RAVSim

Run-time Analysis and Visualization Simulator

A quick start user manual,

Current RAVSim Version: v2.0,

Released in November 2023,

For help, please contact:

Sanaullah (sanaullah@hsbi.de)

Thorsten Jungeblut (thorsten.jungeblut@hsbi.de)

Industrial IOT Lab,

Department of Engineering and Mathematics,

Bielefeld University of Applied Sciences and Arts, Germany

https://www.hsbi.de/

Contents

About RAVSim

RAVSim, short for Run-time Analyzing and Visualization Simulator, is an interactive tool designed for the analysis and simulation of spiking neural network models. Developed on the LabVIEW (Laboratory Virtual Instrument Engineering Workbench) platform [1], RAVSim uses a multi-core architecture to provide users with a versatile environment for exploring the behavior of spiking neural networks.

RAVSim offers both deterministic (by solving Ordinary Differential Equations (ODEs)) [2] and stochastic (using stochastic leaky integrate-and-fire (LIF) neurons algorithm) simulations [3]. This dual simulation capability enables users to gain a comprehensive understanding of spiking neural network dynamics through various approaches. What sets RAVSim apart from other tools is its unique ability to execute, analyze, extract, and validate models using image-based datasets. This empowers users not only to analyze and simulate spiking neural networks but also to create custom datasets tailored to their specific image pixel, quality, and extension preferences.

Furthermore, RAVSim has recently introduced new features to enhance the user experience. It now offers a convenient option to generate weights for your spiking neural network models using image classification training. This feature eliminates the need for complex coding; you can simply utilize RAVSim to train your model and generate the required weights effortlessly.

In addition to weight generation, RAVSim facilitates the comparison of different spiking neural network models. Users can dynamically update sets of parameter values for each model at runtime and obtain comparative results. This functionality aids in identifying the most suitable model for your specific spiking neural network application.

To learn how to effectively use this tool at runtime experiments, this document provides a comprehensive guide on using RAVSim for performing run-time experiments. You can also watch a video demonstration at the following link:

RAVSim v1.0: https://www.youtube.com/watch?v=Ozv0MXXj89Y

RAVSim v2.0: https://www.youtube.com/watch?v=J9UR2IUA5Bc

Basic Requirements

A graphical programming approach LabVIEW Runtime 2021 or above.
A minimum computer requirement:

- Processor: i3 CPU @ 2.3 GHz

- RAM: 8.0 GB
- System Type: 32/64-bit OS
- Operating System: Mac OS, Window

To install Runtime LabVIEW:

https://www.ni.com/kokr/support/downloads/softwareproducts/download.labview.html#305931

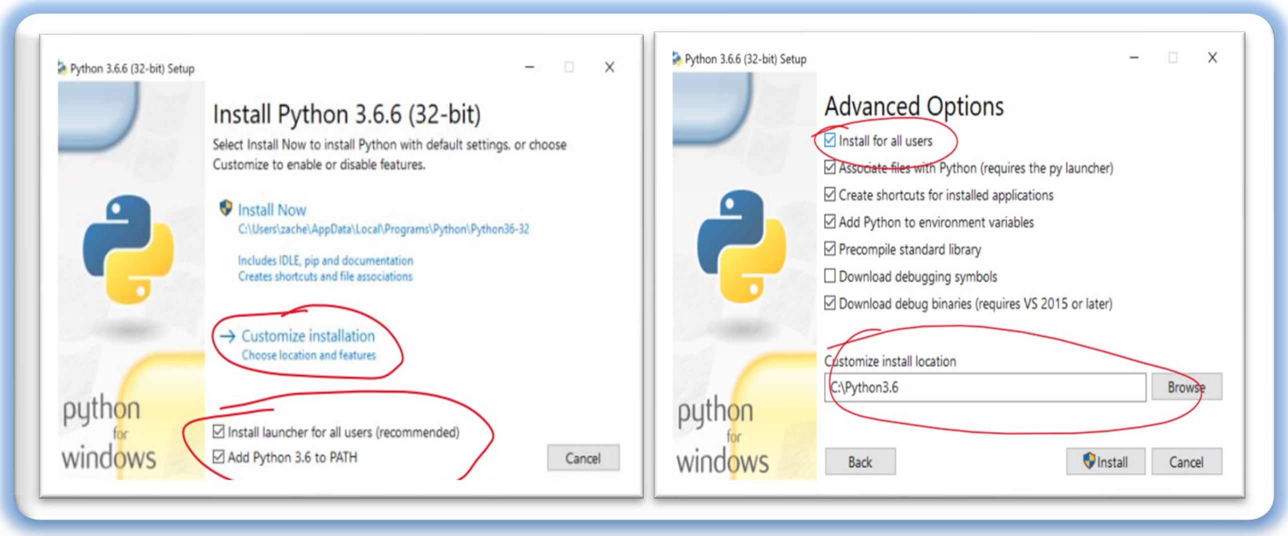
To install Runtime LabVIEW:

Users May also need to install the Python version for image classification VI. As LabVIEW requires a very specific Python Installation and is not compatible with many installers. It requires you to download Python 2.7 or 3.6 from "**Python.org**". Since most individuals polled still use LabVIEW 32-bit that version is referenced.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		9a080a86e1a8d85e45eee4b1cd0a18a2	22930752	SIG
XZ compressed source tarball	Source release		c3f30a0aff425dda77d19e02f420d6ba	17156744	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	c58267cab96f6d291d332a2b163edd33	28060853	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	3ad13cc51c488182ed21a50050a38ba7	26954940	SIG
Windows help file	Windows		e01b52e24494611121b4a866932b4123	8139973	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	7148ec14edfdc13f42e06a14d617c921	7186734	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	767db14ed07b245e24e10785f9d28e29	31930528	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	f30be4659721a0ef68e29cae099fed6f	1319992	SIG
Windows x86 embeddable zip file	Windows		b4c424de065bad238c71359f3cd71ef2	6401894	SIG
Windows x86 executable installer	Windows		467161f1e894254096f9a69e2db3302c	30878752	SIG
Windows x86 web-based installer	Windows		a940f770b4bc617ab4a308ff1e27abd6	1293456	SIG

Installing Python for LabVIEW:

After downloading the above-specified version, use the following method for installation,



After the successful installation, the windows user might need to set the system path. To set the path, please use the following link,

- https://learn.microsoft.com/en-us/previous-versions/office/developer/sharepoint-2010/ee537574(v=office.14)#to-add-a-path-to-the-path-environment-variable.

1. Analysing the SNN models using RAVSim

It's worth mentioning that RAVSim has introduced new features, such as the ability to generate weights for spiking neural network models through image classification training, making it easier for users to generate model weights without the need for extensive coding. Additionally, RAVSim now supports the comparison of different spiking neural network models. Users can dynamically adjust parameter values for each model at runtime and obtain comparative results, aiding in the selection of the most suitable model for their specific spiking neural network application. In this section, you will learn how to analyze and visualize the SNN models in RAVSim.

1.1. Launching RAVSim

The front interface/welcome page of RAVSim, shown in Figure 1, will appear when you double-click on **RAV*Sim.app*** (for MAC OS) or **RAV*Sim.exe*** (for Windows OS).

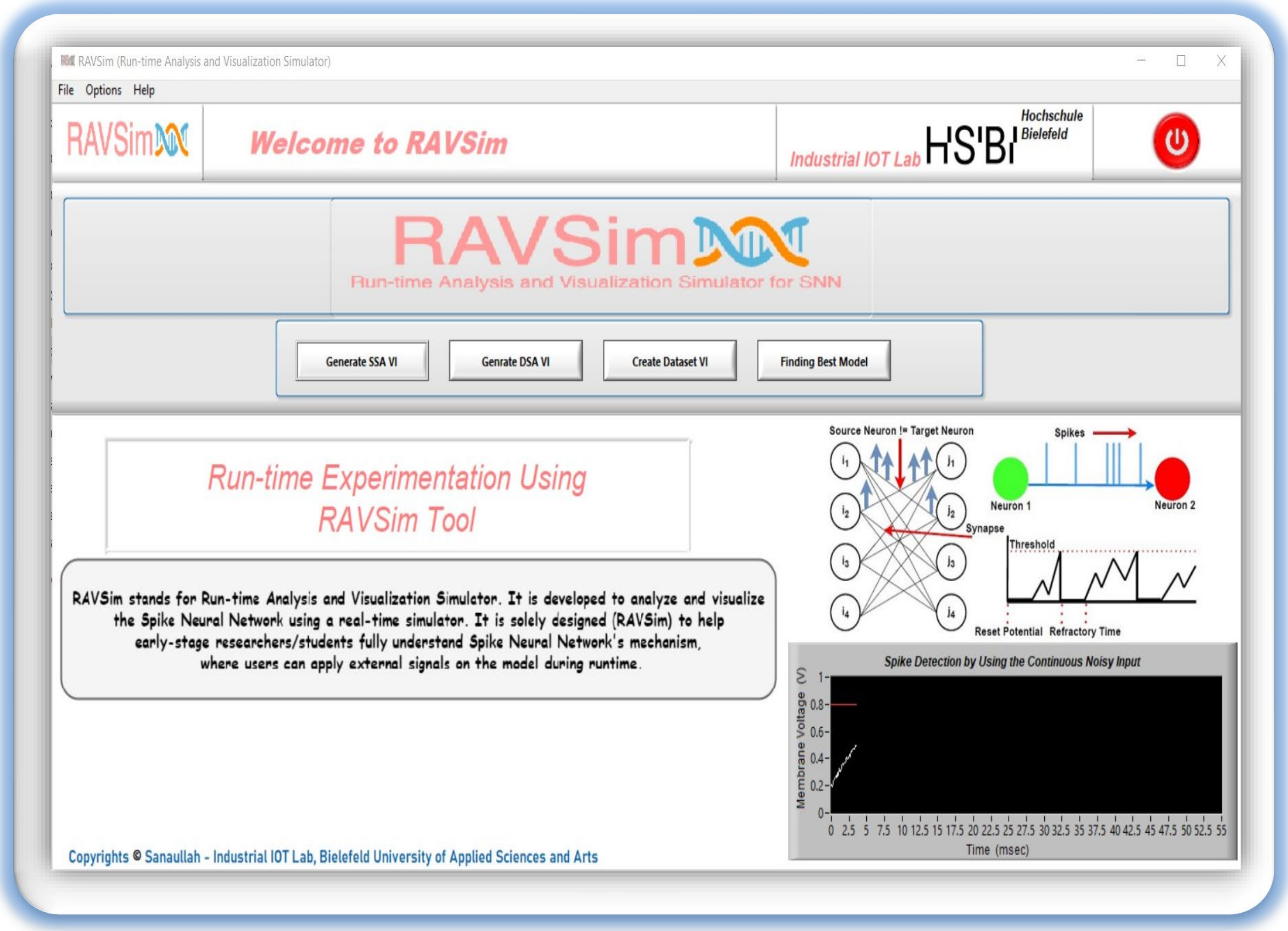


Figure 1: Front interface/welcome page of RAVASim.

On the bottom right side of the corner in Figure 1, the simulation is only for demonstration purposes by using the NLIF model. However, the same model is also used in the run-time simulation VI, where a user can analyze and visualize the model by increasing/decreasing the parameters, all during runtime.

1.2. RAVSim Menu Bar

A menu bar is **a graphical control element that contains drop-down menus**.

The menu bar's purpose is to supply a common housing for window- or application-specific menus that provide access to such functions as closing the RAVSim tool, interacting with a WTA network model manually, and looking at the default parametric values of the LIF model and image classification model simulation or displaying help or contact information.

Figure 2: Menu tab showing the drop-down listed features of a RAVSim tool.

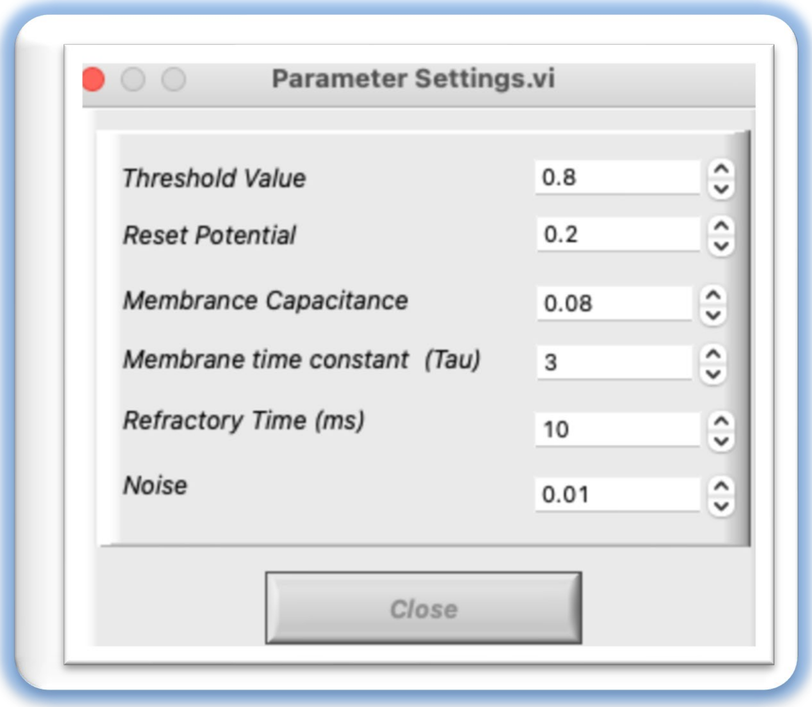


Figure 3: By using the drop-down menu,
users can see the default parametric values of a Runtime Interaction VI.

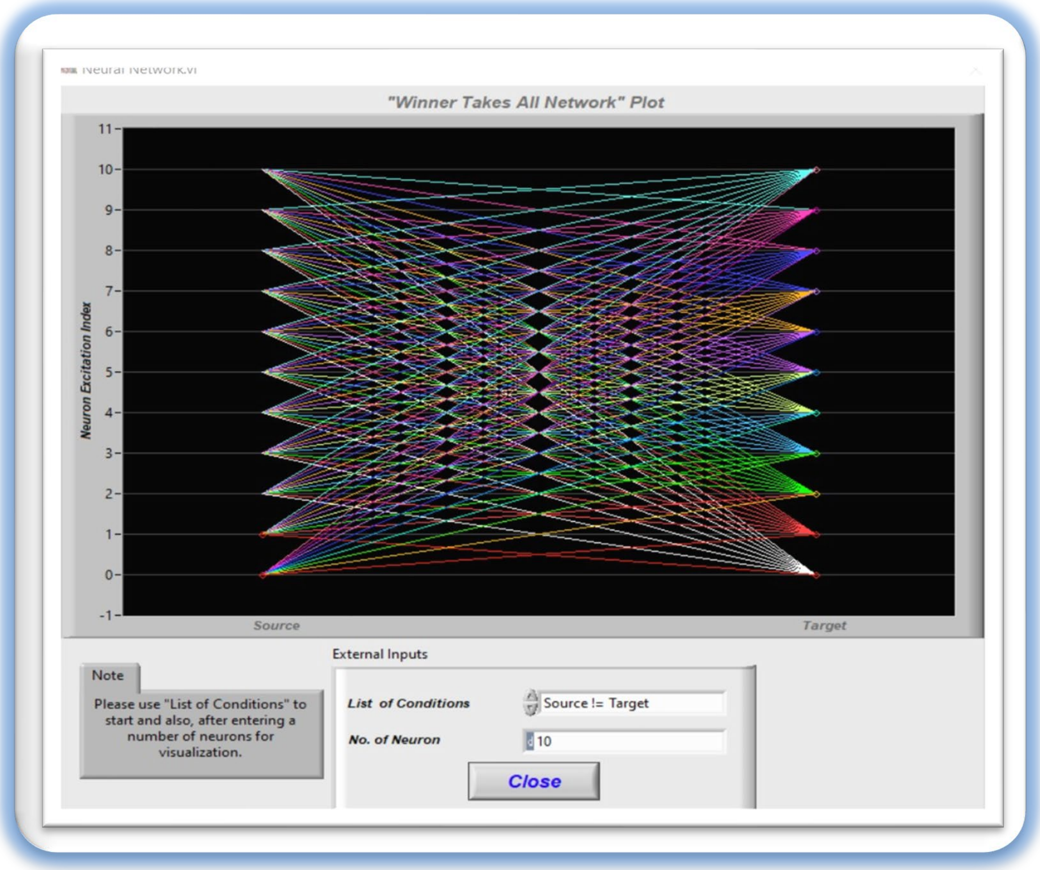


Figure 4: By using the drop-down menu, users can use a manually generated WTA network.

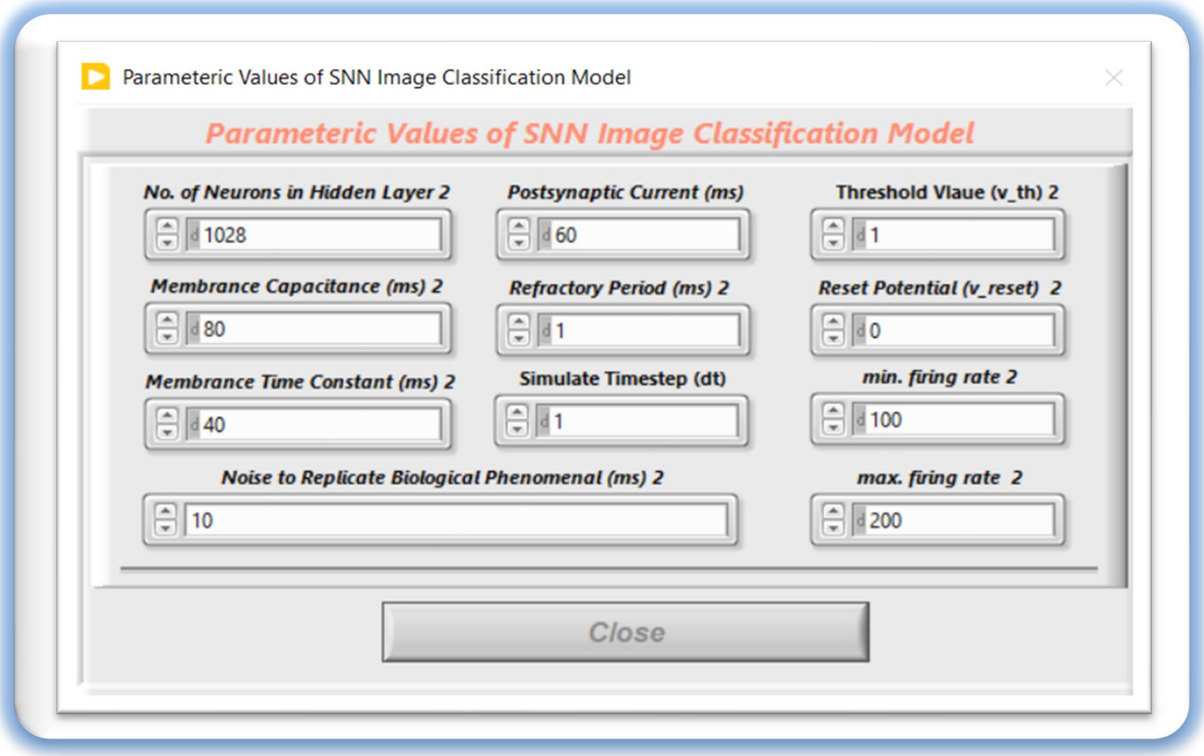


Figure 5: By using the drop-down menu,
users can see the default parametric values of an Image classification VI.

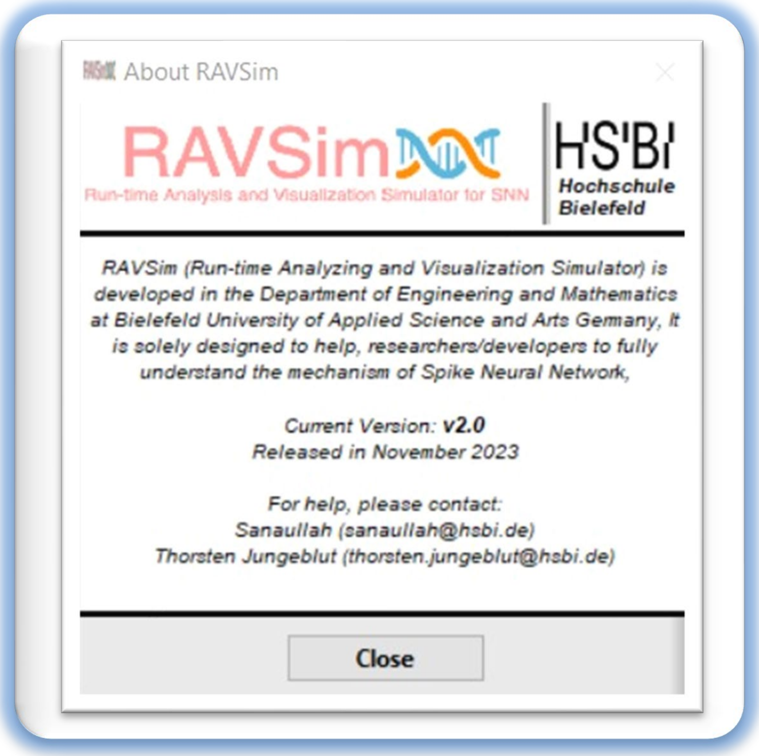


Figure 6: By using the drop-down menu from a **Help list**, users can see the relevant information

2. Generating a Run-time Interaction VI:

Upon clicking the "Generate SSA VI" button, which triggers the appearance of a virtual instrument as depicted in Figure 7, users will transition into the runtime interactive environment, a pivotal feature of RAVSim. This environment provides a dynamic platform for spiking neural network analysis and visualization.

Within this runtime interactive environment, a "Run_Spikes.vi" dialogue box will appear, offering users a comprehensive range of options for tailoring their simulations. This dialogue box serves as the control canter, allowing users to finely adjust simulation parameters. Users can input specific values for parameters related to the neural network model, including but not limited to membrane potential, firing thresholds, and time constants.

Notably, users have the flexibility to enter their preferred parameter values immediately, or they can choose to utilize the default values as a starting point. This adaptability is a significant advantage, as it accommodates both experienced users who may have precise parameter requirements and those who are exploring spiking neural network simulations for the first time.

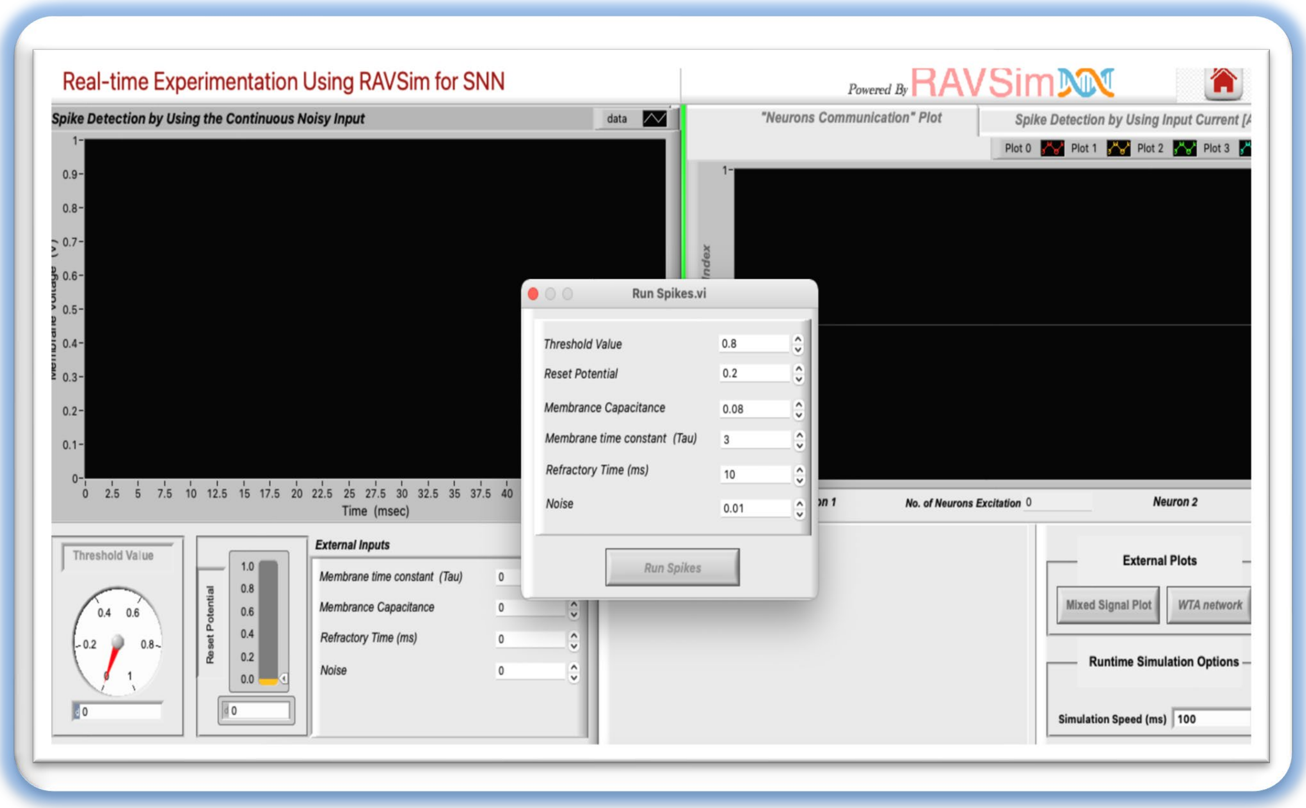


Figure 7: RAVSim Virtual Instrument specifically for a LIF model

2.1. Runtime Experimentation VI

Run-time experimentation of the RAV-Simulator provides a lot of flexibility for the user to interact with the SNN model at any point in time. Moreover, the runtime interactive environment is designed to facilitate real-time adjustments to parameter values during the simulation. This live modification capability empowers users to observe the immediate effects of parameter changes, enhancing their understanding of how different variables impact the behavior of the spiking neural network.

Figure 7 shows the simulator's screen overview before hitting the “Run Spike” button. Plots in Figure 8, visualize the spikes generated with the values that the user has inserted. And they can change the values of the parameters like the threshold, reset potential, etc during the run-time. The right plot shows the spike detection by using continuous noisy input and the left plot shows the users, how neurons are communicating with each other, and how many times one neuron is interacting with the other, also the right plot menu option 2nd, visualizes the plot that shows the detection of spikes using the input current. Once the simulation is started, the user can increase or decrease the parametric values of a model by using the left bottom “external Parameter Values” table and observing their effects on the SNN model during run time.

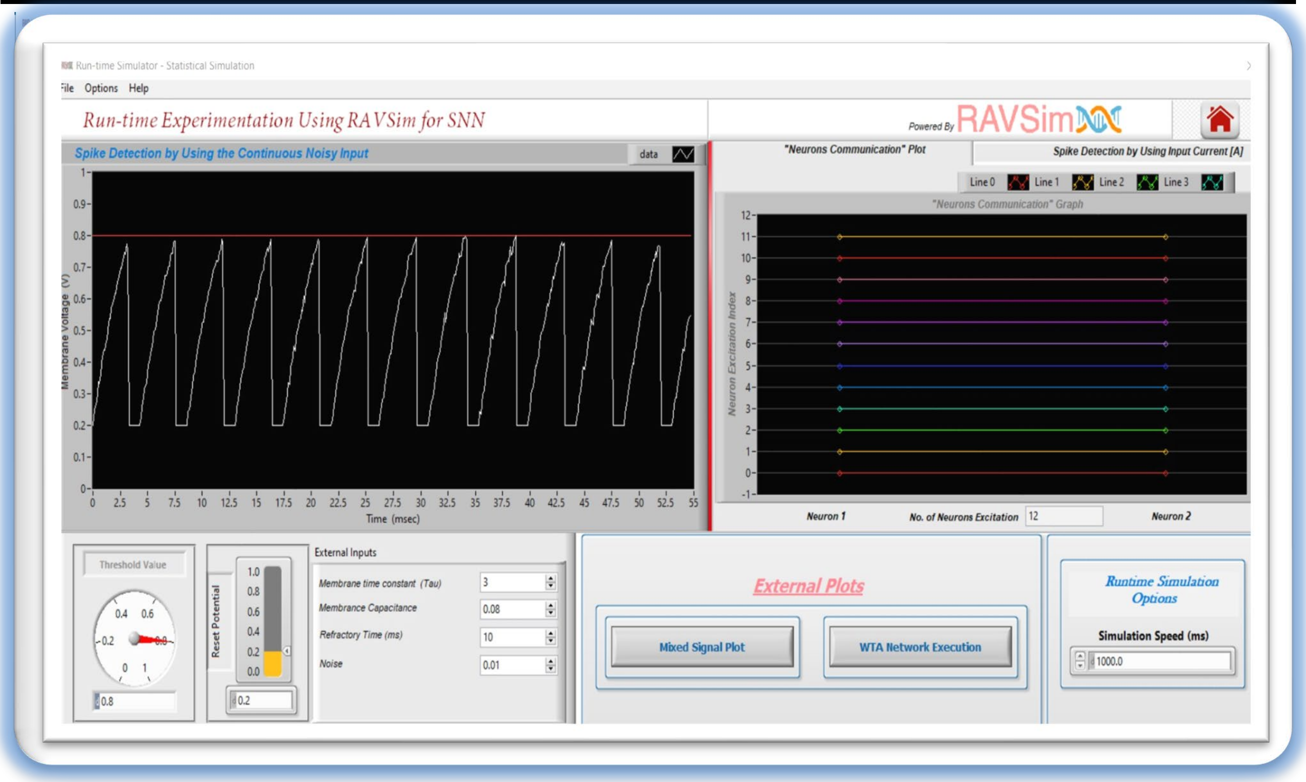


Figure 8 (A): Run time Experimentation of RAV-Simulator for SNN,

Runtime interaction is visualized on the left side plot, while on the right side, neuron communication based on the Winner-Takes-All (WTA) "i==j" network scheme is displayed.

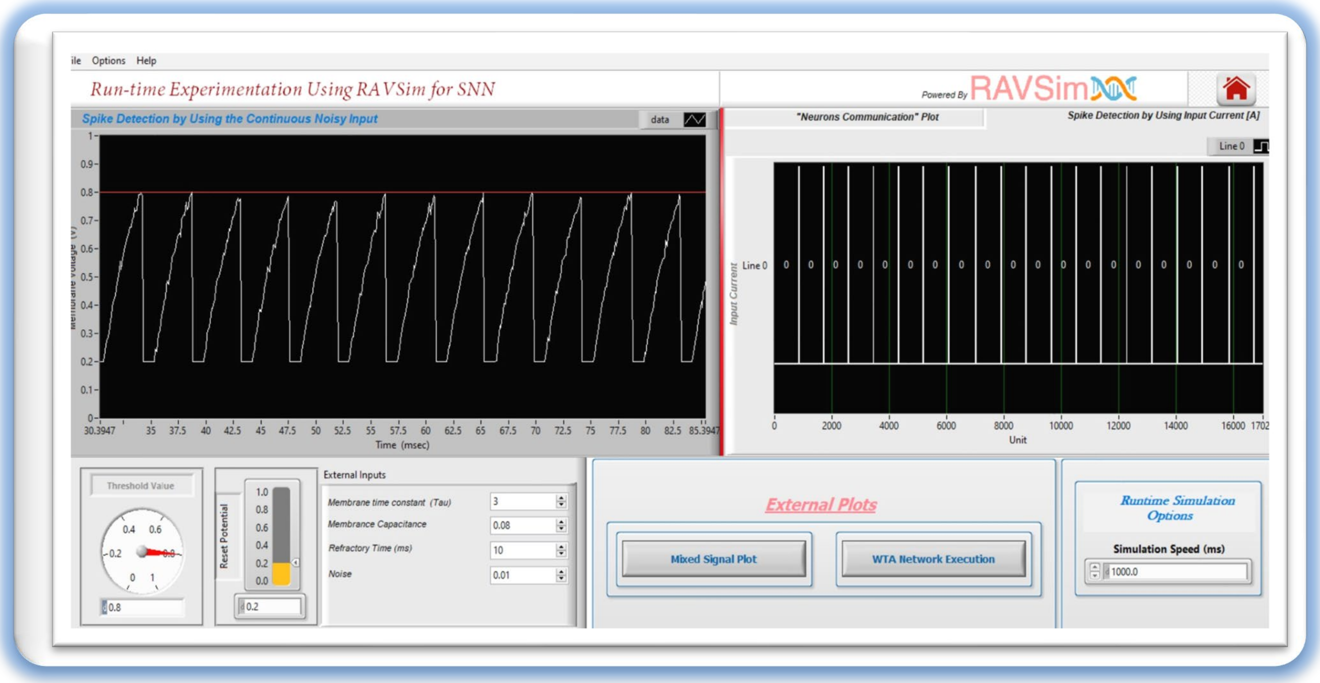


Figure 8 (B): Run time Experimentation of RAV-Simulator for SNN,

Runtime interaction is visualized on the left side plot, while on the right side, spike detection is performed using input current.

In Figure 8 (A-B), a screenshot of the run-time simulation of a LIF model using continuous noisy input is presented, along with a randomly captured neuron communication plot. This figure illustrates that, in contrast to Noise-LIF events with default parameter values, users have the capability to interact with the model and adjust the concentration of the model's input to any desired level at any moment during the simulation. Moreover, when instantaneous triggering of input model concentrations to the threshold level is necessary, the neuron communication plot visually presents the real-time excitation results of the neurons.

3. Generating an Image Classification VI:

When you click the "Generate DSA VI" button, this action triggers the appearance of a dedicated virtual instrument, as illustrated in Figure 9. This virtual instrument serves as the gateway to the image classification simulation environment. Within this environment, users have the capability to input their preferred parameter values, crucial for the analysis and visualization of image classification processes. This functionality allows users to explore and experiment with different datasets. Upon initiating the image classification simulation by pressing the "Start Analysis" button, the initial green lights will transition to red, and the status will change to "In Progress." This alteration in status serves as a clear indicator to the user that the simulation is actively running and analyzing the chosen image classification tasks.

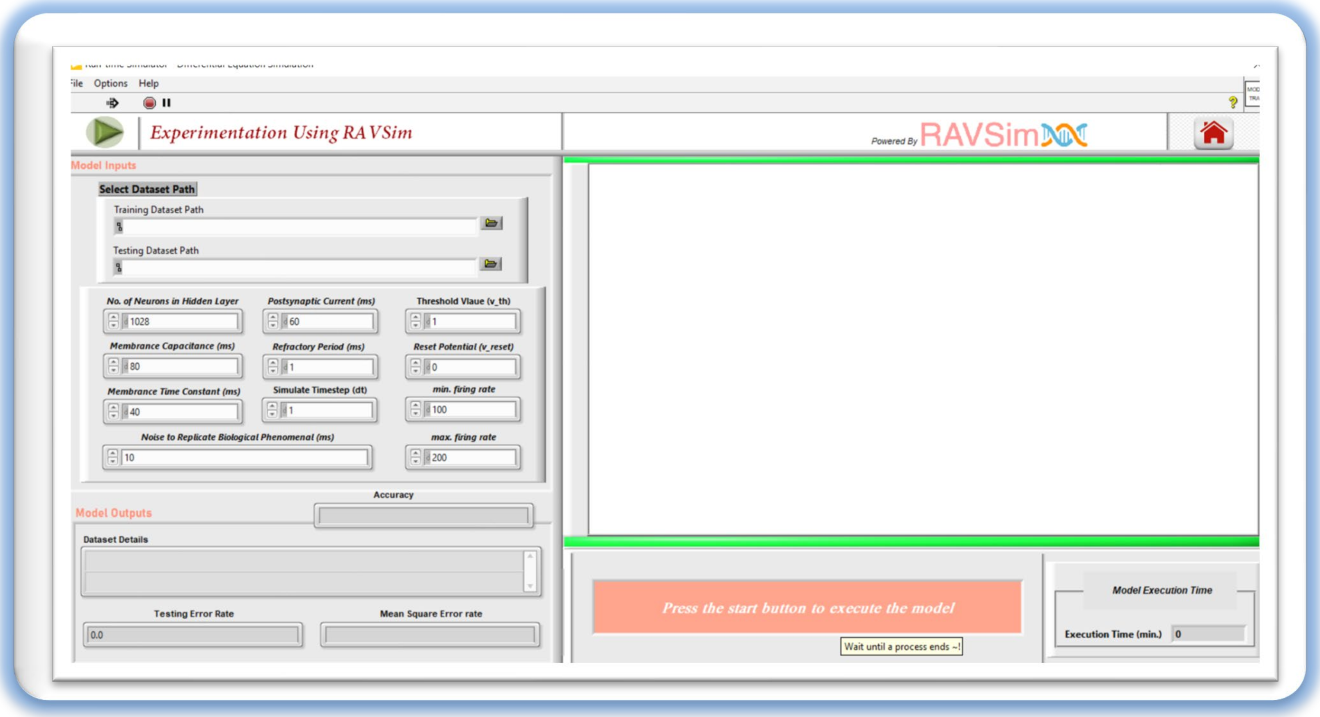


Figure 9: Image classification VI of RAVSim

Upon completion of the analysis, the system will present the user with a comprehensive set of results and details for their review. These details include:

- Model Accuracy: This is visualized through plot representations, providing an in-depth view of how well the model performs in the image classification task.
- Dataset Details: Users can access detailed information about the dataset used for the analysis. This information can be invaluable for understanding the context and characteristics of the data.
- Testing Error Rate: The analysis also reveals the testing error rate, calculated using Mean Squared Error (MSE). This metric offers insights into the accuracy of the model's predictions.
- Model Execution Time: The time it takes for the model to complete the image classification task is also reported. This can be crucial for assessing the efficiency of the model in practical applications.

Randomly generated sample results showcasing these details are available in Figure 10. These results provide a visual representation of the model's performance and the impact of the chosen dataset on accuracy and execution time. This comprehensive feedback empowers users to make informed decisions and refine their image classification models.

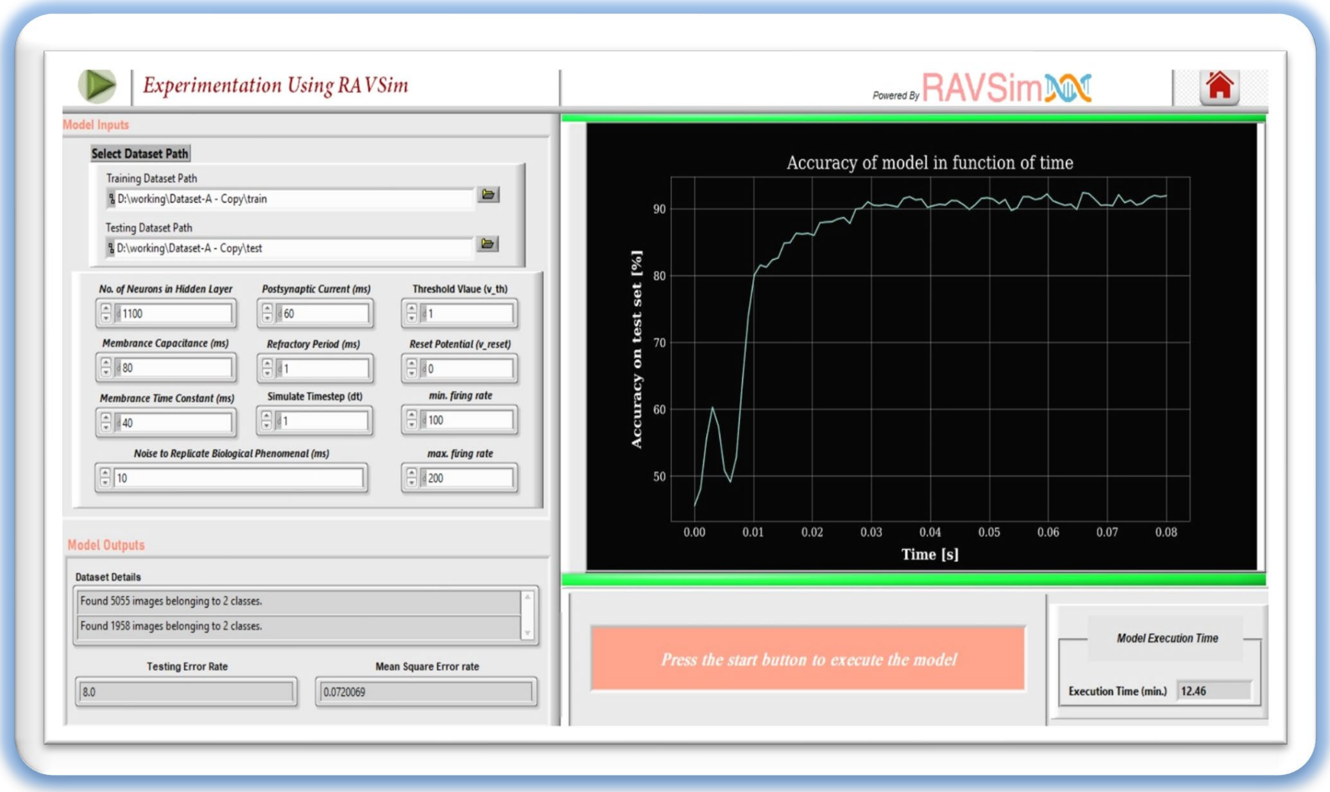


Figure 10: Image classification VI after analysis

Users have the flexibility to simulate the image classification model with various datasets, specifically those containing RGB images. However, it's essential to note that the default parameter values provided in the tool are optimized for a specific dataset mentioned in the research paper [5, 6]. When users need to use a dataset other than the default, users may need to fine-tune the model's parameters to achieve optimal performance. This adaptability enables researchers and practitioners to apply the image classification model to a wide range of tasks, ensuring that it can effectively handle diverse datasets and real-world applications.

3.1. Weight Generation Feature:

RAVSim introduces an innovative feature that simplifies the process of weight generation for spiking neural network models, as depicted in Figure 11. This enhancement empowers users to effortlessly create the necessary weights without the need for intricate coding.

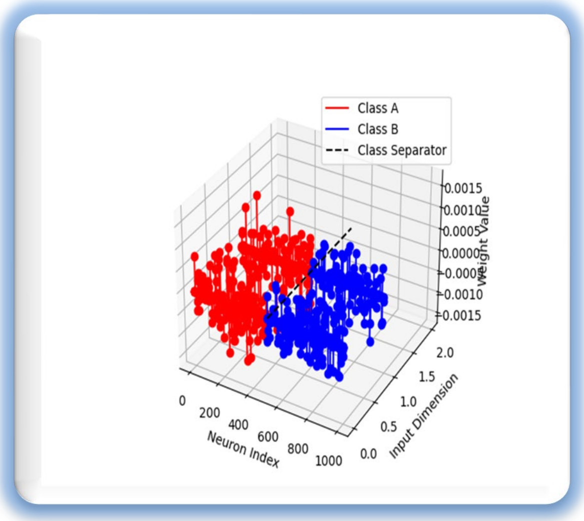


Figure 11: RAVSim the weight generation feature.

Figure 11, illustrate RAVSim's commitment to enhancing user experiences and increasing the efficiency of weight generation for SNN models. This feature promotes accessibility and flexibility, making it a valuable asset for users working with neural network models.

4. Preprocess Custom Dataset VI:

When the "Create Dataset VI" button is activated, this action triggers the appearance of a dedicated virtual instrument, as shown in Figure 11. This virtual instrument serves as the entry point to dataset preprocessing VI, offering users an adaptable platform for preprocessing custom datasets. Within dataset preprocess VI, users are allowed to use downloaded images for creating a scalable dataset. This flexibility includes the ability to use images with varying extensions, sizes, and qualities, pixel to the user's specific requirements. The tool supports a wide variety of image formats, accommodating datasets that suit diverse research and application needs. Users can thus construct datasets that are both representative of their target domain and capable of effectively training and testing image classification models.

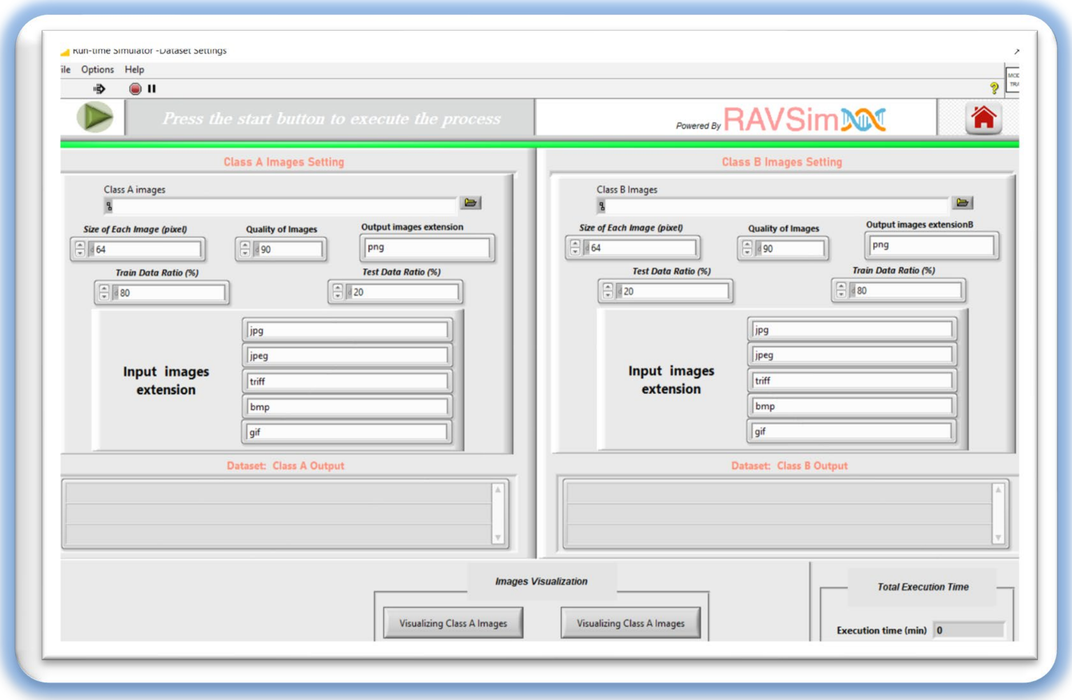


Figure 12: Dataset Creation VI of RAVSim

RAVSim also includes a friendly feature that enables users to preview images both before and after creating a dataset. This feature is designed to ensure that users can visually assess the content and quality of the images without the need to open each one individually. This time-saving and user-friendly feature provides a quick and comprehensive overview of the images, allowing users to verify their selection and make any necessary adjustments before finalizing the dataset preprocessing process.

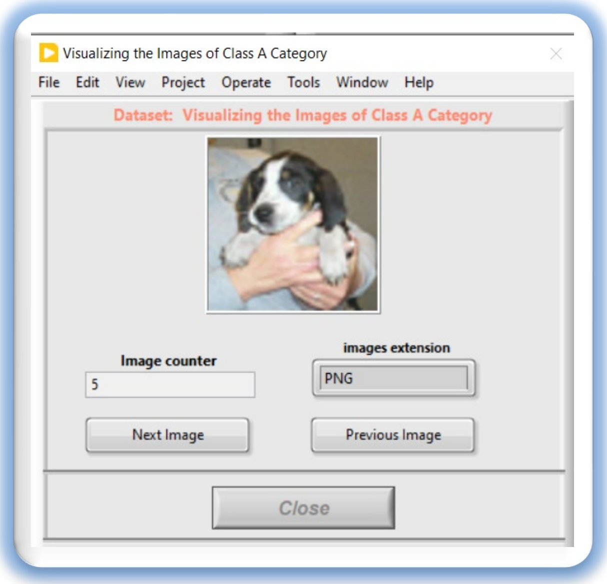


Figure 13: Image Visualization of Dataset Creation VI

5. Comparative Model Analysis Feature VI:

RAVSim presents an advanced feature, as shown in Figure 13, allowing users to conduct a comprehensive comparative analysis of various SNN models [7]. This feature significantly enhances the understanding of model performance and provides assistance in the selection of the most suitable model for specific SNN applications.

Key Features and Benefits:

- **Dynamic Parameter Updates:** RAVSim empowers users to dynamically adjust sets of parameter values for each spiking neural network model during runtime. This real-time flexibility provides users with the means to fine-tune models according to evolving requirements.
- **Comparative Analysis:** The tool facilitates side-by-side comparisons of multiple SNN models, enabling users to assess their performance and behavior. Users can evaluate how changes in parameter values impact model outcomes and efficiency.
- **Performance Evaluation:** Users can gain valuable information about which model best aligns with their specific SNN application. By comparing and contrasting the models, it becomes easier to identify which one offers superior performance or accuracy.
- **Tailored Applications:** This feature accommodates a wide range of applications, from research and experimentation to real-world deployments. Users can fine-tune models to meet the unique demands of their projects, leading to more efficient and effective neural network implementations.

Once the user clicks on the "Neural Model Comparison" button, a user-friendly pop-up window is promptly presented, as shown in Figure 14. Within this window, the user is given the flexibility to make a choice regarding the type of model comparison they wish to perform. The options available are as follows:

- **One-to-One Model Comparison:** By selecting this option, the user can conduct a detailed comparative analysis between two specific neural models. This mode is particularly valuable when pinpointing subtle differences and performance variations between individual two models is essential.
- **All Models Comparison:** Alternatively, the user can opt for the "All Models Comparison" mode, which streamlines the process of comparing all available neural models simultaneously. This feature is especially useful when seeking an overall assessment of model performance across the entire set, making it an efficient choice for broader evaluations and quick insights.

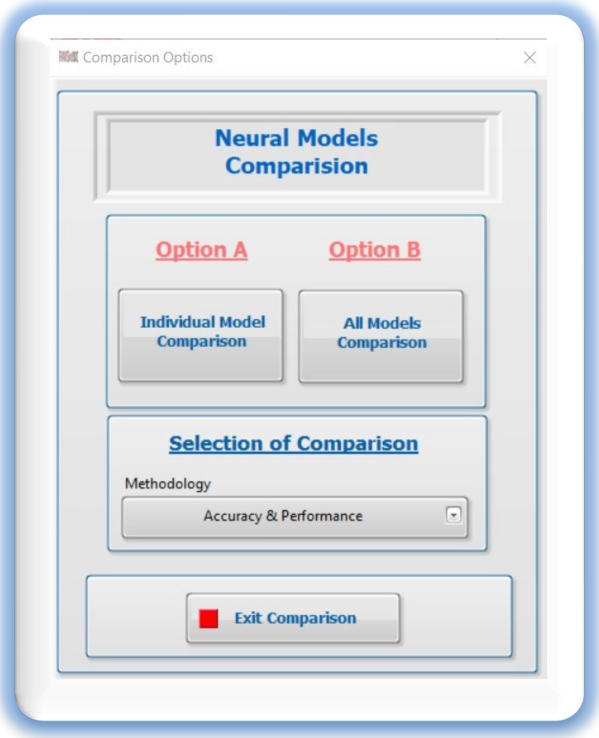


Figure 14: Neural Model Comparison options VI

Once the user clicks on the "One-by-One Model Comparison" option, a pop-up window appears, enabling the user to select two different models for the purpose of comparing their performance loss, accuracy, and error rates.

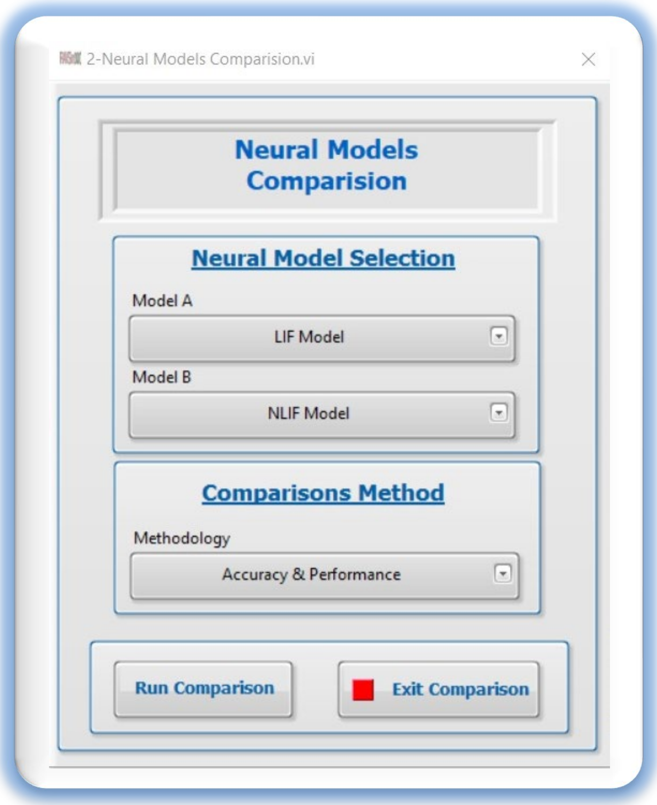


Figure 15: One-by-One Model Comparison VI

After clicking the "Run Comparison" button in Figure 15 VI, the user enters the "Two-Model Comparison VI." Within this VI, the user has the capability to update the parameter values for each model, perform comparisons using different parameter sets, and visualize the results through both text-based data and two distinct graphical plots.

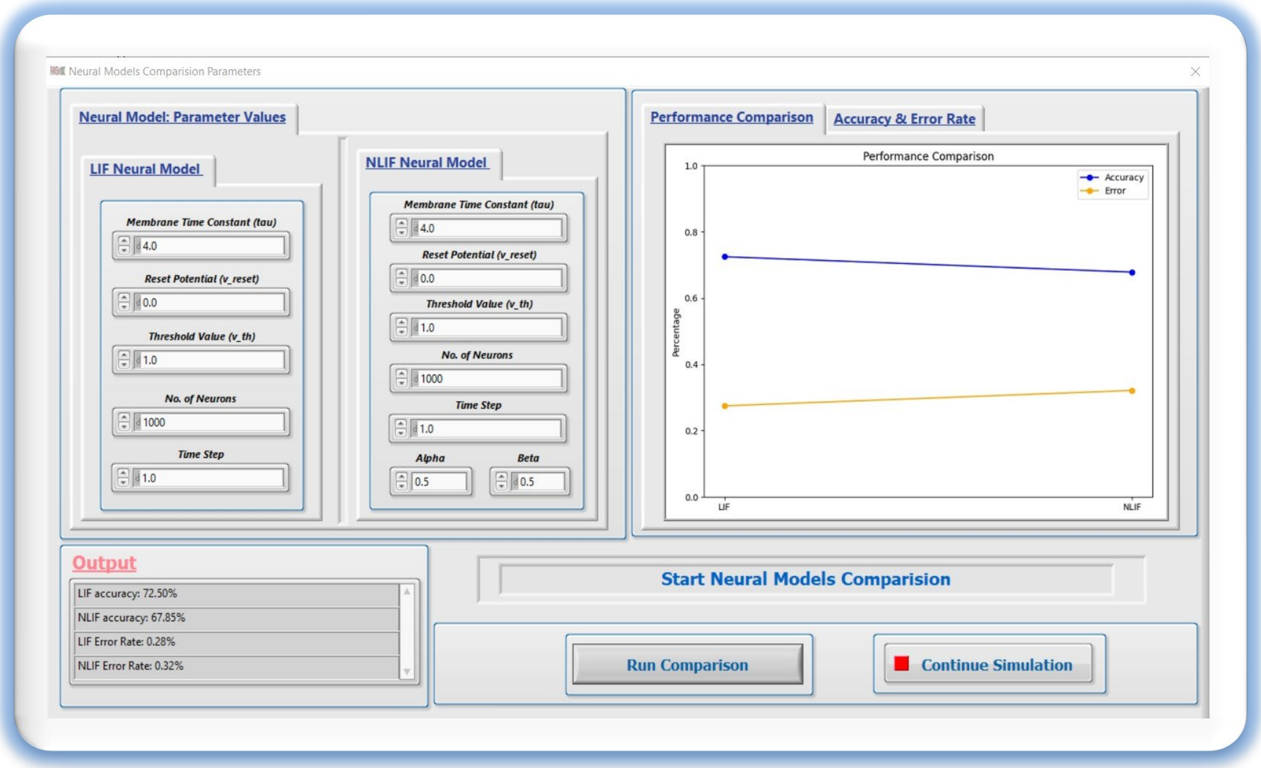


Figure 16: One-by-One Model Comparison visualization VI

Upon clicking the "All Comparison" button, the user gains access to a diverse array of four distinct comparison options, each catering to specific facets of model analysis:

- **Accuracy and Performance:** This mode allows the user to study the accuracy and overall performance of each neural model, offering valuable information about how well each model performs in its classification task.
- **Spiking Activity:** Users can select this option to go through the spiking activity of the models, providing a detailed view of how neurons within each model fire and communicate in response to stimuli.
- **Computational Complexity:** This mode offers a comprehensive analysis of the computational complexity of the neural models, shedding light on the resource demands and efficiency of each model.
- **Biological Plausibility:** For users interested in the biological relevance of the models, this mode explores the extent to which the models mimic real-world neural behavior, providing valuable information about their potential applicability in biological contexts.

Once the user selects their desired comparison option, they are seamlessly directed to the "All Model Comparison VI." Within this VI, users have the flexibility to fine-tune the parameter values for different sets of models, allowing for customized comparisons. The results of these comparisons are presented both in a text format for detailed analysis and in two distinct graphical plots for a visual representation of the findings. This multifaceted approach empowers users to comprehensively assess and refine their neural models based on their specific research or application requirements.

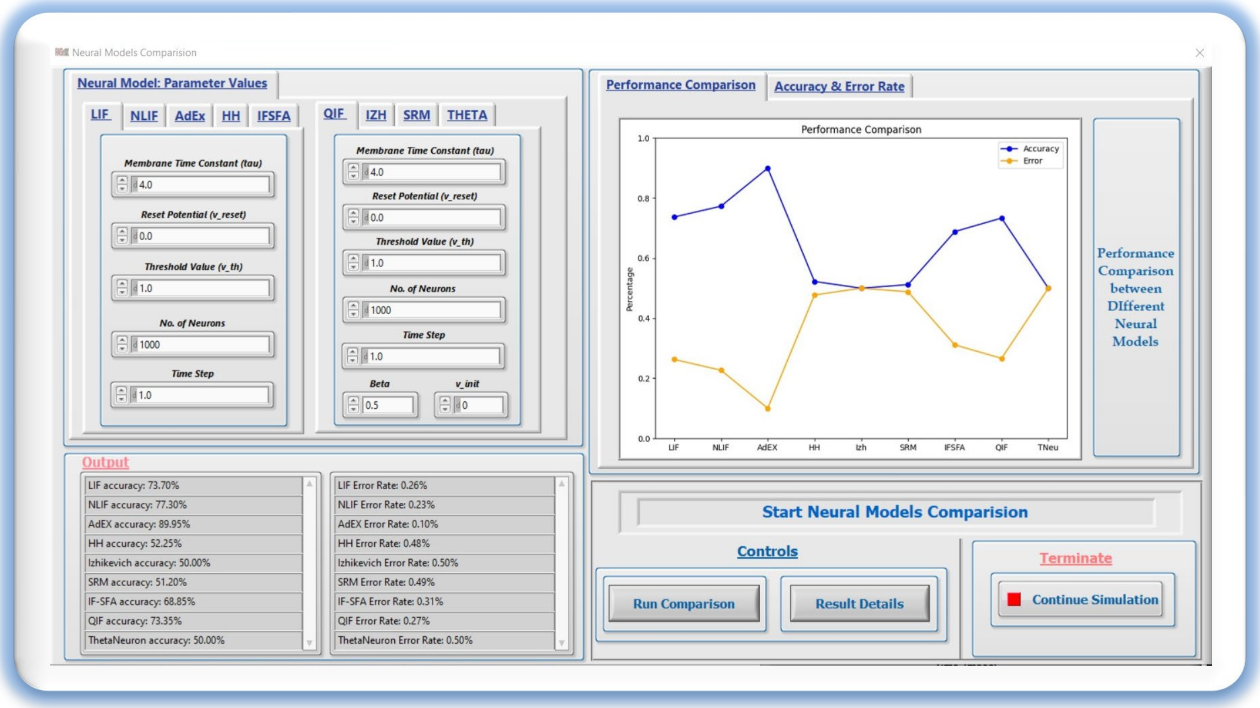


Figure 17: All Model Performance and Accuracy Comparison Visualization VI

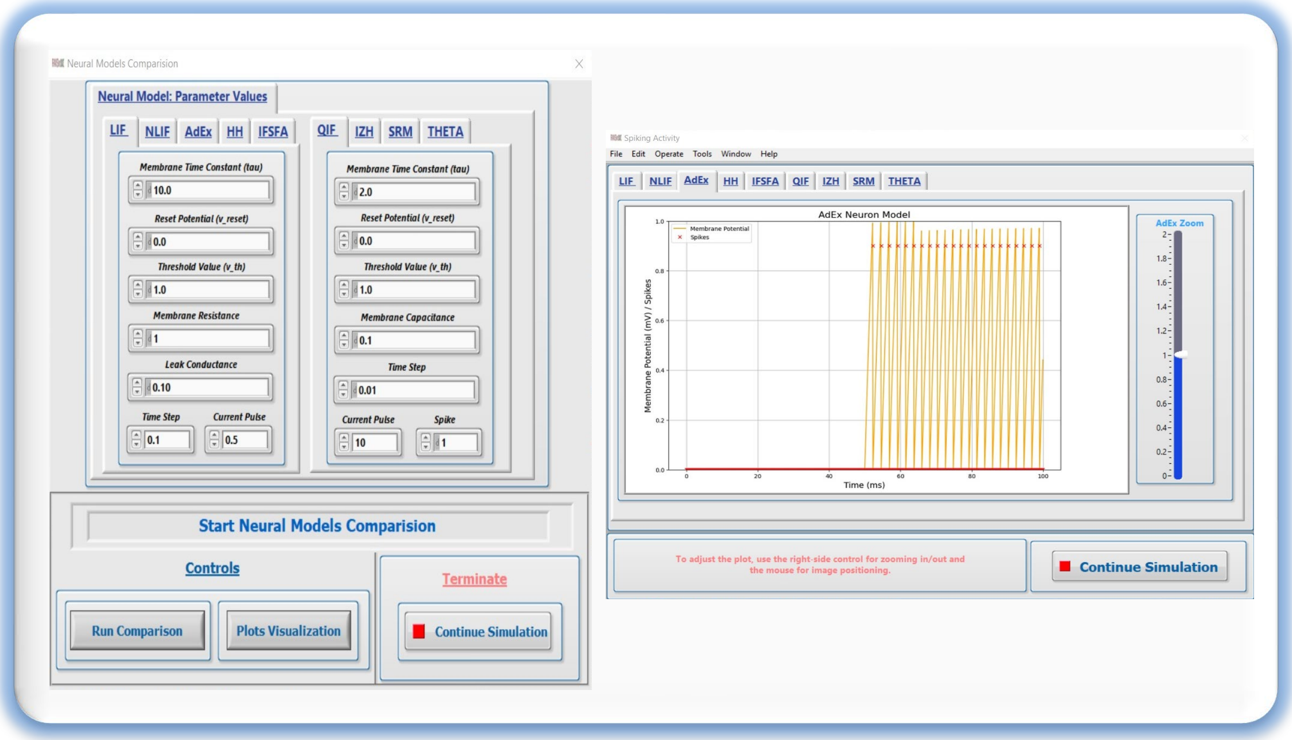


Figure 18: Visualization and Result VI for Spiking Activity Comparison of All Models

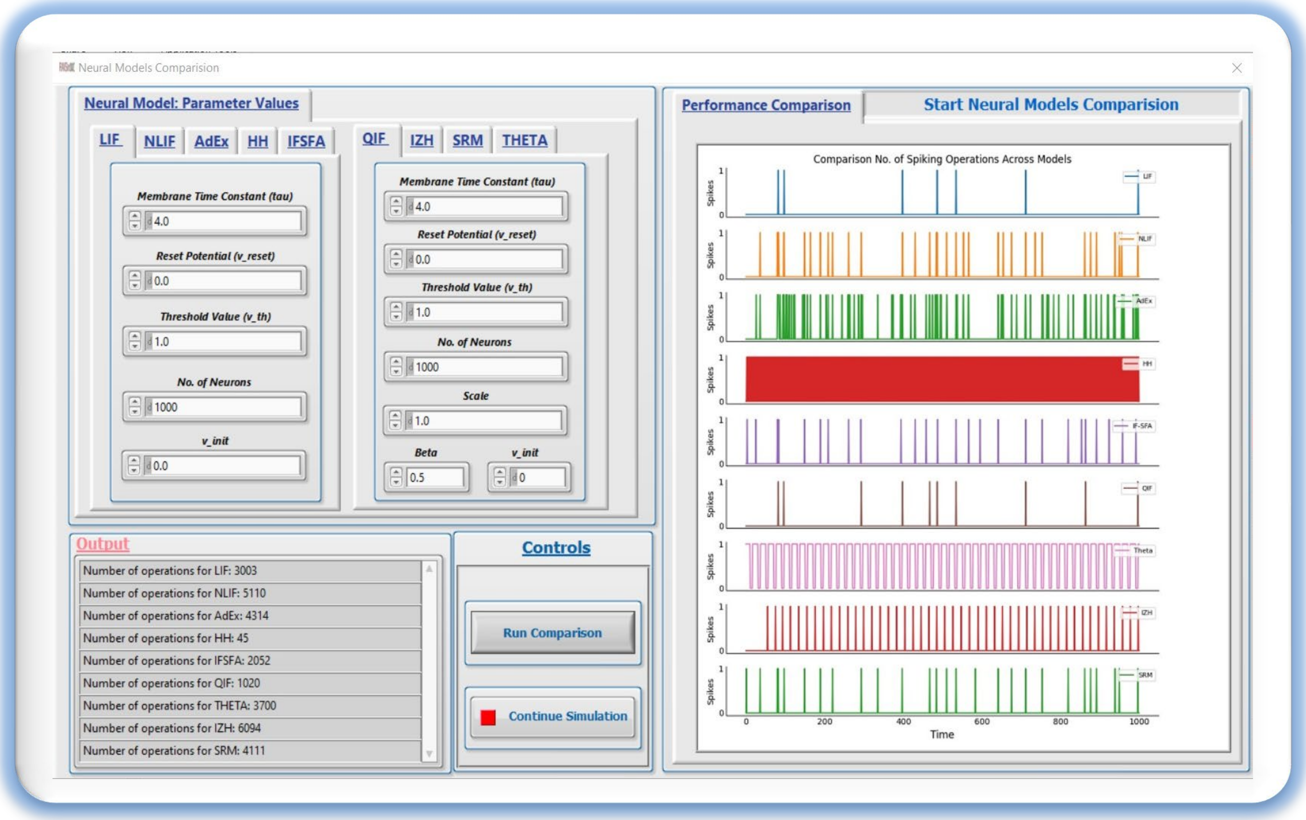


Figure 19: Visualization and Result VI for Computational Complexity Comparison of All Models

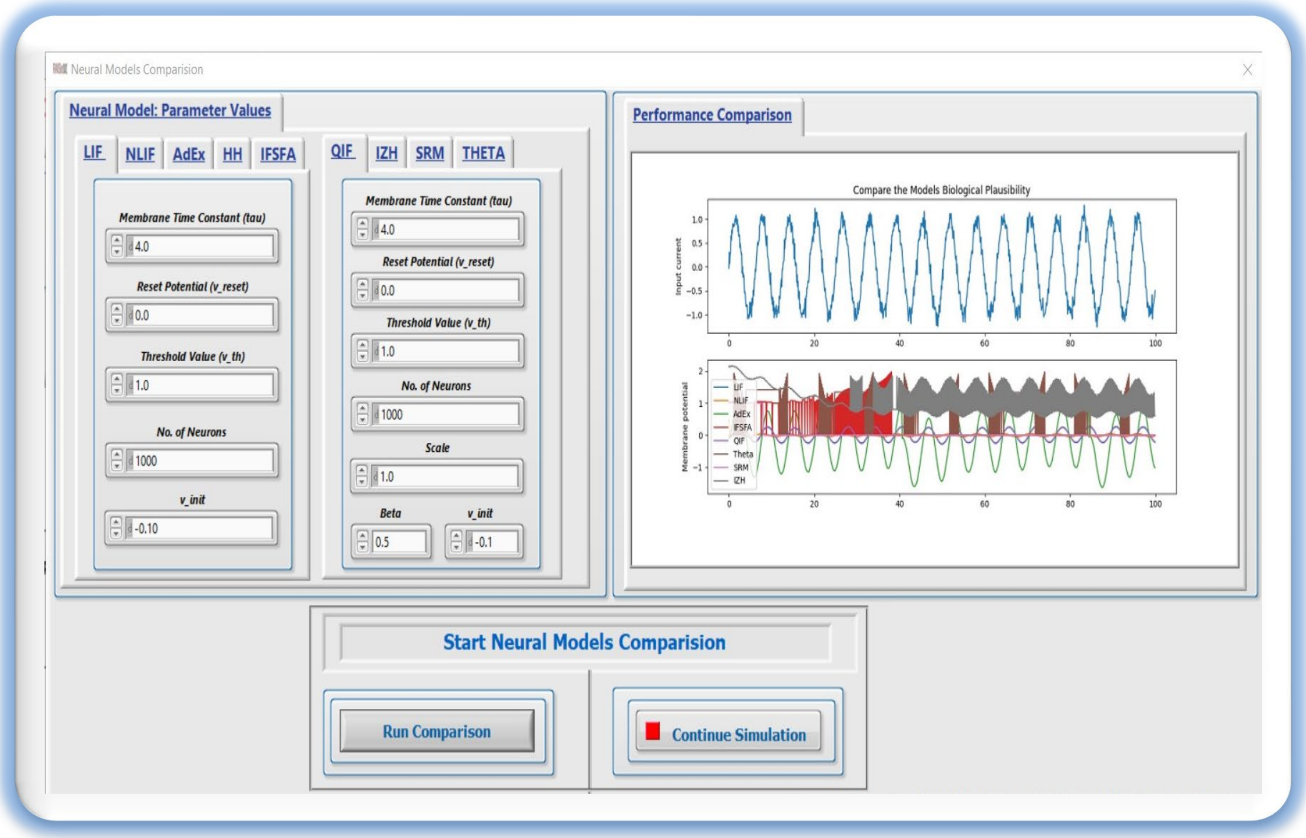
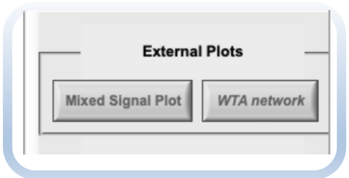


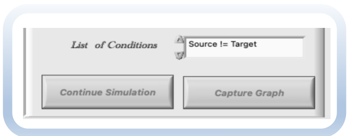
Figure 20: Visualization and Result VI for Biological Plausibility Comparison of All Models

6. Additional features in RAVSim

If a **STOP** button (icon located at the top right corner in Figure 7.) is pressed, “RAVSim VI” navigates the user back to the home screen.



In Figure 7, the bottom right we have provided the selection of the external plots, mixed-signal plots, and WTA network for the ease of the user and also change the runtime simulation values. With the mixed-signal plot, one can visualize the graphs of the input currents and the continuous signals values which are depicted in figure 8. It also allows the user to capture the graphs for documentation or reporting purposes which will be saved in the relative path of the simulator. In order to go back to the run-time environment, the user needs to click on the “Continue Simulation” button.



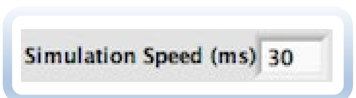
And for navigating to the network connectivity mechanism the user must click on the “WTA Network” in external plots and a window will pop up, where we can visualize the neuron communication graph. Two types of communication graphs can be analyzed at this moment with two different conditions. The first one is the source index equal to the target index and the second one is the source index of the neuron is not equal to the target index. We can select the required graph from the “list of conditions” dropdown. We also capture the graphs or we can continue back to the simulation.

Each screenshot is saved at the location, where “Documents/Screenshot/Screenshot_name”.

Alert – When the **STOP** button is pressed, the user navigates to the home page and the stop button on the home page can be used to terminate the RAVSim tool immediately.

Tip – The functionality of graphical plots, including simulation colors, can be changed by using a left-click on the plot legend and selecting the desired drip-down option.

The runtime simulation option, available in RAVSim v1.1, is shown at the bottom right corner of a VI in Figure 7. The option is described below:



- **Simulation Speed**: This option allows the user to increase or decrease the simulation speed by entering the numerical values which are considered as milliseconds values.

Alert – Make sure the triggered threshold value is always greater than the membrane reset potential.

References

[1]. NI LabVIEW, https://www.ni.com/de-de/shop/labview.html.

[2]. Gardner, B., Gr¨uning, A.: Supervised learning in spiking neural networks for precise temporal encoding. PloS one 11(8), e0161335 (2016)

[3]. Morro, A., Canals, V., Oliver, A., Alomar, M.L., Gal´an-Prado, F., Ballester, P.J., Rossell´o, J.L.: A stochastic spiking neural network for virtual screening. IEEE transactions on neural networks and learning systems 29(4), 1371–1375 (2017)

[4]. Gr gory Dumont, Jacques Henry, and Carmen Oana Tarniceriu. Noisy threshold in neuronal models: connections with the noisy leaky integrate-and-fire model. Journal of mathematical biology, 73(6):1413–1436, 2016.

[5]. Sanaullah, Koravuna, S., Rückert, U., & Jungeblut, T. (2022, June). SNNs model analyzing and visualizing experimentation using RAVSim. In *International conference on engineering applications of neural networks* (pp. 40-51). Cham: Springer International Publishing

[6]. Sanaullah, Koravuna, S., Rückert, U., & Jungeblut, T. (2023). Evaluation of Spiking Neural Nets-Based Image Classification Using the Runtime Simulator RAVSim. *International Journal of Neural Systems*, 2350044-2350044.

[7]. Sanaullah, Koravuna, Shamini, Ulrich Rückert, and Thorsten Jungeblut. "Exploring spiking neural networks: a comprehensive analysis of mathematical models and applications." *Frontiers in Computational Neuroscience* 17 (2023).