
q = quant_type.lower()
if q == 'float32':
    return arr.astype(np.float32)
elif q == 'int8':
    # Map float [-1,1] -> int8 range [-127,127]
    out = np.round(arr * 127.0).astype(np.int8)
    return out
elif q == 'int16':
    # Map float [-1,1] -> int16 range [-32767,32767]
    out = np.round(arr * 32767.0).astype(np.int16)
    return out
else:
    raise ValueError(f"Unsupported quant_type: {quant_type}. Use float32, int8 or int16.")
```

Validating the Pre-processing Script

To validate your pre-processing script on EdgeBench, go to Adding a Pre-processing Script.

Creating a Post-processing Script

When entering the post-processing step, the script editor displays a template similar to the following:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```python
import argparse               # Standard library for parsing command-line arguments
from pathlib import Path      # Safer, cross-platform path handling
import numpy as np            # Numerical operations and .npy file handling
                              # These three imports are mandatory for all postprocessing script
files

def postprocess(file_path: str,quant_type: str) -> np.ndarray:
    """
    Main postprocess function which applies the postprocessing logic
    on the model output

    This function is expected to:
    1. Load a NumPy (.npy) file from path
    2. Apply any task-specific transformations
    3. Return a NumPy array ready to be passed in the pipeline

    Parameters
    ----------
    1. file_path : str
        Absolute Path to the input .npy file containing raw model outputs.
        Example: "outputs/raw_output.npy"

    2. quant_type: str
        Quantization type to use while processing the model outputs
        Example: "int8"


    Returns
    -------
    np.ndarray
        The postprocessed NumPy array.

    Notes
    -----
    Replace the placeholder logic below with real postprocessing steps,
    such as:
    - Normalization or scaling
    - Casting to a specific dtype (e.g., int8, int16, float32)
    - Reshaping or filtering values
    - Converting logits to probabilities
    """
    # TODO: Load file and apply real postprocessing logic
    # Example:
    # data = np.load(file_path)
    # processed = data / np.max(data)
    # return processed

    return  # Placeholder return


if __name__ == "__main__":
    """
    Entry point when running this file as a script.

    Example command-line usage:
    ---------------------------
    python postprocess.py \
        --source outputs/raw_output.npy \
```

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```
        --output_file_path outputs/final_output.npy \
        --quant_type float32
    """

    # Create the argument parser object
    # This defines which command-line arguments the script accepts
    parser = argparse.ArgumentParser(
        description="Postprocess model outputs and save them as a NumPy (.npy) file"
    )

    # Required argument: input model output file
    parser.add_argument(
        "--source",
        required=True,
        help=(
            "Absolute Path to the input .npy file containing raw model outputs."
        )
    )

    # Required argument: output file path (including filename)
    parser.add_argument(
        "--output_file_path",
        required=True,
        help=(
            "Absolute Path where the postprocessed .npy file will be saved."
        )
    )

    # Required argument: quantization type
    parser.add_argument(
        "--quant_type",
        required=True,
        help=(
            "Quantization type: int8, int16, float32"
        )
    )

    # Parse arguments provided via the command line
    args = parser.parse_args()

    # Extract argument values
    source_path = args.source
    output_file_path = Path(args.output_file_path)
    quant_type = args.quant_type

    # Run postprocessing on the input file
    result = postprocess(source_path, quant_type)

    # Print array shape for debugging or validation
    print(f"Shape of postprocessed data: {result.shape}")

    # Save the final NumPy array to disk
    np.save(output_file_path, result)

    print(f"Saved postprocessed data to: {output_file_path}")
```

The core of this template is the `postprocess()` function, which accepts the following parameters:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

1. **file_path: str** - Absolute path to the raw output file. An example is `"outputs/raw_output.npy"`.
2. **quant_type: str** - Quantization type applied to dequantize the data. An example is `"int8"`.

The post-processing script **must** output a NumPy array and write it to a `.NPY` file for UI presentation.

Pre-processing typically performs the following steps:

1. Loading raw model outputs
2. Dequantizing values (if applicable)
3. Transforming results for task-specific presentation

The following discussion uses an image classification post-processing script as an example.

Loading Model Outputs

Load the raw model output data from inference using np.load.

```
import numpy as np

p = Path(file_path)
if not p.exists():
    raise FileNotFoundError(f"Model output file not found: {file_path}")

# Load raw model outputs
data = np.load(p)
```

Dequantizing Outputs

If applicable, dequantize data as shown below:

```
q = quant_type.lower()
if q == 'float32':
    outputs = data.astype(np.float32)
elif q == 'int8':
    # int8 was encoded as round(float * 127)
    outputs = data.astype(np.float32) / 127.0
elif q == 'int16':
    # int16 was encoded as round(float * 32767)
    outputs = data.astype(np.float32) / 32767.0
else:
    raise ValueError(f"Unsupported quant_type: {quant_type}. Use float32, int8 or int16.")
```

Transforming Outputs for Display

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The post-processing logic to prepare output data for UI display vary by task. The table below shows the expected post-processing output for each task:

Domain	Task	Expected Output	Structure
Vision	**Classification**	Probabilities mapped to the model's labels	`np.array([prob_1, prob_2, prob_3])` **Shape:** `(1, n)`, `n` being the number of classes
Vision	**Image Segmentation**	Two image masks (such as, one for background, one for subject)	`np.array([[[val_1, val_2, ...], [val_1, val_2, ...], ...], [[val_1, val_2, ...], [val_1, val_2, ...], ...]])` **Shape:** `2, H, W`, corresponds to 2 masks of input image with height `H` and width `W`
Vision	**Object Detection**	Array of dictionaries for each detected object specifying bounding box, label, and confidence score	`np.array([{"box": box, "label": label, "score": score}])` **Shape:** `(n, 1)`, `n` being the number of objects detected **Bounding box format:** `(x_ref, x_length, y_ref, y_length)`
Vision	**Visual Wake Word**	Probabilities mapped to the model's labels	`np.array([prob_1, prob_2, prob_3])` **Shape:** `(1, 2)`, corresponding to detected or not detected
Audio	**Denoising**	Audio data in NumPy format	`np.array([data_1, data_2, data_3...])` **Shape:** `(n,)`, `n` being the number of audio samples
Audio	**Genre Identification**	Probabilities mapped to the model's labels	`np.array([prob_1, prob_2, prob_3])` **Shape:** `(1, n)`, `n` being the number of genres
Audio	**Keyword Spotting**	Probabilities mapped to the model's labels	`np.array([prob_1, prob_2, prob_3])`

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Domain	Task	Expected Output	Structure
			Shape: `(1, n)`, `n` being the number of keywords
Time Series	**Motor Fault Detection**	Time-series data	`np.array([<same shape as input>])` **Shape:** Same shape as the input data. The UI adapter computes Mean Squared Error (MSE) loss between input data and the output reconstructed data, hence, they **must** have the same shape.
Time Series	**Anomaly Detection**	Time-series data	`np.array([<same shape as input>])` **Shape:** Same shape as the input data. The UI adapter computes Mean Squared Error (MSE) loss between input data and the output reconstructed data, hence, they **must** have the same shape.

Example logic for image classification task, converting from logits to probabilities:

```
exp_shift = np.exp(outputs - np.max(outputs, axis=1, keepdims=True))
probs = exp_shift / np.sum(exp_shift, axis=1, keepdims=True)

return probs
```

Validating the Post-processing Script

To validate your post-processing script on EdgeBench, go to Adding a Post-processing Script.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

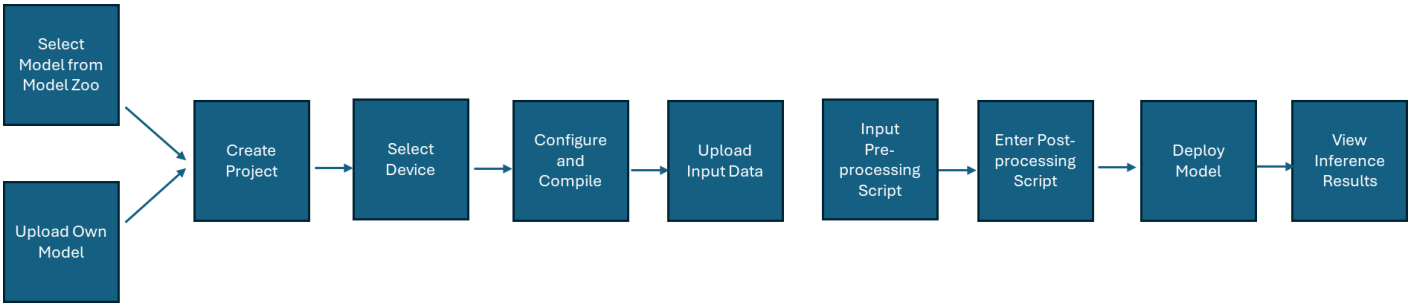
Build Process

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Build Process Overview

The **EdgeBench** platform offers a streamlined, step-by-step workflow for deploying AI models to edge hardware. The journey begins with the **User**, who acts as the central driver of the process—creating projects, uploading models, configuring builds, and generating results.

This section outlines the step-by-step process for deploying AI models using the platform. The process consists of these key steps:



1. Start with a Project

The user initiates the workflow by creating a **Project**, which serves as a container for organizing related models and builds. Each project is defined by its name, task type, and framework.

2. Choose or Import a Model

The user can either select a pre-trained model from the **ADI Model Zoo** or upload a custom model using the **BYOM** feature. Uploaded models must follow specific format and packaging requirements and are stored within the **Workspace**.

3. Select Target Device

With the model in place, the user chooses the **Device** on which the model will be deployed. This selection determines hardware-specific compilation settings.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

4. Configure and Compile

The user sets **Compilation Flags** and **Optimization Levels** to tailor the build for performance, power efficiency, or compatibility. These settings are saved as part of the **Configuration**, which is used to generate a **Build**—a deployable unit combining the model and device settings.

5. Upload Input Data

The user provides **Input Files** for inference, which may include images, audio, or time-series data. These files are used to test the model's performance on real-world data.

6. Add Pre-Processing Script (Required for BYOM)

To ensure compatibility between input data and model expectations, the user uploads a **Pre-Processing Script**. This script transforms raw input into a format suitable for inference.

7. Add Post-Processing Script (Required for BYOM)

The user defines how the model's output should be interpreted using a **Post-Processing Script**, which formats results for analysis or downstream use.

8. Run Inference

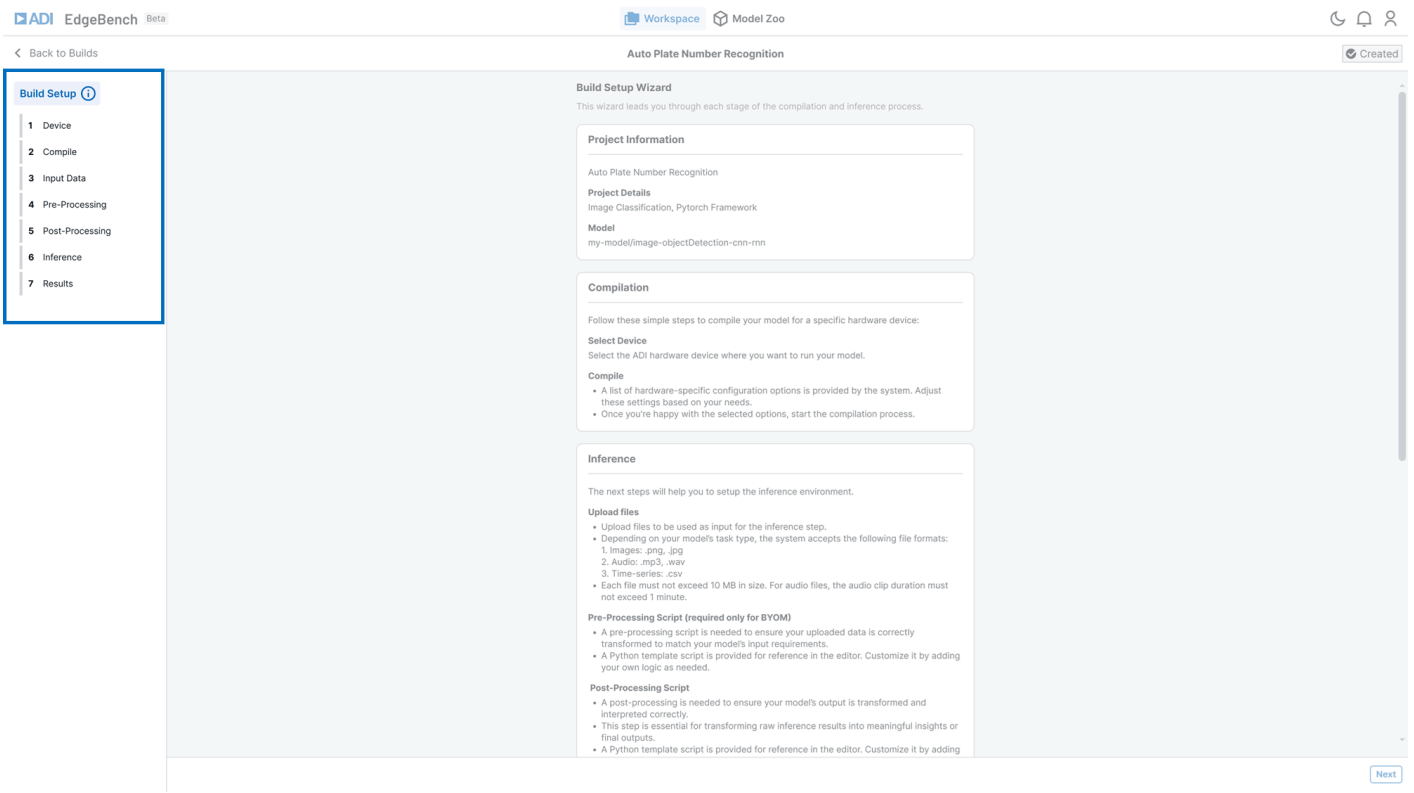
With all components in place, the user initiates **Inference** on the selected device. A progress monitor provides real-time feedback during execution.

9. View Results

Once inference is complete, the user accesses the **Results**, which include output data, performance metrics, and accuracy scores. These results are timestamped and organized by batch for easy tracking.

Throughout the journey, a fixed navigation panel remains visible, highlighting the current step and helping users stay oriented as they move through the workflow.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



The detailed setup instruction for each step follows in the succeeding sections.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Device Selection

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

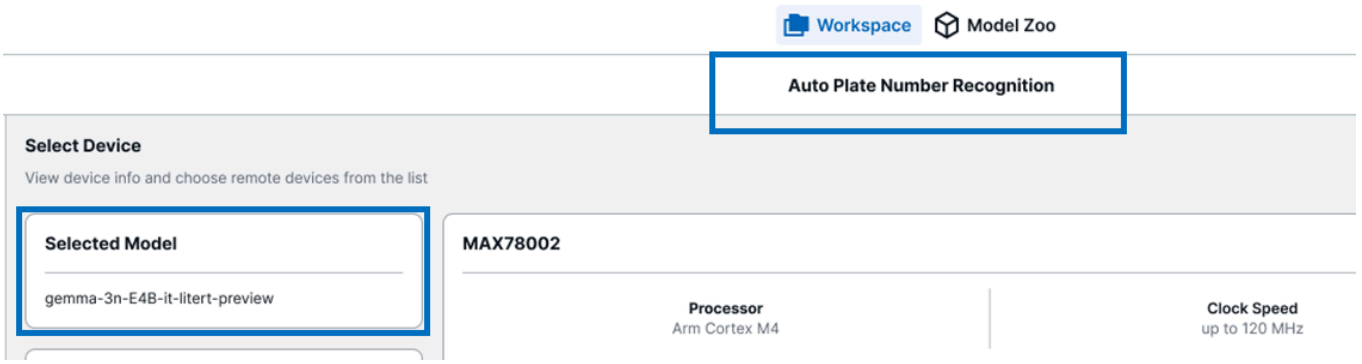
Device Selection

To begin the **build setup**, users must select a **device** for deploying their **AI model**. The list of available boards shows devices currently hosted in the **ADI Remote Board Farm** that are compatible with the selected **task type**.

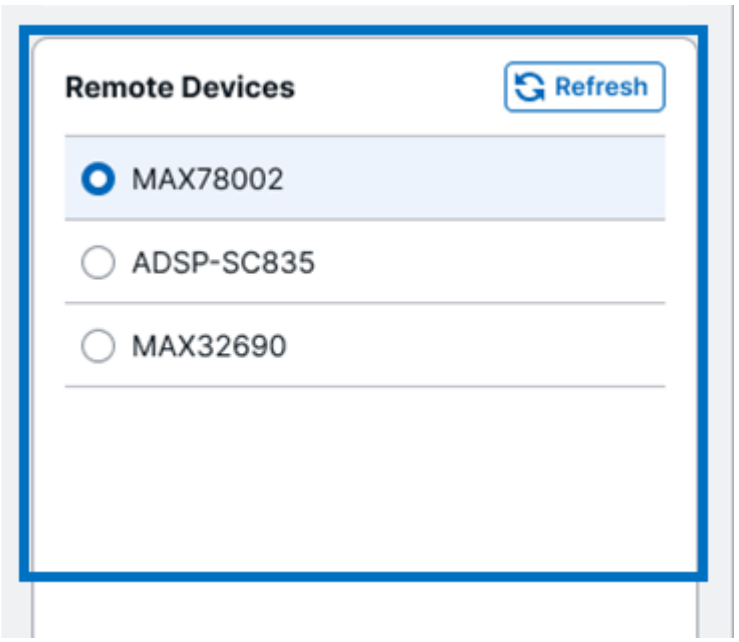
Before proceeding, users should ensure that the model they are deploying is compatible with the selected device's **processor** and **specifications**. The **device selection** step is **required**, and the **build setup** process cannot proceed until a **compatible device** is selected.

Steps for selecting a device for model deployment

1. Review **Project** and **Model** information.



2. Browse the list of available **remote devices**.



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

> **Note**
>
> The device list is dynamic and may change based on **availability**. Always check for the latest options before proceeding. Click the **Refresh** icon to update the device list if needed.

Check device availability based on the following criteria:

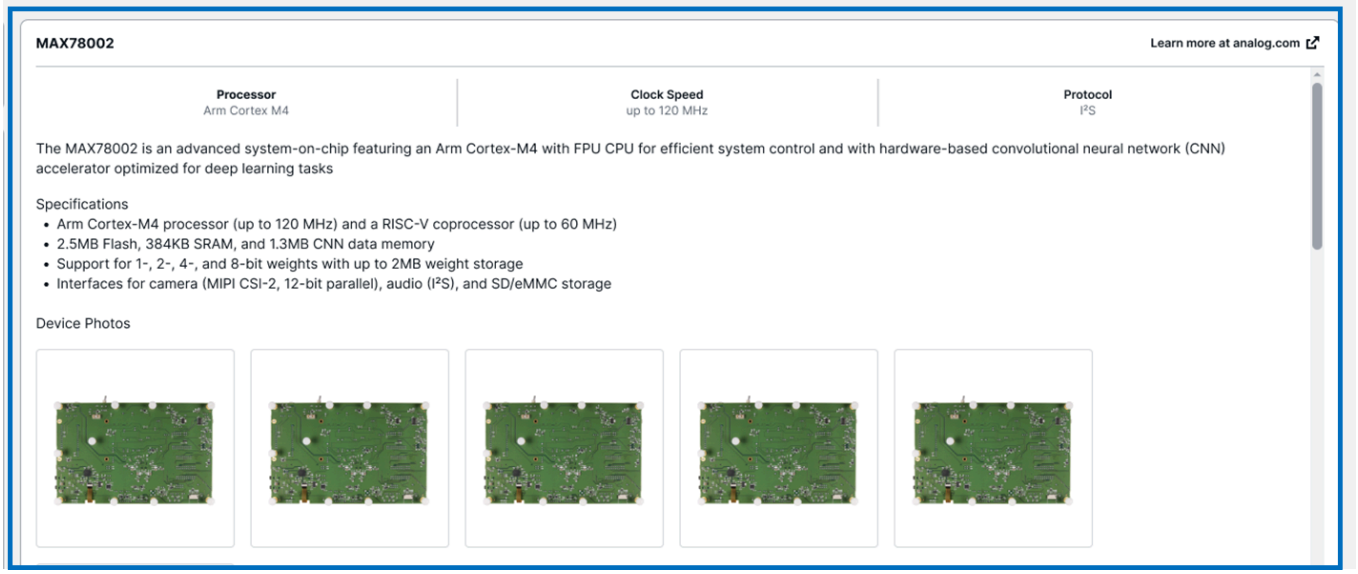
If a board is **selectable**, it means the board is:

- **Available** for use, and
- **Compatible** with the model's **framework** and **task type**.

Non-selectable devices may be:

- **In use** by another session, or
- **Incompatible** with the selected model.

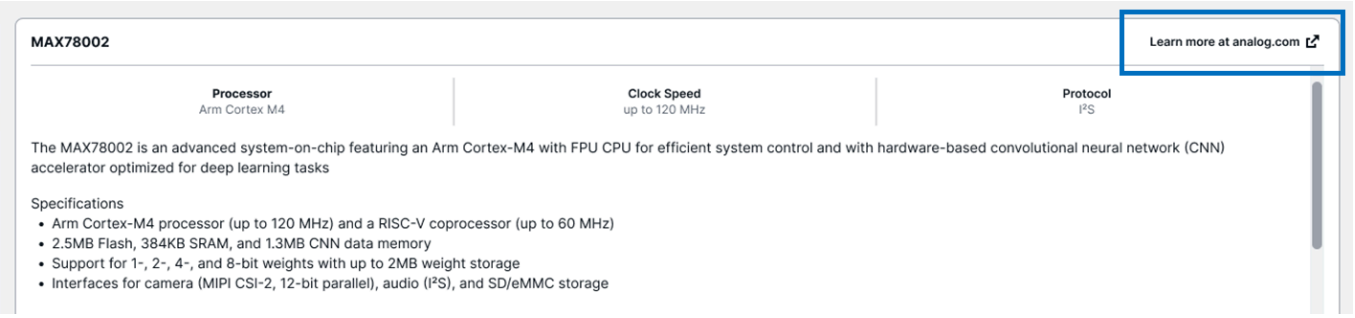
3. Select a Device. Click on a device to view its details in the **Device Info Panel**, including:
   - Processor type
   - Clock speed
   - Supported protocols
   - Board photos
   - General specifications



4. Ensure the selected device meets your model's requirements in terms of:
   - Processing architecture
   - Memory capacity
   - Supported AI framework
   - Task type

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

5. *(Optional)* Click the **Learn more on analog.com** link to open the product page for full specifications.



6. Once a **compatible device** is selected and reviewed, click **Next** to continue to the **Compile** step.

Tip

If the selected device becomes **unavailable** during **Step 6: Deploy and Inference**, users can go back to **Step 1: Device Selection** and choose a different device. This can be done without restarting the entire setup process.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Configure and Compile

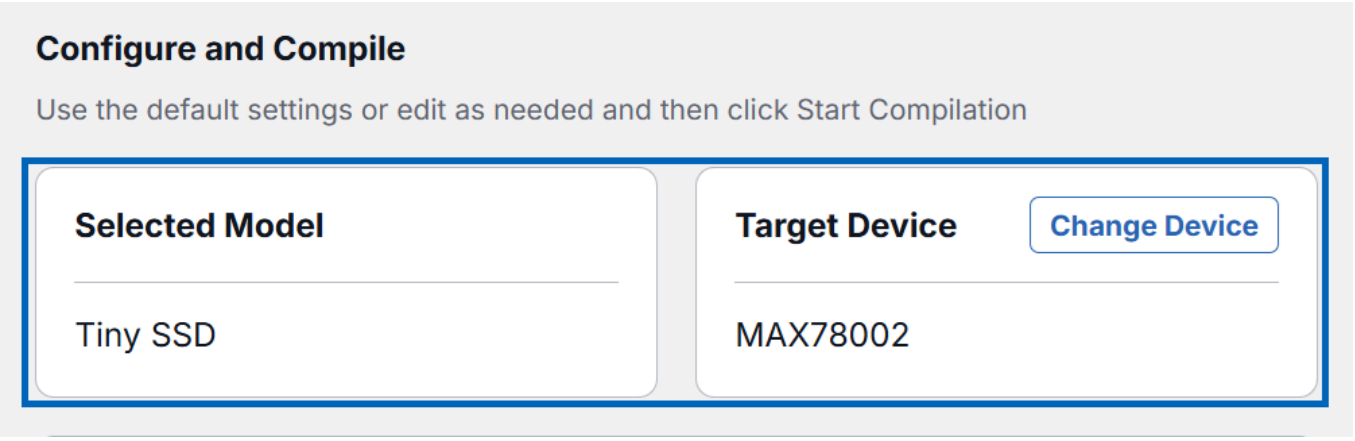
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Configure and compile

The configure and compile step enables users to configure hardware-specific compilation settings, allowing for optimized model performance on the selected target hardware. It provides flexibility to experiment with different configurations, which can significantly influence how the model performs once deployed on the actual hardware.

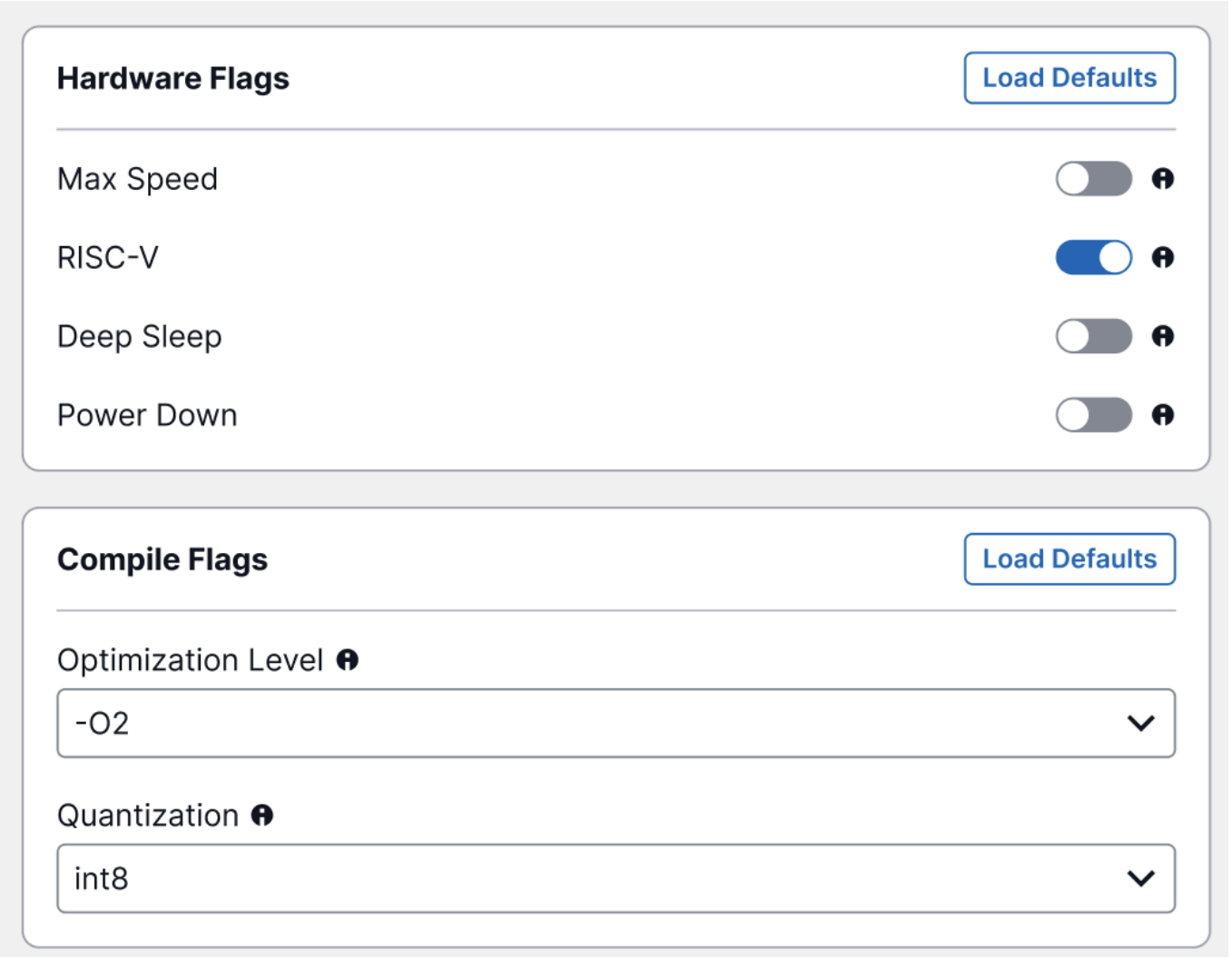
Starting the compilation process

1. Review **Selected Model** and **Target Device**. Ensure you have selected the correct model and device to avoid compiling the wrong model.



2. (Optional) If you want to change the target device, click the **Change Device** button to select a different target hardware. This returns you to Step 1 Device Selection to select another device.
3. Access the **Hardware Flags** and **Compile Flags** sections to customize these parameters as needed to suit the specific requirements of your model. See the Compilation Flag Settings Overview section for more details.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

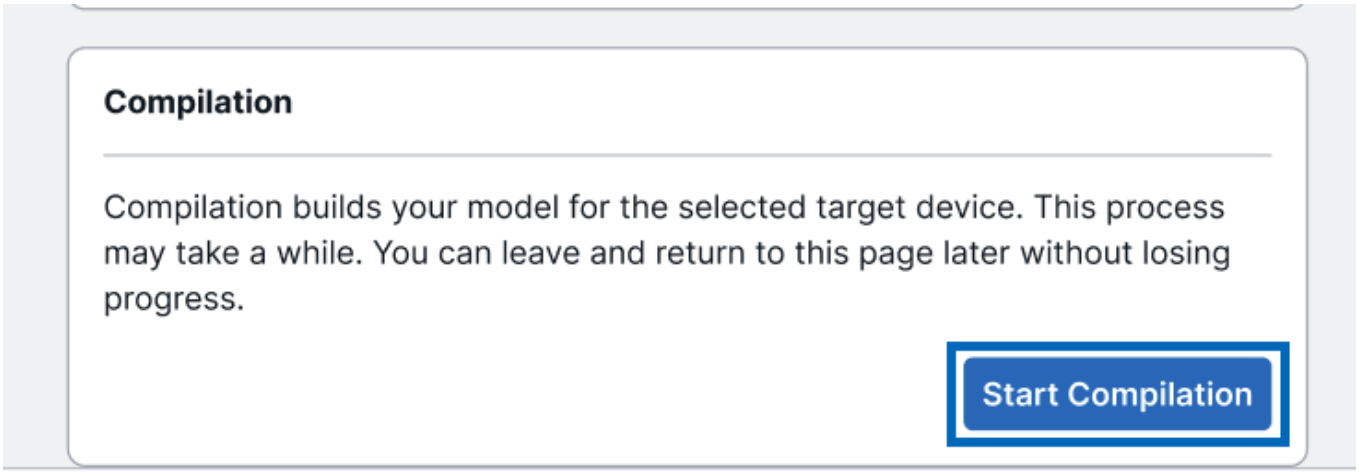


Note

Upon selecting a device, the default hardware settings provided by ADI are automatically loaded. These settings dynamically change based on the target hardware selected. If you want to revert all changes and restore the recommended default values, click the **Load Defaults** button.

4. Review the settings to make sure they are correct.
5. Click **Start Compilation** to begin building the model for the selected device.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Once the compilation process begins, the interface displays real-time feedback to help users monitor progress and detect issues. These indicators include:

- **Progress Indicators**

  A progress bar and icon show the current status of the compilation. Please see Compilation Bar and Icon Progress Indicators section for a description of each status.

- **Failure Indicators**

  If the compilation fails, both the progress bar and the top-right status icon will indicate the failure, with detailed logs available for troubleshooting.

- **Logs Pane**

  Detailed logs are displayed in the bottom pane, providing a summary of background activities and any errors encountered. These logs can be collapsed or expanded and are useful for troubleshooting and understanding what went wrong during compilation.

Important

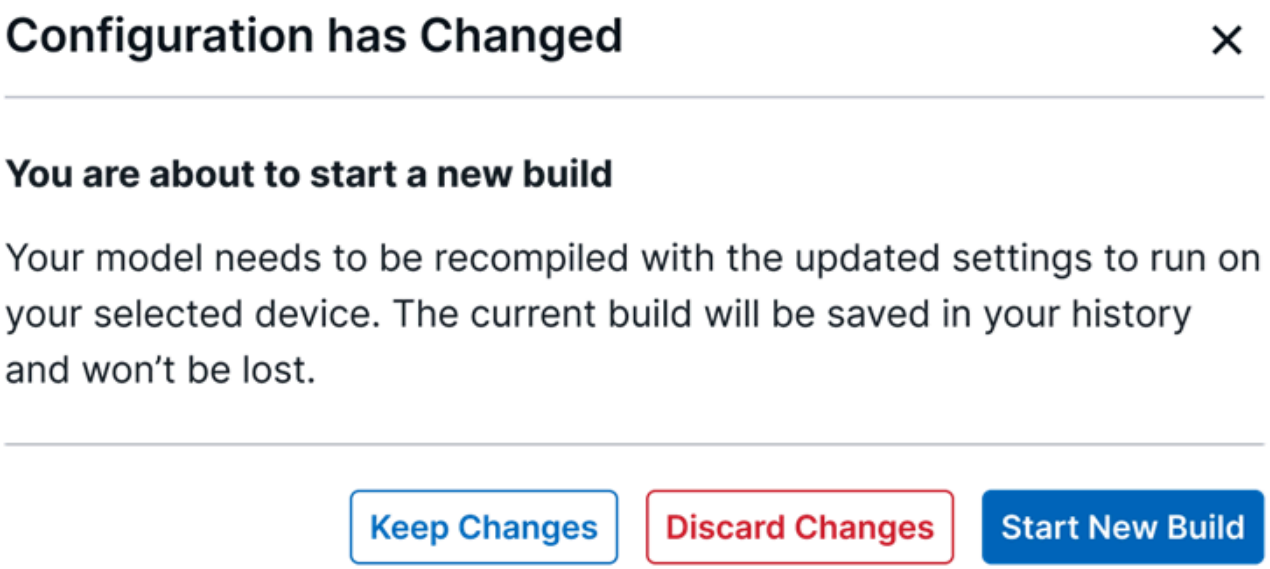
During compilation, changes to any of the build steps will not be permitted. Users can only modify the build step settings after the Compilation process is completed or if the process is aborted.

Tip

You can click **Abort Compilation** anytime to stop the process. Stopping the compilation will not reset any changes you made.

When you reconfigure compilation settings after starting a build, a confirmation window appears, informing about recompiling a new build with the updated settings. Note that starting a new build will save the previous build in your history.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Confirm whether to keep, discard, or accept changes using the following buttons:

- **Start New Build** – accept the configuration changes and initiate a new build.
- **Discard Changes** – use previously loaded configuration prior to the recent settings modification and do not accept any changes; no new build will be started when this option is selected.
- **Keep Changes** – select this to retain the changes to configuration, but do not start a new build.

6. Once the compilation is complete, click **Next** to proceed.

> **Note**
>
> The Next button will remain disabled until compilation is complete.

> **Important**
>
> You can now also export the project and import it into CodeFusion Studio. For more information, see Exporting Builds for Codefusion Studio.

> **Tip**
>
> Steps 3, 4, and 5 may be completed in any order.

Compilation flag settings overview

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Optimization levels

Flag	Description
-O0	Reduces compilation time and ensures debugging produces expected results. (Default)
-O2	Performs nearly all supported optimizations that do not involve space-speed tradeoffs.

Quantization options

Type	Description
int8	Model weights and activations use 8-bit integer data type.
int16	Model weights and activations use 16-bit integer data type.
float32	Model weights and activations use 32-bit floating point data type.
implied	Model weights and activations are defined by the user-uploaded model.

Important: For custom models or BYOM workflow, ensure the selected quantization matches the quantization used during model training.

Compilation flags

Flag	Description
max-speed	In pipeline mode, loads weights and input data on PLL clock ÷1 instead of ÷4. ~50% faster, but ~200% energy usage.
riscv	Uses RISC-V processor.
deepsleep	Puts ARM core into deep sleep during inference.
powerdown	Powers down unused MRAM instance.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Flag	Description
kernel	Specifies kernels to be used by the inference runtime.

Kernel options

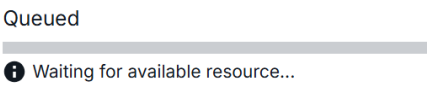
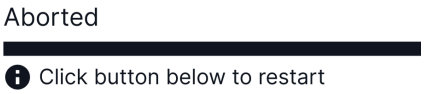
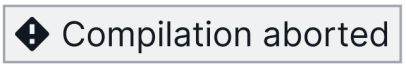
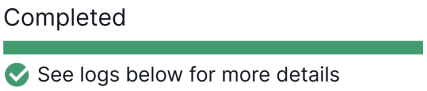
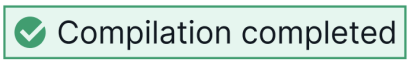
Option	Description
Reference	Uses default reference kernels from TFLite.
CMSIS-NN	Uses optimized kernels for ARM Cortex-M core.
ADI-SHARCFX-NN	Uses optimized kernels for SHARCFX core.

Compilation bar and icon progress indicators

Status	Description
Queued/Compilation Queued	EdgeBench has initiated your compilation request; waiting to begin processing.
Running/Compilation Running	EdgeBench is compiling your model for the selected target device.
Failed /Compilation Failed	The compilation process has failed due to an error. Please see the logs for additional details. Ensure that you fix the error before restarting the process.
Aborted /Compilation Aborted	The compilation process was manually stopped after clicking the **Abort Compilation** button. Click **Start Compilation** to start compiling again.
Completed /Compilation Completed	EdgeBench has finished compiling your model. You can now proceed to the next step.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Visual tag

Status	Progress Bar Indicator	State Indicator
Queued/Compilation Queued		
Running/Compilation Running		
Failed /Compilation Failed		
Aborted /Compilation Aborted		
Completed /Compilation Completed		

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Upload Input Data

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Upload Input Data

AI model inference requires input data to generate predictions or outputs based on learned patterns and context specific to that input. This step allows users to upload input data. Refer to the table below for the supported formats.

Supported formats

Input Data	Supported Formats	Maximum File Size
Image	PNG, JPEG	<10 MB
Audio	MP3, WAV (Keyword Spotting Task Types maximum duration is 1 second; other task types are 5 seconds)	<10 MB
Time-Series	CSV	<10 MB

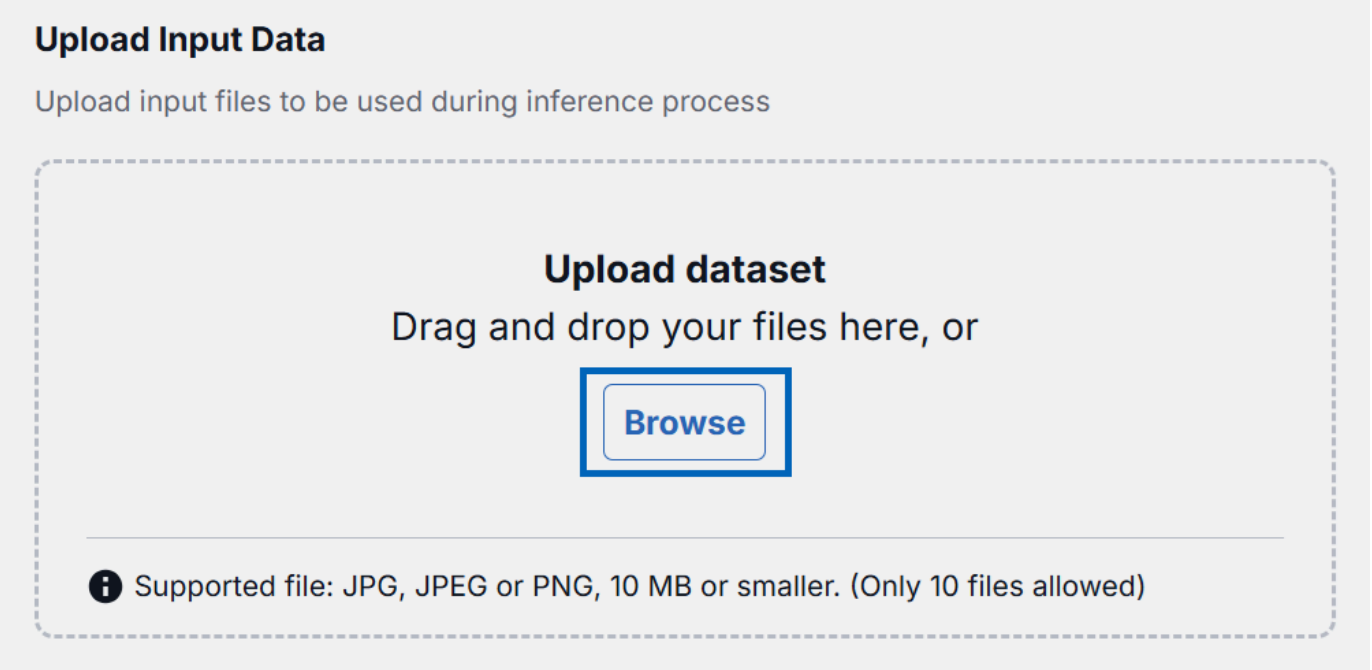
Important

The maximum number of files you can upload is 10.

Uploading files

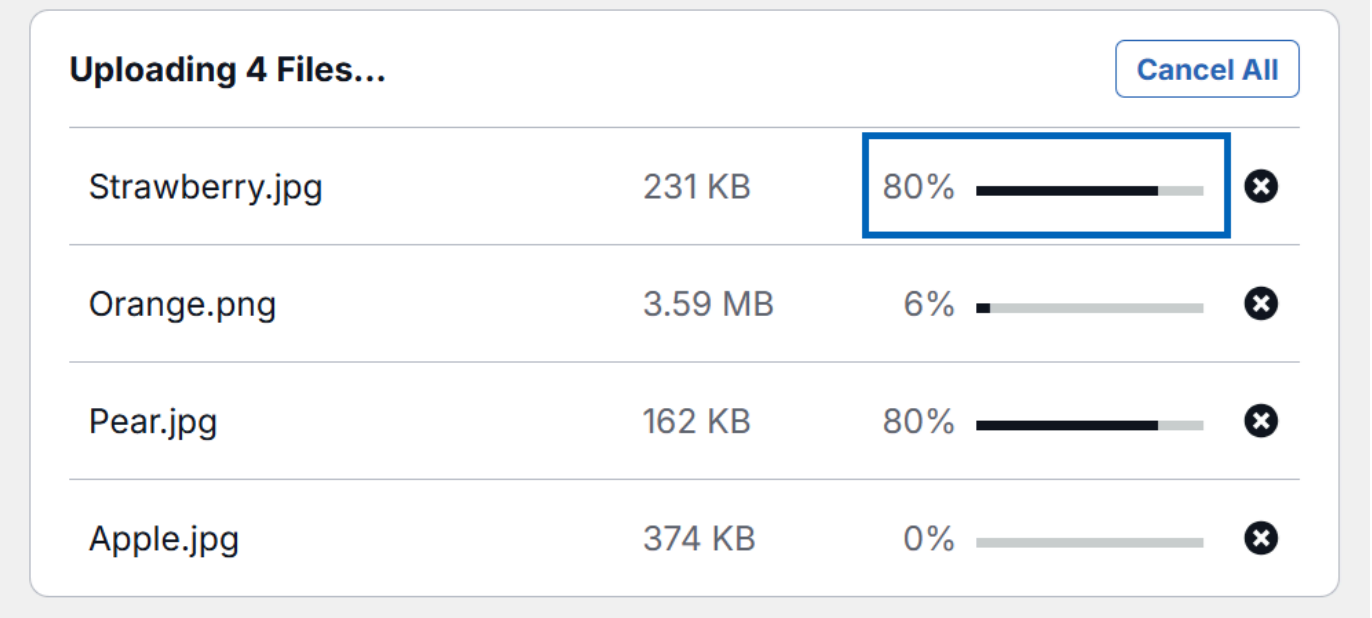
- Drag and drop files into the uploader pane or click **Browse** to manually select files via File Explorer and, after choosing the files, click **Upload**.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



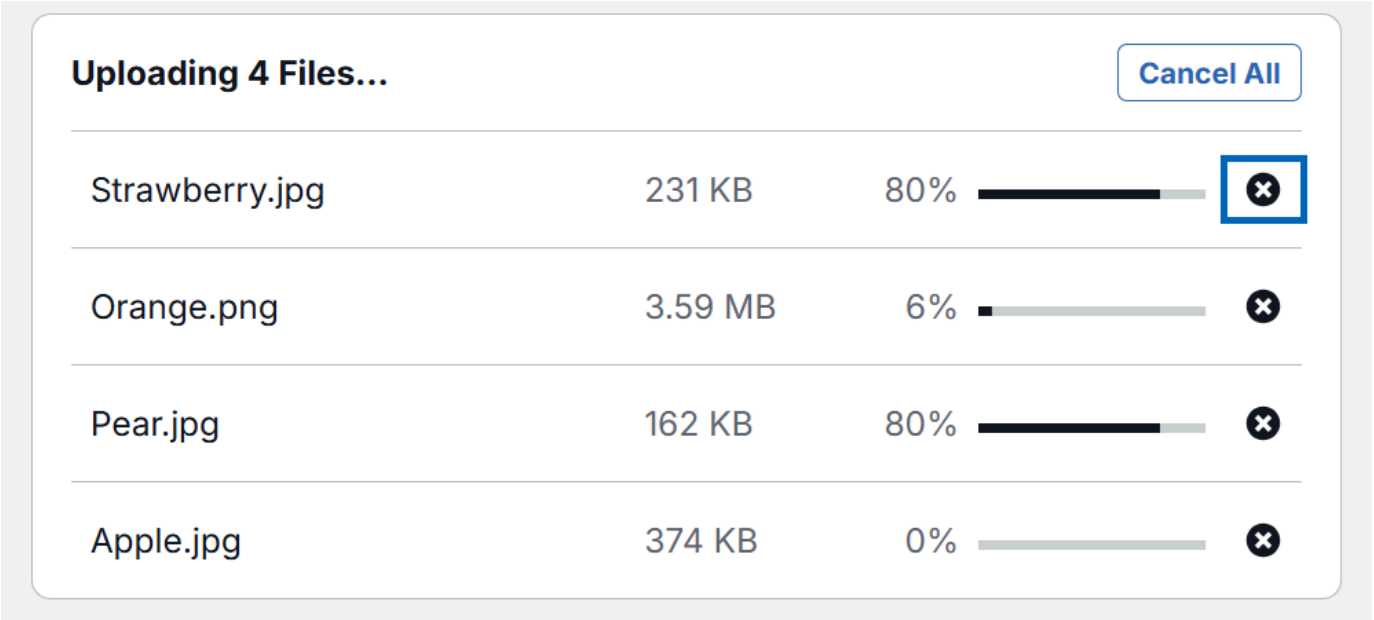
- A progress bar will appear during the upload process. The Uploaded Files pane will appear after a file is successfully uploaded.

> **Note**
>
> You can continue to upload files while there is an ongoing upload as long as the total number of files doesn't exceed the maximum limit of 10.

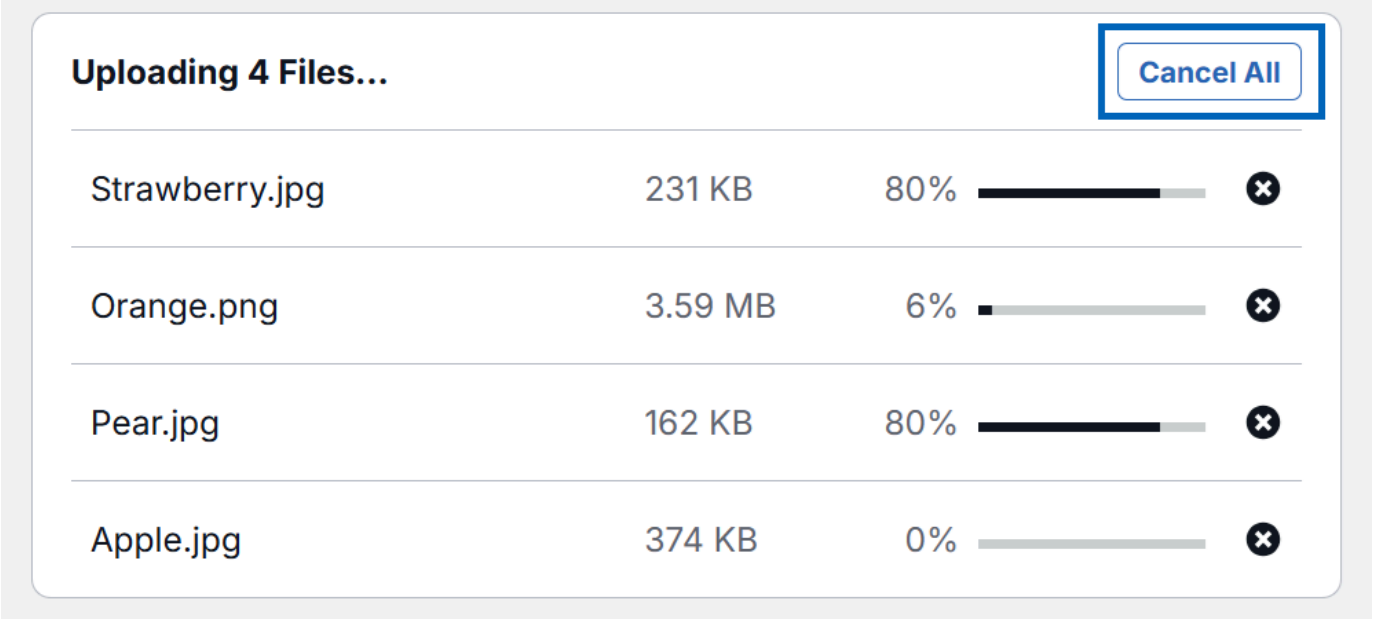


- Cancel individual file uploads by clicking on the **X** icon.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

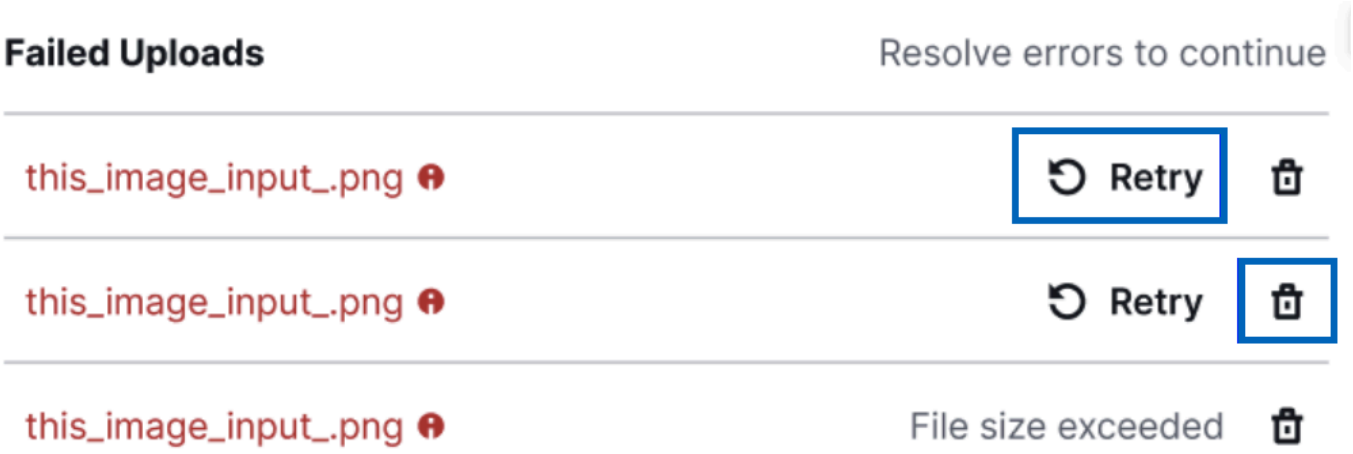


- To cancel all ongoing uploads, click the **Cancel All** button.



- A Failed Uploads pane will appear after an error occurs during the upload process. Click the **Retry** button if you want to try and upload the files again, or you can click the **Trash** icon to remove the failed upload.

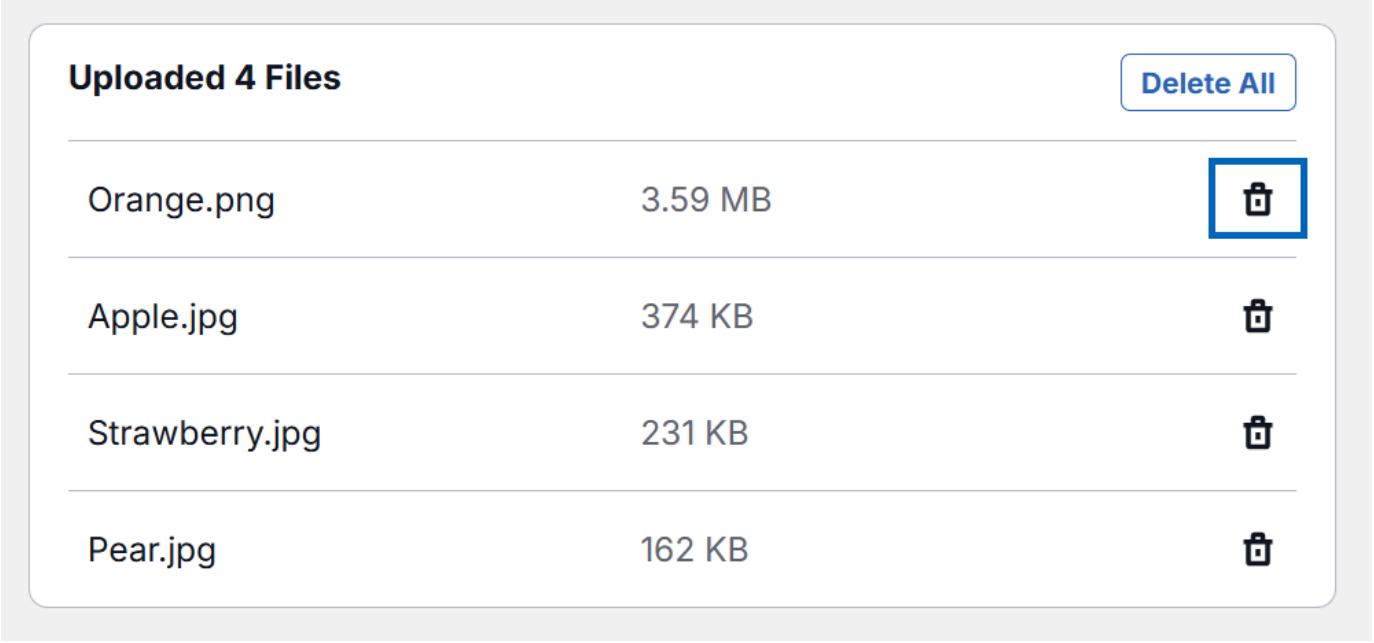
Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Managing uploaded files

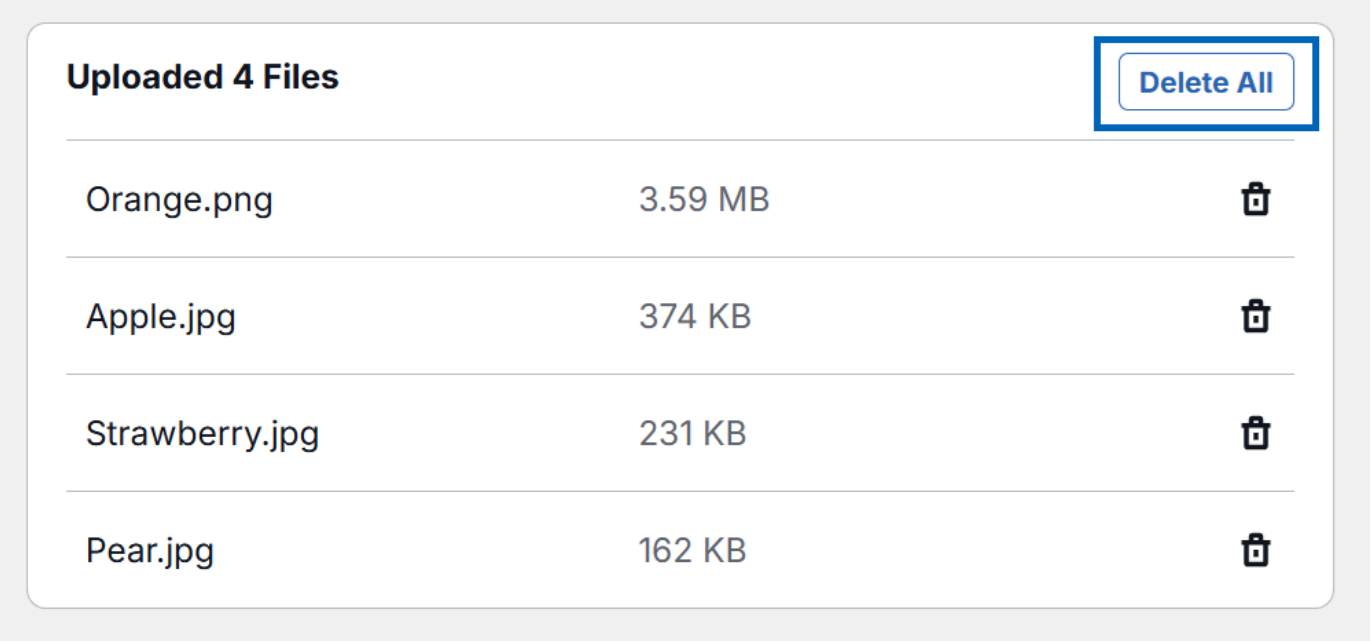
Successfully uploaded files will be listed in the Uploaded Files pane.

- Delete individual files by clicking on the **Trash** icon.



- Remove all files at once by clicking **Delete All**.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



Once all required files are uploaded, click the **Next** button to proceed.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Add a Pre-processing Script

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Pre-Processing Script

This step is intended for adding a pre-processing script designed to transform uploaded files with the model's input shape requirements, ensuring input compatibility. The pre-processing script step is **mandatory** when using the **Bring Your Own Model (BYOM)** feature.

> **Important**
>
> If using a model from ADI's Model Zoo, this step will not appear in the build setup process.

To learn how to create custom data processing scripts, see the Data Processing page.

Script editor environment

The script editor operates within a Jupyter-like Python environment, and these three imports are mandatory for postprocessing script files:

- argparse
- numpy
- pathlib

These are the only supported libraries in the runtime environment:

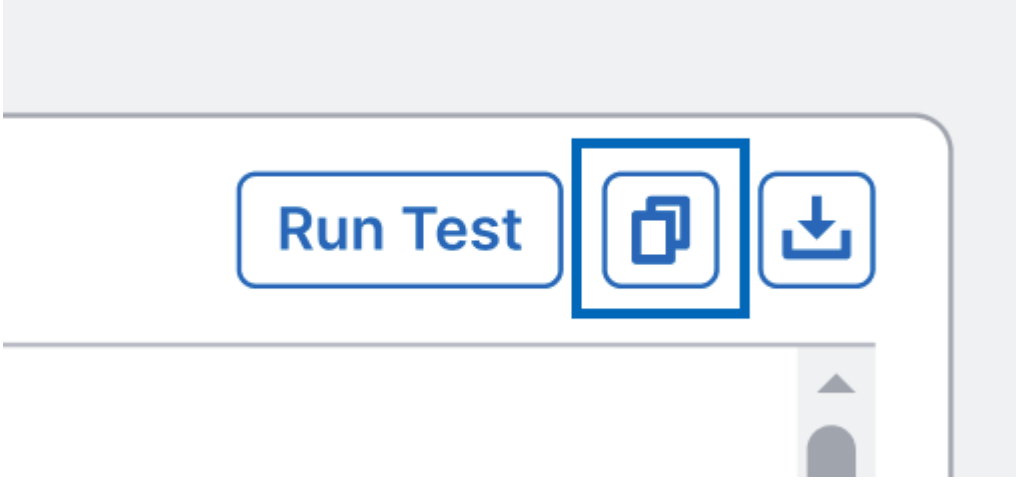
- torch (2.1.2+cpu)
- torchvision (0.16.2+cpu)
- tensorflow-cpu (2.15.0)
- requests (2.32.5)
- pyyaml (6.0.3)
- pandas (2.3.3)
- scipy (1.15.3)
- matplotlib (3.10.8)
- opencv-python (4.11.0.86)

It is recommended to develop your script outside EdgeBench. This environment allows users to test custom post-processing logic directly within the platform.

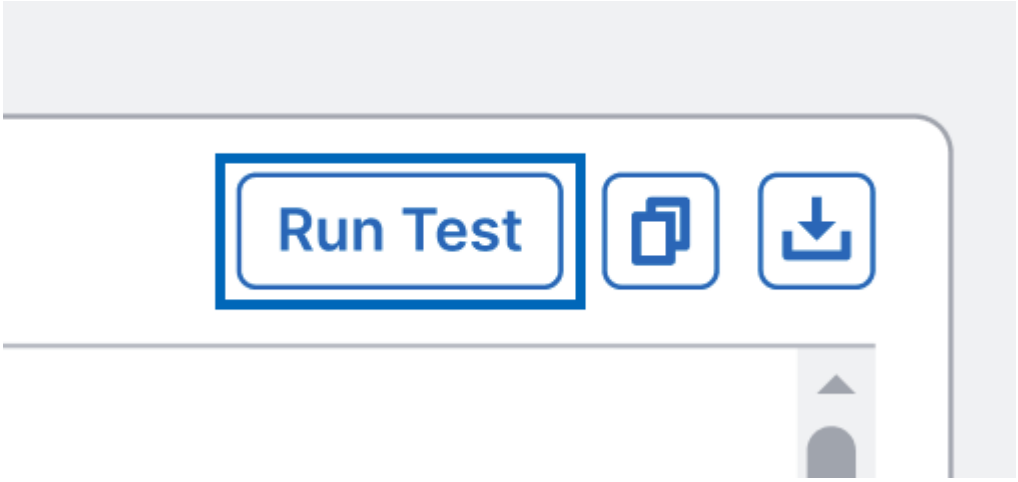
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Adding a pre-processing script

1. Use the provided **Python template** in the editor to transform input data to match your model's expected format. Customize the logic as needed.
2. Click the **Copy Script** icon to save the current script to your dashboard for future reference and easy sharing.



3. Click **Run Test** to validate the script against sample data.



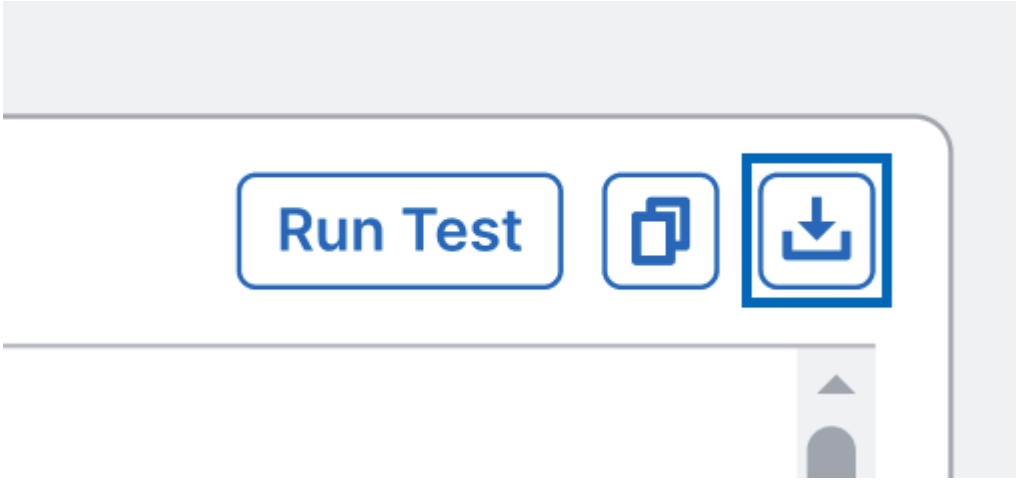
If the test compilation detects any errors or warnings:

- Review the **Logs Pane** for detailed messages.
- Resolve errors or debug warnings before re-running the test.

> **Note**
>
> Users can iteratively update and re-run the script until the pre-processing completes successfully.

4. Once the test compilation passes successfully, click the **Download Script** icon to store your script.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



5. Click **Next** to proceed to the next step.

> **Note**
>
> The **Next** button will remain disabled until the test compilation completes successfully and the script is saved.

Logs pane

The **Logs Pane**, located at the bottom of the interface, displays detailed logs summarizing background activities, errors, and warnings encountered during compilation. This pane can be collapsed or expanded and is a valuable tool for troubleshooting and understanding compilation behavior.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Add a Post-processing Script

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Post-Processing Script

This step is for inputting a post-processing script to ensure that the output from the AI model is correctly interpreted and formatted for analysis and presentation. Note that this step is **required** for the Bring Your Own Model (BYOM) feature.

> **Important**
>
> If using a model from ADI's Model Zoo, this step will not appear in the build setup process.

To learn how to create custom data processing scripts, see the Data Processing page.

Script editor environment

The script editor operates within a Jupyter-like Python environment, these three imports are mandatory for postprocessing script files:

- argparse
- numpy
- pathlib

These are the only supported libraries in the runtime environment:

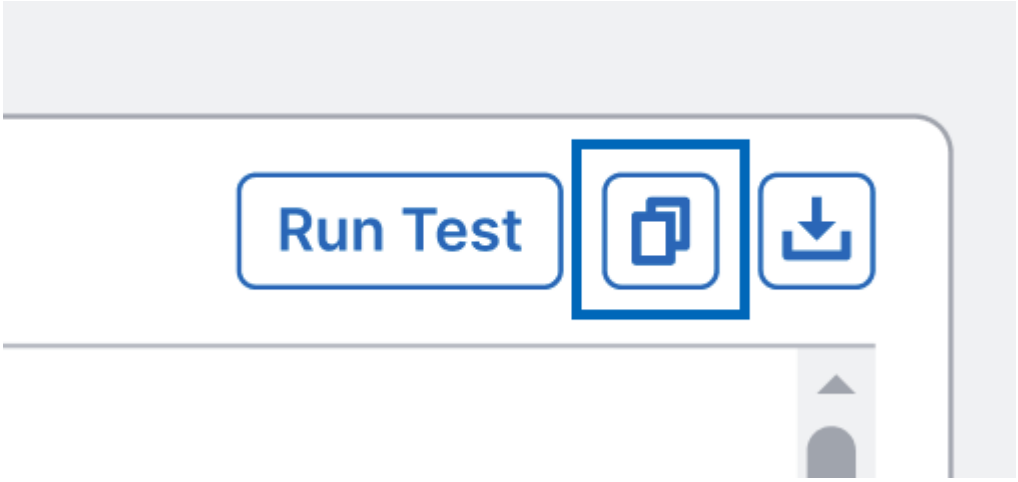
- torch (2.1.2+cpu)
- torchvision (0.16.2+cpu)
- tensorflow-cpu (2.15.0)
- requests (2.32.5)
- pyyaml (6.0.3)
- pandas (2.3.3)
- scipy (1.15.3)
- matplotlib (3.10.8)
- opencv-python (4.11.0.86)

It is recommended to develop your script outside EdgeBench. This environment allows users to test custom post-processing logic directly within the platform.

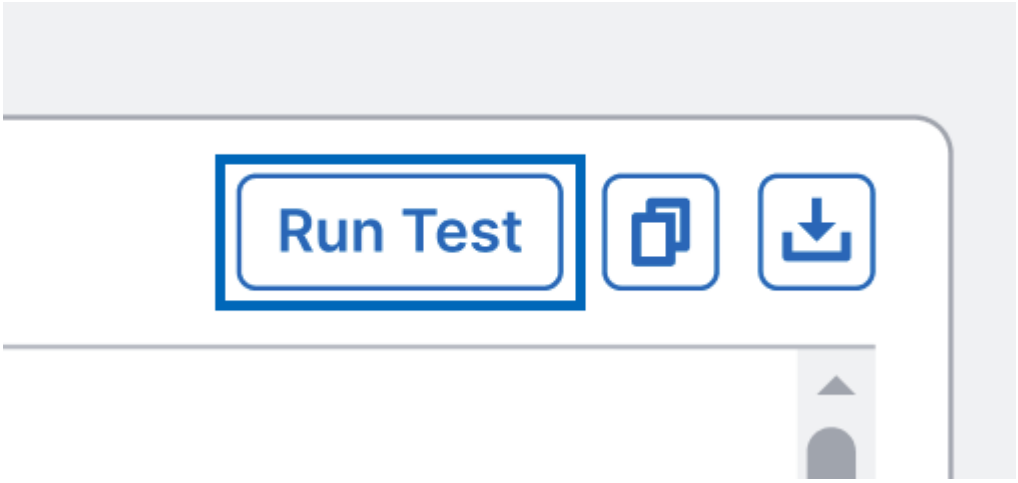
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Adding a post-processing script

1. Use the provided **Python template** in the editor to interpret and format the model's output. Modify the logic to suit your use case.
2. You can click the **Copy Script** icon to save the script to your dashboard.



3. Click **Run Test** to validate the script against sample data.

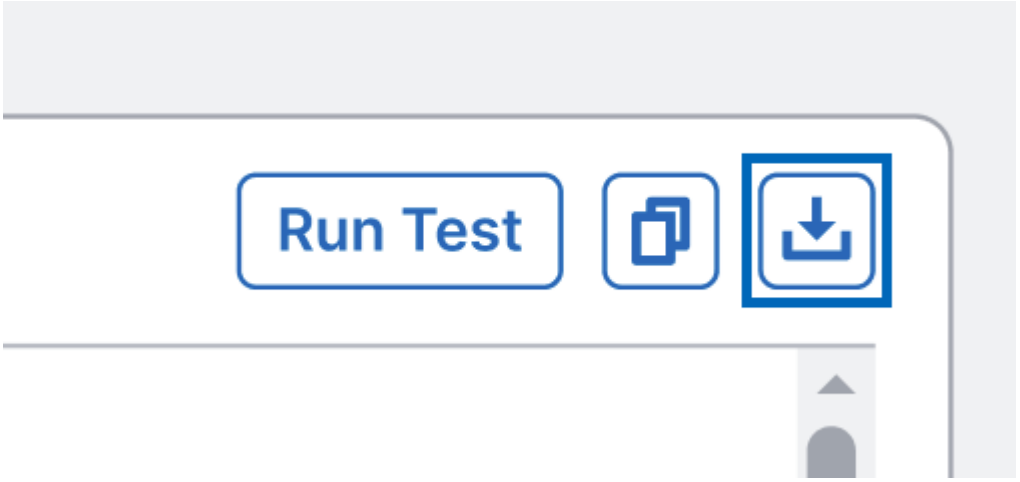


4. If the test compilation finds any errors/warnings, check the **Logs** pane and resolve the errors or debug the warnings before running the test again.

> **Note**
>
> Users can iteratively update and re-run the script until the pre-processing completes successfully.

5. Once the test compilation passes successfully, click the **Download Script** icon to save your script.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



6. Click **Next** to proceed to the next step.

> **Note**
>
> The **Next** button will remain disabled until the test compilation completes successfully and the script is saved.

Logs pane

The **Logs Pane**, located at the bottom of the interface, displays detailed logs summarizing background activities, errors, and warnings encountered during compilation. This pane can be collapsed or expanded and is a valuable tool for troubleshooting and understanding compilation behavior.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Deploy Inference

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

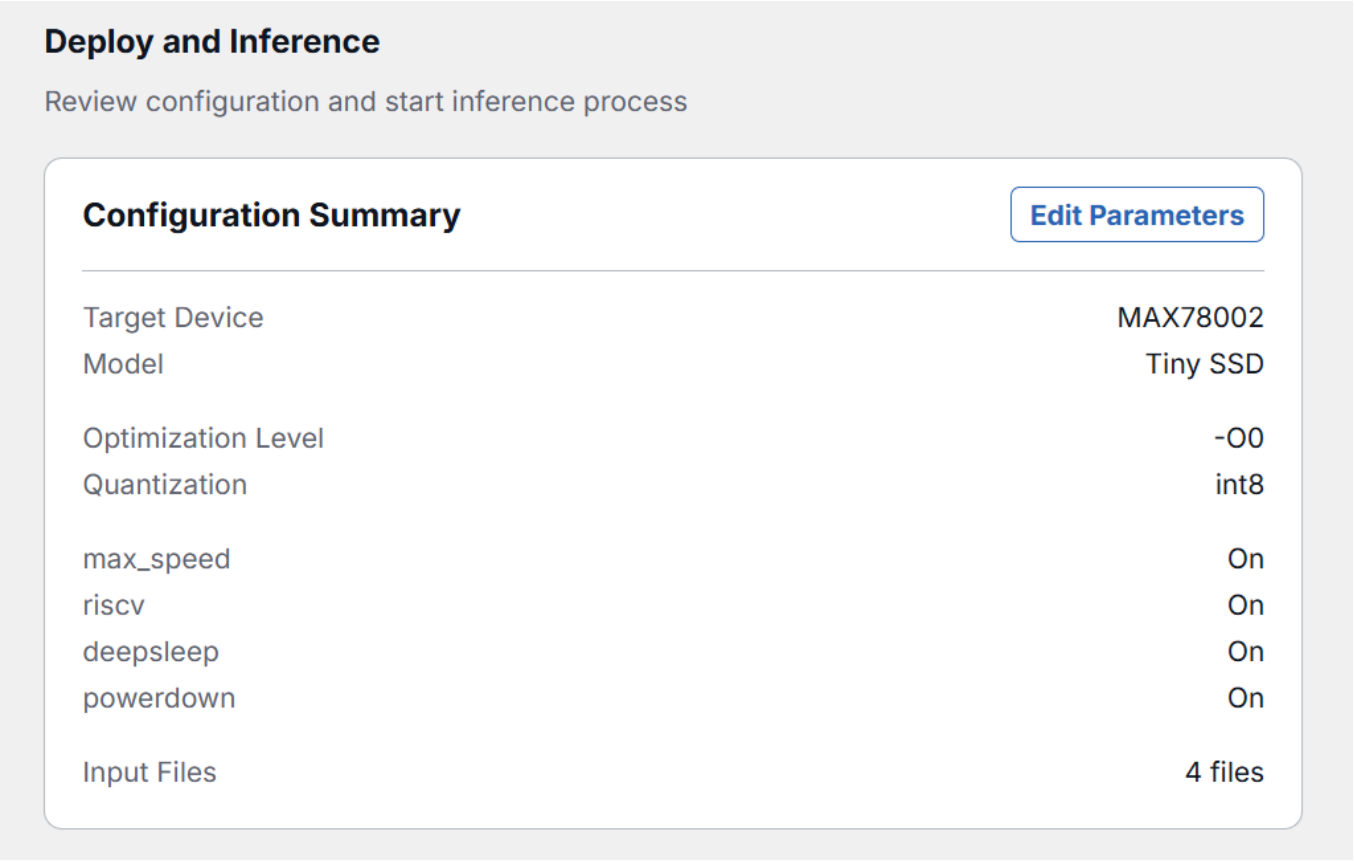
Deploy and Inference

The Deploy and Inference step allows users to run their build and perform inference on the selected remote device.

Note that this step is only unlocked when Step 3: Upload Input Data, Step 4: Pre-Processing Script (only if BYOM), and Step 5: Post-Processing Script (only if BYOM) are completed.

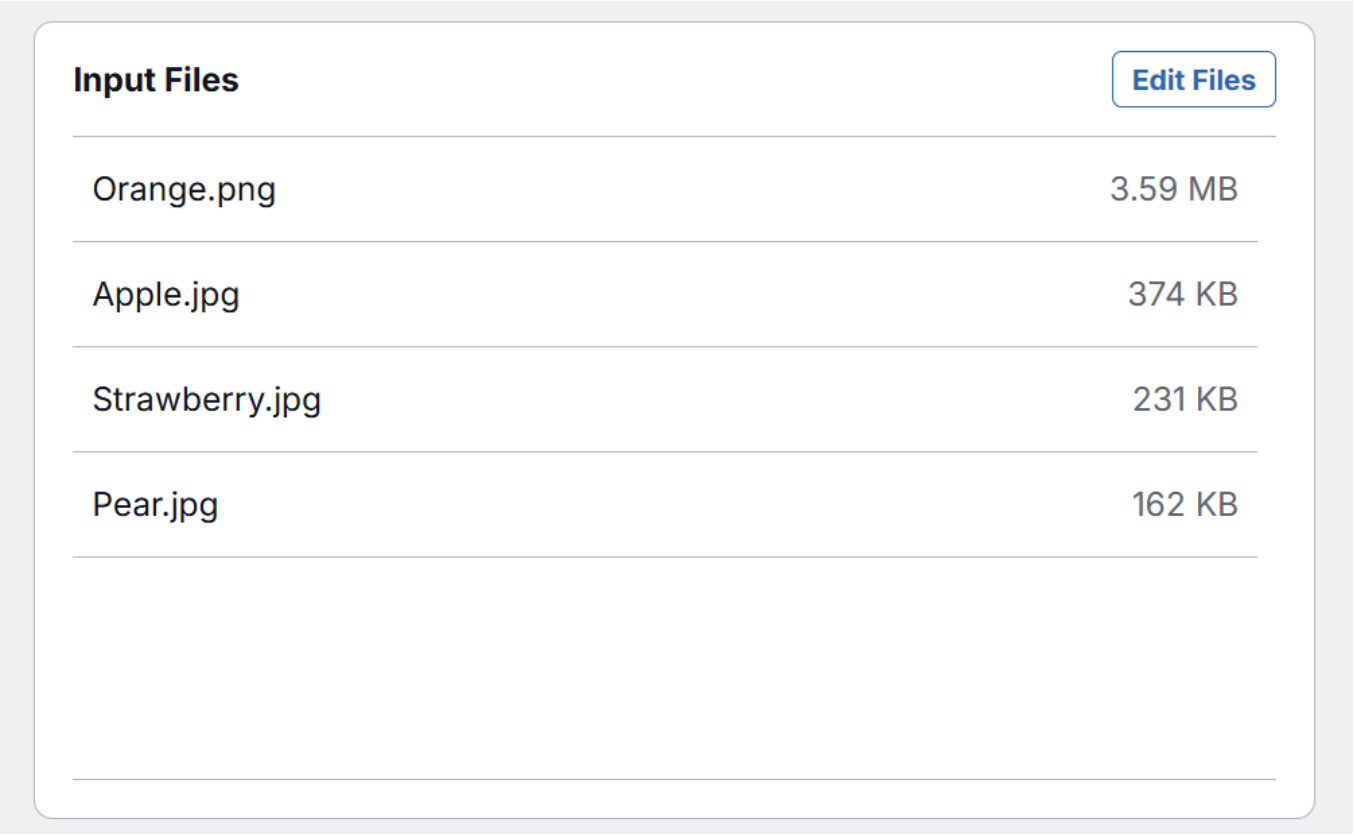
Starting the inference process

1. Review your customized hardware-specific settings in the configuration summary pane. A sample is shown below, but these parameters should match the selection from the **Configure & Compile** step.

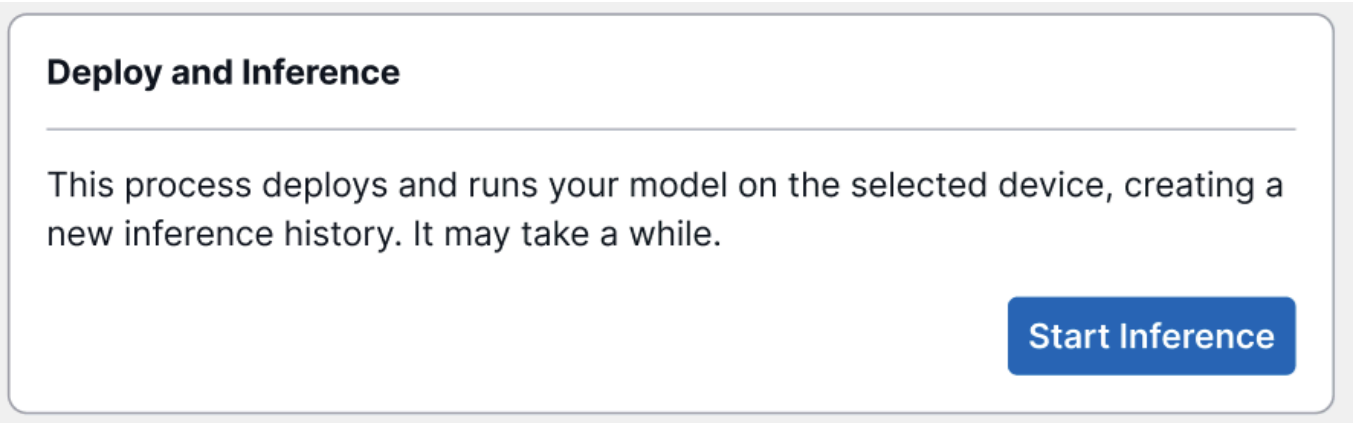


2. (Optional) If you want to make changes, click **Edit Parameters**. This action will redirect the user to **Step 2 – Configure and Compile**.
3. Review all of your uploaded files that will be included in this particular inference session.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



4. (Optional) If you want to make any edits, click **Edit Files**. This action will redirect you to **Step 3 - Upload Input Data**.
5. After you have reviewed the configuration settings and the input files, click **Start Inference**.



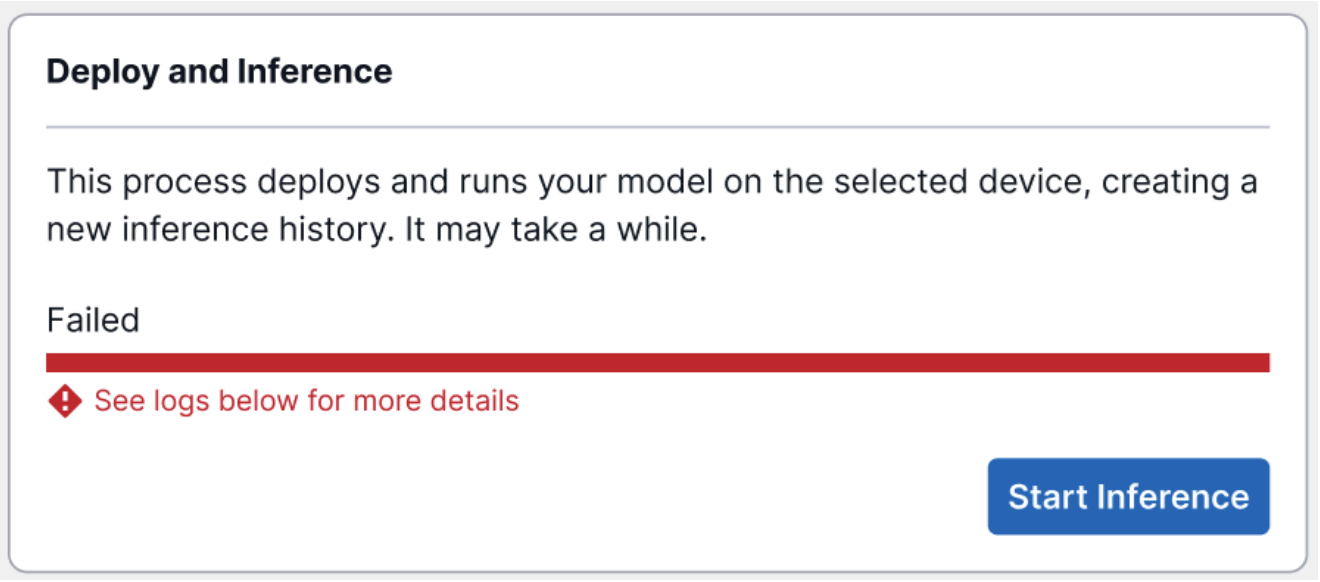
Note

Your request will be added to the queue. Queue times may vary depending on hardware availability. The inference status will change to **Running** once the request is processed by the remote board farm.

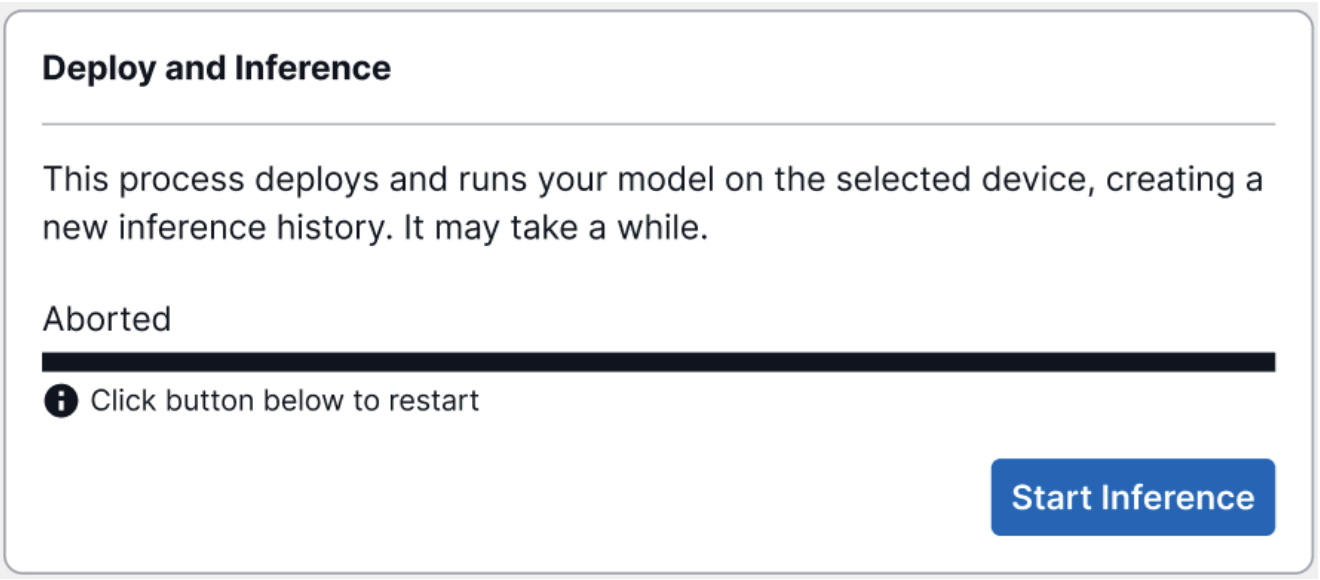
Take note of the Inference Statuses:

- **Failed:** Errors must be resolved before re-sending the request. All errors are shown in the logs pane.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



- **Aborted:** Your request will be removed from the queue and/or any ongoing progress on the remote device will be lost. The user must click **Start Inference** to make a new request.



- **Completed:** A new entry is added to the **Inference History**.

6. Once inference is complete, click **Next** to view the results.

> **Note**
>
> The Next button remains disabled until the inference process is successfully completed.

Logs pane

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Detailed logs are displayed in the bottom pane, providing a summary of background activities and any errors encountered. These logs can be collapsed or expanded and are useful for troubleshooting and understanding what went wrong during compilation.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Check Inference Results

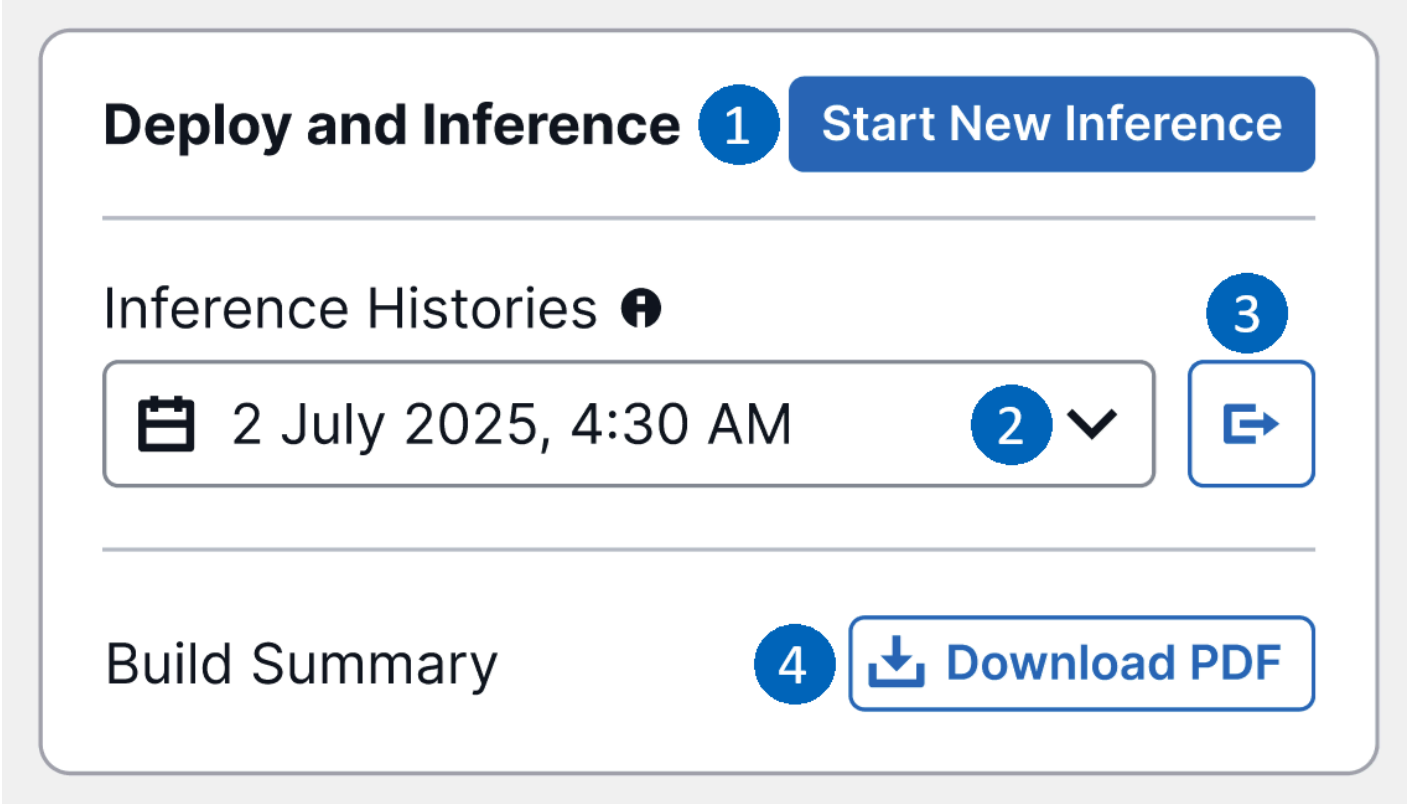
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Inference Results

Once inference is complete, results are displayed for each input file to help users analyze model accuracy and performance.

Deploy and inference pane

This pane lists all past inference sessions related to this project build, including the date and time of creation. The most recent session is automatically shown, and you won't see any inference sessions from another project.



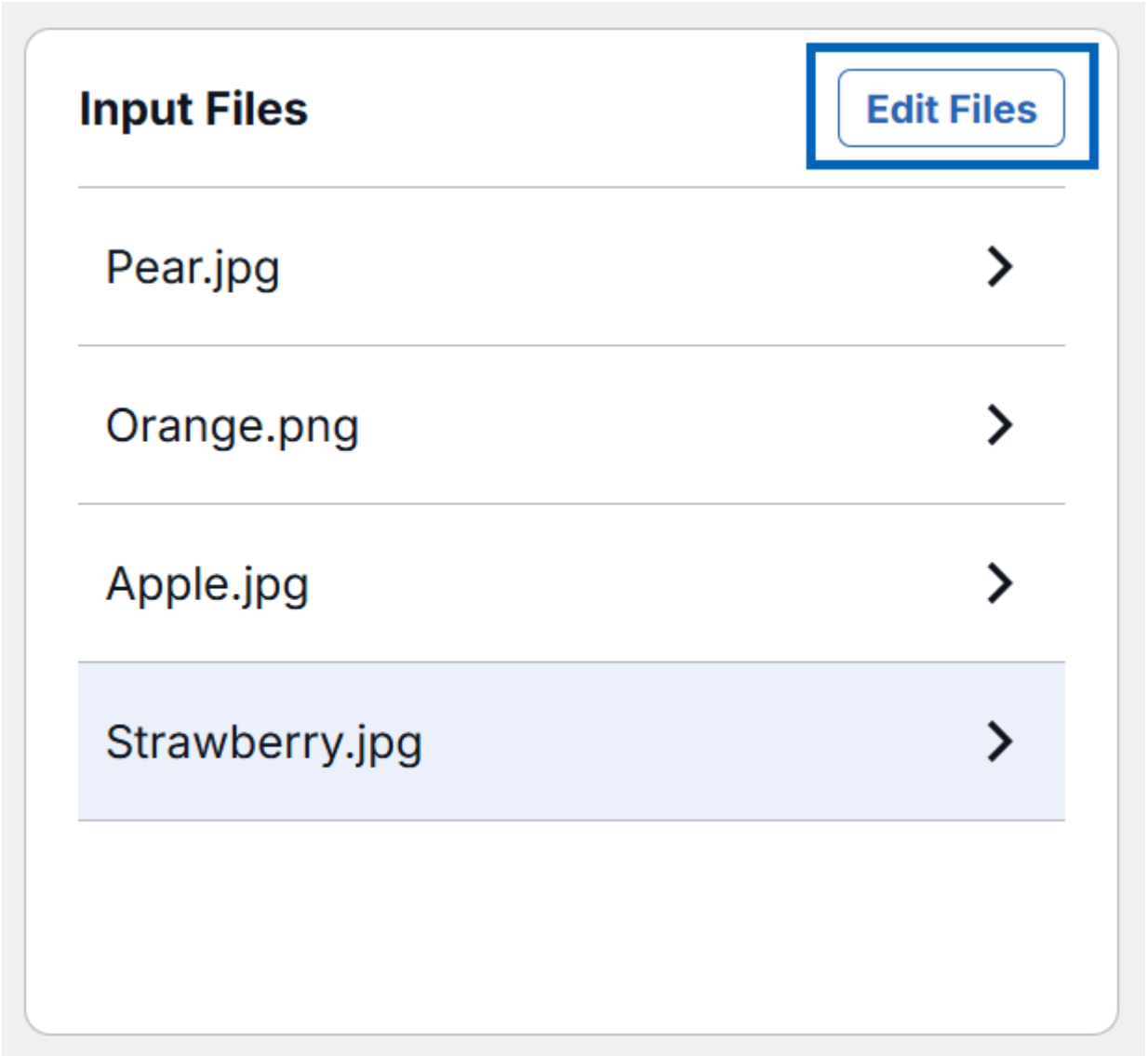
1. To initiate a new inference, click **Start New Inference**. This action redirects the user to **Step 1 – Device Selection**.
2. Use the **Dropdown Menu** to view and select past inference sessions and click the Proceed icon to load the specific inference session.
3. Click the **Export Inference** button to download the file containing information such as output data, source data, and real output.
4. Click the **Download PDF** button to get a copy of the Build Summary.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

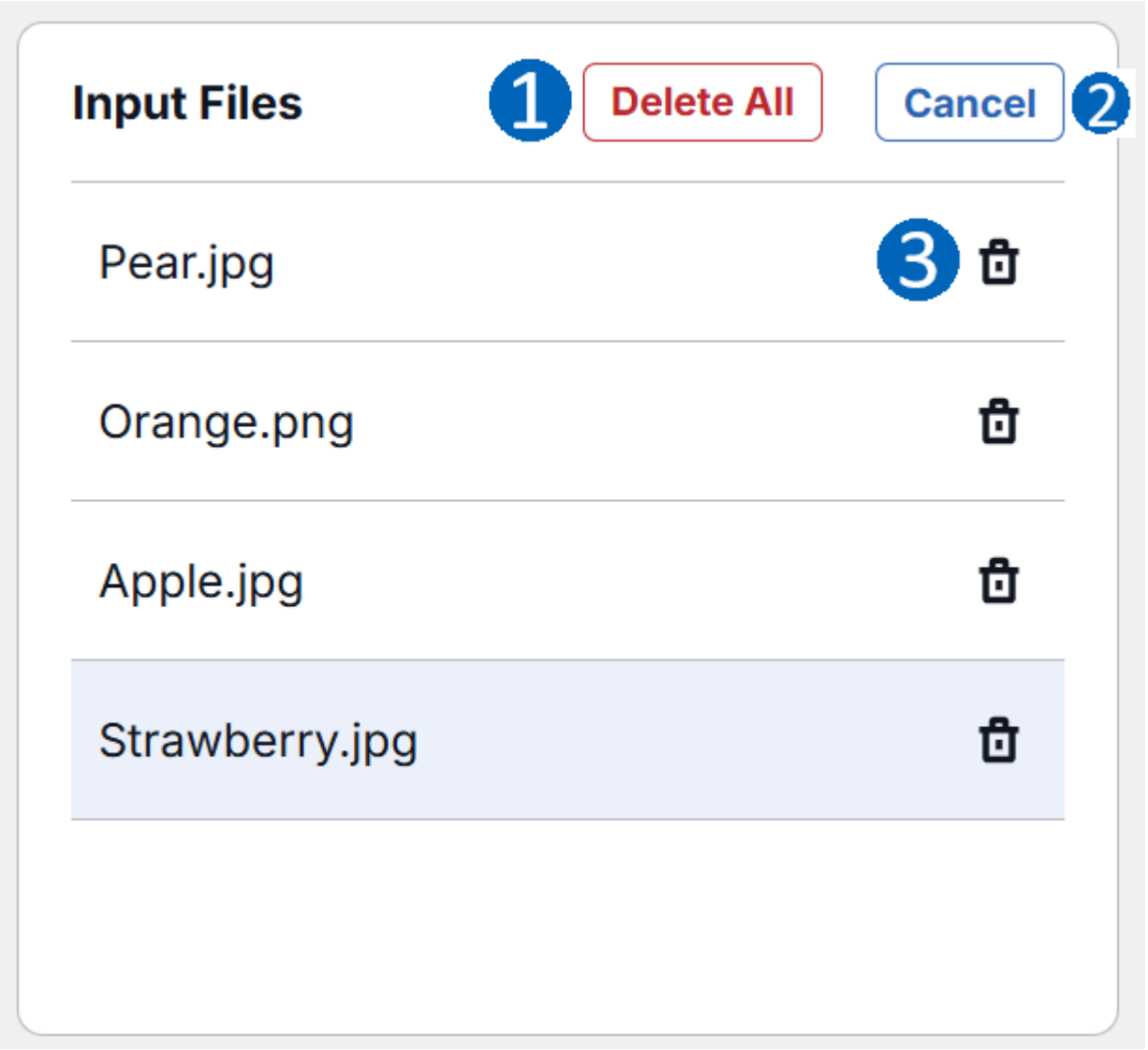
Input files pane

This pane shows the files used as input for this inference session. By clicking each individual input file, users can see the corresponding inference result or output.

To manage the files, click **Edit Files**.



Copyright © 2026 Analog Devices Inc. All Rights Reserved.



1. Remove all files at once by clicking **Delete All**.
2. Click **Cancel** to stop editing the files.
3. Delete individual files by clicking on the **Trash** icon.

Inference results pane

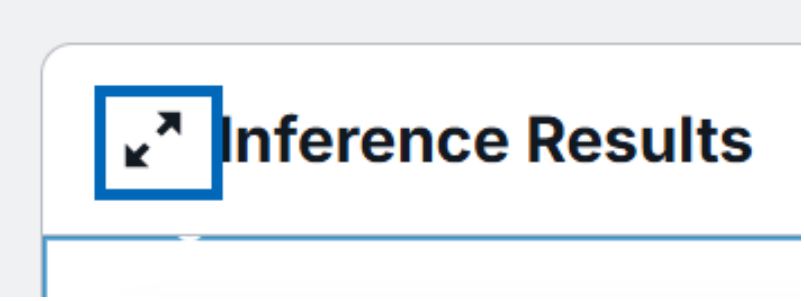
The interface adapts based on the type of AI model used. Supported result types include:

- Vision
- Audio
- Time-Series

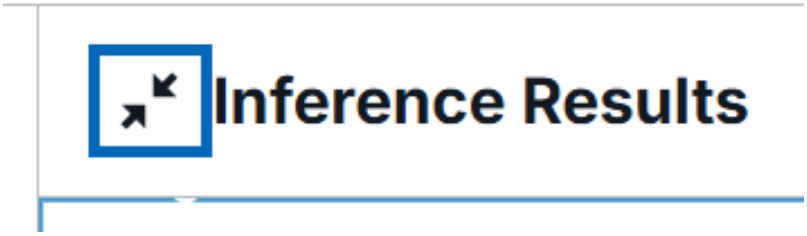
Users can resize the pane using:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Expand** icon – enlarges the pane and reveals the **Export Results** button. This action also hides the build setup side pane.



- **Collapse icon** – reduces the pane size.



All result views also display Device Performance Metrics, as shown in the table below.

Flag	Description	Unit
Latency	The length of time it took for the device to process the data.	µs
Power	The amount of energy the device used to perform the inference.	mW
Memory	The size of the model used to store the data of this specific inference result.	kb

Model-specific result views

Vision inference results

Vision inferences are categorized into:

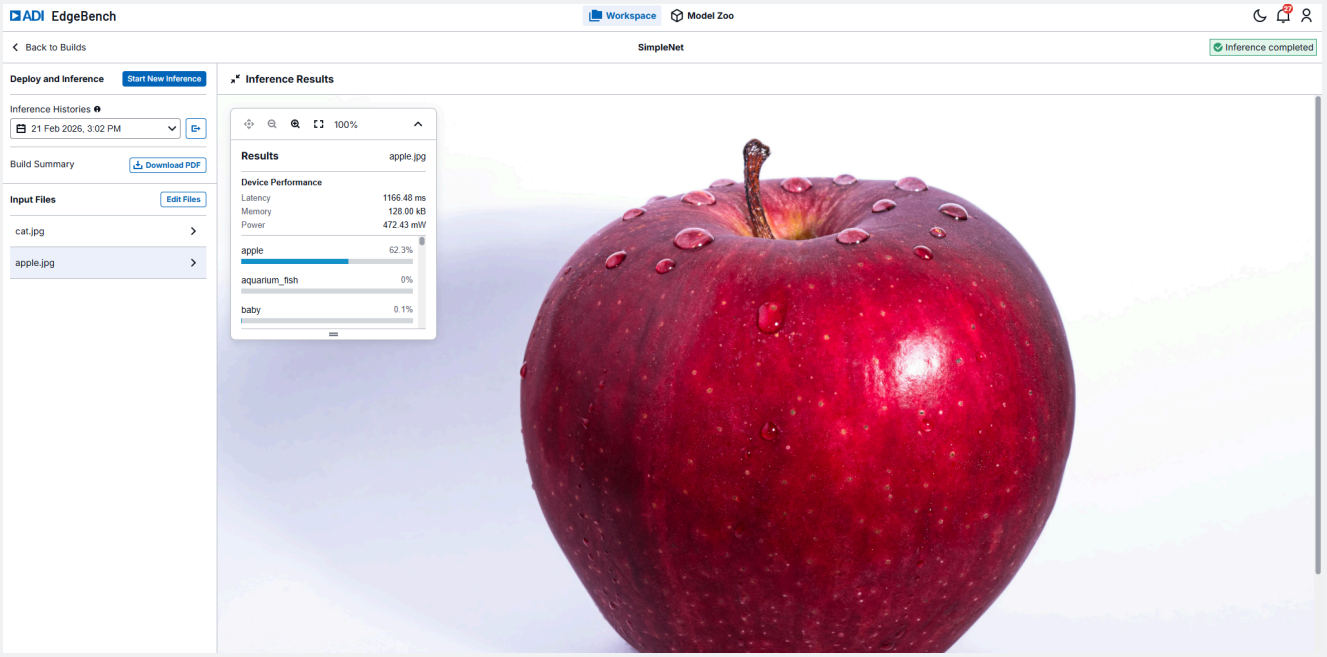
- **Classification (Image Classification and Image Segmentation)**

  Identifies the primary object(s) in an image and returns a label with a confidence score.

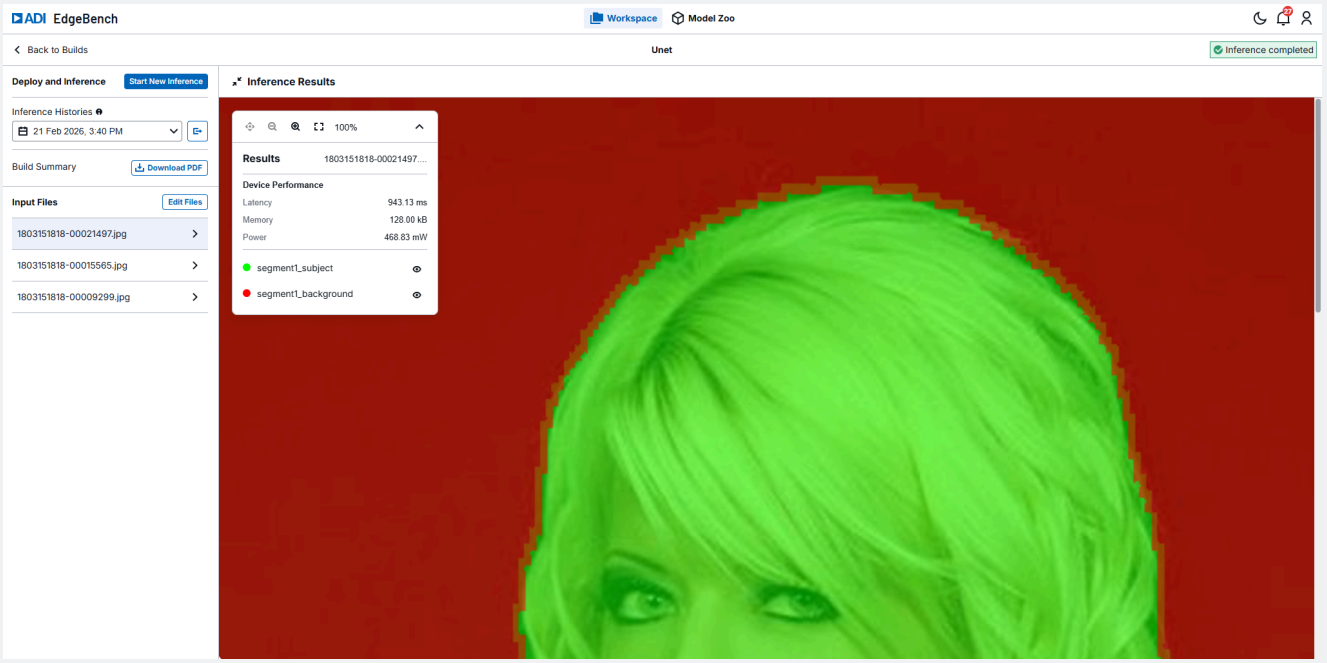
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Example

Example 1: Image Classification



Example 2: Image Segmentation



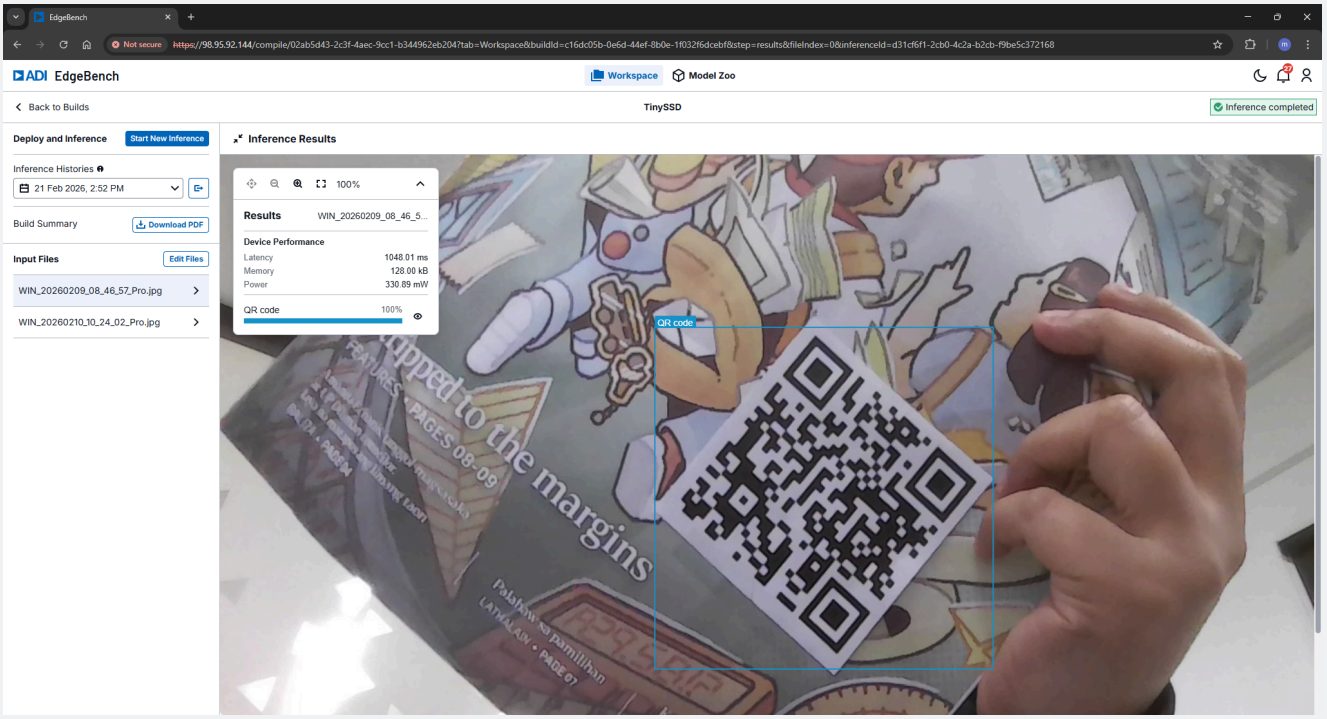
- **Detection (Visual Wake Work, Face ID, and Object Detection)**

  Combines classification with object localization using bounding boxes.

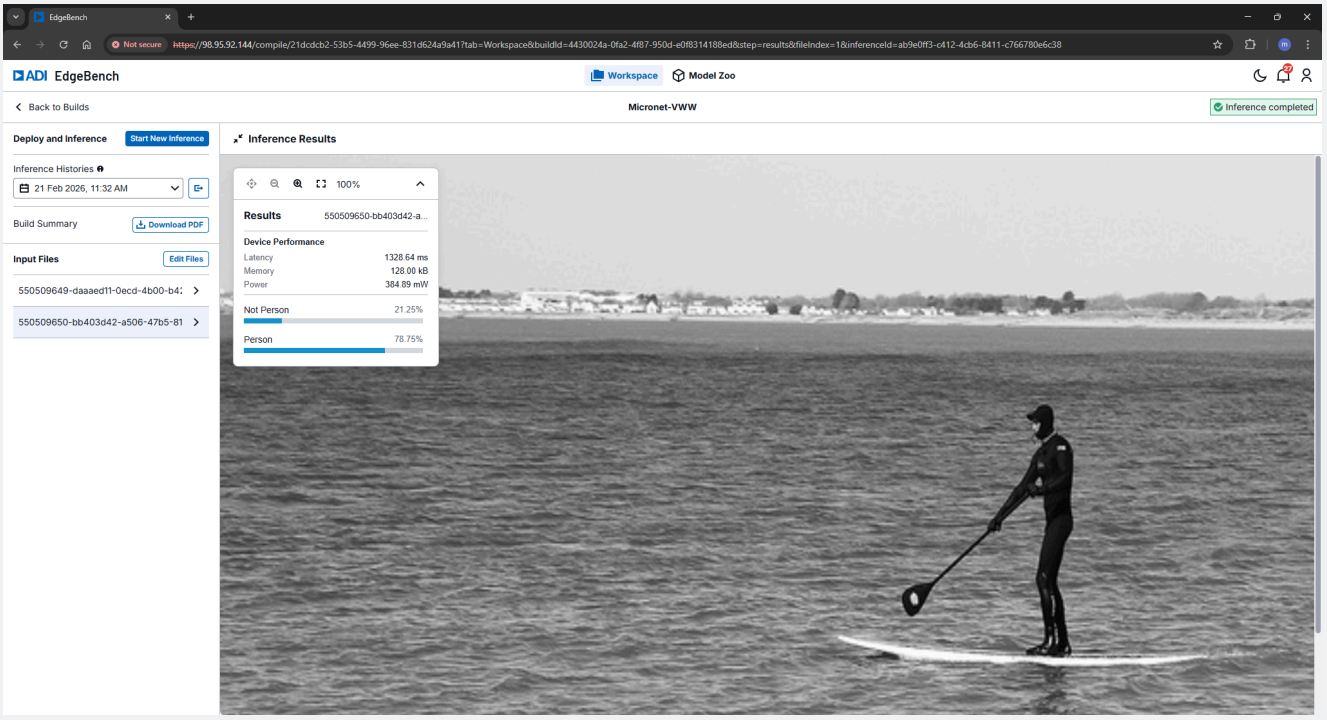
Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Example

Example 1: Object Detection



Example 2: Visual Wake Word



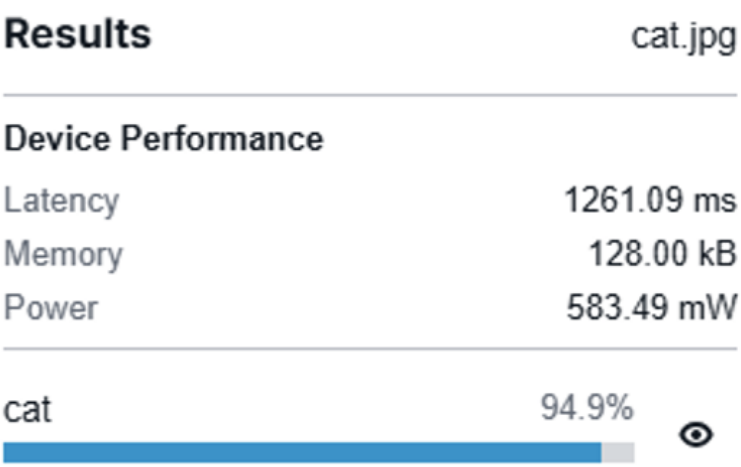
Vision Viewer Controls (from left to right):

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



- **Drag the Results Screen** – Turn on the ability to click and drag the image area to move around when zoomed in
- **Zoom Out** – Reduces the image size to see the full image or larger context.
- **Zoom In** – Enlarges the image to view finer details more clearly.
- **Maximize Window** – Expands the image results panel to full screen for better visibility.
- **Zoom Percentage** – Displays the current zoom level of the image (e.g., 100%, 200%).

The Results Section shows model predictions, confidence scores, and device performance metrics.



Audio inference results

Audio inferences are categorized into two parts:

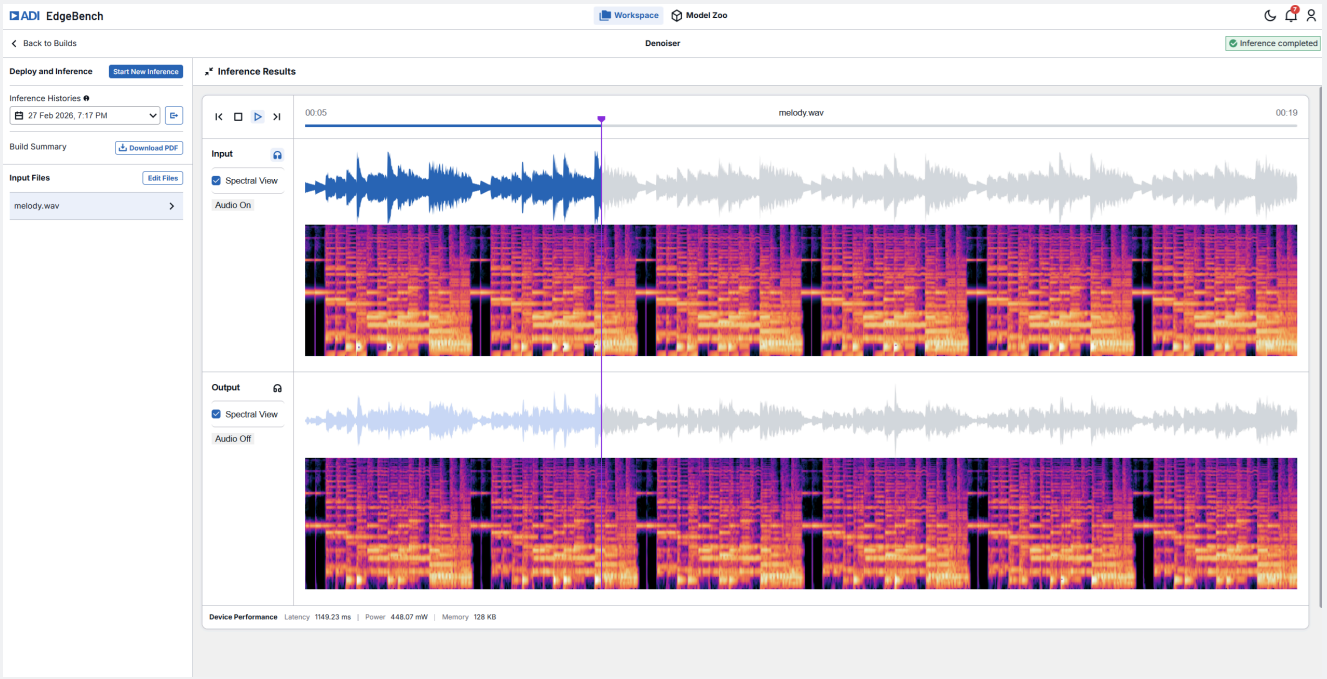
- **Denoiser (Before-After)**

  Compares original and processed audio using two synced players: Input and Output.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Example

Denoiser

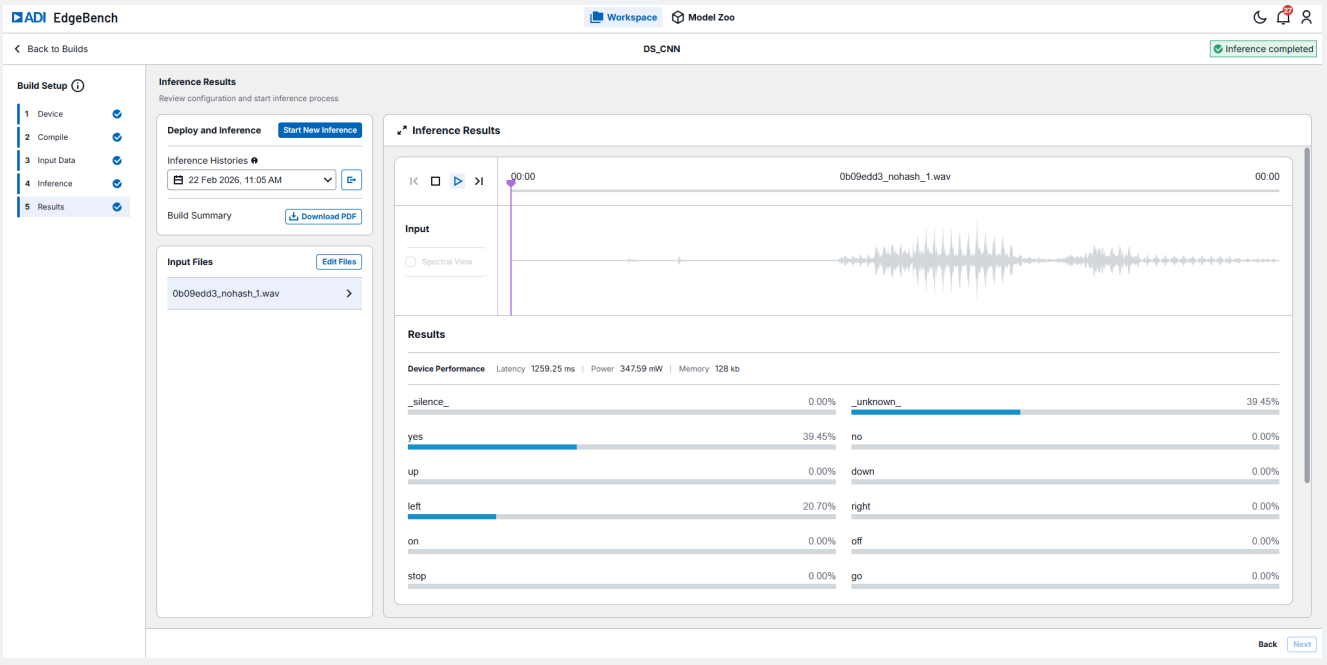


- **Detection (Keyword Spotting and Genre Identification)**

  Identifies spoken keywords or music genre with confidence scores using a single audio player.

Example

Keyword Spotting

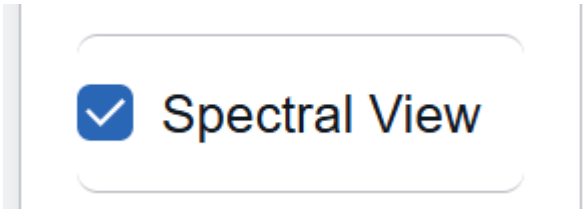


Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Audio Viewer Features:

- **Spectral View**

  Displays the spectogram which represents the frequency content as it varies with time.



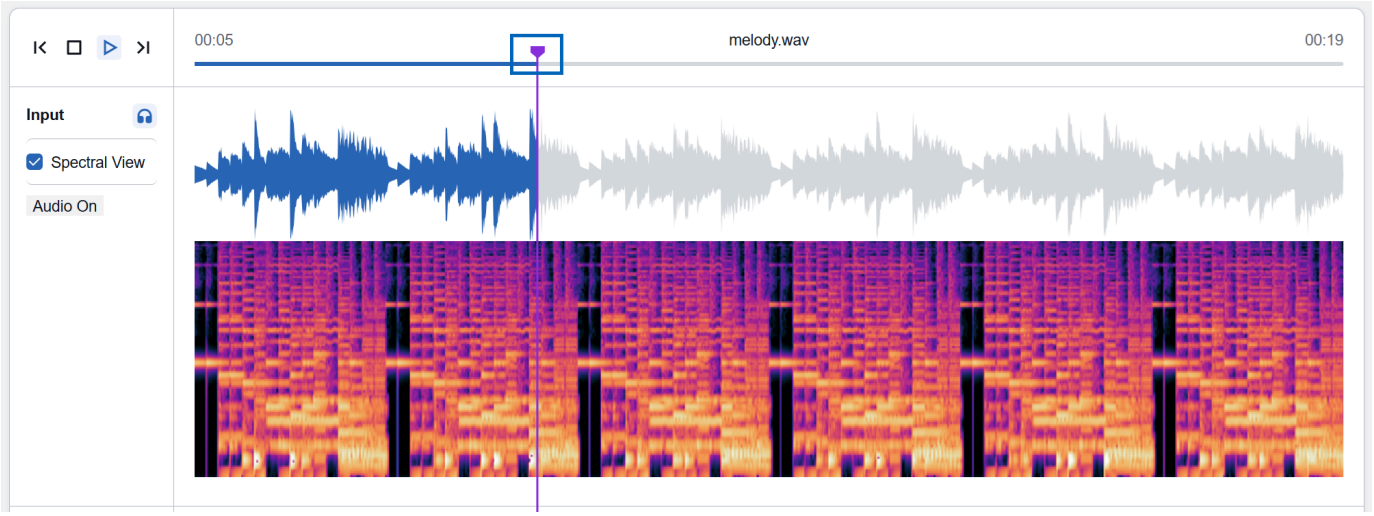
- **Headphone Icon**

  Indicates active playback.



- **Drag Bar (Seek Bar / Timeline Cursor)**

  Allows the user to click and drag the vertical line on the waveform to move to a specific timestamp in the audio.



Audio Player Controls (from left to right):



Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Backward (Rewind)** – Moves the playback position backward.
- **Stop** – Stops the audio and resets playback to the beginning.
- **Play** – Starts playing the selected audio file from the current position.
- **Pause** – Temporarily stops the audio playback while keeping the current position.
- **Forward (Skip Ahead)** – Moves the playback position forward.

Time-series inference results

Time Series inferences are separated into two categories:

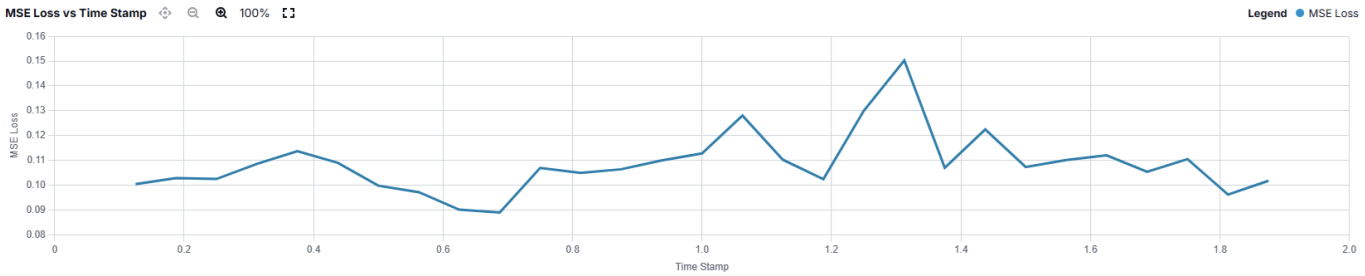
- **Point Anomaly Detection** - where the anomalies in the data are only marked as various red dots in the graph.
- **Region Anomaly Detection** – where an entire section of the graph is marked red.

The Time Series inference results will show two graphs:

- **Input vs. Timestamp** – Input Represents the raw signal or data values fed into the system or model. Timestamp represents the time at which each input value was recorded. Together, they help visualize how input data changes over time.



- **Mean Squared Error (MSE) Loss vs. Timestamp** – represents the difference between predicted and actual values. When plotted against Timestamp, it shows how model error changes over time.



MSE Loss Threshold

An MSE loss threshold is a cutoff value that defines the maximum acceptable prediction error for the model.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

The default value is 0.5 and you can set the value in the Threshold field.



Note

The threshold value for each input file has to be applied independently, and refreshing the page will set all the values back to default.

Important

When exporting the CSV file, the default 0.5 threshold value will be used. Users can only edit the data after it has been exported.

Legend – X Channel, Y Channel, Z Channel Checkboxes



These checkboxes allow users to show or hide specific signal channels.

- X Channel – Displays X-axis signal data.
- Y Channel – Displays Y-axis signal data.
- Z Channel – Displays Z-axis signal data.

Enable or disable a channel by clicking on its respective box to easily differentiate between the plots/lines.

Time-Series Graph Controls (from left to right):

Copyright © 2026 Analog Devices Inc. All Rights Reserved.



- **Pan** – Turn on the ability that allows users to move the graph left, right, up, or down without changing the zoom level.
- **Zoom Out** – Reduces the image size to see the full image or larger context.
- **Zoom In** – Enlarges the image to view finer details more clearly.
- **Maximize Window** – Expands the image results panel to full screen for better visibility.
- **Zoom Percentage** – Displays the current zoom level of the image (e.g., 100%, 200%).

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Frequently Asked Questions

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Frequently Asked Questions

This article guides you through the Workspace, helps you understand platform features, and answers common issues you may encounter with storage, model uploads, data formats, devices, compilation, deployment, inference, and file management.

Use this FAQ as a quick reference to troubleshoot problems and navigate the system more efficiently.

Workspace limitations

Q: Is there a storage limit per user?
A: Yes. Each user has a **10 GB** storage limit. Once you exceed the limit, you will no longer be able to create new projects or upload additional models.

Q: Where can I check my total storage usage?
A: The Workspace displays your current storage usage in the top-left corner.

Q: What does the "Free up storage space" message mean?
A: This message appears when you have reached your storage limit.
To continue working, delete unused items such as:

1. Old or unused projects
2. Previous builds

Supported model formats

Q: I am trying to upload a model file using the Upload Model option, but it is not uploading. What are the supported model formats?

A: The platform supports the following frameworks:

PyTorch

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Framework	Case	Required Files	Remarks
PyTorch	Case 1: Single Model File	- `PT` or `*PTH` (complete model architecture + trained weights) - `input_shape.txt`	`input_shape.txt` file must only contain a valid input shape in one of the accepted formats below: Tuple: `(1, 3, 224, 224)` Array: `[1, 3, 224, 224]` CSV: `1, 3, 224, 224`
PyTorch	Case 2: Model Definition + Weights	- `MODEL.PY` (full model definition) - `*PT` or `*PTH` (weights) - `input_shape.txt`	- Must include `model.py` containing the full architecture definition - The **class name** in `MODEL.PY` must match the trained model - `input_shape.txt` uses the same accepted formats: Tuple: `(1, 3, 224, 224)` Array: `[1, 3, 224, 224]` CSV: `1, 3, 224, 224`

```
MODEL.PY Requirements:

  * Must contain the complete architecture class definition.
  * The architecture class name must match the one used during training.
  * No syntax errors—file must load properly.
```

Important

- If you upload weights only (PT/PTH), you must also include model.py with the complete architecture class definition.
- The architecture class name in model.py must match the name used during training.
- Ensure that:
  - MODEL.PY has no syntax errors.
  - You can load the model without runtime errors in PyTorch.
  - The input shape must match the model's expected dimensions.

TensorFlow

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Framework	Case	Required Files	Remarks
TensorFlow	Case 1: Model in H5 format	- `H5`	Complete model (architecture + weights).
TensorFlow	Case 2: Model in PB format	- `PB`	Complete model with architecture. Must represent a **frozen graph**.
TensorFlow	Case 3: Model in `SAVED_MODEL` directory format	- `SAVED_MODEL/` directory	Directory must contain the **complete exported model** (do not modify or delete internal files).

Important

- Ensure the model is properly exported from TensorFlow.
- Do not upload checkpoint-only or partial models.
- The model must be loadable using standard TensorFlow methods:
- H5 for full models,
- PB for frozen graphs,
- saved_model/ for full SavedModel exports (keep internal structure intact)

ONNX

Framework	Case	Required Files	Remarks
ONNX	Model in ONNX format	ONNX	Must contain the complete model graph and all trained parameters.

Warning

- ONNX models must use opset version 13 or lower

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Important

- Export with all required operators included (no missing ops). Check the list of operators here.
- Ensure ONNX version compatibility with your target runtime.
- Verify the file is not corrupted or partially exported.

Q: What additional files are required based on task type?

A: Some tasks require extra metadata files. Refer to below requirements:

Additional Files (Based on Task Type)

Some tasks require extra metadata files so the system can interpret outputs correctly. Filenames are case sensitive and must match exactly.

Tasks requiring `classes.txt`:

- Keyword Spotting
- Genre Identification
- Image Classification
- Image Segmentation
- Object Detection
- Action Recognition
- Visual Wake Word

classes.txt rules:

Rule	Description
One per line	Write **one class label per line**
No extras	Don't include commas, brackets, extra spaces, or numbers (unless part of the label)
No empty lines	The file must not contain blank lines
Order matters	Class order must **match the model's output order**
Count must match	The **number of labels** must equal the model's **output dimension**

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

```
Example (classes.txt):

* cat
* dog
* car
```

Tasks requiring `sampling_rate.txt`:

- Denoiser

sampling_rate.txt rules:

Rule	Description
First line only	Must contain **one integer** on the first line
No units	Do **not** include “Hz” or any text
No extra characters	No symbols or trailing spaces
Must match training	Value must match the **training sampling rate**

```
Example (sampling_rate.txt):

* 16000
```

Q: How does the system detect the framework of my BYOM model?
A: After uploading, the system automatically detects the framework and displays it in the Framework tab under both Create Project (BYOM) and Upload Model.

Q: Why is my PyTorch model not uploading?
A: Ensure that:

1. The file is in ZIP or TAR.GZ format
2. The file size is under 100 MB

Q: Why am I getting a “Missing Class” error during upload?
A: The selected task requires a class file included with the model package. Add the appropriate class file, then upload again.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Q: I've already created a PyTorch model and added the classes.txt file, but why is the upload still failing?

A: For PyTorch models, the input shape .txt file is mandatory. Even if the model file and classes.txt are correctly added, the upload will fail without a valid input shape file. Refer to below table for the requirements:

Input Shape File Requirements

Requirement	Details
File type	`.txt`
File name	Any valid name (`input.txt`, `shape.txt`, etc.)
Content	Must contain only a valid input shape and match the model's expected input size
Supported formats	**Tuple:** `(1, 3, 224, 224)` **Array:** `[1, 3, 224, 224]` **CSV:** `1, 3, 224, 224`

If the input shape file is missing or incorrectly formatted, the PyTorch model upload will fail.

Q: Where can I view my uploaded models?
A: Go to the **My Models** tab.

Supported input data formats

Q: Why can’t I upload certain file types on the Input Data page?
A: Only file types compatible with the selected task type and the compiled model are allowed. Files that do not meet the model’s requirements will be rejected.

Example: Vision models only accept image based file formats

Q: What input file formats are supported after compilation?
A: Supported formats vary by model and may include:

- WAV
- CSV
- JPG

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- JPEG
- PNG

Device selection

Q: Where can I view compatible devices for an ADI model?
A: Follow below steps to see the compatible devices for each model:

1. Open ADI's Model Zoo.
2. Select a model.
3. Go to the Model Details page.
4. Expand the `Performance` dropdown → view `Compatible Platforms`.

Q: Why are some devices not listed?
A: A device will appear only if it:

1. Is compatible with the selected model
2. Is online
3. Is sending a heartbeat signal (indicating active connection)

   A heartbeat is a periodic signal indicating the device is online and ready.

Compilation

Q: Why is the Start Compilation button disabled after completing a compilation?
A: You cannot run another compilation with the exact same parameters. Modify at least one parameter to enable the button and start a new compilation.

Q: Why can’t I change parameters while a compilation is running?
A: Parameters are locked during an active compilation.
To make changes:

1. Abort the current compilation
2. Update the parameters

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

3. Restart the compilation

Q: How do I abort a compilation?
A: Click `Abort Compilation` during an active compilation.

Q: Can I restart an aborted compilation?
A: Yes. After aborting, simply click `Start Compilation` again.

Q: How can I find the reason for a failed compilation?
A: Open the Logs section on the Compilation page.

Pre- and post-processing scripts

Q: My Pre processing scripts are failing, what are the mandatory requirements for preprocessing scripts?

A: The pre-processing script must:

- Include required imports:
- argparse
- Path
- numpy
- Accept command-line arguments:
- --source
- --output_file_path
- --quant_type
- Load raw input data.
- Apply task-specific preprocessing.
- Return a valid NumPy array.
- Save output as .npy
- Print output shape.

Note that the return value cannot be None.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Deployment

Q: Why can't I start inference after uploading input files? Why are more steps required?
A: For BYOM models, you must complete:

1. Pre processing
2. Post processing

Inference can begin only after both steps complete successfully.

Q: Why can't I proceed to Post-Processing after entering the Pre-Processing script?
A: The Pre-Processing script must run without errors. Check the Logs section for details.

Inference

Q: Where can I view my inference history?
A: Use the dropdown menu on the Inference Result page.

Q: Can I start another inference using the same compiled model?
A: Yes. Click `Start New Inference` on the Inference Result page.

Q: Can I download inference results?
A: Yes. Select `Download PDF` on the Inference Results page.

Q: How can I export inference results?
A: Use `Export Inference` on the Inference Results page.

File management

Q: Where can I view all builds under a project?
A: Open the project in the Project Workspace.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Q: How can I export a project?
A: In Project Workspace, use the `Export Build` option in the Project Status panel.

Q: Where can I view the projects I created?
A: Go to Workspace → Projects.

Q: How can I filter created models and projects?
A: Use the Filters panel on the left to filter by modality or framework.

Q: Where can I find all projects associated with a specific model?
A: Open the model in `Workspace > Models tab` and check the `Associated Projects` section.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Support

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Support Resources

This section provides resources to support you while using the EdgeBench application. Whether you are looking for official documentation, community discussions, or direct technical assistance, the links below will guide you to the appropriate support channels offered by Analog Devices.

Support Type	Details
Online Documentation	Comprehensive online documentation is available on developer.analog.com.
Support Forum	The ADI EngineerZone is a public online community forum where you can search previously answered questions or post general technical inquiries. **NOTE: Do not share sensitive or confidential information**, as the forum is publicly accessible and intended for general support discussions.
Technical Support	To request technical support, submit this online support request form.
EdgeBench Application Page	To access the EdgeBench application, visit https://edgebench.app.analog.com.
Support Email	If you prefer not to post questions in a public forum, you may send EdgeBench-related inquiries directly to EdgeBench.Support@analog.com.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Terms of Use

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Terms of Use

1. Your Agreement to These Terms of Use

Analog Devices, Inc., a Massachusetts corporation with its principal office in Wilmington, Massachusetts (“ADI”), is making its cloud hosted platform and services provided through it (collectively referred to as the “Services”) available to you under these Terms of Use (the “Terms”) for use as described herein. The Terms also govern the use of any software provided by ADI or an ADI-approved distributor for purposes of accessing and using the Services (collectively referred to as the “Software”), except to the extent that the Software is otherwise provided by ADI pursuant to separate license terms, in which case such other terms shall, in addition to these Terms, apply to the use of that Software (and any conflict between the provisions of such terms and these Terms shall be resolved in favor of such terms).

By selecting the “I ACCEPT/AGREE” button, or by otherwise accessing or using the Services or Software, you represent and warrant that you are authorized to represent the company or organization you are acting for (the “Customer”) and you conclude a legally binding contract based on these Terms with ADI on behalf of yourself and the Customer. (For purposes of these Terms, the term “you” shall refer to you and the Customer.) If you do not agree to be bound by and comply with all of these Terms, you may not use the Services or Software.

2. Use of the Services

2.1 User Account and Authorized Users

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Using credentials provided by ADI, you must establish one or more unique access accounts to access and use the Services (each a “User Account”). You represent and warrant that you are authorized by the Customer to access the Services on behalf of Customer and have received account credentials from the Customer or directly from ADI in your capacity as a Customer representative. You are responsible for all of your employees and users who you authorize to access the User Account or to whom you supply any account credentials (“Authorized Users”). You are responsible for protecting against unauthorized access to your User Account(s), maintaining the confidentiality of your account credentials, and all activities that occur under your User Account(s) and your Authorized Users’ account credentials. You represent that all registration and other information provided to ADI is true, correct, and complete in all respects, and you agree to promptly update your account information if there are any changes. ADI is not responsible for any harm caused by your Authorized Users. Unless otherwise expressly agreed in writing by ADI, no more than a single concurrent Authorized User may access a User Account at a time. You must notify ADI immediately upon learning of any unauthorized use. ADI reserves the right to limit, suspend, terminate, or take other actions against your accounts, credentials, and use of the Services and Software in the event of any improper access or use of the Services or Software.

2.2 Lawful Use of Services and Software

You may not use the Services or Software unless you are in a jurisdiction where local laws allow such use, and you must comply with these Terms and applicable laws at all times. You agree only to use the Services on behalf of the Customer within the scope of your employment relationship with the Customer. You acknowledge and agree that the Customer, and not ADI, is responsible for ensuring that your use is legal and for any notice and consent requirements that may apply to you in any employer-employee relationship.

2.3 Services

Subject to your ongoing compliance with these Terms and the documentation made available by ADI for the Services (“Documentation”), You may use and access the Services for a period of 30 days from the date when ADI makes the Services available to You (“Demonstration Period”). ADI may provide updates to the Services, Software, and/or Documentation including automatic and over-the-air updates, and use of Services may require you to install and use the most recent update of Software made available by ADI. In addition, ADI may terminate or modify the Services if the termination or modification is generally applicable to all ADI customers using the subject Services.

2.4 Usage Limits

ADI shall not be obligated to increase storage capacity for any account.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

2.5 Restrictions

You may only use the Services and any reports or data generated by the Services ("Reports") for Customer's internal business purpose of evaluating the Services and Software. You shall not (and shall not permit or assist anyone else to) directly or indirectly: (i) reverse engineer, decompile, disassemble, or otherwise attempt to discover the object code, source code, or underlying ideas or algorithms of the Services or Software; (ii) modify, translate, or create any derivative works based on any element of the Services or Software; (iii) rent, lease, distribute, sell, resell, assign, or otherwise transfer any rights to use, access or install any Services or Software; (iv) use any Services or Software for timesharing purposes or otherwise for the benefit of anyone other than the Customer; (v) use any automated means, including without limitation, agents, robots, scripts, scrapers, spiders, or any other data mining technology to access, monitor, extract, or copy any part of the Services, Software, or content made available thereon; (vi) upload to or transmit through the Services or Software any computer viruses, Trojans, worms, or malicious code, or any Customer Data that is illegal, offensive, malicious, harmful, or otherwise contrary to Internet standards or social norms; (vii) remove or alter any copyright or proprietary notices or disclaimers from any Services, Software, Documentation, or Reports; (viii) use the Services or Software to access the data of any other ADI customer; (ix) publish or disclose to third parties any Reports or any other output, interfaces, or evaluation of the Services or Software without ADI's prior written consent; (x) use the Services, Software, or Reports for any purpose other than their intended purpose; (xi) load or penetration test the Services or Software, or otherwise use the Services or Software in any way that is, or could reasonably be expected to interfere with, disrupt, or otherwise be detrimental to the integrity or performance of the Services or Software or ADI's ability to provide either; or (xii) attempt to gain unauthorized access to the Services or Software. If your use of Services or Software or other actions exceed(s) the limits on your account, violate(s) these Terms, or otherwise interfere(s) with or pose risk to the Services, Software, or ADI's systems, ADI shall be entitled to take all reasonable steps to protect the Services, Software, and ADI's systems, including without limitation suspension or termination of your account.

3. Third Party Providers; Use of Other Products

You are responsible for (i) obtaining, deploying and maintaining all computer hardware, software, and other communications equipment necessary for you to access and use the Services via the Internet (as applicable); (ii) contracting with third party ISP, telecommunications, and other service providers to access and use the Services via the Internet (as applicable); and (iii) paying all fees and access charges incurred in connection with the foregoing. You acknowledge that ADI engages third party solutions and services in connection with the hosting and operating of the Services and ADI will have no warranty, support, or other obligation with respect to such third party solutions and services.

4. Customer Data

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

All data, content, materials, software and other information uploaded, submitted, stored, processed, or analyzed by you or your Authorized Users in connection with the Services ("Customer Data") shall, as between ADI and Customer, remain the property of Customer. You are responsible for backing up any Customer Data, and ADI shall not have any responsibility or obligation relating thereto or for any loss thereof. You are responsible for obtaining any and all consents and other rights necessary to collect, submit, use, display, reproduce, and process Customer Data in connection with the Services, Software, and Reports. By uploading, submitting, storing, or processing any Customer Data, you represent that you have all required rights and permissions to do so and to permit the use thereof as set forth in these Terms. ADI reserves the right to block, limit, remove, or delete any Customer Data from the Services at any time for any reason. The ADI Privacy Policy (maintained at https://www.analog.com/en/who-we-are/legal-and-risk-oversight/data-privacy/privacy-policy.html) describes the administrative, physical, technical, and other safeguards that are applied to Customer Data, and used to protect against unauthorized disclosure of Customer Data to third parties, other than for the purposes permitted under these Terms.

5. Data Privacy

By submitting any personal data, including health related data, ("Personal Data") to the Services, You represent that you have all applicable rights to provide such Personal Data to the Services, have obtained any necessary consents from the individual data subjects, and have disclosed to the data subjects that their Personal Data may be processed by a third party such as ADI.

6. Proprietary Rights and Licenses

6.1 License to Software and Documentation

Any Software and Documentation provided hereunder is nonexclusively licensed and not sold (and any Software shall be provided only in object code form). Software and Documentation may only be used for the purpose of accessing and using the Services in accordance with these Terms during the Demonstration Period. Except as expressly set forth in these Terms, no other license or rights are granted to you and all such rights are expressly reserved to ADI.

6.2 Trademarks

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

You may not use any trademarks of ADI or its affiliates ("ADI Marks") without ADI's prior written consent. If consent is provided, it may be withdrawn upon written notice by ADI and is deemed withdrawn in the event of termination or expiration of these Terms. Use of ADI Marks is subject to compliance with ADI trademark usage policies as may be in effect from time to time. You acknowledge and agree that ADI is and shall remain the sole and exclusive owner of the ADI Marks, and you shall not contest or take any action that would jeopardize or diminish ADI's exclusive rights in the ADI Marks.

6.3 Rights to the Services

You agree that, as between ADI and you, and except for the rights expressly granted to you in these Terms, ADI shall exclusively own and retain all right, title, and interest in and to the Services, Software, and Documentation, as well as all generated output (except for Customer Data incorporated therein), adaptations, updates, revisions, enhancements, or improvements thereto (including, but not limited to, underlying or associated algorithms, processes, models, tools, interfaces, works of authorship, inventions, developments and intellectual property rights with respect to the foregoing), and that ADI is not restricted or limited from using or exploiting any of the foregoing in any way.

6.4 License to Customer Data

You hereby grant to ADI a perpetual, irrevocable, royalty-free, fully-paid, transferable, sublicensable (through one or more tiers), worldwide, nonexclusive right and license to use, host, store, transfer, display, perform, reproduce, modify, create derivative works of, distribute, and otherwise exploit Customer Data in order to (i) provide, analyze, and improve the Services, Software, and other ADI products and services; (ii) monitor your use of the Services; and (iii) compile, use, and otherwise exploit aggregate or de-identified data, statistics, measurements, or other metrics derived from Customer Data (including in combination with the aggregate or de-identified data of other ADI customers) for its own purposes. ADI shall use Customer Data in accordance with the ADI Privacy Policy. ADI shall exclusively own all analyses, reports, models, statistics, and other information created, compiled, generated, or derived by ADI in connection with the use of Customer Data pursuant to the aforementioned license and/or the delivery of the Services (collectively, "Analyses"), provided that any Customer Data included in any Analysis shall remain the property of Customer (subject to the license granted to ADI in this Section 8.4).

6.5 Feedback

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

You are not obligated to provide ADI with any error reports, evaluations, suggestions, enhancements, recommendations, or other feedback (collectively referred to as “Feedback”). If you or any of your Authorized Users make or provide any Feedback relating to the Services, Software, Documentation, or any other ADI products or services, then you hereby grant to ADI a royalty-free, worldwide, transferable, sublicensable, perpetual, irrevocable, right and license to use, copy, modify, distribute, and otherwise exploit without restriction any such Feedback and any products, software, or services based on such Feedback.

6.6 Third Party Software

Software may be accompanied by, include, or rely on the use of software made available by one or more third parties (collectively referred to as “Third Party Software”). Third Party Software may include commercial software licensed by third parties, “open source” software meeting the Open Source Definition as promulgated by the Open Source Initiative, or other software programs released by third parties. Each component or portion of Third Party Software is subject to its own separate software license terms and conditions (“Third Party Licenses”). The Third Party Licenses for Third Party Software made available as part of the Software may be set forth or identified (by URL or otherwise) in the Documentation or such other location customarily used for such terms. Third Party Software is provided on an “as is” basis without any representation, warranty or liability of any kind.

6.7 Federal Government End User Provisions

ADI provides the Services, including related Software and technology, for ultimate federal government end use solely in accordance with the following: Government technical data and software rights related to the Services and/or Software include only those rights customarily provided to the public as defined in these Terms. This customary commercial license is provided in accordance with FAR 12.211 (Technical Data) and FAR 12.212 (Software) and, for Department of Defense transactions, DFAR 252.227-7015 (Technical Data – Commercial Items) and DFAR 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation). If a government agency has a need for rights not granted under these Terms, it must negotiate with ADI to determine if there are acceptable terms for granting those rights, and a mutually acceptable written addendum specifically granting those rights must be included in any applicable agreement.

7. Confidentiality

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Except as expressly permitted under these Terms, you agree not to disclose or permit any third party to access (i) any ADI software, technology, programming, specifications, materials, data, reports, or other information regarding the Software, Services, or Documentation, or (ii) any other information designated in writing by ADI as "Confidential" or an equivalent designation any other information which a reasonable person would understand is of a confidential or proprietary nature (collectively referred to as "Confidential Information"). Nothing in these Terms will be construed as preventing disclosure of information that is legally compelled to be disclosed by any governmental investigative or judicial agency pursuant to proceedings over which such agency has jurisdiction, provided that you promptly notify ADI of any such requirement to disclose Confidential Information to the extent legally permitted and cooperate with ADI in obtaining a protective order or other available protection against such disclosure.

8. Disclaimers and Exclusions of Warranties

The SERVICES, SOFTWARE, DOCUMENTATION, AND ALL ASSOCIATED CONTENT AND OUTPUT are provided "AS IS" and "AS AVAILABLE" withOUT warranty of any kind (express oR implied), AND ADI HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, ORAL, OR WRITTEN, WITH RESPECT TO THE SERVICES, SOFTWARE, DOCUMENTATION, AND ALL ASSOCIATED CONTENT AND OUTPUT, INCLUDING WITHOUT LIMITATION ALL WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE.

WITHOUT LIMITING THE FOREGOING, ADI DOES NOT WARRANT THAT THE SERVICES OR SOFTWARE WILL BE UNINTERRUPTED OR ERROR-FREE, OR THAT ANY DATA, CONTENT, OR OUTPUT PROVIDED BY OR THROUGH THE SERVICES OR SOFTWARE WILL BE ACCURATE OR SUITABLE FOR YOUR INTENDED PURPOSES. from time to time, you may experience downtime and erroRs in the operation, functionality, or performance of the services or software.

ADI MAKES NO REPRESENTATIONS OR WARRANTIES REGARDING, AND EXPLICITLY DISCLAIMS THE APPROPRIATENESS OR APPLICABILITY OF, THE USE OF THE SERVICES, SOFTWARE, DOCUMENTATION, OR ANY DATA, CONTENT, OR OUTPUT ACCESSED OR GENERATED THEREFROM FOR ANY SPECIFIC PATIENT CARE OR TREATMENT. YOU UNDERSTAND AND AGREE THAT COMMUNICATIONS SENT WITH USE OF THE SERVICES AND/OR SOFTWARE ARE NOT MONITORED IN REAL-TIME, AND ADI DOES NOT GUARANTEE ANY RESPONSE BY OR INTERACTION WITH ANY HEALTH CARE PROFESSIONAL OR ORGANIZATION. ADI IS NOT A MEDICAL PROVIDER; AND ADI, THE SERVICES, SOFTWARE, AND DOCUMENTATION DO NOT OFFER OR CONSTITUTE MEDICAL OR CLINICAL SERVICES OR ADVICE.

9. Indemnity

9.1 Indemnification by Customer

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Customer shall defend ADI and its affiliates, and their respective directors, officers, employees, consultants, hosting and other service providers, successors and assigns (collectively, “ADI Indemnitees”) against any Claim and indemnify and hold ADI Indemnitees harmless from all associated losses, liabilities, damages and expenses (including reasonable attorneys’ fees) incurred in connection with such Claim where the Claim arises from or relates to: (i) use of the Services, Software, Documentation, or Reports by you or any of your Authorized Users (except to the extent that the Claim falls within the scope of ADI’s obligation of defense and indemnification as detailed in Section 11.1); (ii) any Customer Data, Personal Data, or other information or materials provided by you or any of your Authorized Users, or the authorized processing of it through the Services ; or (iii) any breach or violation of these Terms by you or any of your Authorized Users.

9.2 Indemnification Procedures

A party seeking any defense or indemnification under this Section 11 shall (i) provide the indemnifying party with prompt written notice of any Claim and give the indemnifying party the right to control the defense, provided, however that any delay in or failure to provide such notice shall relieve the indemnifying party of its obligation only to the extent that such failure or delay results in material prejudice to the indemnifying party and (ii) reasonably cooperate with the indemnifying party in assisting the defense of the Claim and in the negotiations or settlements of any Claim. No settlement shall be entered into which would admit wrongdoing of any kind by or on behalf of the party being defended.

10. Limitations of Liability

ADI AND ITS AFFILIATES, AND THE OFFICERS, DIRECTORS, EMPLOYEES, AGENTS, AND REPRESENTATIVES OF EACH OF THE FOREGOING, (COLLECTIVELY, THE “ADI PARTIES”) WILL NOT BE LIABLE FOR ANY LOST DATA, LOST PROFITS, LOST REVENUE, LOST BUSINESS, COST OF COVER, OR INCIDENTAL, CONSEQUENTIAL, PUNITIVE, SPECIAL, OR OTHER INDIRECT DAMAGES OF ANY KIND FOR ANY REASON WHATSOEVER INCLUDING, BUT NOT LIMITED TO, DAMAGES BASED UPON, CONTRACT, WARRANTY, TORT, NEGLIGENCE, STRICT LIABILITY, OR ANY OTHER THEORY, WHETHER OR NOT SUCH DAMAGES ARE FORESEEABLE AND EVEN IF ADI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. ADI’S LIABILITY FOR DAMAGES UNDER OR IN CONNECTION WITH THESE TERMS AND THE SERVICES AND SOFTWARE PROVIDED HEREUNDER, REGARDLESS OF THE FORM OF CAUSE, CLAIM, OR ACTION, WILL NOT (EXCEPT AS MAY BE REQUIRED BY APPLICABLE LAW) EXCEED ONE HUNDRED U.S. DOLLARS ($100). THE EXISTENCE OF MULTIPLE CAUSES, CLAIMS, OR ACTIONS SHALL NOT ENLARGE OR EXTEND THIS LIMIT. IN NO EVENT WILL ADI’S AFFILIATES OR ANY SUPPLIER OF ADI OR ITS AFFILIATES HAVE ANY LIABILITY FOR ANY DAMAGES WHATSOEVER.

11. Term and Termination

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Unless earlier terminated as permitted herein, these Terms shall continue for a period of thirty (30) days from the date on which ADI makes the Services available to you. ADI may terminate these Terms at any time by giving written notice to Licensee. You may terminate these Terms at any time by written notice to ADI, provided that you discontinue any further access to or use of the Services, Software, and Documentation. In the event of your breach of these Terms or the Order Terms, the termination rights are in addition to, and not in lieu or limitation of, any other right or remedy that may be available to ADI. Upon expiration or termination of these Terms, your license and access rights to the Software, Services, and Documentation shall cease. Upon any termination by either you or ADI, you must (i) immediately cease all use of the Services, Software, and Documentation; (ii) promptly destroy all Reports and other data or output downloaded or otherwise obtained from using the Services, Software, or Documentation, as well as all copies of the Software, Documentation, and Confidential Information; and (iii) certify to ADI in writing that you have completed the actions described in (i) and (ii). Upon any such termination, ADI may delete Customer Data and other information related to your account. The following provisions survive the expiration or termination of these Terms for any reason whatsoever indefinitely: 2.1, 2.2, 2.5, and 4 -12.

12. General

12.1 Force Majeure

Any nonperformance by ADI shall be excused to the extent that performance is rendered impossible by strike, fire, flood, governmental action, failure of suppliers, earthquake, pandemic, or any other reason where failure to perform is beyond ADI’s reasonable control.

12.2 Relationship of the Parties

The parties are independent contractors. There is no relationship of agency, partnership, joint venture, employment, or franchise between the parties. Neither party has the authority to bind the other or to incur any obligation on its behalf.

12.3 Choice of Law and Jurisdiction

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

These Terms (including validity and interpretation) will be governed by, construed and enforced in accordance with the laws of the Commonwealth of Massachusetts, without reference to any conflict of law principles. The exclusive place of jurisdiction for all disputes or other case or matter arising out of the contractual relationship between the parties shall be the Commonwealth of Massachusetts, USA; provided that ADI may seek equitable (including injunctive) relief and enforce judgements in any venue of its choosing. The parties hereto specifically exclude the United Nations Convention on Contracts for the International Sale of Goods and the Uniform Computer Information Transactions Act from these Terms.

12.4 Complete Agreement

These Terms constitute the entire agreement between you and ADI relating to your access to and use of the Services and Software. Anything contained in or delivered through the Services that is inconsistent with or conflicts with these Terms is superseded by these Terms. These Terms shall not be deemed or construed to be modified, amended or waived, in whole or in part, except by written agreement signed by both parties hereto. The failure of either party, in any one or more instances, to enforce any of the terms of these Terms shall not be construed as a waiver of future enforcement of that or any other term. If any provision of these Terms is unenforceable, such provision shall be enforced to the extent possible under applicable law, and the remaining provisions will remain in effect.

12.5 Compliance

You must comply with all applicable laws and regulations relating to your use of the Services and Software. Without limiting the generality of the foregoing, you must comply with all export control and trade laws and refrain from exporting or re-exporting the Services or Software or any information related thereto in violation of applicable law.

12.6 Assignability

You agree that these Terms and all incorporated agreements between you and ADI may be automatically assigned by ADI, in its sole discretion, without restriction. You may not assign any rights or delegate any duties hereunder, and any purported assignment or delegation shall be null and void and constitute a material breach.

12.7 Changes to Terms of Use

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

You acknowledge and agree that ADI may update, modify, replace, or change these Terms, in whole or in part, at any time at its sole discretion, with or without prior notice to you. Your continued access to, viewing, contribution to or other use of the Services will constitute your acceptance of any new terms and conditions.

12.8 Contact Information

Any notice or communication from one party to the other shall be in writing and sent by electronic mail and shall be deemed given when received. All notices to ADI shall be sent to Legal.Notices@analog.com.

Appendix A

This product uses third-party software, the licenses and notices for which can be found at the following location.

- https://download.analog.com/adi-edgebench/versions.html

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Release Notes

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Release Notes

- 1.2.0 Release Notes
- 1.1.0 Release Notes
- 1.0.0 Release Notes

See Help for details on how to get support on EdgeBench.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

1.2.0 Release notes

About this release

This release introduces the EdgeBench Build Export feature, enabling users to export EdgeBench project builds as CodeFusion Studio workspace projects. Exported workspaces can be imported directly into CodeFusion Studio, streamlining the transition from cloud-based model benchmarking and evaluation into local development and customization.

What's new

Platform capabilities

- **EdgeBench Build Export to CodeFusion Studio** - Users can now export EdgeBench project builds as fully configured CodeFusion Studio workspace projects. Exported builds include all necessary source and configuration files, enabling developers to seamlessly build, flash, debug, and continue development directly on their target hardware using CFS. This feature streamlines the transition from cloud-based evaluation and prototyping in EdgeBench to local development, reducing setup effort and accelerating time to productivity.
- **Security & Maintenance Updates** - Upgraded third-party packages and libraries to the latest supported versions, incorporating security patches, vulnerability fixes, and stability improvements.

Known limitations

Recompile any EdgeBench projects created before version 1.2.0 to generate the CFS workspace files.

Support and resources

- **CFS User Guide and Documentation:** https://developer.analog.com/docs/codefusion-studio/latest/user-guide/build-and-flash/

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- **Developer Documentation:** https://developer.analog.com
- **EngineerZone Support Forum:** https://ez.analog.com/dsp/software-and-development-tools/edgebench
- **Technical Support:** https://support.analog.com
- **EdgeBench Application:** https://edgebench.app.analog.com
- **Support Email:** EdgeBench.Support@analog.com

Legal

This version release falls under the EdgeBench Terms of Use and applicable Analog Devices licensing agreements.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

1.1.0 Release notes

About this release

EdgeBench 1.1.0 is a minor version update that focuses on security fixes and minor web app patches. This version also added a demonstration video to the front page.

What's new

- Performed dependency updates to ensure the platform remains secure, reliable, and up to date.
- Enhanced user documentation and added a demonstration video to the homepage to improve onboarding and product discoverability.
- This release includes no feature changes.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

1.0.0 Release notes

About this release

EdgeBench 1.0.0 is the first release of Analog Devices' cloud-based Edge AI benchmarking platform. This release introduces a browser-accessible workflow that enables users to evaluate, compile, deploy, and benchmark AI/ML models on real ADI embedded hardware hosted in the ADI Remote Board Farm.

What's new

Platform capabilities

- **Cloud-Based Edge AI Evaluation**
  Evaluate and benchmark AI/ML models directly on ADI embedded devices through a web browser—no local toolchains or hardware required.
- **Remote Board Farm Integration**
  Secure, 24/7 access to ADI evaluation boards hosted in the Remote Board Farm, enabling hardware-accurate inference results under consistent test conditions.
- **End-to-End Guided Workflow**
  A step-by-step build process covering project creation, device selection, compilation, inference, and results analysis.

Model support

ADI Model Zoo Integration

Access a curated collection of ADI-optimized, pre-trained AI models spanning vision, audio, and time-series applications. Models available in EdgeBench are also accessible through Analog Devices' public GitHub repository: ADI Model Zoo

Bring Your Own Model (BYOM)

Upload and evaluate custom pre-trained models using:

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- PyTorch
- TensorFlow
- Open Neural Network Exchange (ONNX)

Quantization and Optimization Options

Configure compilation with int8, int16, float32, and select optimization flags for speed, power, or kernel selection.

Supported devices

EdgeBench 1.0.0 supports deployment and inference on the following ADI platforms hosted in the Remote Board Farm:

- **MAX78002**–AI accelerator SoC with convolutional neural network (CNN) hardware
- **MAX32690**–Arm Cortex-M4 MCU with Bluetooth LE and RISC-V coprocessor
- **ADSP-SC835**–SHARC-FX DSP with Arm Cortex-M33 connectivity core

---

Data and inference

Input Data Support

- Vision: PNG, JPEG
- Audio: WAV, MP3
- Time-series: CSV

Pre- and Post-Processing Scripts

- Python-based scripting environment (required for BYOM)
- Jupyter-like editor with test and validation support

Inference Results and Metrics

- Hardware-accurate latency
- Memory usage
- Power consumption
- Batch-based inference tracking and history

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

User experience

Workspace Management

- Projects, builds, and models organized in a centralized workspace
- 10 GB personal cloud storage per user

Search, Filter, and Sort

- Quickly locate projects and models using metadata, framework, task type, and size filters

Light/Dark Theme Support

- Toggle UI themes using the top-bar controls

Host environment support

EdgeBench 1.0.0 is accessible through modern web browsers on the following platforms:

- macOS: Safari
- Windows 11: Chrome, Microsoft Edge
- Ubuntu 22.04: Firefox

Known limitations

Single active project

Users may work on only one active project at a time.

Model compatibility

Not all user-supplied models may successfully compile or deploy due to size, architecture, unsupported operations, or framework limitations.

BYOM constraints

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

Users can upload and deploy custom-trained models to EdgeBench using the BYOM feature. However, successful compilation and flashing depend on adhering to the following constraints.

Embedded resource limitations

- Remote boards are memory-constrained edge devices; large models may fail compilation or deployment.
- Models should be trained with embedded constraints in mind (for example, smaller input resolutions, lightweight architectures).
- High-resolution inputs are not recommended.

Input/output tensor size

- The target firmware uses **nanopb** (Protocol Buffers for embedded systems) to transmit model inputs and outputs.
- Nanopb supports messages up to **64 KB** by default.
- Models with input or output tensors exceeding this limit will fail during firmware compilation—even if the model itself is valid.
- **int8 quantization is strongly recommended** to reduce tensor size.

Framework and export requirements

Supported frameworks for BYOM:

- PyTorch
- TensorFlow
- ONNX (opset ≤ 13)

Additional requirements:

- Models must be exported with complete architecture and trained weights.
- Checkpoint-only or partial exports are not supported.
- ONNX models must include all required operators and parameters; missing or unsupported operators may cause build failures.

Dependency compatibility

Best results are achieved when training and exporting models using the following versions:

- Python ≥ 3.10

Copyright © 2026 Analog Devices Inc. All Rights Reserved.

- TensorFlow ≥ 2.12
- PyTorch ≥ 2.1.2
- ONNX ≤ 1.13.0

Metadata requirements

- Certain tasks require additional metadata files for correct output interpretation (for example, `classes.txt`, `sampling_rate.txt`).
- Missing, malformed, or mismatched metadata may result in build errors or incorrect inference results.

Support and resources

- **Developer Documentation:** https://developer.analog.com
- **EngineerZone Support Forum:** https://ez.analog.com/dsp/software-and-development-tools/edgebench
- **Technical Support:** https://support.analog.com
- **EdgeBench Application:** https://edgebench.app.analog.com
- **Support Email:** EdgeBench.Support@analog.com

Legal

This release is governed by the EdgeBench Terms of Use and applicable Analog Devices licensing agreements.

Copyright © 2026 Analog Devices Inc. All Rights Reserved.