

# Development of a Neuromorphic-Friendly Spiking Neural Network for RF Event-based Classification

Michael J. Smith  
US Air Force Institute of  
Technology (AFIT)  
Wright-Patterson AFB OH  
[michael.smith2@afit.edu](mailto:michael.smith2@afit.edu)

Michael A. Temple  
US Air Force Institute of  
Technology (AFIT)  
Wright-Patterson AFB OH  
[michael.temple@afit.edu](mailto:michael.temple@afit.edu)

James Dean  
US Air Force Institute of  
Technology (AFIT)  
Wright-Patterson AFB OH  
[james.dean@afit.edu](mailto:james.dean@afit.edu)

## Abstract

*This paper provides details for the most recent step taken in RndF-to-CNN-to-SNN classifier transition activity supporting an envisioned RF “event radio” concept. Successful results here include the transition from CNNs to neuromorphic-friendly CNN-derived SNNs and pique sufficient interest for pursuing next-step hardware demonstrations. Consistent with earlier RndF and CNN works that used the same experimentally collected WirelessHART signals, SNN results here show that two-dimensional event-based fingerprinting is best overall using events detected in burst Gabor transform responses. The approximate  $\%C_{\Delta} \approx -2\%$  decrease in average percent correct classification performance resulting from RF eventization encoding is effectively offset by a complementary  $\%C_{\Delta} \approx +2\%$  to  $+3\%$  increase that occurs with the CNN-to-SNN transition. This level of neuromorphic-friendly SNN performance is promising when considering the potential 10X-100X energy efficiencies that remain to be demonstrated.*

**Keywords:** RF fingerprinting, neuromorphic processing, spiking neural network, edge processing, WirelessHART.

## 1. Introduction

RF communication networks that use edge computing stand to gain considerable benefit as more energy efficient neuromorphic algorithms and hardware emerge. The anticipated benefits include decreased latency (accelerated inference) at lower energy consumption levels (Joules-per-inference), which are essential for performing reliable and timely device identity (ID) verification. Edge device ID verification and authentication is “an important but often overlooked aspect of sensing” (Jian, et al., 2022) and RF fingerprinting is among the useful options to achieve reliable device authentication at the network physical (PHY) layer (Soltanieh, et al., 2020).

However, PHY-based RF fingerprinting in edge devices poses certain challenges given the processing power and memory storage constraints in size-constrained edge devices. Thus, achieving efficient RF fingerprinting in edge devices requires designers to consider the need to achieve sufficiently high classification accuracy within device hardware constraints (Jian, et al., 2022).

In this work, a neuromorphic-friendly data transformation developed in (Smith, et al., 2024a) is paired with a neuromorphic-friendly Spiking Neural Network (SNN) classifier founded upon a prior Convolutional Neural Network (CNN) methodology presented in (Smith, et al., 2024b). A proven CNN-to-SNN process is used for the conversion (Voelker, et al., 2021), providing consistency in expected average correct classification (%C) performance pre- and post- conversion and maintaining a fair performance and energy use comparison to earlier Random Forest (RndF) (Smith, et al., 2024a) and CNN (Smith, et al., 2024b) classifiers developed using the WirelessHART dataset. Initial accuracy and latency performance of the SNN is presented using the Nengo SNN simulation framework (Bekolay, et al., 2014).

The demonstration dataset is based upon the signals collected in (Maier, et al., 2023) using RF bursts from  $N_{Dev} = 8$  WirelessHART devices. 1D and 2D Gabor transforms are considered, which undergo a first-derivative operation followed by a threshold-based eventization process. The %C performance is considered for both eventized and non-eventized datasets. As in (Smith, et al., 2024b), %C performance results show that 2D fingerprinting is best overall using detected events in burst Gabor transform responses. The  $\%C_{\Delta} \approx -2\%$  to  $\%C_{\Delta} \approx -3\%$  decrease in classification performance resulting from RF eventization encoding is effectively offset by a near-equivalent increase that occurs with the CNN-to-SNN transition. Further, it is shown that 2D classifiers have far less inference latency than 1D alternatives; for results here the 2DSNN architectures result in an

average inference latency decrease of approximately 61% when compared with the 1D SNN architectures.

### 1.1. Research motivation

The graphic in Figure 1 effectively embodies the research motivation that is based on prior [P], recent [R] and future [F] activity supporting RF event radio development and demonstration. Although not all inclusive, the list of noted activities in Figure 1 address why this research is important and includes:

[P1] Consideration of neuromorphic processing benefits that have been realized in event-based camera systems—these benefits are likewise desirable for RF-based systems and include very high temporal resolution, low latency, very high dynamic range, and low power consumption (Gallego, et al., 2022; Eliasmith & Anderson, 2004; Davies, et al., 2018);

[P2] Consideration of SNNs that have gained the attention of machine learning (ML) and artificial intelligence (AI) communities (Javanshir, et al., 2022)—the desire exists for RF-based hardware-compatible models to perform real-time inference using much less energy than traditional ANN and CNN implementations (Mastella & Chicca, 2021; Acharya, et al., 2022);

[R1] Introduction of the RF “event radio” concept and device discrimination using fingerprints derived from RF eventized signals (Smith, et al., 2024a)—fingerprints based on 2D Gabor Transform (GTX) responses of WirelessHART signals were superior using a RndF classifier;

[R2] Completion of an interim RndF-to-CNN classifier transition as a next step to obtain SNN capability (Smith, et al., 2024b)—the marginal impact of RF eventization on CNN classification accuracy was most promising using GTX-Derivative (GTX-DER) fingerprint features;

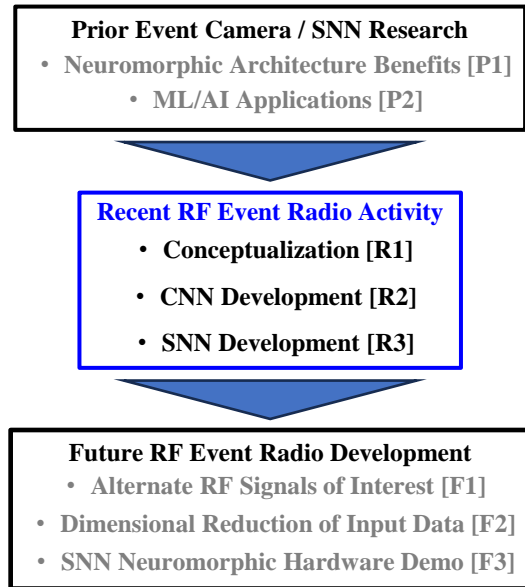
[R3] Completion of the final CNN-to-SNN classifier transition and demonstration in this paper—an important step toward realizing a neuromorphic-friendly SNN capability to achieve brain-inspired neuromorphic processing goals (Rathi, et al., 2023) and adapting event-based camera processing for RF event radio datasets;

[F1] Replication of [R2]-[R3] for alternate signals of interest—important for demonstrating the extensibility of RF event radio methods into application spaces beyond edge communications;

[F2] Reduction in the number of eventized fingerprint features required to achieve a given

level of SNN classification performance—fingerprint dimensional reduction corresponds to a direct reduction in required computational resources and/or operations; and

[F3] Transition to hardware demonstration using a neuromorphic SNN architecture—compare energy consumption and latency metrics with other prominent edge-AI compute architectures.



**Figure 1. Relationship of recent [R] RF event radio development and demonstration activity to prior [P] related and future [F] envisioned activity.**

### 1.2. Relationship to prior research

This work expands prior CNN development work in (Smith, et al., 2024b) by considering the transition to a neuromorphic-friendly SNN classifier using eventized data. The concept of an “event” occurring within time-series data has been studied at the signal waveform level as well as the machine-processing level (Zhu, et al., 2014; Guralnik & Srivastava, 1999; Benko, et al., 2022). Various methods can be used to implement eventization encoding on time series data, and are broadly categorized as rate coding and temporal coding methods (Auge, et al., 2021). For a given application, a preferred encoding method may be empirically selected based on information loss (Petro, et al., 2020), classifier accuracy (Yarga, et al., 2022), or some other metric. As achieved in the event camera arena, key advantages of eventization include very high temporal resolution, low latency, very high dynamic range, and low power consumption (Gallego,

et al., 2022). It is desired to achieve similar advantages in neuromorphic RF systems.

A threshold-based encoding method was used to eventize RF burst responses given its successful application in event-based classification of Free Spoken Digit (FSD) audio datasets in (Forno, et al., 2022). Eventization of RF signals is still largely uninvestigated, and the audio samples used in (Forno, et al., 2022) are perhaps the closest alternative signal type to the WirelessHART Frequency Hopped Time Division Multiplexing (FH-TDMA) signals used here.

In this work we start with an existing trained CNN classifier and use an established process to perform CNN-to-SNN conversion (Diehl, et al., 2015).

### 1.3. Paper organization

The remainder of this paper is organized as follows. The Demonstration Methodology is presented in Section 2 and includes selected details on WirelessHART signals, RF signal eventization encoding, and fingerprint formation. Details for the CNN-derived, one-dimensional (1D) and two-dimensional (2D) SNN classifiers are presented in Section 3. The CNN-to-SNN conversion process is described in Section 4. The device classification results are presented in Section 5. The paper ends with a research summary and conclusions in Section 6.

## 2. Demonstration methodology

### 2.1. WirelessHART signals

The WirelessHART signals were collected using the experimental hardware shown in Figure 2 and combined using the frequency hopped, time division multiple access (FH-TDMA) process detailed in Figure 3 (Maier, et al., 2023). The FH-TDMA processing induced Cross Channel Interference (CCI) effects that are important to consider given they 1) exist during normal network operation and, 2) significantly increase the device fingerprint discrimination challenge when compared with non-CCI channel conditions.

As shown in Figure 2, the RF signals were collected from  $N_{Dev} = 8$  wireless devices, including four D1-D4 AW210 (Siemens, 2012) and four D5-D8 Bullet (Pepperl+Fuchs, 2015) adapters. The devices operated through an Emerson 1410 gateway (Emerson, 2020) controlled by a laptop computer. The signals were collected using a USRP X310 Software Defined Radio (SDR) (Ettus Research, 2024) tuned to a center frequency of 2.4 GHz. The adapters operated across  $N_{CH} = 8$  WirelessHART channels (IEEE

Standards Association, 2011) within the Industrial, Scientific, and Medical (ISM) band, with each channel occupying 5.0 MHz of bandwidth. The X310 collection receiver employed an  $f_s = 100$  Mega-Samples-per-second (MSps) sample rate and a total of 2,500 bursts were collected per device in each of the selected  $N_{CH} = 8$  WirelessHART channels.

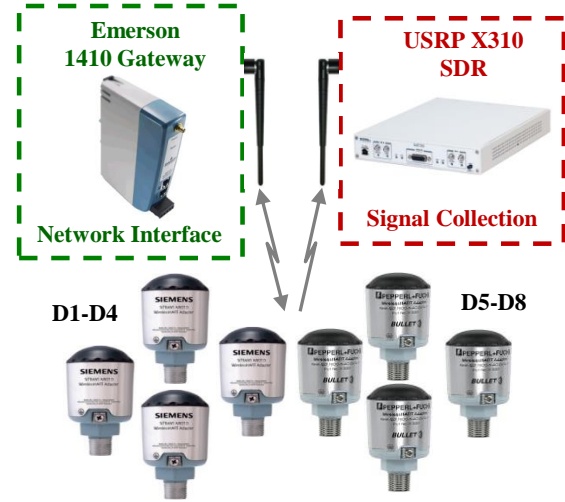


Figure 2. Experimental WirelessHART signal collection hardware.

The FH-TDMA processing in Figure 3 was used to induce CCI signaling effects. As illustrated, each of the  $N_{TS}$  time slots (TS) is jointly occupied by signals from the  $N_{Dev} = 8$  devices operating in a randomly assigned frequency slot (FS). The highlighting of D2 across time is provided to illustrate the FH nature of device operations. The composite FH-TDMA burst responses for each time slot (illustrated in the top of Figure 3) are input to the signal eventization process to form the fingerprints used for device discrimination.

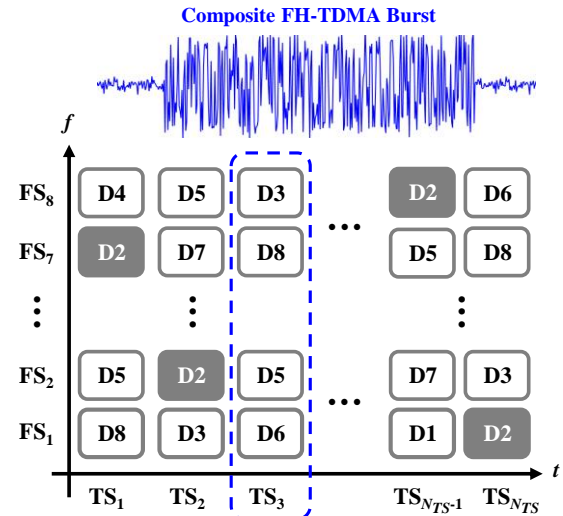


Figure 3. FH-TDMA signal generation.

For a given timeslot in the FH-TDMA processing, the average collected signal ( $S$ ), background noise ( $N_{\text{Bck}}$ ) and CCI interference ( $N_{\text{CCI}}$ ) powers are fixed. The channel variation effects across time slots are assessed using an analysis of Signal-to-Interference-plus-Noise Ratio (SINR) calculated as  $\text{SINR} = S / (N_{\text{Bck}} + N_{\text{CCI}} + N_{\text{AWGN}})$ . The  $N_{\text{AWGN}}$  factor is the average power of Additive White Gaussian (AWGN) that is 1) filtered (bandlimited) across the 5.0 MHz WirelessHART channel bandwidth, 2) power-scaled, and 3) added to the FH-TDMA bursts. Consistent with (Smith, et al., 2024a; Smith, et al., 2024b), the  $N_{\text{AWGN}}$  power was set to achieve  $+6.5 > \text{SINR}(\text{dB}) > +26.5$  for channel variation assessments.

## 2.2. RF signal eventization

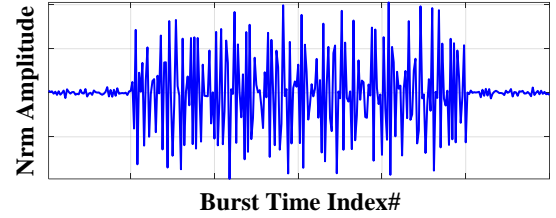
Earlier RF eventization efforts (Smith, et al., 2024a; Smith, et al., 2024b) considered both 1D time domain (1D-TD) and 2D Gabor (2D-GTX) features. The assessments here focused on 2D-GTX features given their general superiority. Selected 2D-GTX details from (Smith, et al., 2024a) are presented for completeness and explained with the aid of Figure 4.

A discrete GTX transform (Qian & Chen, 1993) was used to transform WirelessHART sample sequences  $\{b_{\text{Col}}(n)\}$  from  $N_{\text{Dev}} = 8$  devices into GTX output matrices. The GTX transform was implemented using  $K_{\text{Tim}} = 16$ ,  $M_{\text{Frq}} = 128$ , and  $N_{\Delta} = 64$  time shift parameters with a Gaussian Window (GW) width of  $W_{\text{GW}} = 0.01$ . The row-based (frequency) first-derivative of the resultant GTX matrix was taken and used to generate the GTX-DER event-based fingerprints. The GTX-DER responses were eventized using threshold-based encoding (Forno, et al., 2022) with a variable threshold value applied.

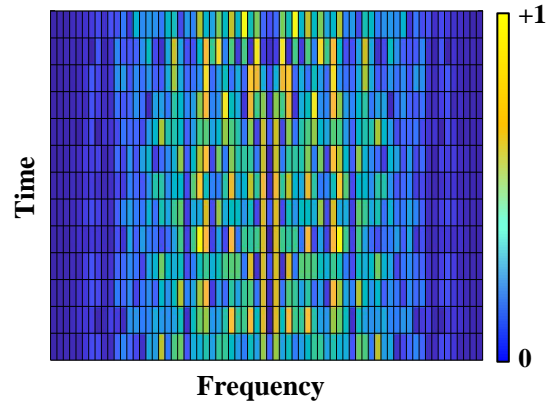
Figure 4 shows the different GTX-DER encoding steps starting with the original burst time domain response in Figure 4a. The 2D-GTX representation of this burst is shown in Figure 4b. After the derivative operation and eventization, elements of the resultant eventized matrix in Figure 4c can be input directly in the 2D classifiers, or be row-wise concatenated into a vector for input to the 1D classifiers.

**2.2.1. GTX-based eventization.** For GTX-DER eventization, a first derivative operation (element-by-element differencing) is first applied to the  $(M_{\text{Tim}} \times K_{\text{Frq}})$ -dimensional GTX matrix. The resultant difference matrix is centered (mean removed) and then normalized using  $-1 \leq \text{abs}[\text{GTX}_{\text{Ctr-Nrm}}^{\text{DER}}(m, k)] \leq +1$  for all  $(m, k)$ .

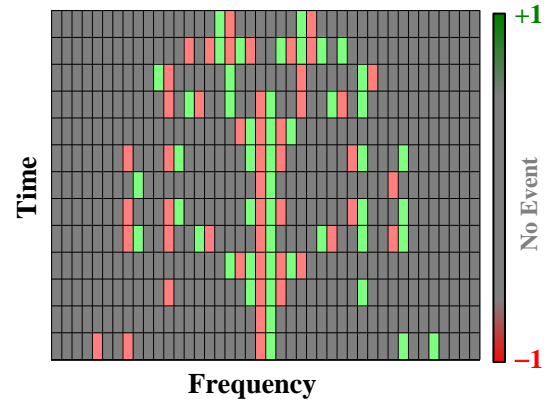
A contrast encoding threshold value of (Thr) is applied element-by-element to the  $\text{GTX}_{\text{Ctr-Nrm}}^{\text{DER}}$  matrix



(a) 1D Time Domain Response.



(b) 2D-GTX Response.



(c) 2D Eventized GTX-DER Response.

**Figure 4. Illustration of 2D-GTX eventization process for a representative burst.**

and detected events placed in corresponding eventized matrix denoted as  $\text{GTX}_{\text{EV}}^{\text{DER}}$ . Event detection polarity is assigned using  $p = \text{sign}[\text{GTX}_{\text{Ctr-Nrm}}^{\text{DER}}(m_p, k_p)] = \pm 1$  where  $(m_p, k_p)$  are  $\text{GTX}_{\text{EV}}^{\text{DER}}$  elements where  $\text{abs}[\text{CtrNrm}(m_p, k_p)] \geq \text{Thr}$ . All other elements in the eventized matrix are assigned a value of 0 indicating that no event has been detected. Thus, the final eventized  $\text{GTX}_{\text{EV}}^{\text{DER}}$  derivative matrix has the same dimension as its non-eventized counterpart.

**2.2.2. GTX-based fingerprint generation.** Two different types of GTX fingerprints were considered, including non-eventized and eventized. The non-eventized GTX fingerprints ( $\vec{F}_{\text{NonEv}}^{\text{GTX}}$ ) were formed as vectorized ROI elements of the  $\text{GTX}_{\text{Ctr-Nrm}}^{\text{DER}}$  matrix before eventization was applied. These fingerprints were formed according to

$$\vec{F}_{\text{NonEv}}^{\text{GTX}} = [e_1 \ e_2 \ \dots \ e_i \ \dots \ e_{N_{\text{ROI}}}]_{1 \times N_{\text{Feat}}} \quad (1)$$

where  $e_i$  are  $\text{GTX}_{\text{Ctr-Nrm}}^{\text{DER}}$  ROI elements for  $i = 1, 2, \dots, N_{\text{Feat}} = N_{\text{ROI}}$  and  $N_{\text{ROI}}$  is the total number of matrix ROI elements.

The eventized  $\vec{F}_{\text{Ev-DEr}}^{\text{GTX}}$  fingerprints were formed using the threshold polarity assignment described above. The  $p(n) \in [-1 \ 0 \ +1]$  polarity assignments were used to form eventized fingerprints according to

$$\vec{F}_{\text{Ev}}^{\text{GTX}} = [p_1 \ p_2 \ \dots \ p_i \ \dots \ p_{N_{\text{ROI}}}]_{1 \times N_{\text{Feat}}} \quad (2)$$

where  $p_i$  correspond to the  $\text{GTX}_{\text{Ev-DEr}}^{\text{DER}}$  eventized ROI elements for  $i = 1, 2, \dots, N_{\text{Feat}} = N_{\text{ROI}}$  and  $N_{\text{ROI}}$  is the total number of matrix ROI elements. Thus, the  $i^{\text{th}}$  fingerprint feature in (2) is assigned as either 1)  $p_i = \pm 1$  for a positive/negative detected event, or 2)  $p_i = 0$  when no event is detected.

### 3. CNN architectures

As used previously, the two-dimensional time-frequency GTX (Bastiaans, 1980; Bastiaans & Geilen, 1996) of a one dimensional sequence of time-series samples effectively represents a grid-like topology that is well-suited for CNN classifiers (Goodfellow, et al., 2017).

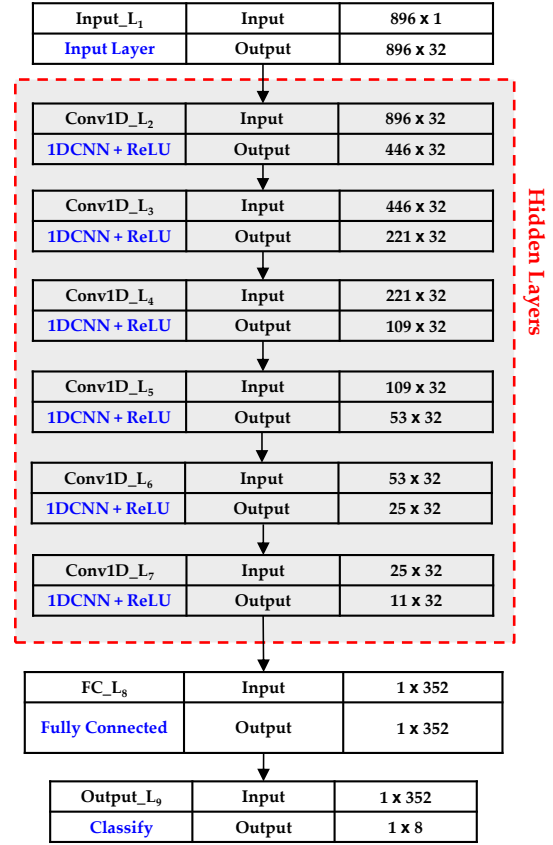
The 1DCNN (Figure 5) and 2DCNN (Figure 6) used here for demonstration were adopted directly from (Smith, et al., 2024b) to ensure consistency with progressive event radio demonstration objectives.

As shown in Figure 5, the 1DCNN input is a (896x1)-dimensional vector of GTX-DEr matrix elements that have been reshaped from the original (16x56)-dimensional matrices. The classifier uses six CNN layers implemented with an identical Rectified Linear Unit (ReLU) activation function given by

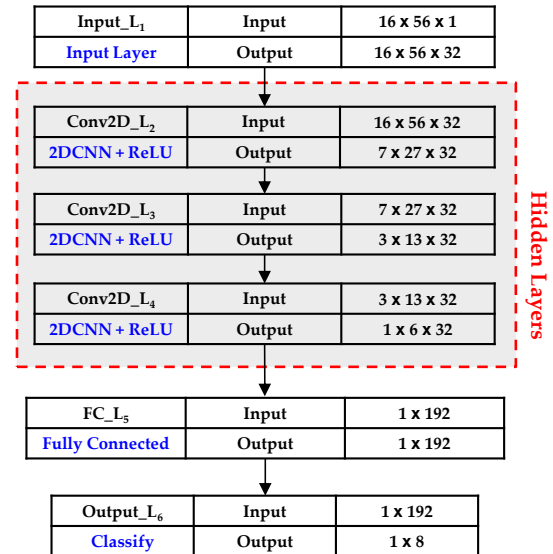
$$f(x) = \max(x, 0) . \quad (3)$$

Convolution within each 1DCNN layer is performed using  $N_{\text{KF}} = 32$  kernel filters of size  $N_{\text{SZ}} = 5$  and a kernel stride of  $N_{\text{KS}} = 2$  elements between convolution operations. As implemented, the 1DCNN includes a total of 28,776 trainable parameters.

The 2DCNN architecture in Figure 6 extends the 1DCNN using the (16x56)-dimensional input GTX matrices,  $N_{\text{KF}} = 32$  (3,3)-dimensional kernel filters



**Figure 5. Architecture used for 1DCNN-to-1DSNN conversion demonstration (Smith, et al., 2024b).**



**Figure 6. Architecture used for 2DCNN-to-2DSNN conversion demonstration (Smith, et al., 2024b).**

and a kernel stride of  $N_{KS} = (2,2)$  between convolution operations. The 2DCNN includes three 2DCNN layers with each layer using the same ReLU activation function given in (1). As implemented, the 2DCNN includes a total of 20,360 trainable parameters.

#### 4. CNN-to-SNN conversion process

A CNN-to-SNN conversion process is used here to develop a neuromorphic-friendly SNN that is suitable for event radio development and demonstration. A CNN is first trained through conventional backpropagation methods using a CPU or GPU. Once the desired classification performance level is achieved, the trained CNN network weights are saved for import into the SNN. The only network component that changes in the SNN architecture is the activation function, replacing the ReLU with the Spiking ReLU (Stanojevic, et al., 2023).

The ‘temporally diffuse quantizer’ conversion algorithm used here (Voelker, et al., 2021) is implemented with the Nengo Python software package (Bekolay, et al., 2014). Nengo uses an additional spike rate scale factor ( $\gamma$ ) which is applied to the activation and raises the spike rate of every neuron in the network. Increasing  $\gamma$  is necessary to stabilize the network output in a reasonable amount of time.

A synaptic delay ( $\tau$ ) is also set to establish characteristics of a low-pass filter that is applied to the SNN Spiking ReLU function output according to

$$h(t) = \tilde{a}_t * (\tau^{-1} e^{-t/\tau}) \quad (4)$$

where the  $*$  symbol denotes time domain convolution,  $h(t)$  is the filtered output, and  $\tilde{a}_t$  is the estimated output of the original activation function  $f(x)$ , as generated from the conversion algorithm.

The CNN-to-SNN conversion process was applied to the 1DCNN and 2DCNN networks using the GTX-DER input dataset. The empirically determined conversion values for  $\gamma$  and  $\tau$  are shown in Table 1 and indicate that larger networks with more layers require a higher  $\gamma$  value.

**Table 1. CNN-to-SNN conversion parameters**

	Spike Rate Scale Factor ( $\gamma$ )	Synaptic Delay ( $\tau$ )
1DCNN	150	0.04
2DCNN	120	0.04

An additional critical parameter that effects the SNN output performance is the number of time steps-per-inference ( $N_{Ts/Inf}$ ).  $N_{Ts/Inf}$  must be large enough to ensure the network has converged sufficiently.

### 5. SNN classifier performance

Classifier performance assessments include 1) classification accuracy versus latency, 2) overall cross-class classification accuracy using an arbitrary benchmark of  $\%C = 90\%$ , and 3) gains and losses resulting from CNN-to-SNN conversion and RF eventization. For all presented results, data marker sizes have been adjusted to encompass the  $CI = 95\%$  confidence intervals (Park & Leemis, 2019) which are the basis for drawing comparative same (data marker CIs overlap) and different (data marker CIs do not overlap) conclusions (Park & Leemis, 2019).

#### 5.1. Classifier latency

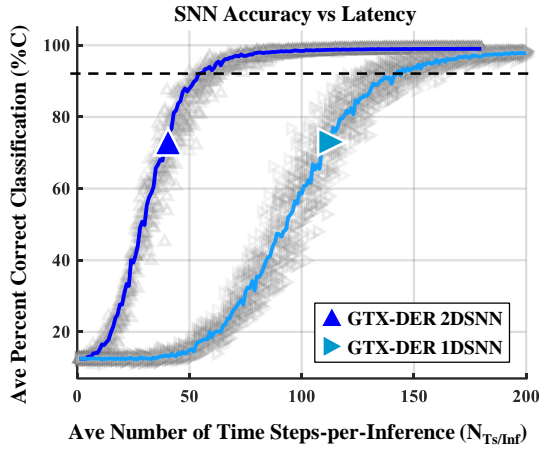
Total energy consumption for a given classifier is directly related to  $N_{Ts/Inf}$ . The classification accuracy ( $\%C$ ) versus latency ( $N_{Ts/Inf}$ ) tradeoff is important for characterizing overall SNN performance (Ngu & Lee, 2022; Wu, et al., 2022) and assessing the Joules-per-inference energy consumption for a given classifier. Assessment of the average  $\%C$  vs.  $N_{Ts/Inf}$  latency relationship is shown for CNN-derived SNNs using non-eventized GTX features—similar assessments can be done using eventized GTX features if desired.

Ten CNN-to-SNN model realizations were run to capture the accuracy versus latency for the GTX-DER dataset using 1DCNN-to-1DSNN and 2DCNN-to-2DSNN classifiers. Figure 7a shows results for the 1D classifier run for  $0 < N_{Ts/Inf} < 200$  and the 2D classifier run for  $0 < N_{Ts/Inf} < 180$ . The average  $\%C$  accuracy across the ten iterations (grayed-out symbols) is shown as a solid line (7,600 total test evaluations across the range of  $N_{Ts/Inf}$ ). For  $\%C = 90\%$  the average  $N_{Ts/Inf}$  for the 2DSNN classifier ( $\blacktriangle$ ) is  $N_{Ts/Inf} \approx 55$  and  $N_{Ts/Inf} \approx 140$  for the 1DSNN classifier ( $\blacktriangleright$ ). This represents an approximate 61% reduction in latency.

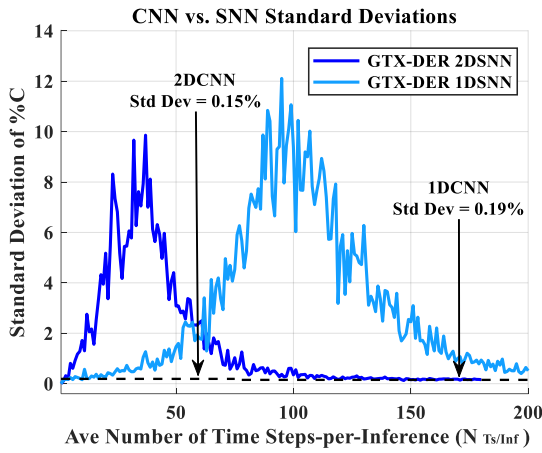
SNN processing may be terminated based on any number of factors. Some termination criteria includes processing until 1) a predetermined number of  $N_{Ts/Inf}$  steps has been completed, 2) a specific  $\%C$  accuracy has been reached, or 3) the  $\%C$  accuracy has reached an asymptotic maximum. In this work we propose using standard deviation as a termination metric.

Figure 7b shows the standard deviation ( $\sigma$ ) about the average  $\%C$  value for the ten model realizations and every  $N_{Ts/Inf}$  value. A summary of minimum ( $\sigma_{SNN}^{Min}$ ) and maximum ( $\sigma_{SNN}^{Max}$ ) standard deviation extremes for the 1DSNN and 2DSNN classifiers in Figure 7 is provided in Table 2. Note that these minimum  $\sigma$  values are considered only for  $N_{Ts/Inf}$  values after the peak standard deviation has occurred;

this ensures the network has had sufficient time to converge on a stable output class estimate.



(a) Individual models and cross-model averages



(b) Cross-model standard deviations of %C

**Figure 7. SNN performance using non-eventized features: (a) Classification accuracy (%C) vs latency ( $N_{Ts/Inf}$ ) for ten model realizations and (b) cross-realization standard deviation for each  $N_{Ts/Inf}$ .**

**Table 2. Standard deviation extremes for CNN-derived SNNs using non-eventized data.**

	CNN-Derived SNN			CNN
	Feature Type	$\sigma_{SNN}^{Min}$	$\sigma_{SNN}^{Max}$	
1D	GTX-DER	0.40%	12.11%	0.19%
2D	GTX-DER	0.13%	9.86%	0.15%

Table 2 values indicate that 1) for the 1DCNN models, the minimum  $\sigma_{SNN}^{Min}$  never achieves the 1DCNN  $\sigma_{CNN}$  within the range of  $0 < N_{Ts/Inf} < 200$  steps for the GTX-DER features. This means that the 1DCNN model may need to be run for  $N_{Ts/Inf} > 200$ , or the 1DSNN model has too much inherent model

variance; and 2) for the 2DSNN, the  $\sigma_{SNN}^{Min} < \sigma_{CNN}$  at  $N_{Ts/Inf} = 147$  using GTX-DER features.

The relationship between CNN standard deviation  $\sigma_{CNN}$  and SNN maximum  $\sigma_{SNN}^{Max}$  standard deviation for a given number of  $N_{Ts/Inf}$  steps may be useful in establishing SNN stopping conditions. Matching CNN accuracy is the ideal outcome in a CNN-to-SNN conversion (Ngu & Lee, 2022; Wu, et al., 2022). However, model variance during deployment is usually considered more important than the accuracy itself, and thus is a good choice for selection of  $N_{Ts/Inf}$ .

## 5.2 CNN vs. SNN classifier performance using eventized data

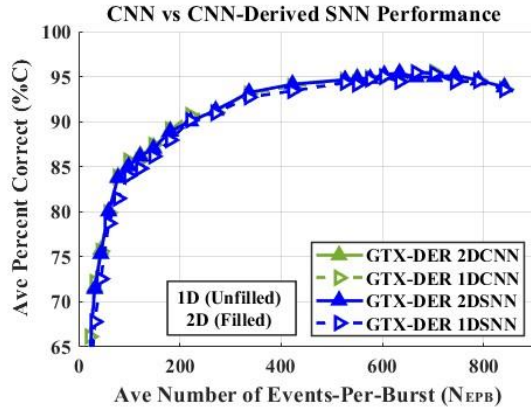
The eventization of RF fingerprints is next considered by looking at the minimum number of average events per burst ( $N_{EPB}$ ), calculated across all bursts and all devices, required to achieve maximum average %C classification performance. This was addressed by varying the eventization threshold for the GTX-DER features and calculating the resultant %C classification performance. Values of  $N_{Ts/Inf} = 180$  and  $N_{Ts/Inf} = 200$  were used for the 2DCNN and 1DCNN classifiers, respectively. The minimum average  $N_{EPB}$  required to achieve the maximum asymptotic %C is shown in Table 3 for both CNNs and the corresponding CNN-derived SNNs.

Results in Table 3 are among those appearing in Figure 8 across the range of average  $N_{EPB}$  obtained by varying the eventization threshold value between  $0.01 < Thr < 0.75$  ( $\Delta Thr = 0.05$ ). The table also shows the threshold value used to achieve the maximum %C with the minimum  $N_{EPB}$ .

Relative to achieving the minimum  $N_{EPB}$  objective, it is expected that CNN and corresponding CNN-derived SNN %C classification performance would be near equivalent (Ngu & Lee, 2022; Wu, et al., 2022). This is apparent in Table 3 and Figure 8 classification performance for  $N_{EPB} \geq 600$  where near statistically equivalent %C is achieved between the 1DCNN and 2DCNN classifiers and their SNNs.

**Table 3. Minimum average number of  $N_{EPB}$  for CNNs and CNN-derived SNNs to achieve maximum %C using eventized data and GTX-DER features.**

Feature Type	Threshold (Thr)	Classifier	Min $N_{EPB}$	Max %C
GTX-DER	0.04	1DCNN	703.0	95.51%
	0.05	SNN	666.8	95.45%
	0.06	2DCNN	634.0	95.37%
	0.06	SNN	634.0	95.42%



**Figure 8.** CNN and CNN-derived SNN classification performance versus average Number of Events Per Burst ( $N_{EPB}$ ) using RF eventized data.

### 5.3. Performance gains and losses

A more detailed analysis of  $N_{EPB} = 600$  results is performed using results in Figures 9 and 10. These figures show CNN-to-SNN classifier conversion gains ( $\%C_{\Delta}^{SNN}$ ) and eventization losses ( $\%C_{\Delta}^{EV}$ ) for the 1D and 2D classifiers. For a given CNN and CNN-derived SNN pairing, the conversion gains are calculated as  $\%C_{\Delta}^{SNN} = \%C_{SNN} - \%C_{CNN}$ . For a given classifier, the RF eventization losses are calculated as  $\%C_{\Delta}^{EV} = \%C_{Non-EV} - \%C_{EV}$ . Results in Figures 9 and 10 are summarized in Table 4. Observations for the 1D classifier gains and losses in Figure 9 include:

1) For non-eventized features (filled markers), the 1DCNN-to-1DSNN conversion gain (loss) is  $\%C_{\Delta}^{SNN} \approx -1.00\%$ . For eventized features (unfilled markers), there is a marginal 1DCNN-to-1DSNN conversion benefit that includes  $\%C_{\Delta}^{SNN} \approx +0.57\%$ .

2) RF eventization losses of  $\%C_{\Delta}^{EV} \approx -4.21\%$  and  $\%C_{\Delta}^{EV} \approx -2.64\%$  for the 1DCNN and 1DSNN classifiers, respectively. The 1DSNN RF eventization loss is approximately 2% lower (better) than its 1DCNN counterpart.

Observations for the 2D classifier gains and losses in Figure 10 include:

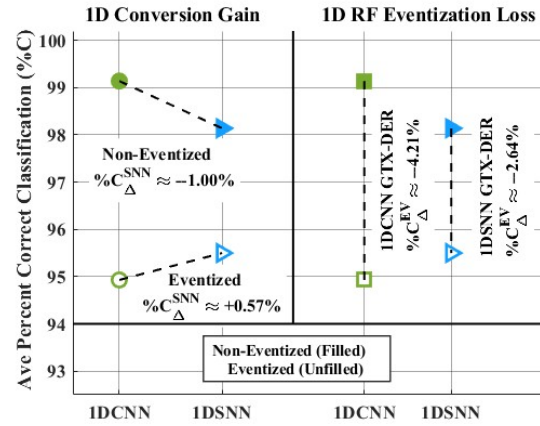
1) For non-eventized features (filled markers) features, there is a marginal 2DCNN-to-2DSNN conversion gain (loss) of  $\%C_{\Delta}^{SNN} \approx -0.25\%$ . For eventized features (unfilled markers), there is a clear conversion benefit of  $\%C_{\Delta}^{SNN} \approx +1.75\%$ .

2) RF eventization losses of  $\%C_{\Delta}^{EV} \approx -4.00\%$  and  $\%C_{\Delta}^{EV} \approx -2.00\%$  for the 2DCNN and 2DSNN classifiers, respectively. The 2DSNN RF eventization loss is approximately 2% lower (better) than its 2DCNN counterpart.

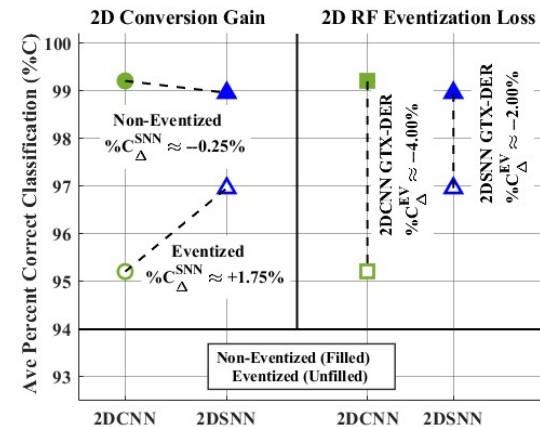
As summarized in Table 4, the  $\%C_{\Delta}^{SNN}$  conversion gains (losses) and  $\%C_{\Delta}^{EV}$  eventization losses for the CNN-derived SNN classifiers are generally superior to their CNN counterparts. It remains to be demonstrated in future work that the marginal SNN performance losses will likely be offset by the energy savings that has been demonstrated for other SNN architectures (Christensen, et al., 2022).

**Table 4.** Summary of CNN-to-SNN conversion gains ( $\%C_{\Delta}^{SNN}$ ) and RF eventization losses ( $\%C_{\Delta}^{EV}$ ) in Figures 9 and 10 for GTX-DER features.

Figure	Classifier	$\%C_{\Delta}^{SNN}$	$\%C_{\Delta}^{EV}$
9	1DCNN	-1.00%	-4.12%
	1DSNN	+0.57%	-2.64%
10	2DCNN	-0.25%	-4.00%
	2DSNN	+1.75%	-2.00%



**Figure 9.** 1DCNN and CNN-derived 1DSNN classification for  $N_{Ts/Inf} = 200$  and  $N_{EPB} = 600$ .



**Figure 10.** 2DCNN and CNN-derived 2DSNN classification for  $N_{Ts/Inf} = 180$  and  $N_{EPB} = 600$ .

## 6. Summary and conclusions

The CNN-to-SNN transition method used here is effective and yields a reliable CNN-derived SNN architecture. This includes average percent correct classification (%C) that is statistically equivalent to its CNN counterpart (Smith, et al., 2024b) using eventization thresholds for GTX responses that yield an average number of Events-Per-Burst of  $N_{EPB} \geq 600$ . Considering the average number of time steps per inference ( $N_{Ts/Inf}$ ) as a measure of latency, it is shown that the 2DCNN-derived SNN results in an approximately 61% lower latency when compared to the 1DCNN-derived SNN. Thus, there is a correspondingly lower amount of consumed energy (Joules-per-inference) for the 2D SNN. It is also shown that the 2D SNN is superior to the 1D SNN for  $N_{EPB} = 600$ , with better %C scores for both non-eventized and eventized data.

Additionally, the CNN-derived SNNs yielded eventized classification performance that is on average  $\%C_{\Delta} \approx +2\%$  to  $+3\%$  higher when compared to their 1DCNN and 2DCNN counterparts using GTX-DER (derivative) features. GTX-DER features achieved  $\%C \geq 90\%$  for  $N_{EPB} \geq 200$  events.

Results here are encouraging and motivate the continued development and demonstration of an RF event radio. Depending on community feedback, there are several next-step research activities that could be pursued, including: 1) repeating CNN activity in (Smith, et al., 2024b) and CNN-to-SNN transition activity here using RF eventized features for alternate signals of interest; 2) pursuing potential benefits of fingerprint compression and event labeling methods to support adoption of event-camera processing for RF event radio processing; and 3) shifting emphasis from algorithm development to hardware demonstration. This is envisioned as including implementation on multiple neuromorphic and non-neuromorphic devices such as Intel's Loihi chip (Davies, et al., 2018), Google Coral USB AI accelerator chip (Coral.ai, 2019), and Intel Neural Compute Stick 2 USB accelerator (Intel, 2019). This will enable comparison of SNN and neuromorphic performance factors such as inference time, energy-per-inference, classification accuracy, and computational load.

## 11. Acknowledgements

The views and conclusions in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S.

Government. This paper is approved for public release, Case Number 88ABW-2024-0504.

## 12. References

- Acharya, J., et al. (2022). Low precision local learning for hardware-friendly neuromorphic visual recognition. *2022 IEEE Int'l Conf on Acoustics, Speech and Signal Processing (ICASSP)*8937-8941.
- Auge, D., et al. (2021). A survey of encoding techniques for signal processing in spiking neural networks. *Neural Processing Letters*, 53(6) 4693-4710.
- Bastiaans, M.J. (1980). Gabor's expansion of a signal into Gaussian elementary signals. *Proc. IEEE*, 68(4), 538-539.
- Bastiaans, M.J. & Geilen, M.C.W. (1996). On the discrete Gabor Transform and the discrete Zak transform. *Signal Processing*, 49,151-166.
- Bekolay, et al. (2014). Nengo: a Python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7.
- Benko Z., et al. (2022). Model-free detection of unique events in time series. *Scientific Reports*, 1(12)(1).
- Christensen, D.V., et al. (2022) 2022 Roadmap on Neuromorphic Computing and Engineering. *Neuromorphic Computing and Engineering*. Retrieved Aug 16, 2024, from <https://iopscience.iop.org/article/10.1088/2634-4386/ac4a83/pdf>.
- Coral.ai (2019). USB Accelerator datasheet. V1.4, G950-01456-01 / G950-06809-01. Retrieved Aug 18, 2024, from <https://coral.ai/static/files/Coral-USB-Accelerator-datasheet.pdf>
- Davies, M., et al. (2018). Loihi: A Neuromorphic Manycore Processor with On-chip Learning. *IEEE Micro*, 38, 82-99.
- Diehl, P., et al. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold Balancing. *Int'l Joint Conf on Neural Networks (IJCNN)*.
- Eliasmith, C. & Anderson, C.H. (2004). Neural engineering: Computation, representation, and dynamics in neurobiological systems. *The MIT Press*.
- Emerson (2020). Emerson Wireless 1410 Gateway. Reference Manual 00809-0200-4410, Rev CA. Retrieved Aug 18, 2024, from <https://www.emerson.com/documents/automation/manual-emerson-smart-wireless-gateway-1410-en-77632.pdf>.
- Ettus Research (2024). USRP™ X300 and X300 X Series. Specification Sheet. Retrieved Aug 18, 2024, from [https://www.ettus.com/wp-content/uploads/2024/01/X300\\_X310\\_Spec\\_Sheet\\_2024-01-23.pdf](https://www.ettus.com/wp-content/uploads/2024/01/X300_X310_Spec_Sheet_2024-01-23.pdf).
- Forno E., et al. (2022). Spike encoding techniques for IoT time-varying signals benchmarked on a neuromorphic classification task. *Front. Neurosci.* 16:999029.
- Gallego, G., et al. (2022). Event-Based Vision: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154-180.

- Goodfellow, I., et al. (2017). *Deep Learning*; MIT Press: Massachusetts.
- Guralnik, V. & Srivastava, J. (1999). Event detection from time series data. Proceedings of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining.
- IEEE Standards Association (2011). Part 15.4: Low-rate Wireless Personal Area Networks (LR-WPANs). IEEE Std 802.15.4™-2011, IEEE, New York, NY, USA.
- Intel (2019). Intel® Neural Compute Stick 2, Product Brief, Retrieved Aug 18, 2024, from <https://cdrdv2-public.intel.com/749742/neural-compute-stick2-product-brief.pdf>.
- Javanshir, A., et al. (2022). Advancements in Algorithms and Neuromorphic Hardware for Spiking Neural Networks. *Neural Computation*, 34:1289-1328.
- Jian, T., et al. (2022). Radio Frequency Fingerprinting on the Edge. *IEEE Trans on Mobile Computing*, 21(11):4078-4093.
- Maier, M.J., et al. (2023). Ensuring the longevity of WirelessHART devices in industrial automation and control systems using distinct native attribute fingerprinting. *Int'l Journal of Critical Infrastructure Protection (IJCIP)*, Elsevier, Vol. 43(C).
- Mastella, M. & Chicca, E. (2021). A Hardware-Friendly Neuromorphic Spiking Neural Network for Frequency Detection and Fine Texture Decoding. *2021 IEEE Int'l Symposium on Circuits and Systems (ISCAS)*.
- Ngu, H.C.V. & Lee, K.M., (2022). Effective Conversion of a Convolutional Neural Network into a Spiking Neural Network for Image Recognition Tasks. *MDPI Applied Sciences*, 12:5749.
- Park H. & Leemis, L. (2019). Ensemble Confidence Intervals for Binomial Proportions, *Statistics in Medicine*, 38(18):3460.
- Pepperl+Fuchs. (2015). WHA-BLT-F9D0-N-A0-\*, WirelessHART Adapter. TDOCT-4909\_ENG Manual. Retrieved Aug 18, 2024, from <https://tinyurl.com/peppplusfucwirelesshart>.
- Petro, B., et al. (2020). Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks. *IEEE Trans on Neural Networks and Learning Systems*, 31(2):358-370.
- Qian, S. & Chen, D. (1993). Discrete Gabor transform. *IEEE Trans. Signal Process.* 41(7):2429-2438.
- Rathi, N., et al. (2023). Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware. *ACM Comput. Surv.* 55(12)(243).
- Siemens (2012). WirelessHART Adapter, SITRANS AW210, 7MP3111. DM1101020UBB User Manual. Retrieved Aug 18, 2024, from <https://tinyurl.com/yyjbgybm>.
- Smith, M.J., et al. (2024a). Effects of RF Signal Eventization Encoding on Device Classification Performance. *MDPI Electronics*. 2024; 13(11):2020.
- Smith, M.J., et al. (2024b). Edge Device CNN Classification Using Eventized RF Fingerprints. *ACM Int'l Conf on Neuromorphic Systems (ICONS)*, Jul 2024, Arlington VA, USA.
- Soltanieh, N., et al. (2020). A Review of Radio Frequency Fingerprinting Techniques. *IEEE Journal of Radio Frequency Identification*, 16(3):222-233.
- Stanojevic, A., et al. (2023). An Exact Mapping from ReLU Networks to Spiking Neural Networks. *Neural Networks*, 168:74-88.
- Voelker, A.R., et al. (2021). A Spike in Performance: Training Hybrid-Spiking Neural Networks with Quantized Activation Functions. Retrieved Aug 18, 2024, from <https://arxiv.org/abs/2002.03553>.
- Wu D., et al. (2022). A Little Energy Goes a Long Way: Build an Energy-Efficient, Accurate Spiking Neural Network From Convolutional Neural Network. *Frontiers in Neuroscience*, 16:759900.
- Yarga, S., et al. (2022). Efficient Spike Encoding Algorithms for Neuromorphic Speech Recognition. Retrieved Aug 18, 2024, from <http://arxiv.org/abs/2207.07073>.
- Zhu B., et al. (2014). Event-based Time Series Data Preprocessing: Application to Traffic Flow Time Series. Proceedings Int'l Work-Conference on Time Series (ITISE). Retrieved Aug 18, 2024, from [https://oa.upm.es/36830/1/INVE\\_MEM\\_2014\\_195277.pdf](https://oa.upm.es/36830/1/INVE_MEM_2014_195277.pdf).