Lattice sensAI Edge Vision Engine Tool

User Guide

FPGA-UG-02230-1.2

August 2025

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
3DHP	3-Dimensional Head Position
AI	Artificial Intelligence
CPU	Central Processing Unit
DMS	Driver Monitoring System
EVE	Edge Vision Engine
FHD	Full High Definition
FPGA	Field-Programmable Gate Array
FPS	Frames Per Second
GPU	Graphics Processing Unit
HDMI	High-Definition Multimedia Interface
HMI	Human Machine Interface
IR	Infrared
NPU	Neural Processing Unit
OS	Operating System
PC	Personal Computer
RAM	Random Access Memory
RGB	Red, Green, Blue representing a colored image
RoI	Region Of Interest
SDK	Software Development Kit
SoC	System on Chip
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
VVML	Voice and Vision Machine Learning

1. Introduction

The Lattice™ sensAI™ Edge Vision Engine (EVE) tool is a software library that performs edge artificial intelligence (AI) and computer vision on an image to extract data about your presence. EVE utilizes a combination of central processing unit (CPU), graphics processing unit (GPU), and neural processing unit (NPU) to achieve maximum efficiency. EVE focuses on attention-sensing algorithms, computing your distance from the camera, detecting signs of fatigue, assessing your gaze direction, and more. The data computed by EVE is available to other applications via the EVE software development kit (SDK). EveVisualizer is a software application that enables you to visualize different components of EVE directly on the camera feed.

2. Hardware Requirements

2.1. Windows 11 PC

A Windows® 11 personal computer (PC) with the following minimum requirements is needed to run EVE:

- CPU:
  - Intel® i5 processor or better
  - AMD® Ryzen™ 5 series or better
- GPU:
  - Intel Integrated Graphics with OpenCL™ 2.0 support
  - AMD Integrated Graphics with OpenCL 2.0 support
- Memory:
  - Minimum 8 GB of random access memory (RAM)
- Storage:
  - Minimum 1 GB
- Camera:
  - Lenovo® 510 FHD (RGB camera)

2.2. Raspberry Pi 5 Arm

A Raspberry Pi 5 with the following minimum requirements is needed to run EVE:

- CPU:
  - Raspberry Pi 5 (Broadcom® BCM2712 2.4 GHz quad-core 64-bit Arm® Cortex®-A76)
- GPU:
  - Raspberry Pi 5 (VideoCore VII GPU)
- Memory:
  - Minimum 8 GB of RAM
- Storage:
  - 32 GB microSD card or larger
- Camera:
  - Lenovo 510 FHD (RGB camera)
- Power supply:
  - 5 A USB-C power supply
- Cooling system:
  - Any active heat sink designed for Raspberry Pi 5

2.3. Other Hardware Recommendations

2.3.1. Camera Recommendations

Consider the following recommendations when using EVE:

- Recommended RGB webcam for general evaluation: Lenovo 510 FHD.
- Recommended IR camera for evaluation: To be defined in a future release.

2.3.2. External Display Recommendations for Raspberry Pi

For better viewing experience, Lattice recommends using a full HD display with high-definition multimedia interface (HDMI) ports. The Raspberry Pi 5 only outputs a display via the two micro-HDMI ports. To avoid using signal converters, Lattice recommends using a micro-HDMI to HDMI cable.

3. Requesting a Product Key

A product key is required to activate EVE. This product key may have already been provided to you. Otherwise, you can request a product key by emailing *evekey@latticesemi.com* with the following information:

- Full name and email address of recipients for a product key.
- Name and contact information of the distributor (or other sales contact) that introduced you to EVE.
- Name of the organization you represent.
- Summary of your intended use of EVE and/or the problem you want to solve with EVE.

A product key can be provided within 1–2 business days.

4. Software Setup

4.1. Installing EVE on Windows x64

To install EVE on Windows x64, follow these steps:

1. Download the latest revision of the *Lattice-sensAI-EVE-Tool_v6.X_YYY-MM-DD.zip* file from the Lattice sensAI Edge Vision Engine Tool web page.
2. Unzip the package into an empty folder.
3. Open a File Explorer window in the folder that is unzipped.
4. Run *TurboActivate.exe* and enter the product key.

You can now test EVE by running the EveVisualizer program. See the Launching EveVisualizer on Windows x64 section.

4.2. Installing EVE on Raspberry Pi 5 Arm

To install EVE on a Raspberry Pi 5 Arm, follow these steps:

1. Insert the destination microSD card into a computer.
2. Download the Raspberry Pi image (*Lattice-sensAI-RPi-EVE-Tool_v6.X_YYY-MM-DD.img.gz)* from the Lattice sensAI Edge Vision Engine Tool web page.
3. Download and install the Raspberry Pi Imager from the Raspberry Pi web page.
4. Open the Raspberry Pi Imager.
5. Click **CHOOSE OS**, scroll down, and select **Use custom.**
6. Locate and select the file downloaded in step 2.
7. Click **CHOOSE STORAGE** and select the microSD as the destination.
8. Click **NEXT**.
9. Click **NO** on the pop-up window to apply OS customization.
10. Wait for the imaging process to complete.
11. Remove the microSD card from the computer when a pop-up window from the Raspberry Pi Imager shows the process is complete.
12. Insert the microSD card into Raspberry Pi 5 and turn the power on.
13. Log in using the following credentials:
    a. Username: demo
    b. Password: Drive2Lattice
14. Open a terminal and go to */usr/local/bin* using the following command:
    ```
    cd /usr/local/bin
    ```
15. Run EveAuthenticator with your product key using the following command:
    ```
    ./EveAuthenticator -k [key]
    ```

You can now test EVE by running the EveVisualizer program. See the Launching EveVisualizer on Raspberry Pi 5 Arm section.

5. CrossLink-NX-33 VVML Board Integration

EVE acts as a bridge between you and the CrossLink™-NX-33 Voice and Vision Machine Learning (VVML) board. EVE can read the data computed and generated by the CrossLink-NX-33 VVML board, display images and outputs from the board for easier debugging, and forward this information to other applications through the SDK. Additionally, EVE can perform further computation on the CrossLink-NX-33 VVML board video feed, augmenting the data sent to external applications. Currently, support for the CrossLink-NX-33 VVML board is only on the Windows platform. Support for Raspberry Pi 5 is planned for a future release.

5.1. CrossLink-NX-33 VVML Board Setup for EVE

To run EVE alongside a CrossLink-NX-33 VVML board, consider the following:

- When the board is connected to the computer, Windows assigns a COM port number to the USB cable. This COM port number must be specified in the EveVisualizer to ensure that EVE can identify the correct port for connection. You can find this information in Windows Device Manager.
- All options, except for the **draw** options, must be set before starting streaming from the CrossLink-NX-33 VVML board. In the current configuration, the CrossLink-NX-33 VVML board does not support interrupts for back-and-forth communications.

For more information on the CrossLink-NX-33 VVML board setup, refer to the FPGA AI Firmware Pipeline User Guide (FPGA-UG-02229).

6. Features and Applications

6.1. EVE Supported Features

The table below shows the features supported across different platforms and camera types.

Table 6.1 EVE Supported Features

Feature	Platform			
	Windows x64 with RGB Camera	Raspberry Pi 5 Arm with RGB Camera	CrossLink-NX-33 VVML with Windows x64	CrossLink-NX-33 VVML with Raspberry Pi 5 Arm
3DHP (x, y, z)	Supported	Supported	Not supported	Not supported
3DHP (pitch, yaw, roll)	Supported	Supported	Supported[1]	Supported[1]
Face Landmarks	Supported	Supported	Not supported	Not supported
Face ID	Supported	Supported	Supported[1]	Supported[1]
Gaze	Supported	Supported	Not supported	Not supported
Fatigue	Supported	Supported	Not supported	Not supported
Depth	Supported	Supported	Supported[1]	Supported[1]
Eyewear Detection	Supported	Supported	Not supported	Not supported
Visual Speech Detection	Supported	Supported	Not supported	Not supported
Person Detection	Supported	Supported	Supported[1]	Supported[1]
Object Detection	Supported	Supported	Not supported	Not supported
RoI Selection	Supported	Supported	Not supported	Not supported

Note:

1. Supported via the CrossLink-NX33-VVML Person Detection AI pipeline.

6.2. EveVisualizer

EveVisualizer is a tool for developing, testing, demonstrating, and evaluating EVE and algorithms. This tool connects to cameras, feeds images to the engine for processing, and receives data to display. EveVisualizer also contains elements that turn all engine features on or off. This application includes a camera manager that enables you to choose from various image resolutions and formats. In EveVisualizer, features and the corresponding settings are organized into tabs.

6.2.1. Launching EveVisualizer on Windows x64

To launch EveVisualizer on Windows x64, follow these steps:

1. Open a File Explorer window in the unzipped folder from the setup in the Installing EVE on Windows x64 section.
2. Double-click the *EveVisualizer.exe* file.
3. Select a camera and resolution from the top left boxes.
4. Click **Start Streaming**.
5. EveVisualizer displays the live feed of the selected camera.
6. Toggle on/off algorithms via tabs on the left of the application.

6.2.2. Launching EveVisualizer on Raspberry Pi 5 Arm

To launch EveVisualizer on a Raspberry Pi 5 Arm, follow these steps:

1. Open a terminal and move to */usr/local/bin* using this command:
   ```
   cd /usr/local/bin
   ```

2. Launch EveVisualizer using this command:
```
./EveVisualizer
```
3. Select a camera and resolution from the top left boxes.
4. Click **Start Streaming**.
5. EveVisualizer displays the live feed of the selected camera.
6. Toggle on/off algorithms via tabs on the left of the application.

6.2.3. EveVisualizer Main Window

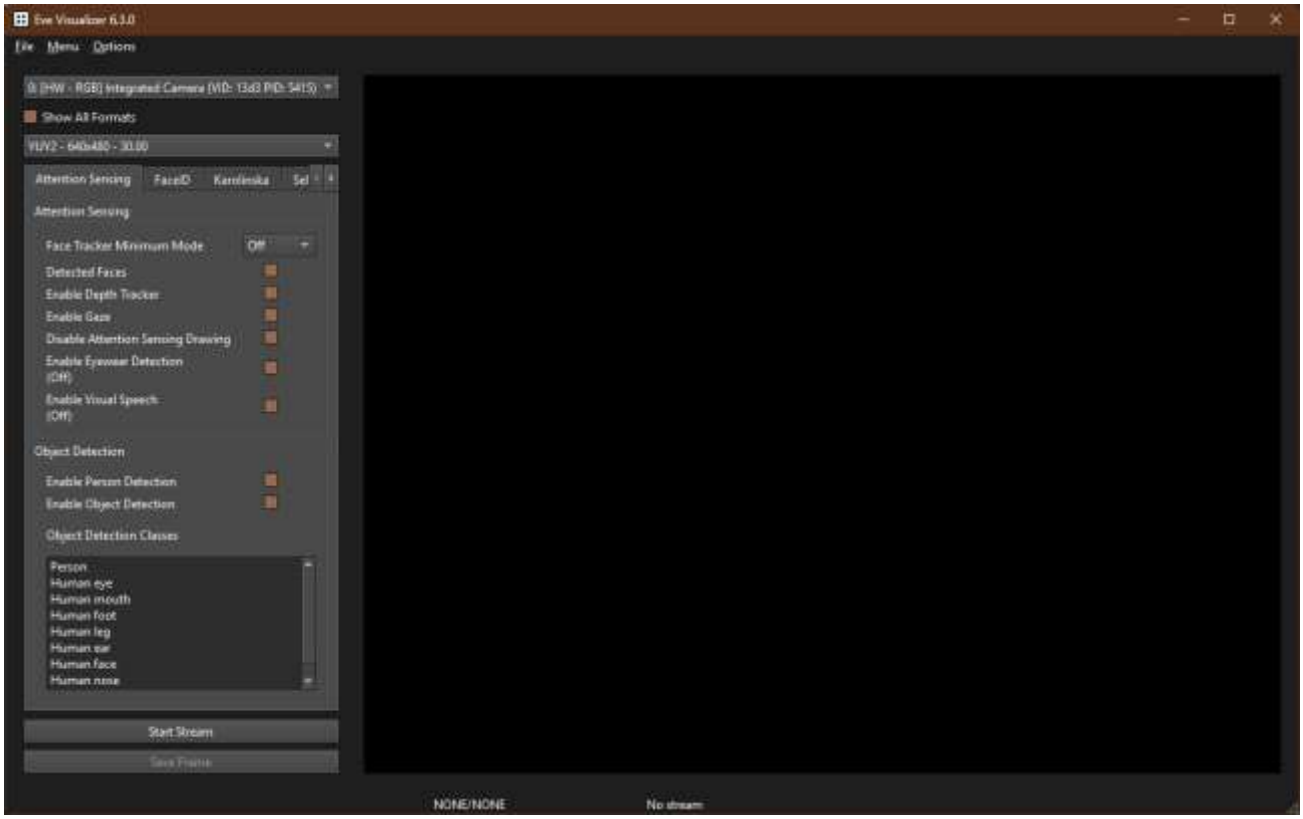

Figure 6.1. EveVisualizer Main Window

The parameters of the **EveVisualizer** main window are as follows:

- Camera list dropdown box—Selects a camera. The dropdown box lists all cameras supported by EVE.
- **Show All Formats**—Filters formats and resolutions by default to display only the preferred formats and resolutions. Turning on this option adds formats and resolutions to the format and resolution dropdown box.
- Format and resolution dropdown box—Selects a format and resolution from a preferred list of formats and resolutions. If **Show All Formats** is selected, the list shows all available formats and resolutions for the selected camera.
- Feature tabs—Contains all options for the various algorithms available in the EVE. Each tab is described in the subsequent sections. The tabs are scrollable from left to right, using the arrows on the top right.
- Start and Stop Streaming
  - **Start Streaming**—Starts the pipeline using an image from the selected camera in the selected format and resolution. Images are displayed in the **Video** tab. Camera, format, and resolution cannot be changed when streaming starts. When streaming starts, this button becomes **Stop Streaming**.

  - **Stop Streaming**—Stops the processing pipeline and camera feed. You can change the camera, format, and resolution.
- **Save Frame**—Saves a single image from the pipeline.
- **Update DMFT log**—This button will be removed in a future update.
- The information at the bottom row is as follows:

Figure 6.2. EveVisualizer Main Window (Bottom Row)

  - fl—Focal length of the selected camera, adjusted to the current resolution.
  - pxsx - pxsy—Pixel size (x and y) of the selected camera, adjusted to the current resolution.
  - Pipeline input format/output format—Displays the format from the input of the pipeline, followed by the format and resolution from the output of the pipeline when streaming starts. EVE can transform the format and resolution of the image passing through its pipeline.
  - Timestamp
    - *V*: viewer frame time—Time taken to acquire the image from the camera.
    - *G*: global frame time—Time taken to perform a complete cycle (acquiring the image and computing EVE pipeline).
    - *P*: pipeline frame time—Time taken to compute active algorithms in the EVE pipeline.
    - The color of the text varies based on the value of *P* over the value of *G* (*P/G*):
      Green: *P/G* < 33%
      Yellow: 33% < *P/G* > 66%
      Red: 66% < *P/G*

6.2.4. FPGA Tab

When using a camera embedded on a CrossLink-NX-33 VVML board, the **FPGA** tab is the only visible tab, with all options available for setting up and connecting to the CrossLink-NX-33 VVML board. Set up options before starting streaming from the CrossLink-NX-33 VVML board. To change any option, you need to close EveVisualizer, unplug the CrossLink-NX-33 device, plug the FPGA board back in the system, and relaunch EveVisualizer. Drawing options can be selected when streaming starts.

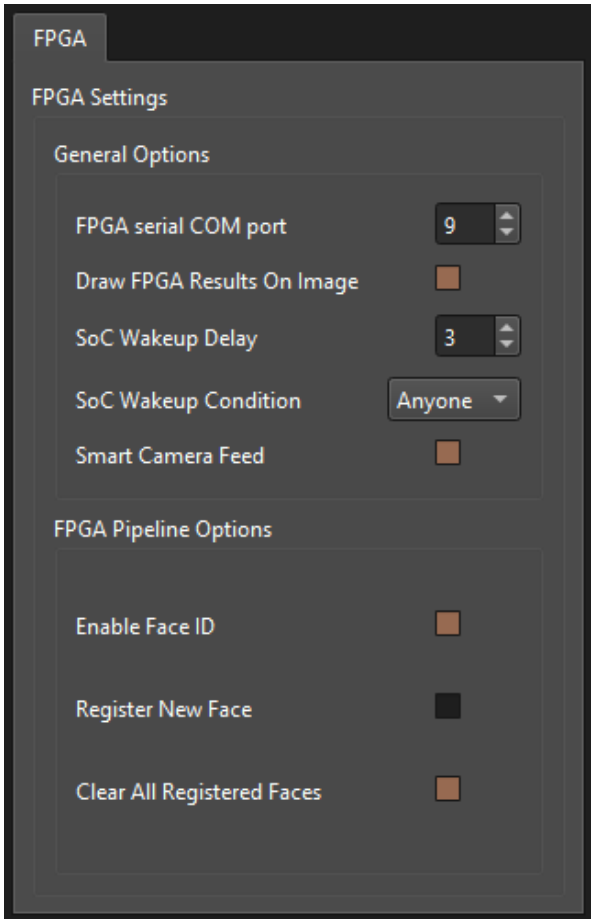

Figure 6.3. EveVisualizer Main Window (FPGA Tab)

The parameters of the **FPGA** tab are as follows:

- **FPGA serial COM port**—Specifies the serial port number (COM1, COM2, and so on) on Windows, that is used to communicate with the board over universal asynchronous receiver-transmitter (UART). If multiple COM ports are available on the system, use the **Device Manager** to identify the port used by the FPGA.

  Note: Using the wrong port may cause rendering issues.
- **Draw FPGA Results On Image**—Selects this option to enable rendering of FPGA results on the image. Multiple bounding boxes are drawn. Bounding boxes from person detection is green for you and dark blue for other detected person. Person detection bounding boxes also have a label *CLOSE* or *FAR* based on the distance of the person from the camera. Bounding boxes from face detection is light blue for your face and red for other detected faces. FaceID data is written below the image, where the results vary from *FaceID: Registered* in greed if the face matches an entry in the gallery, to *FaceID: Unregistered* in red, if the face does not match any entry in the gallery. *FaceID: Unknown* is displayed in red if the algorithm does not run on the detected face.
- **SoC Wakeup Delay**—Specifies the time delay after which the image feed resumes when the wakeup condition is met.
- **SoC Wakeup Condition**—Selects the condition for the Smart Feed feature.
- **Smart Camera Feed**—Selects this option to turn off the image feed when no person is detected.

  Note: Enable this option before streaming starts.
- **Enable Face ID**—Selects this option to register or clear a new face. Data from this feature is shown in the FPGA Data window. You can open the Debug window from **Menu > FPGA Data** or by using the shortcut: Ctrl + F.
- **Register New Face**—To register a new face, selects this option and the FPGA board is reset and waiting. When the streaming starts, the first face detected is registered. Enable FaceID to show this option.

- **Clear All Registered Faces**—Removes all registered faces from the gallery. This option is shown even if FaceID feature is not enabled.

Table 6.2 Supported Features by FPGA Pipeline

Pipeline / Options	HMI
Face Bounding Boxes	Supported
Face Landmarks	Not supported
Face Angles	Supported
Face Center	Not supported
Face Confidence	Not supported
Person Bounding Boxes	Supported

6.2.5. Attention Sensing Tab

Attention-sensing features are features related to you and your attention. EVE computes data about your attention. This feature can be used to augment machine interaction by giving added context to the interaction. This feature also computes fatigue data based on you, the depth, and many other factors.

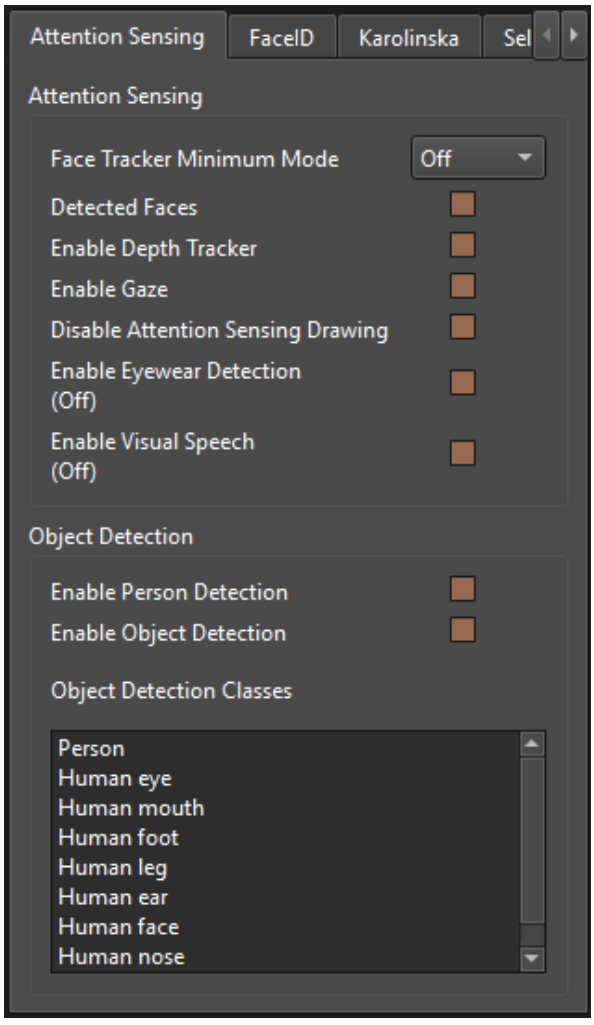

Figure 6.4. EveVisualizer Main Window (Attention Sensing Tab)

The parameters of the **Attention Sensing** tab are as follows:

- **Face Tracker Minimum Mode**—The EVE pipeline is aware of the algorithms and the corresponding dependencies. Activating a feature automatically turns on the dependencies. EVE face tracker features three modes that compute different levels of data to conserve power. This setting forces a minimum mode for the system.
- **Detected Faces**—Draws a white bounding box around each detected user by the algorithm. Each bounding box shows the following data above the box:
  - **Depth**—Estimated depth, in centimeters, at detection time. This value is not updated unless the algorithm runs the detection again.
  - **Detected ID**—Internal ID of the detected face. Not tied to FaceID algorithm.
  - **Elapsed Time**—Time, in seconds, since the last detection.
- **Enable Depth Tracker**—Computes the depth of the tracked person. Displays a scale on the right side of the screen that shows the current position with the minimum and maximum supported ranges.
- **Enable Gaze**—Computes the gaze of the tracked person. This option displays the direction of your gaze with yellow arrows starting from the eye center. Enabling gaze enables eyewear detection implicitly.
- **Disable Attention Sensing Drawing**—Removes all drawings from the image to avoid attention-sensing features.
- **Enable Eyewear Detection**—Detects whether the tracked person is wearing glasses. Updates the label with the value.
- **Enable Visual Speech**—Detects if the tracked person is speaking, based solely on mouth movement. This option does not involve sound processing.
- **Enable Person Detection**—Detects people from the image. This option draws a green bounding box around the detected person and writes, in purple, the ID of the person and the confidence level of the algorithm, ranging from 0.0 to 1.0.
- **Enable Object Detection**—Detects objects on the screen. Like person detection, this method draws a green bounding box around the detected object and writes, in purple, the name and ID of the object, and the confidence level, ranging from 0.0 to 1.0. Detects mobile phones, bottles, mugs, and notebooks.
- **Object Detection Classes**—Enables the drawing of the classes listed in the box. Clicking the objects listed displays the green bounding boxes on the image, and the name, ID, and confidence level of the object, ranging from 0.0 to 1.0.

6.2.6. FaceID Tab

This feature can identify you against a list of registered users. This list is stored in a gallery file. Calibrating users generates the gallery.

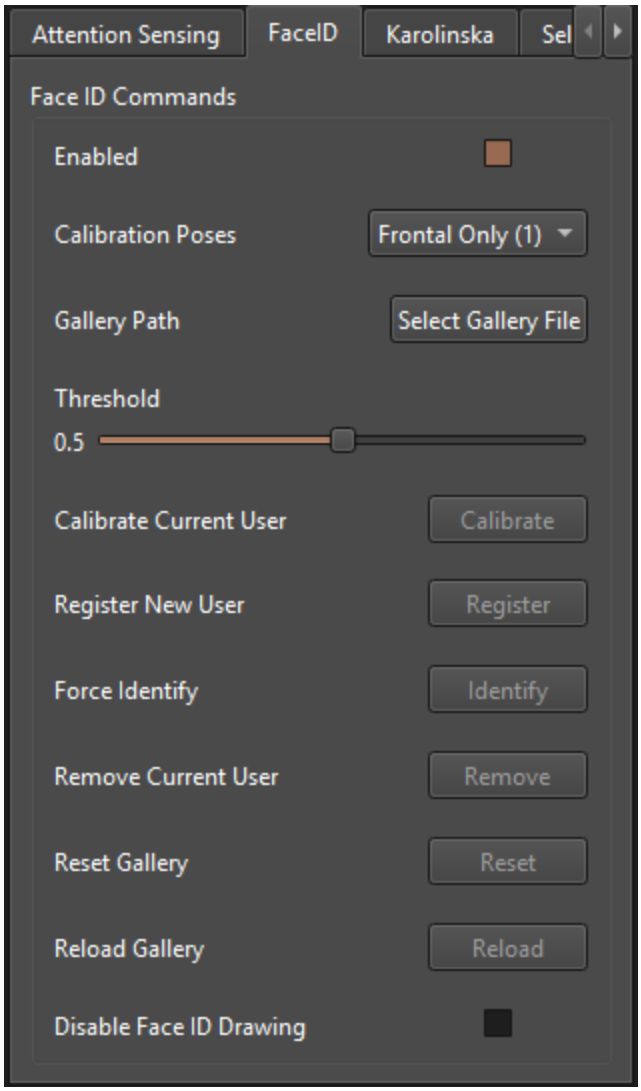

Figure 6.5. EveVisualizer Main Window (FaceID Tab)

The parameters of the **FaceID** tab are as follows:

- **Enabled**—Starts identifying the tracked person on screen. The text appears above the detected person, depending on the current state of the person. If the person is not identified, a red message shows *ID: Not Verified*. If the person matches an entry in the gallery file, a green message shows *ID: [ID NUMBER]*.
- **Calibration Poses**—The current version of the algorithm only supports one form of calibration, Frontal Only, with a single pose. If multiple forms of calibration are supported, the calibrations are listed in this drop-down list.
- **Gallery Path**—The gallery file contains the list of calibrated users. Allows you to provide a gallery file that is created on another computer running the EVE.
- **Threshold**—Slider ranging from 0.0 to 1.0, where you can change the acceptance limit. This value affects the number of *False positive* and *False negative* allowed. The following table shows how the threshold affects identification during testing.
  - **False Positive:** Current user is incorrectly identified as a registered user.
  - **False Negative:** Current user is incorrectly identified as an unregistered user.

Table 6.3 Effect of FaceID Threshold on False Positive and False Negative Based on Internal Dataset

Threshold Value	Number of False Negative Found (Over 2706 Pairs of Images)	Number of False Positive Found (Over 93194 Pairs of Images)
0.4	26	332
0.45	44	104
0.5	72	15
0.55	116	1
0.6	190	0
0.65	283	0

- **Calibrate Current User**—Adds the pose of the current user to the gallery entry. Current users must already have an entry in the gallery file.
- **Register New User**—Creates a new entry in the gallery file for the current user.
- **Force Identify**—Forces the algorithm to perform an identification phase.
- **Remove Current User**—Deletes the current user for the gallery.
- **Reset Gallery**—Deletes every entry from the gallery file.
- **Reload Gallery**—Forces the algorithm to load the gallery file. If this option is used after using the **Gallery Path** option, this option loads in the new gallery file.
- **Disable Face ID Drawing**—Removes the Face ID drawing on the screen while the algorithm continues to run.

6.2.7. Karolinska Tab

The Karolinska Sleepiness Scale is a scale that measures the sleepiness of a tracked user based on various facial fatigue metrics.

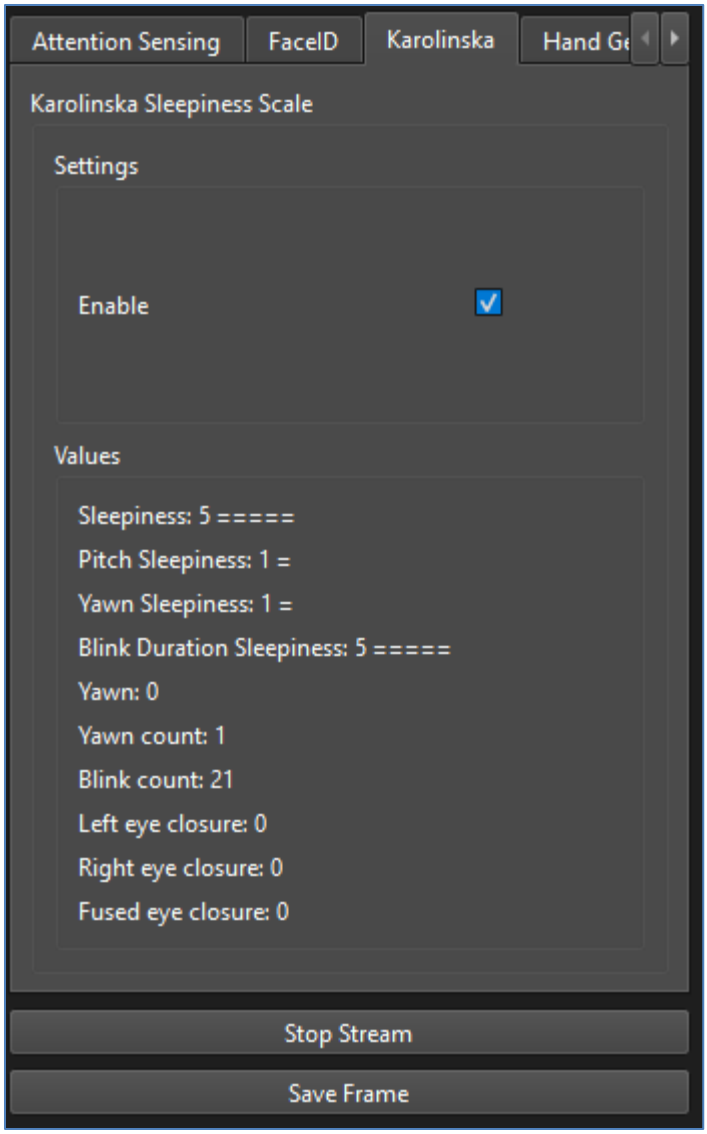

Figure 6.6. EveVisualizer Main Window (Karolinska Tab)

The parameters of the **Karolinska** tab are as follows:

- **Enable**—Turns on or off the Karolinska sleepiness feature. No additional drawing takes place on screen; instead, the values are displayed in the values panel.
- **Sleepiness**—The final sleepiness score on a scale from 1 (most alert) to 9 (fighting for sleep). This value represents the maximum individual sleepiness score.
- **Pitch Sleepiness**—A sleepiness score based on head pitch angle. If the head pitch falls about 30 degrees below the horizon, the sleepiness value goes up to 8. A second event falling below the threshold sets the sleepiness value to 9.
- **Yawn Sleepiness**—A sleepiness score based on the time between yawn events. The first yawn serves as a starting point for measuring the duration between consecutive yawns. The shorter the interval between yawns, the higher the sleepiness score.
- **Blink Duration Sleepiness**—A sleepiness score based on blink duration. The longer a blink lasts, the higher the sleepiness score.
- **Yawn**—A value between 0 and 1 of the mouth opening. 0 means mouth closed, 1 means a complete yawn. Smiles do not affect this score.

- **Yawn count**—The number of yawns counted since the feature is enabled.
- **Blink count**—The number of blinks of both eyes detected since the feature is enabled.
- **Left eye closure**—A scale from 0 to 1 on how much the left eye is closed. A value of 1 means the left eye is fully closed.
- **Right eye closure**—A scale from 0 to 1 on how much the right eye is closed. A value of 1 means the right eye is fully closed.
- **Fused eye closure**—Takes an average of both eyes, considering the confidence (visibility) of each eye to calculate a fused closure of both eyes.

6.2.8. Select RoI Tab

EVE enables the SDK-client application to create regions of interest (RoI) and sends notifications when your gaze interacts with the regions (that is, entering, being in, or leaving a region of interest). EveVisualizer represents regions visually by displaying a dark, transparent overlay over the screen, allowing you to dynamically define regions with mouse clicks and display gaze locations on the screen.

This feature is currently available on the Windows x86 platform. Support for the Raspberry Pi platform will be available in future releases.

The parameters of the **Select RoI** tab are as follows:

- **RoI Selection Enable**—Shows the dark transparent overlay when launched. When enabled, you can perform the following actions:
  - Create RoI by left clicking on the transparent background.
  - Move the selected RoI by clicking and dragging.
  - Remove the RoI by clicking on a RoI and pressing the **Delete** key.
  - Switch between the Fused Gaze mode and the Head Vector mode by pressing the **P** key.
    - Fused Gaze mode: Uses the gaze of the current user to dictate which RoI has the focus.
    - Head Vector mode: Uses the direction of your head to dictate which RoI has the focus.

    When your gaze enters the RoI, the region becomes green, transitioning from transparent to solid based on the duration you remain in the RoI. When you leave the RoI, the region turns yellow and gradually fades to transparent, depending on the time spent outside the RoI.
- **RoI Selection Response Time**—The amount of time gaze needs to stay inside the RoI before triggering the entering phase, and the amount of time gaze needs to stay outside the RoI before triggering the leaving phase.

6.2.9. Main Window Menus

6.2.9.1. File

The list in **File** is as follows:

- **Save Frame** (shortcut: Ctrl + S)—Saves a single image from the pipeline.

6.2.9.2. Menu

The list in **Menu** is as follows:

- **FPGA Data** (shortcut: Ctrl + F)—Opens a window showing data received from FPGA computation.
- **Open Screen and Camera Window** (shortcut: Ctrl + L)—Opens a window that allows you to enter the position of the camera relative to the top-left corner of the screen, in millimeters.

6.2.9.3. Options

The list in **Options** is as follows:

- **Unlimited Pipeline Framerate** (shortcut: Ctrl + U)—This option is not available and does not affect the pipeline running with live feed.
- **Display Framerate in FPS** (shortcut: Ctrl + P)—Changes the framerate display in the main window from a time in milliseconds to a number representing how many frames the EVE is processing per second.

7. EVE SDK Usage

The EVE SDK is a C interface that focuses on the attention-sensing features of the engine. You can control the EVE and decide which features are computed on every frame. EVE SDK also allows you to interact with the CrossLink-NX-33 VVML board. See the CrossLink-NX-33 VVML Board Integration section. You can get the computed data from the CrossLink-NX-33 VVML board through the EVE results.

For more information on the EVE SDK data structures, functions, and code examples, refer to the documentation folder in the package.

7.1. EVE SDK Setup on Windows x64

To communicate with the EVE, an external application must copy the EVE binaries to the bin folder, add the binaries to the project dependencies, and include the EVE header files in the additional include list.

Reference

- FPGA AI Firmware Pipeline User Guide (FPGA-UG-02229)
- CrossLink-NX web page
- CrossLink web page
- Lattice sensAI Solution Stack web page
- Lattice sensAI Edge Vision Engine Tool web page
- Raspberry Pi web page
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.2, August 2025

Section	Change Summary
All	• Changed *EVE Visualizer* to *EveVisualizer*. • Performed minor formatting and editorial edits.
Abbreviations in This Document	Added definition for AI, FHD, FPGA, FPS, HDMI, OS, PC, RAM, SoC, UART, and USB.
Hardware Requirements	Added the minimum requirements for camera in the Windows 11 PC and Raspberry Pi 5 Arm sections.
Requesting a Product Key	Removed the description about the evaluation period.

Software Setup	• Updated the file name in the Installing EVE on Windows x64 section.
	• Updated the Installing EVE on Raspberry Pi 5 Arm section as follows:
	  • Updated the Raspberry Pi image file name.
	  • Added steps to run EveAuthenticator.

Features and Applications	• Updated the supported features in the EVE Supported Features section.
	• Updated the command to launch the EveVisualizer in the Launching EveVisualizer on Raspberry Pi 5 Arm section.
	• Updated the EveVisualizer Main Window section as follows:
	  • Updated Figure 6.1. EveVisualizer Main Window.
	  • Removed the Video and DMFT log tabs from the **EveVisualizer** main window.
	  • Updated Figure 6.2. EveVisualizer Main Window (Bottom Row).
	  • Updated the information at the bottom row.
	• Updated the FPGA Tab section as follows:
	  • Added descriptions on changing options when using the FPGA tab.
	  • Updated Figure 6.3. EveVisualizer Main Window (FPGA Tab).
	  • Updated the *Draw FPGA Results On Image*, *Enable Face ID*, and *Clear All Registered Faces* parameters.
	  • Updated Table 6.2 Supported Features by FPGA Pipeline for HMI instead of Car Sentry.
	• Updated the Attention Sensing Tab section as follows:
	  • Updated Figure 6.4. EveVisualizer Main Window (Attention Sensing Tab).
	  • Added the *Detected Faces* parameter.
	  • Removed the *Use single-output model* parameter.
	• Updated the FaceID Tab section as follows:
	  • Updated Figure 6.5. EveVisualizer Main Window (FaceID Tab).
	  • Added the *Threshold* parameter.
	  • Changed the *Calibrate New User* to *Register New User* parameter.
	• Updated Figure 6.6. EveVisualizer Main Window (Karolinska Tab).
	• Updated the *RoI Selection Enable* parameter in the Select RoI Tab section.
	• Changed *Open Debug Window* to *FPGA Data* parameter in the Main Window Menus – Menu section.

Revision 1.1, April 2025

Section	Change Summary
All	Renamed document from *Lattice sensAI Edge Vision Engine SDK* to *Lattice sensAI Edge Vision Engine Tool*.
Abbreviations in This Document	• Added abbreviations: 3DHP, DMS, IR, and RGB. • Removed abbreviations: AI, IC, OpenCL, and RAM.
Introduction	Updated this section.
Hardware Requirements	• Renamed section *2 Installation* to section 2 Hardware Requirements. • Renamed subsection *2.1 Pre-requisite* to section 2.1 Windows 11 PC.

Section	Change Summary

	• Added the following subsections:
	  • 2.2 Raspberry Pi 5 Arm
	  • 2.3 Other Hardware Recommendations

Requesting a Product Key	Added this section.
Software Setup	Reworked subsection *2.2 Installation Steps* and renamed to section 4 Software Setup.
CrossLink-NX-33 VVML Board Integration	• Reworked section *4 CrossLink-NX-33 Integration* and moved to section 5 CrossLink-NX-33 VVML Board Integration. • Reworked subsection *4.1 Setup* and renamed to subsection 5.1 CrossLink-NX-33 VVML Board Setup for EVE.
Features and Applications	• Added this section. • Reworked section *5 CameraViewer* and renamed to subsection 6.2 EVE Visualizer. • Reworked subsection *5.1 FPGA Tab* and moved to subsection 6.2.4 FPGA Tab. • Reworked subsection *5.2 Attention Sensing Features* and moved to subsection 6.2.5 Attention Sensing Tab.
EVE SDK Usage	• Reworked section *3 Usage* and renamed to section 7 EVE SDK Usage. • Reworked subsection *2.3 EVE SDK Setup* and renamed to subsection 7.1 EVE SDK Setup on Windows x64.
References	Updated references.

Revision 1.0, November 2024

Section	Change Summary
All	Initial release.

www.latticesemi.com