

Article

# Using a Low-Power Spiking Continuous Time Neuron (SCTN) for Sound Signal Processing

Moshe Bensimon <sup>1</sup>, Shlomo Greenberg <sup>1,\*</sup> and Moshe Haiut <sup>2</sup>

<sup>1</sup> School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beersheba 8400711, Israel; bensimmo@post.bgu.ac.il

<sup>2</sup> DSP Group LTD., Herzliya 4659071, Israel; Moshe.Haiut@dspg.com

\* Correspondence: shlomog@bgu.ac.il

**Abstract:** This work presents a new approach based on a spiking neural network for sound preprocessing and classification. The proposed approach is biologically inspired by the biological neuron's characteristic using spiking neurons, and Spike-Timing-Dependent Plasticity (STDP)-based learning rule. We propose a biologically plausible sound classification framework that uses a Spiking Neural Network (SNN) for detecting the embedded frequencies contained within an acoustic signal. This work also demonstrates an efficient hardware implementation of the SNN network based on the low-power Spike Continuous Time Neuron (SCTN). The proposed sound classification framework suggests direct Pulse Density Modulation (PDM) interfacing of the acoustic sensor with the SCTN-based network avoiding the usage of costly digital-to-analog conversions. This paper presents a new connectivity approach applied to Spiking Neuron (SN)-based neural networks. We suggest considering the SCTN neuron as a basic building block in the design of programmable analog electronics circuits. Usually, a neuron is used as a repeated modular element in any neural network structure, and the connectivity between the neurons located at different layers is well defined. Thus, generating a modular Neural Network structure composed of several layers with full or partial connectivity. The proposed approach suggests controlling the behavior of the spiking neurons, and applying smart connectivity to enable the design of simple analog circuits based on SNN. Unlike existing NN-based solutions for which the preprocessing phase is carried out using analog circuits and analog-to-digital conversion, we suggest integrating the preprocessing phase into the network. This approach allows referring to the basic SCTN as an analog module enabling the design of simple analog circuits based on SNN with unique inter-connections between the neurons. The efficiency of the proposed approach is demonstrated by implementing SCTN-based resonators for sound feature extraction and classification. The proposed SCTN-based sound classification approach demonstrates a classification accuracy of 98.73% using the Real-World Computing Partnership (RWCP) database.

**Keywords:** spiking neuron; digital neuron; SNN; SCTN; STDP learning rule; LIF model; MFCC; sound feature extraction



**Citation:** Bensimon, M.; Greenberg, S.; Haiut, M. Using a Low-Power Spiking Continuous Time Neuron (SCTN) for Sound Signal Processing. *Sensors* **2021**, *21*, 1065. <https://doi.org/10.3390/s21041065>

Academic Editor: Wookey Lee  
Received: 30 December 2020  
Accepted: 1 February 2021  
Published: 4 February 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, we are witnessing a shift in technological solutions from the traditional algorithmic approach to multipurpose deep neural networks (DNNs) neuromorphic approach. DNN networks are widely applied to sound recognition and image processing [1,2], presenting a new challenge for low-power and efficient implementation in embedded systems. These types of applications attempt to resemble the operation of the human brain by understanding and imitating the process of human sensory perception. The necessity for low-power and high-performance embedded platforms is increasing [3,4]. Many mobile applications such as the Internet of Things (IoT) are penetrating our life rapidly and require energy-efficient processing units [5], which most of the current NN-based solutions lack. This work presents a new approach based on a spiking neural network for sound preprocessing and classification. The proposed approach is biologically inspired by the

biological neuron's characteristic using spiking neurons, and Spike-Timing-Dependent Plasticity (STDP)-based learning rule. Spiking neural models mimic the biological neurons for information processing purposes [6]. SNN architectures can be used to solve spatiotemporal problems such as in patterns recognition, optimization, and classification problems [7]. SNN have been developed with a neurobiologically plausible computational architecture that incorporates both spatial and temporal data into one unifying model and can be applied for pattern recognition [8]. Recent studies use machine learning methods to integrate the dynamic patterns of spatiotemporal brain data contained in EEG and ERP data. Z. Doborjeh et al. [8] use SNN for evaluation of concurrent neural patterns generated across space and time from electroencephalographic features representing event-related potential (ERP). Unlike classical neurons, a spiking neuron (SN) uses spikes for communication and computations, and therefore has the potential to consume very low power while enabling efficient implementation in terms of silicon area [9]. Bensimon et al. [10] demonstrate that the Spike Continuous Time Neuron (SCTN) model is capable of accurately replicating the behaviors of a biological neuron. A rich diversity of behaviors can be achieved by smart inter-connection of basic neuron blocks. Reordering and connecting blocks of SCTNs into a full SN-based network allow efficient implementation of a large class of cognitive algorithms and voice applications. This work suggests the usage of an innovative energy-efficient hardware-based spiking neuron (SN) presented in [11] as a basic building block in the design of programmable analog electronics circuits.

The proposed SCTN can be considered as a nonlinear device capable of processing an extensive amount of data efficiently. The close resemblance between SN and a biological neuron justifies the evaluation of the substitution of the classical NN model with the SN model. The implementation of SN can be carried out using both analog and digital circuits [12]. Various kinds of SNN-based models and neuromorphic circuits have been recently proposed to support the processing of vast streams of information in real-time [13,14]. A continuous-time spiking neural network paradigm is presented in [15]. The spike latency takes into account that the firing of a given neuron occurs after a continuous-time delay. SNN-based classifier consistently outperforms the traditional Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) neural networks in a temporal pattern classification task [16]. Although RNN and LSTM models capture the temporal transition explicitly, they are hard to train for long sound samples due to the vanishing and exploding gradient problem [17]. Low-power spiking-based neural networks are highly suitable to replace RNN and LSTM as the computational engine for time-series applications utilizing the high correlation between adjacent frames [18], especially if the series of data contains strong temporal dependencies like in the case of sound and video applications [14].

STDP is a biologically plausible learning paradigm inspired by the Hebbian learning principle. STDP is one of the most common activity-driven synaptic weight learning mechanisms [19]. Following the STDP learning rule, the neuron weights are adjusted considering the time difference between presynaptic and post-synaptic spikes. According to this rule, if the presynaptic neuron fires earlier than the post-synaptic one, the synapse is strengthened. It has been shown that the unsupervised STDP rule can be used for the detection of frequent input spike patterns [20,21].

This work presents a new building block approach and SCTN-based spiking neural network for sound preprocessing and classification. The proposed approach is biologically inspired by the biological neuron's characteristic using spiking neurons and STDP-based learning rule. We propose a biologically plausible sound classification framework which uses an SCTN-based network for detecting the embedded frequencies contained within an acoustic signal.

Throughout the years many voice features extraction techniques have been suggested [22,23]. Voice signal identification systems involve the process of converting analog speech waveform into useful features for further processing and classification. Among the leading voice feature extraction techniques are the Mel Frequency Cepstral Coefficient

(MFCC) introduced by Davis et al. [24], the Linear Prediction Coefficients (LPC), and the Linear Prediction Cepstral Coefficients (LPCC). The Hidden Markov Modeling (HMM) is one of the most common techniques for voice classification [25] used for automatic speech recognition.

J. Wu et al. present a biologically plausible framework for sound event classification [16]. They propose to use an unsupervised self-organizing maps (SOM) network for representing frequency contents embedded within the acoustic signals and an event-based SNN for pattern classification. R. Xiao et al. propose a feedforward SNN for sound classification using the temporal learning rule [26]. The classification is based on extracting acoustic features from the time-frequency representation of sound (using FFT). They suggest to convert the representative sound features into a spiking train by a simple mapping rather than using SOM. Then these temporal patterns are classified via SNN using temporal learning rules.

Contrarily to the previous SNN-based approaches for sound classification, we propose a framework based on a combined SNN for both preprocessing, feature extraction, and classification. The feature extraction is carried out using an SCTN-based network avoiding the common external preprocessing stage and time-frequency representation of the sound. Moreover, we suggest direct interfacing of the acoustic sensor with the SCTN-based network avoiding the usage of costly analog-to-digital and FFT conversions. This work also demonstrates an efficient hardware implementation of the SNN network based on the low-power digital SCTN neuron presented in [10,11].

The rest of the paper is organized as follows. Section 2 describes in detail the proposed SCTN model and the STDP learning paradigm. Section 3 presents the proposed SCTN building block approach and demonstrates the use of the SCTN to generate a basic frequency detector and a full resonator. Section 4 describes how the SCTN-based SNN is applied to sound features extraction and classification. Section 5 describes the experimental results, and, finally, Section 6 concludes the paper.

## 2. SCTN-Spike Continuous Time Neuron Model

### 2.1. The Leaky Integrate and Fire (LIF) Neuron Model

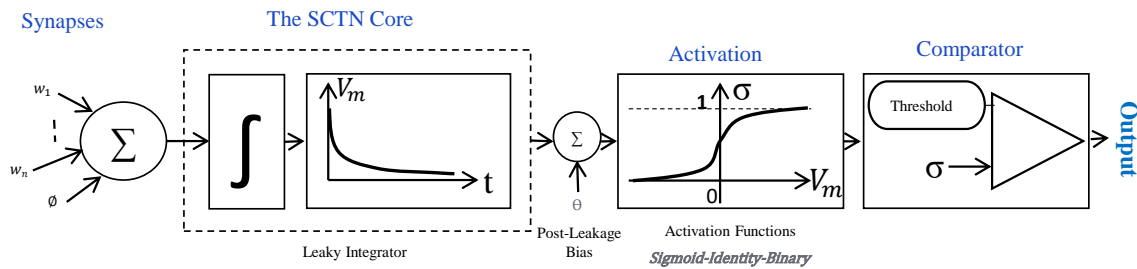
The proposed Spiking Continuous Timing Neuron (SCTN) is a type of continuous-time representation neuron, whose operation is equivalent to the biological neuron [11]. The SCTN model is based on the common Leaky Integrate and Fire (LIF) model with some modifications allowing various neuron configuration using seven different leak modes and three activation functions with a dynamic threshold setting. The various neuron parameters dictate the type of neuron response to a train of spikes in the time domain. The LIF neuron model is given by Equation (1).

$$Vm_j(t) = Vm_j(t-1) + \sum_{i=1}^N w_{ij} \cdot I_i(t) - \mu_j \quad (1)$$

where  $Vm_j(t)$  is the membrane potential at time  $t$ ,  $I_i(t)$  is the binary synaptic inputs (a stream of spikes), and  $w_{ij}$  represents the corresponding weights. The weight  $w_{ij}$  is the strength of the connection from the  $i$ th to the  $j$ th neuron. For each neuron  $j$ , the index  $i$  ranges from 1 to  $N$ , where  $N$  represents the number of the neurons in the previous layer. The neuron leakage is described by the constant parameter  $\mu_j$ . The proposed LIF neuron model uses a dynamic threshold as an adaptive mechanism to control the firing frequency. When the membrane potential ( $Vm$ ) exceeds the threshold value, the neuron fires a spike, and then the membrane potential discharges to a resting potential (like a biological neuron) [6]. This default value is determined empirically. The decision to fire and generating a spike in the neuron output is carried out according to the following condition:

$$if(Vm(t) \geq Threshold) \implies Spike, Vm(t) \leftarrow Vm_{reset} \quad (2)$$

Figure 1 illustrates the mathematical model for the proposed SCTN neuron. The proposed spiking neuron structure is composed of four main components: (a) an adder which performs the weighted input summation, (b) a leaky integrator with a time-constant that is controlled by a “memory” parameter  $\alpha$ , (c) a random generator which is used to implement the sigmoid activation function, and (d) a comparator that checks if the membrane potential reaches the threshold. The spiking neuron fires a pulse in case the weighted input summed by the adder exceeds the current random value (i.e., threshold).



**Figure 1.** The Spiking Continuous Timing Neuron (SCTN) mathematical model.

The SCTN membrane potential,  $V_m(t)$ , is given by Equation (3).

$$V_m(t) = \begin{cases} V_{m0}, & t = 0 \\ (1 - 2^{-LF})(V_m(t-1) + \sum_{j=1}^N w_j \cdot I_j + \phi) + \Theta, & t \cdot \text{mod}(LP) = 0 \\ V_m(t-1) + \sum_{j=1}^N w_j \cdot I_j + \phi + \Theta, & t \cdot \text{mod}(LP) \neq 0 \end{cases} \quad (3)$$

where  $t \in N_0$ , the Leakage Factor ( $LF$ ), and the Leakage Period ( $LP$ ) are the neuron leakage parameters representing the integrator leakage rate and the rate of the integrator operation, respectively, and  $\Theta$  and  $\phi$  are the pre- and post-leakage bias, respectively [10]. The weighted inputs (axons) are accumulated directly into the adder in an iterative process while adding the membrane potential from the previous iteration. The leakage rate is controlled by the  $\alpha$  parameter. The membrane potential is represented by a dedicated internal register. A pseudo-random generator is used to generate the nonlinear activation function. In case the accumulated value (including a bias) of the membrane exceeds the threshold, the neuron fires a pulse in its output. The proposed SCTN model consumes less than 10 nW and requires only 700 ASIC 2-inputs gates for implementing a neuron with 10 synapses inputs [10]. The SCTN neuron architecture is comparable, in terms of area and power, to the IBM SN model used in the neuromorphic chip (TrueNorth), which is composed of 1272 gates [27]. Simulations of the proposed digital spiking neuron also demonstrate its ability to accurately replicate the behaviors of a biological neuron model accurately [10].

## 2.2. STDP Learning Module

Different types of bio-plausible learning rules are adapted for SNN training [28,29]. These learning rules are based on synaptic plasticity training applying unsupervised learning approaches [30]. Among the most commonly used bio-plausible rules are the Spike Timing Dependent Plasticity (STDP) and the Spike-Driven Synaptic Plasticity (SDSP) [7]. Several hardware STDP implementations have recently been proposed [31–34]. We adapt the low-power hardware STDP implementation presented in [10] and integrated it in the new SCTN architecture. Figure 2 shows the hardware implementation of the STDP-based learning module. The module is composed of three main components: (a) Time event matrix, (b) STDP weights register, and (c) Post-synaptic delay line. The event-matrix represents the timing of the events, where each row represents an input synapse.  $Lw$  stands for the Length of the considered time Window. The presynaptic spike times are sampled during the time window  $Lw$  and stored in an event matrix along with the synaptic

index. When the neuron generates a postsynaptic spike, the time of the postsynaptic spike is compared with the stored presynaptic spike times, and the time difference is used to update the weights according to the STDP rule. The time window considers a total of  $2Lw$  time units (pre- and post-firing the spike at the neuron output). Therefore, the event matrix is implemented in hardware using an array of shift registers (one for each input synapse) where each shift-right register is of the length of  $2Lw$  bits. Upon an output spike, the weights of the synapses are updated in relation to the input spike timing and according to the STDP learning rule. Synapses that contribute to the generation of an output spike event should be strengthened. The shifted presynaptic input vector is weighted by  $Bu$  which represents the STDP modification function according to the time difference between the postsynaptic presynaptic times spike. The synapse weight update function  $W_{j_{new}}$  is built using a convolution of the shifted presynaptic input vector weighted by  $Bu$ , where  $-Lw \leq u \leq Lw$ , and the delayed output spike. Different, more complex STDP modification rules can be configured using the proposed B register, eliminating the need for any hardware modification imposed by previous hardware STDP implementation [31]. The STDP learning rule is given by Equation (4).

$$\Delta w_j = \eta \sum_{k \in S_{pre}} \sum_{l \in S_{post}} \begin{cases} +\eta^+ e^{-|t_l - t_k|/\tau_+}, & \text{if } t_l \leq t_k \\ -\eta^- e^{-|t_l - t_k|/\tau_-}, & \text{if } t_l > t_k \end{cases} \quad (4)$$

where  $\eta^{(-/+)}$  is the learning rate and  $\tau_+$  and  $\tau_-$  are the time scales to control synaptic potentiation and depression. The magnitude of the learning rate decreases exponentially with the absolute value of the timing difference. When multiple spikes are fired, the weight change is the sum of the individual change calculated from all possible spike pairs.  $S_{preset}$  and  $S_{post}$  are the sets of spikes of the pre- and post-synaptic neurons, respectively, where  $t_k$  is the time of the spike  $k$  (of the  $S_{preset}$ ) and  $t_l$  is the time of the spike  $l$  (of the  $S_{post}$  set).

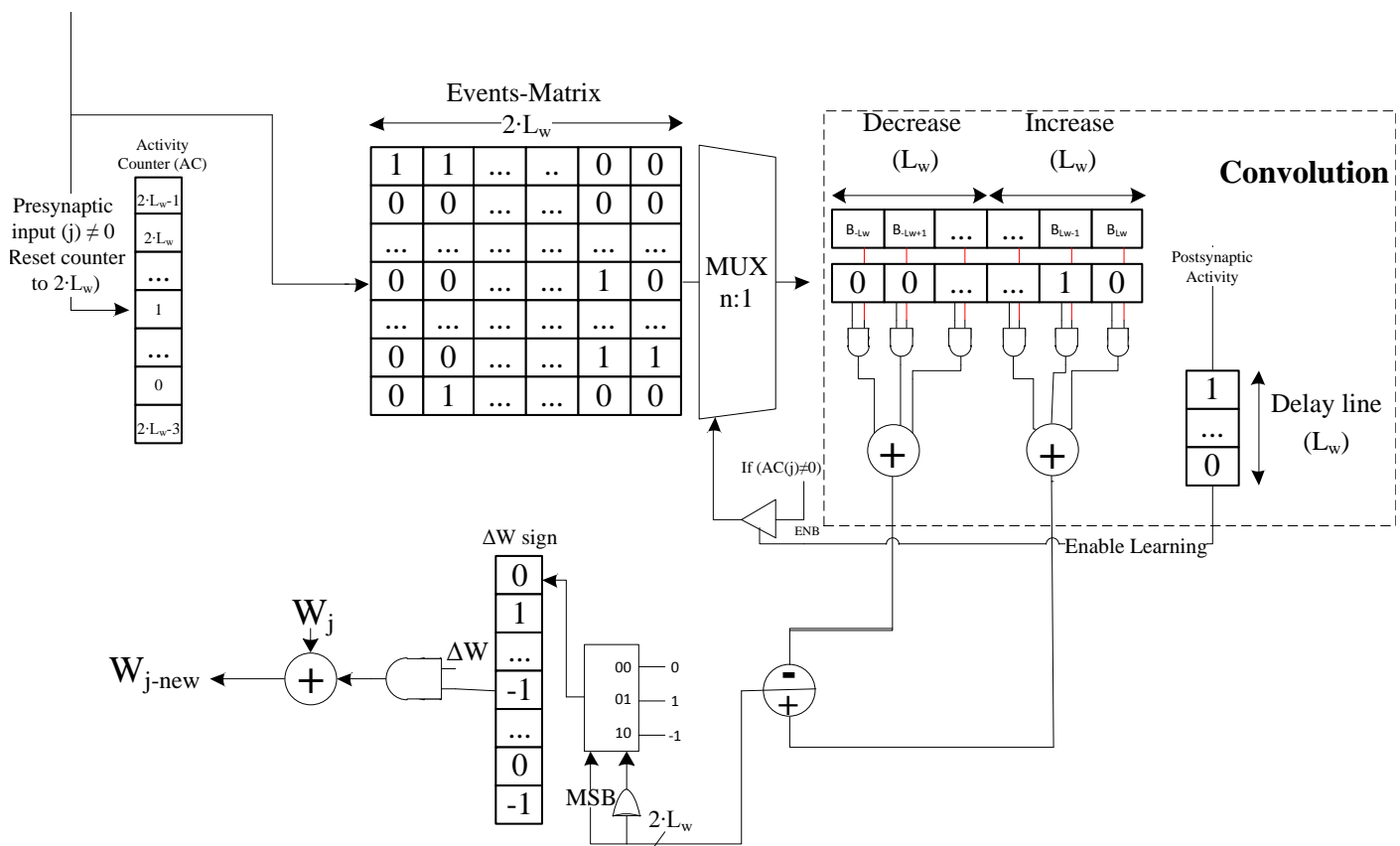


Figure 2. Spike Timing Dependent Plasticity (STDP)-based learning model.

### 3. Building Block Approach

This section presents the proposed building block approach applied to sound feature extraction and classification. We suggest considering the Spike Continuous Time Neuron (SCTN) presented by Bensimon et al. [10] as a basic “building block” in the design of programmable digital electronics circuits. The proposed approach demonstrates the use of the SCTN neuron as an efficient alternative to the design of programmable analog electronics circuits. The SCTN is used as the basic building block to implement the analog-like circuits, utilizing the unique features of the SCTN neuron and smart inter-connections. Usually, a neuron is used as a repeated modular element in any neural network structure, and the connectivity between the neurons located at different layers is well defined. Thus, generating a modular NN structure composed of several layers with full or partial connectivity. The proposed approach suggests controlling the specific behavior of each neuron and uses smart and unique connectivity of each neuron in order to emulate an analog circuit. Unlike existing SNN-based sound classification solutions for which the preprocessing phase is carried out using analog-to-digital and FFT conversions of the raw analog data, we suggest integrating the preprocessing phase into the network. This approach allows referring to the basic SCTN neuron as an analog module (i.e., a building block), and thus enabling the design of simple analog circuits based on SNN with unique inter-connections between the neurons. The proposed approach is applied to sound signal processing demonstrating efficient ultra-low-power implementation of some common analog circuits used for phase detection and sound feature extraction and classification. The benefit of using an SNN-based network for implementing analog circuits is reflected by a generic solution and the reuse ability by changing the network weights. The building block approach is applied to sound signal processing demonstrating efficient ultra-low-power implementation of some common analog circuits used for phase detection, frequency detection, and sound feature extraction and classification.

#### 3.1. SCTN-Based Phase Shifting

The efficiency of the proposed approach is demonstrated by implementing an SNN-based preprocessing phase-shifting circuit and frequency detection. Therefore, enabling a simple and direct sensor interface, saving silicon area, and efficient low-power sound processing in real-time. We examine and demonstrate the ability of the SCTN neuron to generate a phase-shifting emulating a common analog filter circuit. Figure 3 shows a simple first-order RC circuit.

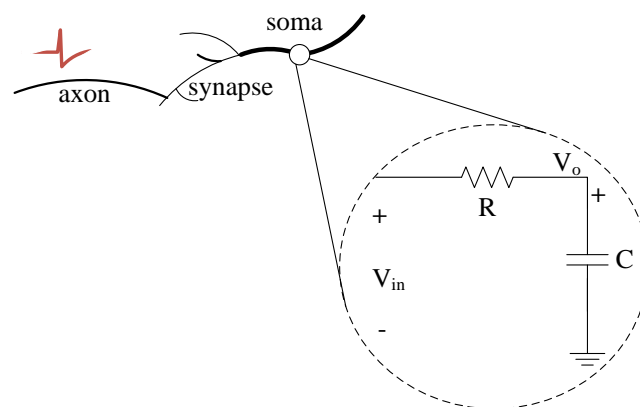


Figure 3. RC circuit filter and its relation/analogy to the SCTN.

The transfer function of the RC circuit is given by

$$H(j\omega) = \frac{V_o}{V_{in}} = \frac{1 - j\omega RC}{1 + (\omega RC)^2} \quad (5)$$



where the radial frequency is  $\omega = 2\pi f$ . For the cutoff radial frequency  $\omega_0 = \frac{1}{RC}$  the magnitude and phase are given by Equations (6) and (7).

$$|H(j\omega_0)| = \frac{1}{\sqrt{2}} \quad (6)$$

$$\arg(H(j\omega_0)) = \frac{-\pi}{4} \quad (7)$$

Therefore, for an input frequency of  $\omega_0$ , a phase shift of  $-45^\circ$  is observed. In analogy with an SN neuron, the charge of the capacitor  $C$  may represent the neuron membrane potential, while the resistor  $R$  may represent the leakage rate of the neuron [35]. Each SCTN cell incorporates a leaky integrator with a time constant that is controlled by a predefined parameter:  $\alpha = 1 - 2^{-LF}$ . The SCTN may postpone the incoming signals according to an expected leakage time constant, i.e., a delay which is given by Equation (8).

$$Delay = T_{PulseCycle} \cdot \frac{1}{1 - \alpha} \cdot (a + LP) \quad (8)$$

where  $T_{PulseCycle}$  is derived from the system clock as

$$T_{PulseCycle} = \frac{x + 1}{2} \cdot (T_{sys=clk}), -1 \leq x < 1 \quad (9)$$

The SCTN delay and the spike rate are derived from Equation (8) and formulated as

$$\tau = T_{Pulses} \cdot \frac{1}{1 - \alpha} \cdot (1 + LP) \quad (10)$$

$$\omega_0 = \frac{1 - \alpha}{T_{Pulses} \cdot (1 + LP)} \quad (11)$$

$$f_{Pulses} = \frac{1}{T_{Pulses}} \quad (12)$$

where  $\tau$  denotes the delay, and the analog cutoff frequency  $f_0$  is given by

$$f_0 = \frac{f_{pulses} \cdot (1 - \alpha)}{2\pi \cdot (1 + LP)} = \frac{f_{pulses}}{2^{LF} \cdot 2\pi \cdot (1 + LP)} \quad (13)$$

Therefore, for an appropriate choice of the LP and LF parameters (which determine the resonance frequency), the SCTN has the ability to generate a phase shift of 45 degrees.

### 3.2. SCTN-Based Phase Shifting

This section presents an SCTN-based resonator circuit which serves as a frequency detection utilizing the SCTN basic phase shifting feature. The proposed resonator is used to extract the representing frequency contents embedded within the acoustic signals.

Figure 4 depicts a unique SCTN-based resonator circuit. The resonator circuit is used to detect the presence of a range of frequencies embedded within the sound signal. The proposed resonator circuit is composed of 17 SCTNs building blocks arranged in 10 layers. A PDM binary stream representing the sound signal serves as an input to the resonator circuit. The first eight layers are composed of eight SCTN neurons (one per layer), where each neuron performs a phase shift of 45 degrees. A feedback path (connected to the 4th neuron) representing a shift of 180 degrees is connected to the input. The final layer contains 8 SCTN neurons: the first four neurons are used for blocking negative signals, while the other four serves as rectifiers configured with binary and identity activation functions, respectively. In the final stage, the output SCTN neuron sums the four positive outputs of each detected phase (0–90–180 and 270 degrees). The output neuron fires a train of pulses as a result of detecting frequencies that are in the range of the resonance

frequency (in a predefined range of frequencies). The maximum pulse rate is achieved at the resonance frequency, and the pulse rate is relatively decreased as the frequency moves away from the resonant mid-frequency.

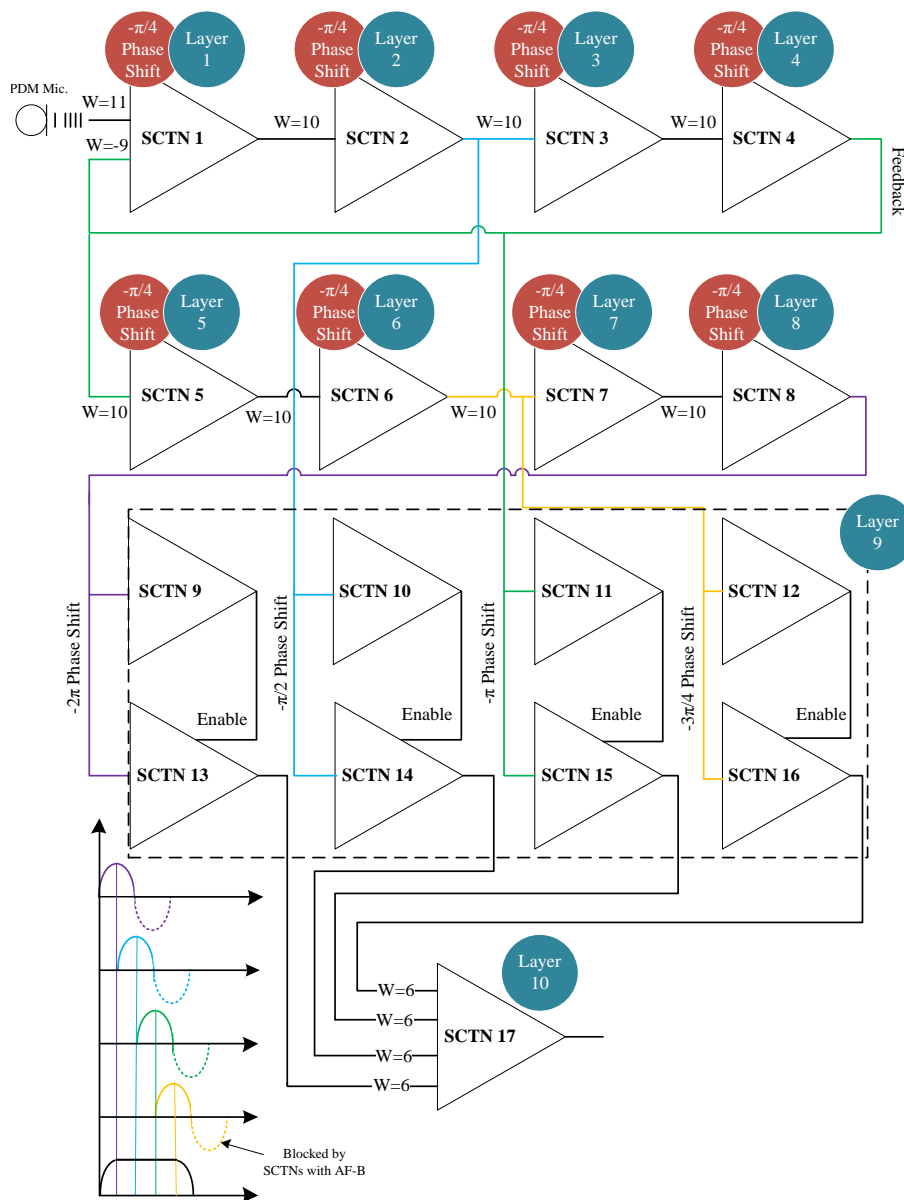


Figure 4. The proposed 10-layer resonator architecture.

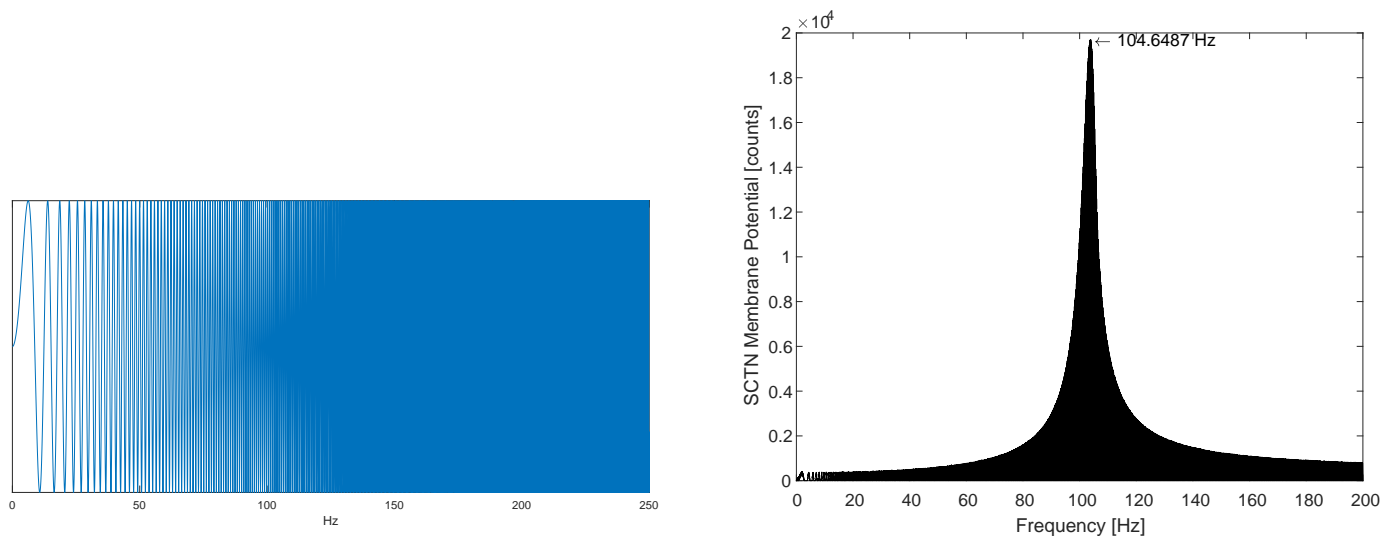
Figure 5 demonstrates the detection of a predefined resonance frequency within a chirp signal containing frequencies in the range of 0 to 250 Hz. The SCTN parameters LP and LF are configured to detect a resonance frequency of  $f_0 = 104.65$  [Hz] using Equation (14) for LF = 5, LP = 73, and system clock  $F_{pulse} = 1.536$  MHz.

$$f_0 = \frac{1.536 \cdot 10^6}{2^5 \cdot 2\pi \cdot (1 + 73)} = 104.65 \tag{14}$$

Figure 5(right) shows that the peak is achieved exactly at the resonance frequency  $f_0$ . In order to detect several frequencies, which represent the sound signal, within a predefined range, we suggest using a bank of filters (i.e., bank of resonators). Each resonator is responsible for detecting a single frequency. For example, to detect 20 predefined

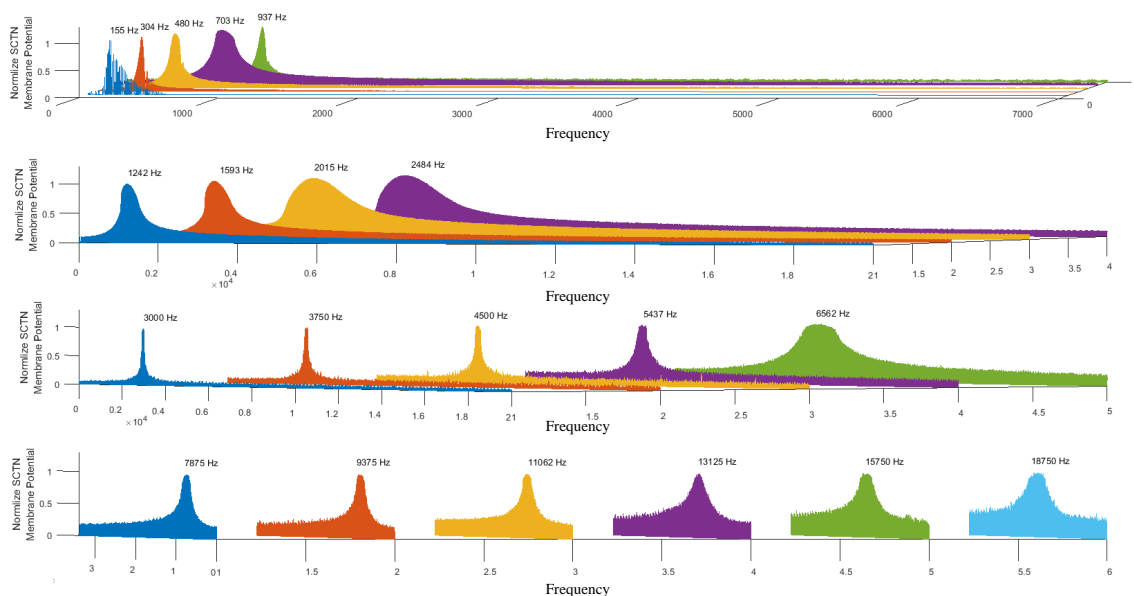


frequencies in the range of [20 Hz, 20 kHz], a bank of 20 filters (composed of  $17 \times 20 = 340$  SCTN neurons) is required.



**Figure 5.** (left) Resonator input: a chirp signal. (right) SCTN-based Resonator output: detected frequency.

Figure 6 illustrates the simulation results for a bank of 20 resonators (based on SCTN) used to detect 20 different frequencies in the audio spectrum. A chirp signal (in the range of 20 to 20k Hz) has been used as an input to each of the 20 resonators. The array of resonators was tuned to detect the following frequencies: [155; 304; 480; 703; 937; 1242; 1593; 2015; 2484; 3000; 3750; 4500; 5437; 6562; 7875; 9375; 11,062; 13,125; 15,750; 18,750].



**Figure 6.** SCTN-based resonators' bank (the parameters are tuned to detect 20 predefined frequencies in the audio spectrum).

This approach can be easily applied for extracting the common MFCC coefficients representing a sound signal. Each resonator can be configured to detect a relatively closed neighborhood of frequencies around the central detected frequency. The bank of filters can be adapted to identify different types of soundtracks (e.g., female and male, background, noises, etc.).

#### 4. SCTN-Based Sound Feature Extraction

##### 4.1. Classical Sound Preprocessing

Traditionally, a complete sound processing framework consists of the following main stages: Pulse Density Modulation (PDM), preprocessing, features extraction, and classification. MFCC is the most widely used feature extraction method for automatic speech recognition. Figure 7 depicts the stages of classical sound preprocessing and features extraction. The digital microphones translate their vibration intensity from an audio wave to digital information. The PDM signal is converted to Pulse Code Modulation (PCM) representation by a digital Low-Pass Filter (LPF) with downsampling to a sampling rate that is appropriate to voice processing (typically 3–16 KHz). The PCM samples are then buffered into 20 [ms] frames and converted to the frequency domain via FFT transform. The extraction of the MFCC coefficients is achieved by multiplying (in the frequency domain) the Mel filters [24] and the spectral transform of the signal. This roughly determines the energy distribution in the frequency domain. Usually, the sound preprocessing stage is implemented using analog hardware [36].

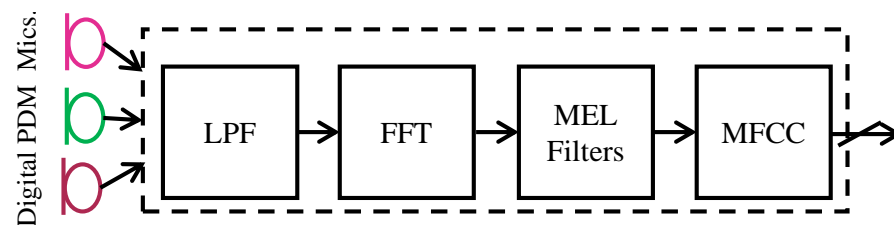


Figure 7. Classical sound preprocessing approach.

##### 4.2. SCTN-Based Resonators Applied to Features Extraction

The usage of the SCTN-based resonator for frequency detection can be extended to extract features representing a sound signal. Each resonator can be reconfigured to detect a relatively closed neighborhood of frequencies around the central detected frequency. The proposed approach disassembles the sound signal into its fundamental spectral components by detecting the frequencies presented in the audio signal. Figure 8 depicts the classical approach for MFCC extraction along with the proposed SCTN-based feature extraction (SN-MFCC).

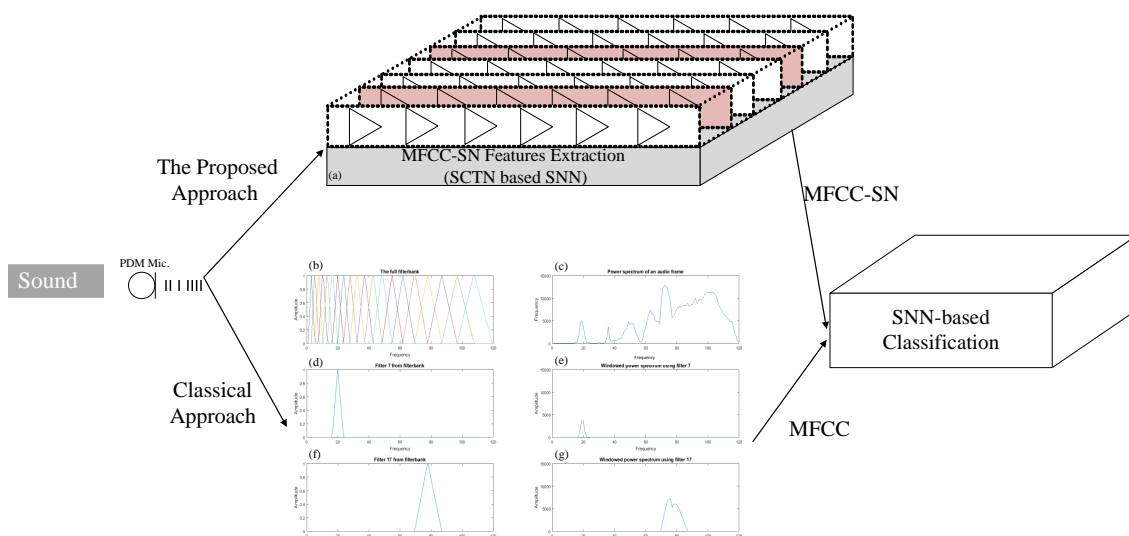


Figure 8. (a) SCTN-based features extraction. (b–g) Classical Mel Frequency Cepstral Coefficient (MFCC) extraction.

Figure 8a shows the proposed SCTN resonator-based filter banks used to extract the SN-MFCC features, while Figure 8b–g demonstrate the classical MFCC extraction.

Figure 8b describes the MEL filters; Figure 8c depicts the FFT domain of a typical sound frame. Figure 8e,g depicts the multiplication of two representative MEL filters shown in Figure 8d,f by the spectral transform of the signal in Figure 8c. The extracted MFCC features are used for sound classification using an SCTN-based classifier, although the extracted SN-MFCC coefficients do not precisely match the classical MFCC.

The SCTN-based classifier is composed of one layer with a dedicated SCTN neuron per class. The SCTN-based classifier has been trained separately for each class using the STDP learning rule. The classical MFCC coefficients should be extracted iteratively in real-time for each sampling frame (typically 20 ms frames with 10 ms overlap). By contrast, the proposed approach suggests processing the audio signal continuously, avoiding the need to split the input signal into frames. The input data can be connected directly to a digital PDM microphones that produces rate coded spikes. Moreover, in contrast to the frame-based system, as the SCTN-based preprocessing is driven by an external stimulus, unnecessary continuous frame processing (like silence frames) is avoided.

## 5. Experimental and Results

The proposed sound classification based on the SN-MFCC extracted features was evaluated using the Real-World Computing Partnership (RWCP) standard database [37]. To allow fair comparisons with other existing SNN-based sound classification systems [16,26] we select the same ten classes from the RWCP database ('cymbals', 'horn', 'phone4', 'bells5', 'kara', 'bottle1', 'buzzer', 'metal15', 'whistle1', and 'ring'). The sound files are recorded at 16 kHz sampling rate in a noisy background. The SCTN-based classification network was trained and tested with 40 randomly selected sound samples per class, where 20 samples have been used for training and 20 samples used for testing. To evaluate the generalization abilities of the proposed SCTN-based classifier the test data set includes only new different sound samples. The test data set contains 200 randomly selected sound samples. A cross-validation, using different randomly partitioning of the dataset for training and testing, is carried out to test the ability of the proposed approach to correctly classify new data that was not used in the training process. The training and testing process was repeated 10 times with different sound samples randomly selected from the RWCP database. To reduce variability, multiple rounds of cross-validation are performed using different dataset partitions, and the validation results are averaged over the 10 rounds.

Figure 9 demonstrates the classification process which is composed of two main stages: the preprocessing phase and the classification stage. First, the SN-MFCC coefficients are extracted using the proposed SCTN-based resonator. An array of resonators is tuned to detect different frequencies in the audio spectrum, where each resonator is configured to detect a specific frequency (in the range of 20 to 20 kHz). Then, the extracted SN-MFCC features are used as an input to the SN classifier. Each SN in the output layer is independently trained, with random weight initialization, to classify one of the ten classes (selected from the RWCP database) using unsupervised learning and STDP learning rule.

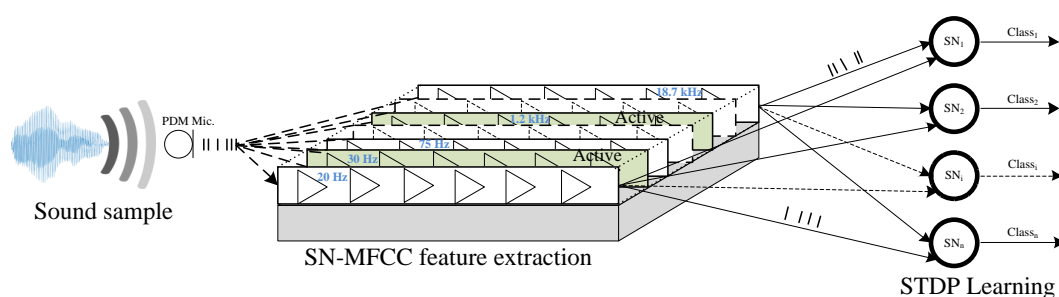
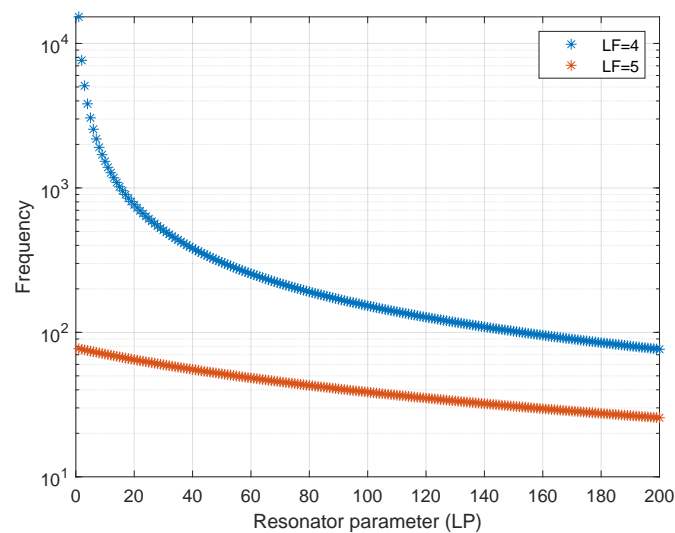


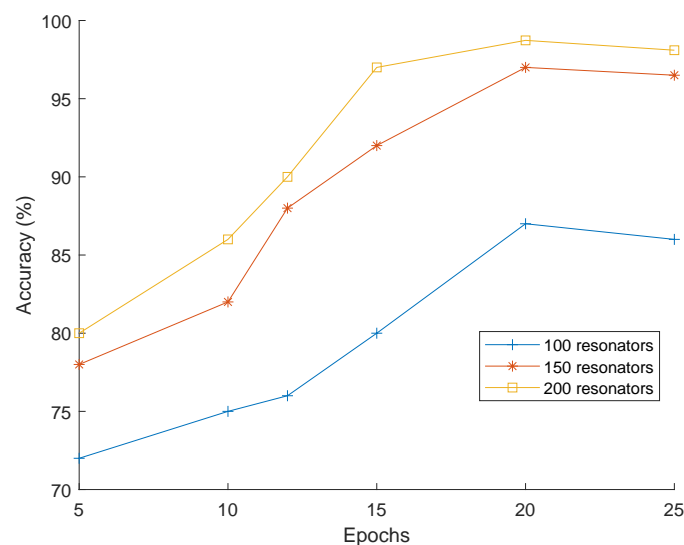
Figure 9. SN-MFCC based sound classification framework .

The STDP module has been configured with a relatively small learning rate  $\eta^+ = 0.005$  and  $\eta^- = 0.004$  to ensure robust learning [25], a time constant of  $\eta^+ = \eta^- = 15$  ms to ensure stable competitive synaptic modification [19], and the  $Lw$  parameter was set to 16. The SCTN-based resonators have been configured to detect frequencies in the range of 38 Hz to 15 kHz using varying LP and LF parameters (LF = 4:5 and LP = 1:200). Figure 10 depicts the derived frequencies as a function of the combination of the LF and LP parameters.



**Figure 10.** Resonators' detection frequencies as a function of LP and LF.

The effect of the number of SN-MFCC coefficients on the classification accuracy has been investigated using 100, 150, and 200 resonators (each composed of 17 neurons). Figure 11 depicts the classification accuracy as a function of the number of SN-MFCC coefficients (for 100, 150, and 200 resonators) and the number of training epochs (for 5, 10, 12, 15, 20, and 25 epochs). An epoch refers to one cycle through the full 200 samples training dataset. Training the network with more epochs leads to better generalization and improved classification accuracy. The average test accuracy achieved with 20 epochs is 87.23%, 97.11%, and 98.73% for 100, 150, and 200 SN-MFCC coefficients, respectively.



**Figure 11.** Accuracy as a function of epochs for different numbers of resonators.

The classification accuracy is compared with two other existing SNN-based approaches [16,26] using the same RWCP database. Table 1 depicts the accuracy results of

our proposed classifier compared to RNN, LSTM [16], and three other SNN-based sound classification [16,26,38].

**Table 1.** Classification accuracy comparison (adapted and completed from the work in [16]).

Model	Accuracy (%)
RNN	95.35
LSTM	98.40
LSF-SNN	98.50
LTF-SNN	97.50
SOM-SNN	99.60
SCTN-SNN	98.73

Simulation results of the proposed SCTN-based classifier show accuracy of 98.73%, which is comparable to the accuracy of 97–99.6% demonstrated by the SOM-SNN approach presented in [16], and outperforms the time-frequency encoding method [26], which shows the accuracy of 96%. One of the main advantages of the proposed approach is that the feature extraction is carried out using SCTN-based resonators, avoiding the common external preprocessing stage and time-frequency representation of the sound. Moreover, we suggest a direct interfacing of the acoustic sensor with the SCTN-based network avoiding the usage of costly digital-to-analog conversions.

## 6. Conclusions

This work presents a new approach based on a spiking neural network for sound preprocessing and classification. We propose a biologically plausible sound classification framework that uses an SNN-based network for detecting the embedded frequencies contained within an acoustic signal. A new SCTN digital neuron is used as a basic building block for constructing a spiking neural network. We demonstrate the use of an SCTN-based network as an efficient alternative to the design of programmable analog electronics circuits. The proposed approach is applied to sound signal processing and classification, suggesting direct interfacing of the analog sensor with the SNN network. We present an efficient ultra-low-power implementation of some common analog circuits used for phase and frequency detection, voice feature extraction, and classification. The SCTN is used as the basic building block to implement analog-like circuits, utilizing the unique features of the SCTN neuron and smart inter-connections. Experimental results show high accuracy of 98.73% achieved for sound classification. The benefit of using an SCTN-based network for implementing analog circuits is also reflected by a generic solution and the reuse ability by changing the network weights. The proposed approach can be applied in future work for efficiently extracting the common MFCC coefficients representing a speech signal.

## 7. Patents

Haiut, M. Neural cell and a neural network. Patent 15/877459, 9 August 2018.

**Author Contributions:** Investigation, M.B., S.G. and M.H.; supervision, S.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the GenPro Consortium within the Framework of the Israeli Innovation Authority's MAGNET Program.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Yepes, A.J.; Tang, J.; Mashford, B.S. Improving classification accuracy of feedforward neural networks for spiking neuromorphic chips. *arXiv* **2017**, arXiv:1705.07755.
2. Tang, J.; Mashford, B.S.; Yepes, A.J. Semantic Labeling Using a Low-Power Neuromorphic Platform. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1184–1188. [[CrossRef](#)]
3. Cao, Y.; Chen, Y.; Khosla, D. Spiking deep convolutional neural networks for energy-efficient object recognition. *Int. J. Comput. Vis.* **2015**, *113*, 54–66. [[CrossRef](#)]
4. Moradi, S.; Qiao, N.; Stefanini, F.; Indiveri, G. A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE Trans. Biomed. Circuits Syst.* **2017**, *12*, 106–122. [[CrossRef](#)]
5. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [[CrossRef](#)]
6. Izhikevich, E.M. Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.* **2004**, *15*, 1063–1070. [[CrossRef](#)] [[PubMed](#)]
7. Kasabov, N.K. *Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2019.
8. Dobarjeh, Z.; Dobarjeh, M.; Crook-Rumsey, M.; Taylor, T.; Wang, G.Y.; Moreau, D.; Krägeloh, C.; Wrapson, W.; Siegert, R.J.; Kasabov, N.; et al. Interpretability of Spatiotemporal Dynamics of the Brain Processes Followed by Mindfulness Intervention in a Brain-Inspired Spiking Neural Network Architecture. *Sensors* **2020**, *20*, 7354. [[CrossRef](#)] [[PubMed](#)]
9. Dytckov, S.; Daneshmand, M. Computing with hardware neurons: Spiking or classical? Perspectives of applied Spiking Neural Networks from the hardware side. *arXiv* **2016**, arXiv:1602.02009.
10. Bensimon, M.; Greenberg, S.; Ben-Shimol, Y.; Haiut, M. A New SCTN Digital Low Power Spiking Neuron. *IEEE Trans. Circuits Syst. II Exp. Briefs* **2021**, submitted for publication.
11. Bensimon, M.; Greenberg, S.; Ben-Shimol, Y.; Haiut, M. A New Digital Low Power Spiking Neuron. *Int. J. Future Comput. Commun.* **2019**, *8*, 24–28. [[CrossRef](#)]
12. Jawandhiya, P. Hardware design for machine learning. *Int. J. Artif. Intell. Appl.* **2018**, *9*, 63–84. [[CrossRef](#)]
13. Jaiswal, A.; Roy, S.; Srinivasan, G.; Roy, K. Proposal for a leaky-integrate-fire spiking neuron based on magnetoelectric switching of ferromagnets. *IEEE Trans. Electron. Devices* **2017**, *64*, 1818–1824. [[CrossRef](#)]
14. Schuman, C.D.; Potok, T.E.; Patton, R.M.; Birdwell, J.D.; Dean, M.E.; Rose, G.S.; Plank, J.S. A survey of neuromorphic computing and neural networks in hardware. *arXiv* **2017**, arXiv:1705.06963.
15. Cristini, A.; Salerno, M.; Susi, G. A continuous-time spiking neural network paradigm. In *Advances in Neural Networks: Computational and Theoretical Issues*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 49–60.
16. Wu, J.; Chua, Y.; Zhang, M.; Li, H.; Tan, K.C. A spiking neural network framework for robust sound classification. *Front. Neurosci.* **2018**, *12*, 836. [[CrossRef](#)] [[PubMed](#)]
17. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Networks Learn. Syst.* **2016**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]
18. Diehl, P.U.; Zarella, G.; Cassidy, A.; Pedroni, B.U.; Neftci, E. Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware. In Proceedings of the 2016 IEEE International Conference on Rebooting Computing (ICRC), San Diego, CA, USA, 17–19 October 2016; pp. 1–8.
19. Song, S.; Miller, K.D.; Abbott, L.F. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* **2000**, *3*, 919–926. [[CrossRef](#)]
20. Brette, R. Computing with neural synchrony. *PLoS Comput. Biol.* **2012**, *8*, e1002561. [[CrossRef](#)] [[PubMed](#)]
21. Masquelier, T. STDP allows close-to-optimal spatiotemporal spike pattern detection by single coincidence detector neurons. *Neuroscience* **2018**, *389*, 133–140. [[CrossRef](#)] [[PubMed](#)]
22. Tirumala, S.S.; Shahamiri, S.R.; Garhwal, A.S.; Wang, R. Speaker identification features extraction methods: A systematic review. *Expert Syst. Appl.* **2017**, *90*, 250–271. [[CrossRef](#)]
23. Prabakaran, D.; Shyamala, R. A Review on Performance of Voice Feature Extraction Techniques. In Proceedings of the 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), Chennai, India, 21–22 February 2019; pp. 221–231.
24. Davis, S.; Mermelstein, P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **1980**, *28*, 357–366. [[CrossRef](#)]
25. Palaz, D.; Magimai-Doss, M.; Collobert, R. End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. *Speech Commun.* **2019**, *108*, 15–32. [[CrossRef](#)]
26. Xiao, R.; Yan, R.; Tang, H.; Tan, K.C. A spiking neural network model for sound recognition. In *International Conference on Cognitive Systems and Signal Processing, Proceedings of the ICCSIP 2016: Cognitive Systems and Signal Processing, Beijing, China, 19–23 November 2016*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 584–594.
27. Cassidy, A.S.; Merolla, P.; Arthur, J.V.; Esser, S.K.; Jackson, B.; Alvarez-Icaza, R.; Datta, P.; Sawada, J.; Wong, T.M.; Feldman, V.; et al. Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–10.



28. Bing, Z.; Meschede, C.; Huang, K.; Chen, G.; Rohrbein, F.; Akl, M.; Knoll, A. End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1–8.
29. Tang, G.; Shah, A.; Michmizos, K.P. Spiking neural network on neuromorphic hardware for energy-efficient unidimensional SLAM. *arXiv* **2019**, arXiv:1903.02504.
30. Li, X.; Wang, W.; Xue, F.; Song, Y. Computational modeling of spiking neural network with learning rules from STDP and intrinsic plasticity. *Phys. A Stat. Mech. Its Appl.* **2018**, *491*, 716–728. [[CrossRef](#)]
31. Cassidy, A.; Andreou, A.G.; Georgiou, J. A combinational digital logic approach to STDP. In Proceedings of the 2011 IEEE international Symposium of Circuits and Systems (ISCAS), Rio de Janeiro, Brazil, 15–18 May 2011; pp. 673–676.
32. Frenkel, C.; Lefebvre, M.; Legat, J.D.; Bol, D. A 0.086-mm 212.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28-nm CMOS. *IEEE Trans. Biomed. Circuits Syst.* **2018**, *13*, 145–158.
33. Diehl, P.U.; Cook, M. Efficient implementation of STDP rules on SpiNNaker neuromorphic hardware. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 4288–4295.
34. Yousefzadeh, A.; Stromatias, E.; Soto, M.; Serrano-Gotarredona, T.; Linares-Barranco, B. On practical issues for stochastic stdp hardware with 1-bit synaptic weights. *Front. Neurosci.* **2018**, *12*, 665. [[CrossRef](#)] [[PubMed](#)]
35. Gerstner, W.; Kistler, W.M.; Naud, R.; Paninski, L. *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*; Cambridge University Press: Cambridge, UK, 2014.
36. Bahoura, M.; Ezzaidi, H. Hardware implementation of MFCC feature extraction for respiratory sounds analysis. In Proceedings of the 2013 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA), Algiers, Algeria, 12–15 May 2013; pp. 226–229.
37. Nakamura, S.; Hiyane, K.; Asano, F.; Nishiura, T.; Yamada, T. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition. In Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece, 31 May–2 June 2000; pp. 965–968.
38. Dennis, J.; Yu, Q.; Tang, H.; Tran, H.D.; Li, H. Temporal coding of local spectrogram features for robust sound recognition. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 803–807.