Ultralow Power Always-On Intelligent and Connected SNN-Based System for Multimedia IoT-Enabled Applications

Qi Liu, *Member, IEEE*, and Zhixuan Zhang

***Abstract*—The recent advances in artificial neural networks (ANNs) have created immense opportunities to achieve excellent results on the Internet of Things (IoT), which serve the uses of various real-time smart applications, such as in computer vision and speech recognition. However, the efficiency of ANNs comes at the expense of a huge number of computational resources. This tends to necessitate larger and wider ANNs unapplicable for embedded systems with limited hardware resources, e.g., mobile and wearable devices. To that end, biologically realistic spiking neural networks (SNNs) are first employed to build an always-on intelligent and connected integrated IoT-enabled system with ultralow power consumption, where we encode the temporal dynamic stimuli into effective, efficient, and reconstructable spike patterns to facilitate the subsequent processing. Herein, neural encoding plays a key role in faithfully describing the temporally rich patterns for downstream cognitive tasks. Therefore, a novel nonlinear piecewise latency coding approach for a fully event-driven SNN system is developed. Moreover, a surrogate postsynaptic potential kernel function is utilized to address the nondifferential nature of the spike generation scheme when using the error backpropagation learning method. The effectiveness of the proposal tandem with SNNs has been corroborated by indicative empirical results on different data sets serving cognitive tasks.**

***Index Terms*—Deep learning, efficient learning, image classification, Internet of Things (IoT), latency coding, speech recognition, spiking neural network.**

I. Introduction

THE RAPID developments in the Internet of Things (IoT) and wireless sensor networks (WSNs) have spurred new capabilities leveraging massively distributed sensing across wide areas [1], and also endowed immense opportunities to connect and expand the communication between humans (or users) and things from the physical world in our daily life. As shown in Fig. 1, the emerging multimedia IoT [2], [3] has attracted on-going interests for researchers and practitioners alike, which integrates inherent capabilities of mobile communication, computer vision, auditory, tactile, and olfactory perception [4], enabling the potentials for edge computing applications, ranging from voice activity detection, text generation, object detection, large-scale image classification to human action recognition, and to name just a few [5]–[7].

The IoT enables the seamless integration of sensors, actuators, and communication devices for real-time sensing, communicating, and remote controlling [8], which, to some extent, is attributed to the advances in deep learning or artificial intelligence (AI). Artificial neural networks (ANNs), as one of the most profound AI approaches, have achieved their fatal success via leveraging big data generated from widespread WSN sensors and ever-growing computing capability. However, with the almost double increase of mobile data traffic every year, e.g., in massive multiple-input–multiple-output system [9], the high requirements of network bandwidth, transmission quality, energy consumption, and processing capacity of nodes limit the enabling IoT applications with deep learning. What is more, it becomes much more challenging owing to the substantially more complex processing required, and the lack of sensor nodes able to operate for a long time without having to replace batteries. Taking the audio processing as an example, the data rate associated with audio signals prohibits the continuous transmission of raw data, as best-in-class radios consuming 5 nJ/bit (i.e., $\sim$500 μW of power) will drain a button cell battery in $\sim$15 days [10], [11]. No significant battery life extension is allowed by on-chip audio compression. To achieve a long lifetime, sensor nodes of WSN require to be intelligent enough that they generally transmit aggregate data, while transmitting samples only events of interest occur. This inspires ANNs potentially being fully event-driven[1] networks instead of data-driven ones.

On the other hand, one fundamental challenge in the distributed IoT enablement with deep learning is the impossibility of hosting the intelligence entirely in the cloud, although this is the mainstream approach followed today in a simple sensing framework. This is because of the increasing density in wireless communications [12]. At the same time, allocating the intelligence entirely to intelligent sensors (e.g., microphones and cameras) is not a good choice either, as this will become too power costly on the side of sensor nodes (i.e., a long lifetime would be severely limited by the energy cost of

Manuscript received 12 September 2021; revised 27 October 2021 and 15 January 2022; accepted 6 February 2022. Date of publication 10 February 2022; date of current version 24 August 2022. *(Corresponding author: Zhixuan Zhang.)*

Qi Liu is with the School of Future Technology, Guangzhou International Campus, South China University of Technology, Guangzhou 511442, Guangdong, China (e-mail: qiliu47-c@my.cityu.edu.hk).

Zhixuan Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China (e-mail: zhangzhixuan77@gmail.com).

Digital Object Identifier 10.1109/JIOT.2022.3150307

[1] Different from the data-driven ANNs, SNNs only trigger a postsynaptic potential (PSP) function when one of spikes is detected.

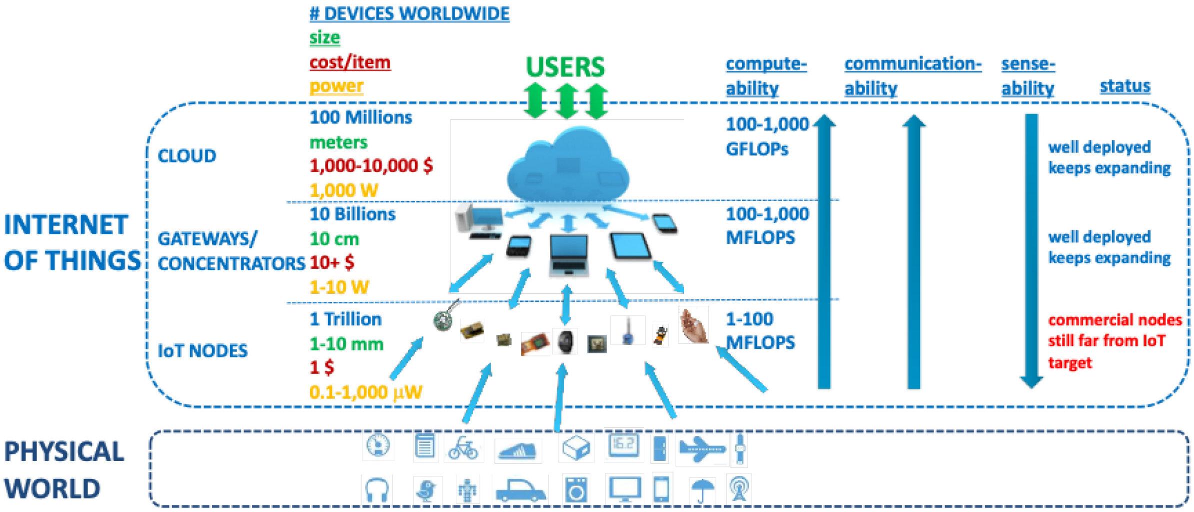

Fig. 1. Block scheme of generic multimedia IoT system.

local computation). In other words, the task should lie in the exploration of innovative processing frameworks that can be optimally distributed between nodes and the cloud (or other intermediate levels of processing). For example, processing at the sensor node level needs to be performed extremely efficient to operate in an event-driven way to drastically limit power consumption and data deluge on the cloud, to further allow a global understanding, detection of acoustic events, human actions, and other activities. To summarize, to deal with the inevitably large amount of data produced by WSN, event-driven schemes are imperative, so that wireless communications from the sensor nodes to the cloud are selectively triggered by activities (e.g., acoustic events), rather than being continuous.

As a compelling use case, strong demand for always-on intelligent and highly connected energy-autonomous integrated systems is rising from the recent convergence of IoT and embedded AI. Such systems continuously monitor sensor signals, detect the occurrence of events of interest, make sense of events through on-chip data analytics, and wirelessly provide the cloud with fine-grain big data for event comprehension. Intelligent and connected devices are expected to be deployed in smart cities, smart homes, industrial plants, wearables, connected cars, and several other applications [13] that benefit from the creation of large networks of connected devices responding intelligently to the stimuli coming from the environment. Nevertheless, with the boost of performance, the whole network architecture from deep learning becomes deeper and wider, which leads to much higher requirements of computational resources and storage space for the inference process.

The neuroscience research offers a bountiful source of inspiration for building human-like computational intelligence systems. Notably, the brain-inspired spiking neural networks (SNNs), which are considered as the third generation of neural network models, have shown great potential with high performance, energy-efficient computing [14]. Unlike traditional ANNs, the biologically realistic SNN models explicitly incorporate the concept of time into the computation. They encode and represent information by the precisely timed spikes, therefore, making SNN a promising candidate for processing temporally rich signals, such as speech and action [15]. By incorporating the time into the computation, it has been demonstrated that SNN models are potentially more efficient than ANN counterparts for data processing [16].

Similar to ANNs, the feedforward computational SNN models for pattern recognition tasks are comprised of three parts: 1) encoding; 2) learning; and 3) readout layers. Herein, the encoding layers are considered as a feature extractor to encode the raw images (or spectrograms for speech recognition) to spike trains, resulting in the spatial–temporal patterns, and they are classified by the learning and readout layers with synaptic weights. Thus, it can be seen that the accuracy of feature extraction directly affects the whole SNN system's performance. Our work is mainly focused on the former encoding part with a novel coding approach. SNNs, mimicking brain functionality, make full use of only addition operations, instead of multiply-and-accumulate (MAC) operations in standard ANNs, and enable to take advantage of significantly reducing the computational complexity. This is in part attributed to different efficient encoding approaches, including two representatives: 1) rate based and 2) temporal based. In rate-based encoding approaches, patterns are encoded by the average number of spikes over a period of time and each spike will trigger memory accesses to load neural network parameters, which require to be fetched from on- or off-chip memory, leading to a relatively high power consumption with the corresponding rate-based neural networks. Thereby, temporal-based encoding techniques are fertile research ground that merits further investigation. Inspired by SNNs, it has been widely used in different cognitive applications in our daily life, including waking up keywords for connected audio devices [17] and classifying images [18]. However, they are built on the basis of computationally demanding rate-based encoding scheme.

In this work, to address the above problems, an ultralow power always-on intelligent and connected system is proposed

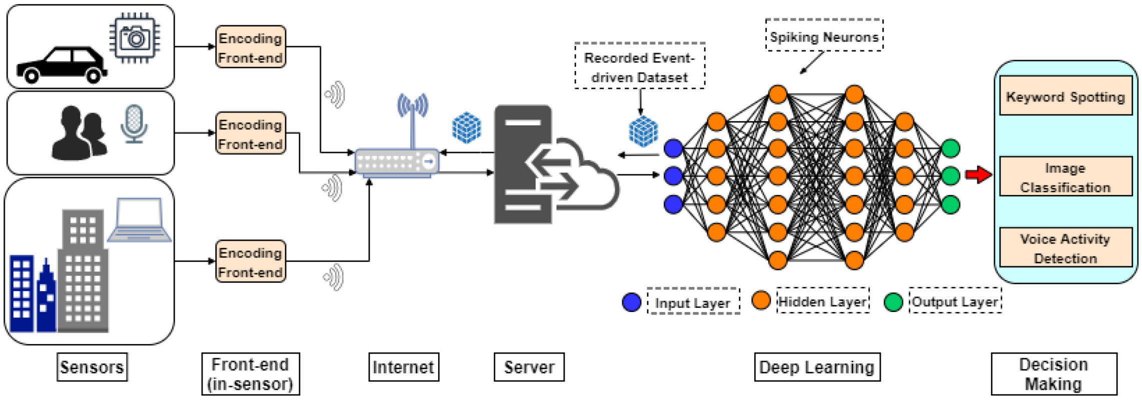

Fig. 2. Block scheme of the proposed multimedia IoT-enabled framework, which composed of a large number of actuators and sensors performing sensing and data processing, and smart-sensing capabilities, such as prediction, categorization, and decision making. With the aid of deep learning, intelligent sensing and decision making will reduce economic losses and improve the safety of smart industrial environments.

from the viewpoint of IoT-enabled applications. Our work takes notable contributions summarized as follows.

1) An always-on intelligent SNN-based auditory and visual system is developed, taking the advantage of IoT enablement.
2) Considering the energy-efficient computing, a novel nonlinear piecewise latency coding method is designed, combined with SNNs to achieve not only efficiently computational complexity but also superior accuracy on the applications of image classification, keyword spotting, and voice activity detection.

The remainder of this article is organized as follows. In Section II, we show the sparse temporal feature encoding using the piecewise latency coding approach. In Section III, we explain each individual system of the proposed human–robot auditory interface in detail. Then, we present the experimental results on the learning capability and energy efficiency of the proposed system in Section IV. Finally, we conclude this article in Section V.

II. Proposed IoT-Enabled Always-On Intelligent System

Complexity is one of the major issues for deep learning models, requiring extra effort to resolve. As many industrial IoT devices are mobile and small in size, they have limited computational power, memory, and battery life, requiring them to offload heavy computations. However, this may not always be possible, for several reasons. Since security and privacy issues have become a great challenge in the IoT [19], critical data should not be transmitted on the Internet. Similarly, offloading the data and computations to the cloud generates high transfer latency, which may not be suitable for many time-critical real-time applications. Thus, it is of utmost importance to devise efficient deep learning algorithms that can process data locally on IoT devices.

Fig. 2 shows the proposed IoT-enabled framework for ultralow power always-on intelligent SNN-based system. The proposed architecture introduces intelligent sensors (e.g., microphones, cameras, and other electronic devices) both as sensing and processing frontend, which converts the recorded raw data into event-based information representation and extracts relevant information on events. The primary motivation is to process information locally (in-sensor) and transmit only aggregate information to the cloud (or server), eliminating the prohibitive power consumption associated with the wireless transmission and communication of detailed samples. We describe individual systems in detail.

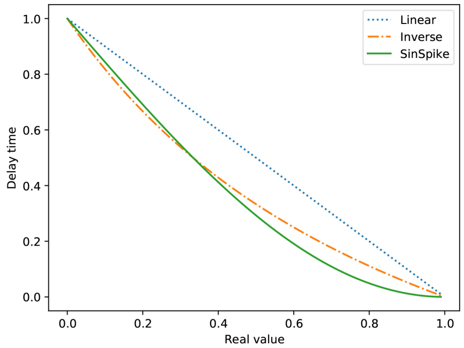

Fig. 3. Illustration of the piecewise latency coding method.

A. Sparse Temporal Feature Encoding

In this section, we aim at generating spike patterns using the proposed nonlinear piecewise latency coding method. Different from the data representation of the ANNs, the information is represented and exchanged via stereotypical action potentials or spikes in the SNNs. The firing rate and temporal structure of the spike train[2] are both considered as important information carriers in the biological neural systems.

In order to process information locally, samples recorded from different sensors should be encoded into events. Herein, the idea of time-to-first-spike (TTFS) scheme can be applied to achieve efficient computation. The conventional TTFS using linear latency coding, as the common representative of temporal encoding method, has been widely used for SNNs. The function of linear latency coding method is plotted in Fig. 3.

[2] A spiking neuron performs a nonlinear transformation over the analog current fed to it through synapses to generate a point process of stereotyped events in its membrane potential, called *action potentials* or *spikes*. The output point process of spikes is also called *spike train* [20].

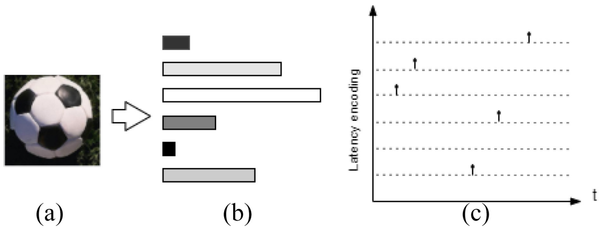

Fig. 4. Illustration of the latency encoding method, where the intensity value is encoded into the spike timing per pixel based on the reverse coding rule. Each horizontal bar represents the intensity value of image pixel proportionally, that is, a brighter pixel corresponding to the longer bar. Thus, the resulting spike fires earlier in the encoding window. (a) Raw image. (b) Pixel luminance. (c) Encoding window time.

It fires only one spike at most per neuron for each input during each inference pass and enjoys fewer spikes in comparison with the rate-based encoding counterpart. Following the reverse coding rule that the larger the pixel is, the shorter the delay time to fire a spike, as illustrated in Fig. 4.

However, it is hoped that those features with smaller real values, such as background noise and edge information, can be encoded more later, while ones with larger real values (i.e., carried more important information) should go into the subsequent neural network earlier and others keep with appropriate delay time. The reason behind that is because we expect the proposed encoding method to enjoy the background noise suppression as well as to preserve the edge information for further performance improvement. Moreover, based on the Weber–Fechner law,[3] the brightness response of the human eye is logarithmic over a wide range of luminance. The logarithmic response is also only valid within the normal luminance range. More detailed studies have shown that the brightness response of the human eye is very complex over a wider range of brightness and has to be described by some piecewise functions.

Inspired by the Weber–Fechner law, a novel nonlinear piecewise latency coding approach for SNNs is designed to achieve not only efficiently computational complexity but also superior accuracy on the applications of image classification and speech recognition. Over a large enough time interval (called time window), the real-valued pixel intensities of input image are mapped to the spike timings for SNNs during inference. The time step is used to keep track of the discrete time, and the total time steps (latency) required are dictated by the desired inference accuracy.

As is well known, the traditional linear latency coding approach has been used intensively in temporal encoding schemes. Given the normalization data X_i, the corresponding spike time for each real value p is shown as

$$t_i = T_{\max} - X_i * (T_{\max} - T_{\min}) \tag{1}$$

where $T_{\max}$ and $T_{\min}$ represent the boundary of the encoding window. From (1), we can see that the larger the real value is, the shorter the delay time is [21]. It is hoped that those features with smaller real values, such as background noise and edge information, can be encoded more later, while ones with larger real values (i.e., carried more important information) should go into the subsequent neural network earlier and the rest keeps with appropriate delay time. The reason behind that is because we expect the proposed encoding method to enjoy the background noise suppression as well as to preserve the edge information for further performance improvement. This motivates us to devise an efficient piecewise latency coding approach for SNNs. To be specific, it is designed as

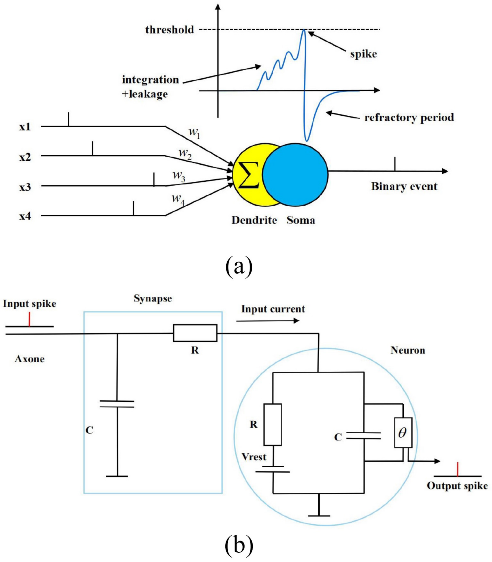

Fig. 5. Brain-like integrate-and-fire model corresponding to its physical circuit. (a) Time course of the membrane potential of a LIF neuron is driven by several constant input currents $\mathbf{x}_i$. The dendrite plays the role of collecting signals from other neurons and transmitting to the "central processing unit," called soma, which performs the function of nonlinear processing. When the sum of total input exceeds a certain threshold, an output spike is generated and then delivered to other neurons by the axon. After firing a spike, the membrane potential is reset to V_{rest}. (b) Equivalent physical circuit of LIF neuron.

$$t_i = \begin{cases} T_{\max} - X_i \times (T_{\max} - T_{\min}), & \text{if } X_i < \alpha \\ \left(\frac{2}{X_i+1} - 1\right) \times (T_{\max} - T_{\min}) + T_{\min}, & \text{if } \alpha < X_i < \beta \\ \left(1 - \sin(\frac{\pi}{2} \times X_i)\right) \times T_{\max}, & \text{if } X_i > \beta \end{cases} \tag{2}$$

where $\alpha = 0.4$ and $\beta = 0.8$ suffice to guarantee the satisfactory performance in our work.

B. Spiking LIF Neuron Model

The leaky-and integrate fire (LIF) neuron model is commonly used for SNNs owing to its simpleness and practicalness, as shown in Fig. 5. In Fig. 5(a), each LIF neuron will accumulate membrane potential from input current and then fire a spike (for single spike SNNs) if its membrane potential reaches the threshold. After that, neurons stay in the refractory period. In Fig. 5 (b), its equivalent physical circuit has been shown with a resistive-capacitor circuit composed of membrane capacitances C_m and membrane resistances R_m. The external input current is used as the driving current to simulate the process of kinetic changes of the membrane potential of LIF neurons. The membrane potential remains the same as the

[3] One application of the Weber–Fechner law is logarithmic coding schemes for neurons.

battery voltage V_{rest} in a resting state without any input current. When the current enters the neuron circuit, the branch current charges the capacitor, and the branch current flows through the resistor to discharge the capacitor.

In order to analyze the circuit, the current conservation law is utilized, leading to two components to form the driving current

$$\begin{aligned} I(t) &= I_R + I_C \\ &= \frac{V(t) - V_{\text{rest}}}{R} + C\frac{dV}{dt} \end{aligned} \tag{3}$$

in which $I_R = (V_R/R)$ and $I_C = C(dV/dt)$ denote the resistive and capacitive currents, respectively. Thus, (3) becomes

$$RC\frac{dV}{dt} = -[V(t) - V_{\text{rest}}] + RI(t). \tag{4}$$

The solution of the linear differential equation is $V(t) = V_{\text{rest}}$ for $t \leq 0$ and given by the membrane potentional of LIF neuron as

$$V(t) = \sum_j w_{ij} \sum_f K(t - t_j^f) + V_{\text{rest}}, \text{ for } t > 0 \tag{5}$$

where the membrane potential $V(t)$ of each neuron obeys the above dynamics. Herein, w_{ij} and t_j^f denote the synaptic connection weight from neuron j to neuron i and the fth spike timing of presynaptic neuron j, respectively. V_{rest} represents the resting potential of LIF neuron. In addition, $K(t - t_j^f)$ is the normalized postsynaptic kernel. In this work, the rectified linear PSP (ReL-PSP) kernel function [22] is applied, defined as

$$K\left(t - t_j^f\right) = t - t_j^f \tag{6}$$

if $t > t_j^f$, and 0 for others.

Different from the conventional ANNs using activation functions (e.g., ReLU) to represent the nonlinear characteristic, SNNs exploit the LIF firing computational mechanism, which is more biologically realistic. The membrane potentials of the presynaptic neurons contribute to the postsynaptic neurons via the positive correlation with the firing time of presynaptic spikes. The action potential, namely, spike, is triggered when the membrane potential $V(t)$ reaching over the threshold (denoted as thr, in general, $thr = 1$) at the arrival time t_r^f. That is

$$V(t_r^f) \geq thr \text{ and } \frac{dV\left(t_r^f\right)}{dt} > 0. \tag{7}$$

$V(t)$ resets to the resting potential V_{rest} after firing and stays at the refractory period for a time period R_a. The conductance of a synapse, a.k.a., synaptic weight, changes depending on the corresponding presynaptic and postsynaptic neurons activities, and the neuron's learning ability is attributed to these activities-dependent synaptic plasticity.

C. SNN-Based Temporal Classifier

First, we apply ON- and OFF-center DoG filters of size $w_1^D \times w_2^D$ on the input patterns, where the DoG filter is a feature enhancement algorithm to increase the visibility of edges and preserve other spatial information. To speed up the training calculation time for feature extraction, we develop a new parallel computing structure composed of three different kernel sizes of convolution maps and max-pooling blocks to further detect the local features, which is motivated by the multiscale theory. Herein, the spike-timing-dependent plasticity (STDP)-based unsupervised learning method [23] is introduced to achieve invariance representation of visual inputs. To employ the STDP rule, the latency of cells is assumed as the firing time of presynaptic and postsynaptic neurons, respectively. That is

$$\begin{cases} \Delta w_{ij} = a^+ * w_{ij} * \left(1 - w_{ij}\right), \text{ if } t_j - t_i \leq 0 \\ \Delta w_{ij} = a^- * w_{ij} * \left(1 - w_{ij}\right), \text{ if } t_j - t_i \geq 0 \end{cases} \tag{8}$$

where $a^+ = 0.004$ and $a^- = -0.003$.

Finally, temporal information are unfolded together for afferent neurons on spike-timing-dependent backpropagation (STDBP)-based SNN classifier [22]. The main derivatives using STDBP learning rule are given by

$$\begin{aligned} \frac{\partial t_j}{\partial w_{ij}} &= \frac{\partial t_j}{\partial V_j\left(t_j\right)} \frac{\partial V_j\left(t_j\right)}{\partial w_{ij}} = \frac{t_i - t_j}{\sum_i w_{ij}} \\ \frac{\partial t_j}{\partial t_i} &= \frac{\partial t_j}{\partial V_j\left(t_j\right)} \frac{\partial V_j\left(t_j\right)}{\partial t_i} = \frac{w_{ij}}{\sum_i w_{ij}} \end{aligned} \tag{9}$$

corresponding to the derivatives of the first spike time t_j with respect to synaptic weights w_{ij} and input spike times t_i, respectively, and $t_j < t_i$.

1) Readout: To output the spikes of interest, the readout part is applied in the last layer of the SNN-based temporal classifier, where each learning neuron corresponds to one category for a classification task. The category of an input pattern will be determined by one of the neurons that generates the lowest spike distance. Here, we utilize the softmax function on the negative values of the spike times in the output layer, to minimize spike times of the desired neurons as well as to simultaneously maximize ones of the undesired neurons. The resulting distance is measured by the cross-entropy loss function

$$\mathcal{L}\left(g, \mathbf{t}^o\right) = -\ln \frac{\exp(-\mathbf{t}^o[g])}{\sum_i \exp(-\mathbf{t}^o[i])} \tag{10}$$

where $\mathbf{t}^o$ represents the vector of the spike times in the output layer and g denotes the desired class index.

III. Experimental Results

We evaluate the proposed system's decision-making capability on cognitive tasks, including keyword spotting, image classification, and voice activity detection. Next, we introduce the experimental setups and stimulus data sets for different cognitive tasks in detail. We perform all the experiments with the Pytorch toolbox, which provides accelerated and memory-efficient training with graphics processing units (GPUs).

A. Evaluation Metrics

1) Energy Calculation: The total computational cost is proportional to the total number of floating-point operations (FLOPs), which is similar to the number of matrix-vector multiplication operations approximately. For per layer l, the FLOPs of ANN can be computed by [24]

$$\mathrm{FLOPs}_{\mathrm{ANN}}(l) = \begin{cases} k^2 \times O^2 \times C_{\mathrm{in}} \times C_{\mathrm{out}}, & \text{if } l \text{ is the convolutional layer} \\ C_{\mathrm{in}} \times C_{\mathrm{out}}, & \text{if } l \text{ is the linear layer} \end{cases} \tag{11}$$

in which k and O denote the sizes of the kernel function and output feature map, respectively. In addition, C_{in} and C_{out} represent the input and output channels, respectively. To calculate the FLOPs of SNN, we define the spiking rate $R_s(l)$ per layer l since SNN only consumes energy when firing spikes, that is

$$\mathrm{R}_s(l) = \frac{\#\text{spikes per layer } l \text{ over all time steps}}{\#\text{neurons per layer } l} \tag{12}$$

which means the average firing rate per neuron. Thus, FLOPS for SNN is

$$\mathrm{FLOPs}_{\mathrm{SNN}}(l) = \mathrm{FLOPs}_{\mathrm{ANN}}(l) \times \mathrm{R}_s(l). \tag{13}$$

Therefore, total inference energy consumption for ANN ($\mathrm{E}_{\mathrm{ANN}}$) and SNN ($\mathrm{E}_{\mathrm{SNN}}$) across all layer is computed as

$$\mathrm{E}_{\mathrm{ANN}} = \sum_l \mathrm{FLOPs}_{\mathrm{ANN}}(l) \times \mathrm{E}_{\mathrm{MAC}} \tag{14}$$

$$\mathrm{E}_{\mathrm{SNN}} = \sum_l \mathrm{FLOPs}_{\mathrm{SNN}}(l) \times \mathrm{E}_{\mathrm{AC}} \tag{15}$$

where E_{AC} and $\mathrm{E}_{\mathrm{MAC}}$ are obtained from a standard 45-nm complementary metal-oxide-semiconductor (CMOS) process, viz., $\mathrm{E}_{\mathrm{MAC}} = 4.6$ pJ and $\mathrm{E}_{\mathrm{AC}} = 0.9$ pJ for 32 bit FP [25].

2) Spike Time Rate:

$$\text{Spike time rate} = \frac{\#t_0}{\#\text{neurons}} \tag{16}$$

where $t_0 < t$ in each time window. Encoding time is the sum of all t_0.

B. Stimulus Data Sets

1) Google Speech Command Data Set v2 [26]: This data set is composed of 65K 1 s long utterances of 30 short keywords, by thousands of different people, with each utterance comprised of only one keyword. We select 10 words out of 30 words in the corpus, which are commonly used commands, viz., “Yes,” “No,” “Up,” “Down,” “Left,” “Right,” “On,” “Off,” “Stop,” and “Go.” The remaining 20 commands are “Bed,” “Bird,” “Dog,” “Cat,” “House,” “Happy,” “Wow,” “Sheila,” “Marvin,” and “Tree,” and ten numbers from 0 to 9. The data set is split into training, validation, and test sets with the ratio of 80% : 10% : 10%, while guaranteeing that the audio utterances from the same person stay in the same set [27]. The utterances are segmented into frames of 40-ms length with a stride of 20 ms, leading to the size of input Mel-frequency cepstral coefficients (MFCCs) (i.e., features) being 49 × 13. The features are extracted and concatenated into a spectrogram, and then encoded via events before inputting to the SNNs for recognition.

2) MNIST [28] and Caltech101 [29] Data Sets: The MNIST data set consists of 60K 28×28 grayscale images (i.e., handwritten digits 0–9) for training and 10K for testing. The Caltech101 data set contains 101 categories and each category has 40–800 300×200 images with complex background noise. We only evaluate all compared models on face and motor bike categories, where 200 randomly selected images per category are used for training and the rest for testing.

3) TIDIGITS [30] and RWCP [31] Data Sets: For the voice activity detection task, RWCP and TIDIGITS data sets are utilized. RWCP data set consists of high-fidelity natural sound samples recorded in the real acoustic environment at a sampling rate of 16 kHz. We use the same 10 environmental sound classes, including ‘cymbals,” “horn,” “phone4,” “bells5,” “kara,” “bottle1,” “buzzer,” “metal15,” “whistle1” and “ring.” We randomly selected 40 samples from each class, wherein 20 samples are used to train an ANN-based sound classifier and the rest are used for evaluation. The TIDIGITS data set comprises of reading digit sequences of variable lengths from 21 dialectical regions of the United States. We use the subset of isolated spoken digits from 11 classes (i.e., “zero” to “nine” and “oh”), which consists of 2464 train and 2486 test utterances. The utterances are spoken by 111 male and 114 female speakers at a sampling rate of 20 kHz.

C. Results

1) Comparison With the Linear Latency Coding Approach: To investigate the efficiency of the proposed nonlinear piecewise latency coding method, we combine with SNN and conduct tests on different data sets, as shown in Table I. Based on the same architecture, yet with different encoding approaches, we can see that the proposed coding method is superior to its counterpart, in terms of accuracy and computationally complexity. In addition, we observe that the reason why ours perform better is that ours can emit more important information, which can be demonstrated by the high spike time rate.

2) Decision Making on Different Cognitive Tasks: Keyword spotting is a critical component for enabling speech-based user interactions on smart devices [33]. For privacy reasons, these devices rely on the user to preface their commands with a keyword, such as “Hey Siri” to wake up iPhones [27]. Due to its always-on nature, keyword spotting application has a highly constrained power budget and typically runs on tiny microcontrollers with limited memory and compute capability. We compare the proposed system with its competitors to detect keywords on Google Speech Command data set v2, as shown in Table II. In [17], a teacher–student training scheme is applied to approximate the discontinuous nature in SNN via sharing the weights between ANN and SNN. Since the errors in SNN do not require to backpropagate and the spikes are utilized to transmit the information in ANN, the whole architecture can achieve efficient inference. Additionally, Masquelier and Thorpe [32] employed the idea of ANN-to-SNN conversion method, and trained an ANN at first, then approximated the pretrained ANN with an SNN equivalent. Although it can achieve comparable performance

TABLE I
COMPARISON RESULTS AMONG DIFFERENT DATA SETS TO EVALUATE THE EFFECTIVENESS AND EFFICIENCY OF THE PROPOSED NONLINEAR PIECEWISE LATENCY CODING METHOD

Dataset / Encoding	MINST			Caltech101		
	Accuracy	Spike time rate	Encoding time	Accuracy	Spike time rate	Encoding time
Linear latency	97.51%	12.88%	706ms	97.83%	59.31%	11194ms
Piecewise latency	**97.94%**	**13.62%**	**542ms**	**97.96%**	**64.78%**	**9019ms**

Dataset / Encoding	TIDIGITS			RWCP		
	Accuracy	Spike time rate	Encoding time	Accuracy	Spike time rate	Encoding time
Linear latency	94.20%	38.04%	4920ms	99.50%	32.50%	5616ms
Piecewise latency	**94.37%**	**43.42%**	**4140ms**	**99.50%**	**36.77%**	**5031ms**

TABLE II
COMPARISON WITH THE EXISTING SNNS IN TERMS OF SPIKE COUNT, ACCURACY, AND INFERENCE ENERGY RATIO. SC AND ACC DENOTE THE NUMBER OF SPIKE COUNT AND ACCURACY, RESPECTIVELY. $\mathbf{E_{SNN}/E_{ANN}}$ REPRESENTS THE INFERENCE ENERGY RATIO BETWEEN SNN AND ANN, WHERE THE ENERGY CONSUMPTION OF ANN IS CONSIDERED AS THE REFERENCE, NAMELY, $\mathbf{E_{ANN}} = 1\times$

Architecture	SC	ACC	$\mathbf{E_{SMLP}/E_{MLP}}$
Rate-bsed SNN [17]	231	73.82%	0.12 ×
ANN-converted SNN [32]	13535	82.22%	4.65 ×
Ours	**83**	**84.24%**	**0.02 ×**
ANN (baseline)	-	84.78%	1 ×

TABLE III
RECOGNITION ACCURACIES OF DIFFERENT MODELS ON THE TIDIGITS DATA SET

Model	Accuracy (%)
Zhang et al. [34]	92.3
Tavanaei et al. [35]	91.0
Abdollahi et al. [36]	78.7
Ours	**94.37**

TABLE IV
CLASSIFICATION ACCURACIES OF DIFFERENT MODELS ON THE MNIST DATA SET

Model	Neural coding	Accuracy (%)
Hussain et al. [38]	Rate-based	90.3
Zhao et al. [39]	Spike-based	91.3
Querlioz et al. [40]	Spike-based	93.5
O'Connor et al. [18]	Rate-based	94.1
Diehl et al. [41]	Spike-based	95.0
Kheradpisheh et al. [37]	Spike-based	98.4
Ours	Spike-based	**97.9**

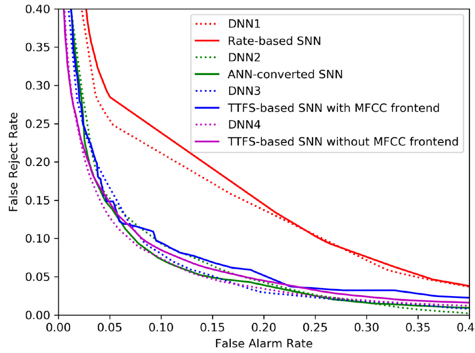

Fig. 6. FRR versus FAR for keyword spotting.

with ours, it is computationally demanding. From Table II, the proposed method outperforms others with less spike count, higher accuracy, and efficient computation, and is also more approaching the deep neural network (DNN) baseline with the same architecture (multilayer perceptron: 49×13-650-10). As illustrated in Fig. 6, the false reject rate (FRR) versus false alarm rate (FAR) is plotted to further evaluate the effectiveness of the proposed method, where we expect to obtain smaller values in both FRR and FAR. That means, the smaller values of FRR and FAR are, the better the classification performance. The results are consistent with the above analysis.

As an integral part of different speech communication systems, e.g., speech encoding and echo cancelation, voice activity detection plays an important role in distinguishing human voices from background noises. In the context of speech and speaker recognition, for example, voice activity detection can avoid unnecessary coding and transmission of silence packets in Voice over Internet Protocol (VoIP) applications via discarding the nonspeech section, thereby saving on computation and network bandwidth. As well, those noise segments can be extracted for the use of noise modeling and speech enhancement. Additionally, it is advantageous to have lower average power consumption in mobile handsets, higher average bit rate for data transmission, or higher capacity on storage chips. As shown in Table III, the proposed method presents the best result among all competitors in terms of accuracy up to 94.37%. Herein, unless otherwise stated, all compared systems employ the same architecture, that is, convolutional neural network (CNN) with a size of 20×30-(4C5-P2, 30C7-P4, 100C9-P6)-240-11.[4] Here, we apply multiscale CNN to parallelly connect.

To further verify the proposed method, we conduct an experiment on the MNIST data set to classify images. From Table IV, the method in [37] achieves the best result, at an accuracy up to 98.4%, and followed by ours (97.9%). The proposed method performs better than other rate-based and spike-based counterparts.

IV. CONCLUSION

In this work, on the basis of IoT enablement, an ultralow power always-on intelligent and connected system is developed, which is based on the deep learning technique, namely, brain-inspired SNN. Herein, the information

[4] C and P denote the Convolution and Pooling layers, respectively.

is represented and exchanged via stereotypical action potentials or spiking events, instead of real values. To that end, a new nonlinear piecewise latency coding method is devised to encode stimuli into spike timings, which contributes to achieving efficient computation. Then, combined with SNN, the proposed system outperforms the existing competitors for keyword spotting, voice activity detection, and image classification, as corroborated by indicative empirical results on different data sets, respectively.

REFERENCES

[1] M. Alioto, *Enabling the Internet of Things: From Integrated Circuits to Integrated Systems*. Cham, Switzerland: Springer, 2017.

[2] S. Rani, S. H. Ahmed, R. Talwar, J. Malhotra, and H. Song, "IoMT: A reliable cross layer protocol for Internet of Multimedia Things," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 832–839, Jun. 2017.

[3] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.

[4] N. Imam and T. Cleland, "Rapid Online learning and robust recall in a neuromorphic olfactory circuit," *Nature Mach. Intell.*, vol. 2, pp. 181–191, Mar. 2020.

[5] A. M. Ortiz, D. Hussein, S. Park, S. N. Han, and N. Crespi, "The cluster between Internet of Things and social networks: Review and research challenges," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 206–215, Jun. 2014.

[6] M. Wang, D. Xiao, and Y. Xiang, "Low-cost and confidentiality-preserving multi-image compressed acquisition and separate reconstruction for Internet of Multimedia Things," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1662–1673, Feb. 2021.

[7] Q. Liu and J. Wu, "Parameter tuning-free missing-feature reconstruction for robust sound recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 1, pp. 78–89, Jan. 2021.

[8] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[9] Q. Liu, X. Li, and H. Cao, "Two-dimensional localization: Low-rank matrix completion with random sampling in massive MIMO system," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3628–3631, Sep. 2021.

[10] C. Pham, P. Cousin, and A. Carer, "Real-time on-demand multi-hop audio streaming with low-resource sensor motes," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw. Workshops*, 2014, pp. 539–543.

[11] G. Zhao, H. Ma, Y. Sun, and H. Luo, "Distributed audio synchronization scheme using audio endpoint in WASNs," in *Proc. IEEE Int. Symp. World Wireless Mobile Multimedia Netw.*, 2011, pp. 1–9.

[12] P. Pace, G. Fortino, Y. Zhang, and A. Liotta, "Intelligence at the edge of complex networks: The case of cognitive transmission power control," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 97–103, Jun. 2019.

[13] J. Branger and Z. Pang, "From automated home to sustainable, healthy and manufacturing home: A new story enabled by the Internet-of-Things and industry 4.0," *J. Manag. Anal.*, vol. 2, no. 4, pp. 314–332, 2015.

[14] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[15] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, Nov. 2019.

[16] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan./Feb. 2018.

[17] E. Yilmaz, Ö. B. Gevrek, J. Wu, Y. Chen, X. Meng, and H. Li, "Deep convolutional spiking neural networks for keyword spotting," in *Proc. INTERSPEECH*, 2020, pp. 2557–2561.

[18] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Front. Neurosci.*, vol. 7, p. 178, Oct 2013.

[19] M. Frustaci, P. Pace, and G. Aloi, "Securing the IoT world: Issues and perspectives," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2017, pp. 246–251.

[20] N. Anwani and B. Rajendran, "Normad-normalized approximate descent based supervised learning rule for spiking neurons," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2015, pp. 1–8.

[21] M. Zhang *et al.*, "An efficient threshold-driven aggregate-label learning algorithm for multimodal information processing," *IEEE J. Sel. Topics Signal Process.*, vol. 14, no. 3, pp. 592–602, Mar. 2020.

[22] M. Zhang *et al.*, "Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks," 2020, *arXiv:2003.11837*.

[23] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through spike timing dependent plasticity," *PLOS Comput. Biol.*, vol. 3, no. 2, pp. 1–11, 2007.

[24] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," 2020, *arXiv:2010.01729*.

[25] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers (ISSCC)*, 2014, pp. 10–14.

[26] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.

[27] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," 2018, *arXiv:1711.07128*.

[28] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[29] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2004, p. 178.

[30] R. G. Leonard and G. Doddington, *Tidigits Speech Corpus*, Linguistic Data Consortium, Philadelphia, PA, USA, 1993.

[31] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proc. 2nd Int. Conf. Lang. Resources Eval. (LREC)*, Athens, Greece, May 2000, pp. 1–4.

[32] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Front. Neurosci.*, vol. 13, p. 95, Mar. 2019.

[33] G. Tucker, M. Wu, M. Sun, S. Panchapagesan, G. Fu, and S. Vitaladevuni, "Model compression applied to small-footprint keyword spotting," in *Proc. Interspeech*, 2016, pp. 1878–1882.

[34] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2635–2649, Nov. 2015.

[35] A. Tavanaei and A. S. Maida, "A spiking network that learns to extract spike signatures from speech signals," *Neurocomputing*, vol. 240, pp. 191–199, May 2017.

[36] M. Abdollahi and S.-C. Liu, "Speaker-independent isolated digit recognition using an AER silicon cochlea," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2011, pp. 269–272.

[37] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Netw.*, vol. 99, pp. 56–67, Mar. 2018.

[38] S. Hussain, S. Liu, and A. Basu, "Improved margin multi-class classification using dendritic neurons with morphological learning," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 2640–2643.

[39] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feedforward categorization on AER motion events using cortex-like features in a spiking neural network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 1963–1978, Sep. 2015.

[40] D. Querlioz, O. Bichler, P. Dollfus, and C. Gamrat, "Immunity to device variations in a spiking neural network with memristive nanodevices," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 288–295, May 2013.

[41] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Front. Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.