



## Article

# A Biologically Inspired Movement Recognition System with Spiking Neural Networks for Ambient Assisted Living Applications

Athanasios Passias <sup>1</sup>, Karolos-Alexandros Tsakalos <sup>1</sup>, Ioannis Kansizoglou <sup>2</sup>, Archontissa Maria Kanavaki <sup>3</sup>, Athanasios Gkrekidis <sup>3</sup>, Dimitrios Menychtas <sup>3</sup>, Nikolaos Aggelousis <sup>3</sup>, Maria Michalopoulou <sup>3</sup>, Antonios Gasteratos <sup>2</sup> and Georgios Ch. Sirakoulis <sup>1,\*</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; apassias@ee.duth.gr (A.P.); ktsakalo@ee.duth.gr (K.-A.T.)

<sup>2</sup> Department of Production and Management Engineering, Democritus University of Thrace, 67100 Xanthi, Greece; ikansizo@pme.duth.gr (I.K.); agaster@pme.duth.gr (A.G.)

<sup>3</sup> School of Physical Education and Sport Science, Democritus University of Thrace, 69100 Komotini, Greece; mkanavak@affil.duth.gr (A.M.K.); agkrekid@phyed.duth.gr (A.G.); dmenycht@affil.duth.gr (D.M.); nagelous@phyed.duth.gr (N.A.); michal@phyed.duth.gr (M.M.)

\* Correspondence: gsirak@ee.duth.gr

**Abstract:** This study presents a novel solution for ambient assisted living (AAL) applications that utilizes spiking neural networks (SNNs) and reconfigurable neuromorphic processors. As demographic shifts result in an increased need for eldercare, due to a large elderly population that favors independence, there is a pressing need for efficient solutions. Traditional deep neural networks (DNNs) are typically energy-intensive and computationally demanding. In contrast, this study turns to SNNs, which are more energy-efficient and mimic biological neural processes, offering a viable alternative to DNNs. We propose asynchronous cellular automaton-based neurons (ACANs), which stand out for their hardware-efficient design and ability to reproduce complex neural behaviors. By utilizing the remote supervised method (*ReSuMe*), this study improves spike train learning efficiency in SNNs. We apply this to movement recognition in an elderly population, using motion capture data. Our results highlight a high classification accuracy of 83.4%, demonstrating the approach's efficacy in precise movement activity classification. This method's significant advantage lies in its potential for real-time, energy-efficient processing in AAL environments. Our findings not only demonstrate SNNs' superiority over conventional DNNs in computational efficiency but also pave the way for practical neuromorphic computing applications in eldercare.

**Keywords:** ambient assisted living (AAL); spiking neural networks (SNNs); reconfigurable neuromorphic processors; elderly activity recognition; energy-efficient processing; real-time processing; activity monitoring



**Citation:** Passias, A.; Tsakalos, K.-A.; Kansizoglou, I.; Kanavaki, A.M.; Gkrekidis, A.; Menychtas, D.; Aggelousis, N.; Michalopoulou, M.; Gasteratos, A.; Sirakoulis, G.C. A Biologically Inspired Movement Recognition System with Spiking Neural Networks for Ambient Assisted Living Applications. *Biomimetics* **2024**, *9*, 296. <https://doi.org/10.3390/biomimetics9050296>

Academic Editor: Simon X. Yang

Received: 31 December 2023

Revised: 20 February 2024

Accepted: 12 March 2024

Published: 15 May 2024



**Copyright:** © 2024 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

Contemporary research conducted by the World Health Organization predicts a significant increase in the elderly population and longer lifespans globally in the coming decades. As more seniors choose to “age in place” instead of moving to nursing homes, eldercare becomes increasingly crucial to supporting their independence and maintaining their health [1,2]. This situation puts significant pressure on the healthcare sector, necessitating the implementation and enhancement of ambient assisted living (AAL) systems [3–6]. AAL systems monitor movements and detect falls in addition to recognizing activities, gestures, and emotions, and they aim to provide an integrated and effective solution for assisted living.

Current approaches predominantly utilize wearable devices, such as wristbands for motion capture and action recognition; audio devices for recording low-level sounds during daily activities; and marker-based systems, such as Vicon and Qualisys, for accurate 3D pose representation and action recognition [7–9]. The emergence of deep neural networks (DNNs) has significantly improved activity recognition in ambient assisted living (AAL) applications [10]. However, these methods are typically characterized by high energy consumption and computational demands, making them less suitable for continuous, real-time applications in domestic environments. In contrast to biological neurons that communicate with spikes, DNNs utilize mathematical calculations between neurons [11,12]. The need for an efficient, adaptive, and less resource-intensive technology is evident, especially one that can accurately recognize and interpret the complex array of human movements that are characteristic of the elderly.

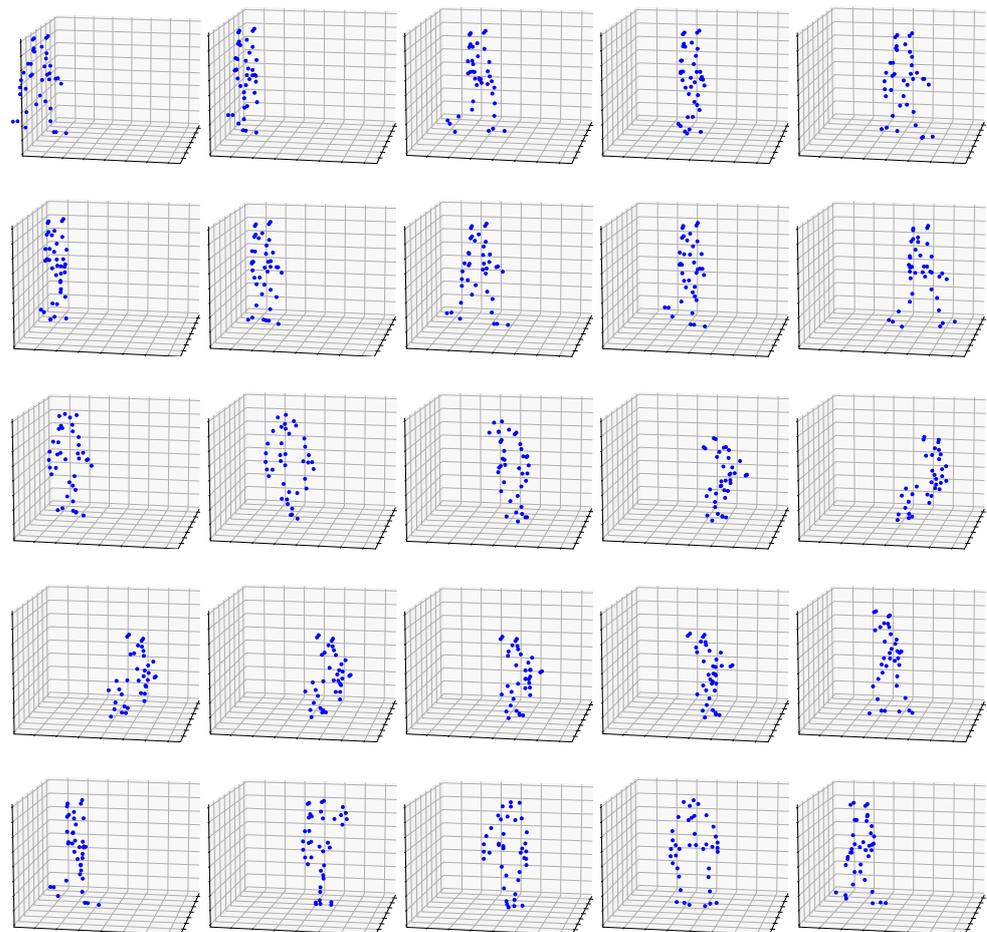
In response to and to address the limitations of DNNs, there is a growing research focus on neuromorphic computing, particularly spiking neural networks (SNNs), which can be implemented in neuromorphic processors and promise a more naturalistic computational paradigm that provides a more energy-efficient approach than DNNs implemented using GPUs [13–15]. Despite their potential and advantages, such as fast inference, analog computation, and low energy consumption, the deployment of SNNs in practical AAL applications remains nascent, with several challenges in their non-differentiable nature, learning efficiency, adaptability, and hardware implementation still requiring further investigation before transferring knowledge from classic AAL approaches to SNNs. This study situates itself within this context, aiming to bridge the gap between the potential of neuromorphic computing and its real-world application in eldercare.

To meet the increasing demand for real-time and large-scale neuromorphic processors, previous studies have proposed a reconfigurable neuromorphic model based on field-programmable gate array (FPGA) technology and asynchronous cellular automata [16–23]. These models offer hardware-efficient solutions for various applications, including Parkinson's treatment emulation, central pattern generation for hexapod robots, spike-timing-dependent synaptic plasticity, neural integrators, tumor immunotherapy, and ergodic cellular automaton neuron models [16–19]. Implementing these models in FPGAs offers lower power consumption and hardware requirements compared with conventional models. The asynchronous cellular automaton neuron (ACAN) model, initially introduced in [24] and further optimized in [22], reproduces neuromorphic behaviors of cortical neurons using discrete-state dynamics, and it requires fewer hardware resources. Its dynamic adjustability after implementation makes it a versatile and suitable solution for implementing SNNs, and it is ideal for real-time neuromorphic applications [25,26], including movement classification tasks.

The contributions of this paper are multifold. Firstly, we propose using the ACAN model as a building block for neuromorphic networks, validating its ability to reproduce a total of 20 cortical spiking patterns on an FPGA and demonstrating its versatility and robustness in diverse neuromorphic modeling scenarios. Secondly, a comprehensive parametric analysis is demonstrated for the MNIST hand-written digit dataset to identify optimal learning configurations for these neuron models by adjusting both the neuron characteristics and learning parameters, significantly enhancing their learning efficiency and applicability. Thirdly, we apply our methodology to our novel movement dataset and the critical task of human movement classification for recognizing basic distinct elderly human movements, such as gait, cutting, standing up, sitting down, and turning. The adoption of the ACAN model and the ReSuMe learning method has been pivotal in achieving precise, adaptive learning for these specific tasks. Our work has yielded promising results for its integration into real-time, holistic AAL systems.

## 2. Dataset

In this section, we describe the data collection methods, dataset division, and preprocessing steps used to train the models discussed in the next section, namely, Section 3. The study involved two senior individuals who repeated five basic action scenarios: *cutting*, *gait*, *sitting down*, *standing up*, and *turning* (Figure 1). The Vicon system assessed the individuals' posture by tracking the position of  $n_j$  joints in the human body, which were represented by 3D coordinates  $(x, y, z)$  and sampled at a consistent rate of 10 ms. The data underwent post-processing and were saved as *.c3d* files, containing joint coordinates, metadata, sensor setups, and specific measurements. After collecting action scenarios for all individuals, we chose to analyze  $n_j = 38$  principal joints. This decision was made since other, additional joints often had missing estimations caused by occlusions during particular activities.



**Figure 1.** Evolving from left to right, the five action scenarios are illustrated in each row, i.e., (1) *cutting*, (2) *gait*, (3) *sitting down*, (4) *standing up*, and (5) *turning*.

Mokka software, which is a tool for analyzing motion kinematics and kinetics [27], was used to transform the *.c3d* files into a tabular format and remove any unnecessary metadata. To maintain consistency, markers that displayed inconsistency or were absent were removed. In order to handle composite movement labels, we conducted manual annotation of the crucial frames that signified a shift in the type of movement, leading to distinct "areas" of movement within the same sample. Given the dataset's constrained and imbalanced nature, the instances were partitioned into segments of fixed length by using a sliding window technique. Segments that intersected with annotated areas were categorized based on their corresponding movement, while the remaining segments were classified as *gait*. The dataset was expanded from 126 to 17,803 samples. In order to maintain balanced classes and prevent any class imbalance, we restricted the number of samples

per class to match the size of the smallest classes for experimentation in Section 4. The dataset was further enhanced by calculating velocity and acceleration data. The samples were then normalized to a range of  $[0, 1]$  as spiking rates. These rates will be provided in the subsequent section.

The basic movement types of *cutting*, *gait*, *sitting down*, *standing up*, and *turning* encompass a combination of gait and labeled movement as described by Menychtas [28]. The original samples were evaluated manually to identify pivotal frames that indicated the shift between various sorts of movements, resulting in the aforementioned 17,803 data points. The samples were divided into segments by using a sliding window of size  $S$  frames, and the frames were categorized into their respective classes. This class was determined either as *gait* or as the specific movement indicated by the label of the original data point. The determination was based on whether the window included the critical area, as identified by the critical frames. More precisely, samples labeled *cutting* and *turning* exhibited a solitary critical area in the middle of the movement. On the other hand, samples labeled *standing up* and *sitting down* featured critical areas at the onset of the movement. Notably, the samples classified as *gait* did not display any critical areas. After the preprocessing step, the number of samples increased, as indicated in Table 1.

**Table 1.** Number of samples by type of movement before and after preprocessing. After preprocessing the original samples, their quantity increased significantly, allowing for more robust training.

	Before Preprocessing	After Preprocessing
Gait	32	3659
Cutting	50	2404
Standing up	22	2751
Sitting down	20	3286
Turning	12	5703
Total	126	17,803

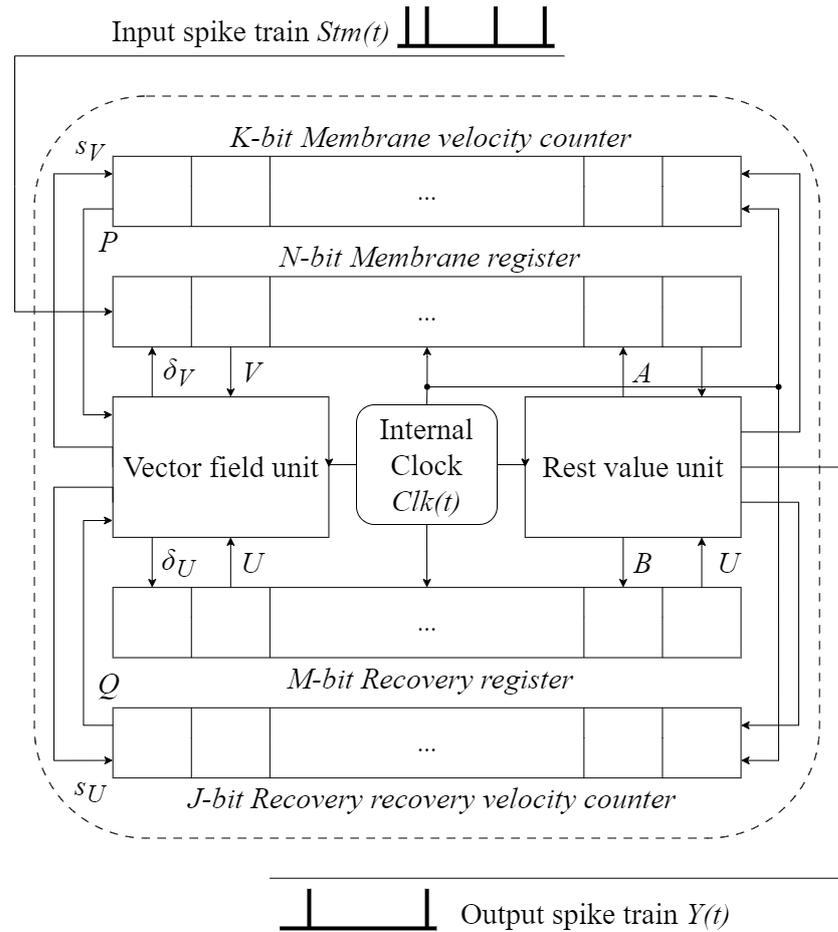
### 3. Method

This section provides an overview of the approach, which encompasses the ACAN model, the *ReSuMe* training method, and the network configuration.

#### 3.1. Asynchronous Cellular Automaton-Based Neuron

The asynchronous cellular automaton-based neuron (ACAN) model is a digital neuron designed for field-programmable gate array (FPGA) optimization, having the ability to replicate various neural activities.

The ACAN architecture, proposed by Matsubara and Torikai in their publication [24], is a digital neuron model that draws inspiration from the Izhikevich model. It is specifically designed for digital systems, with a particular emphasis on its compatibility with FPGAs. The model replicates a range of spiking and bursting patterns observed in cortical neurons [29,30]. The ACAN operates in a generalized configuration where it receives action potentials (spikes) as input, represented by  $Stm(t)$ , modifies its internal variables, and produces output spikes ( $Y(t)$ ). Every ACAN unit is equipped with an internal clock ( $Clk$ ), enabling asynchronous operations among ACAN units as they function independently of a global clock signal (Figure 2).



**Figure 2.** Generalized asynchronous cellular automaton-based neuron (ACAN) model adapted from [22].

The internal state of an ACAN unit is represented by the following four bidirectional shift registers with positive integer bit lengths of  $N$ ,  $M$ ,  $K$ , and  $J$ , respectively:

- The membrane register is an  $N$ -bit bidirectional shift register with an integer state  $V$  in the range of  $(0, \dots, N - 1)$ , representing the membrane potential of the neuron model.
- The recovery register is an  $M$ -bit bidirectional shift register with an internal state  $U$  in the range of  $(0, \dots, M - 1)$ , representing the recovery variable of the neuron model.
- The membrane velocity counter is a  $K$ -bit register with an internal state  $P$  in the range of  $(0, \dots, K - 1)$ , controlling the velocity of membrane potential  $V$ .
- The recovery velocity counter is a  $J$ -bit register with an internal state  $Q$  in the range of  $(0, \dots, J - 1)$ , controlling the velocity of recovery variable  $U$ .

Furthermore, the ACAN’s expected behavior is determined by two logic units, namely, the Vector Field Unit and the Rest Value Unit, which do not retain any memory.

- The Vector Field Unit determines the vector field characteristics for states  $V$  and  $U$ .
- The Rest Value Unit sets the rest values for states  $V$  and  $U$ .

Each field unit comprises logic gates and reconfigurable wires that provide connections between the membrane and rest registers.

The control signals  $(s_V, s_U) \in 0, 1$  and  $(\delta_V, \delta_U) \in \{-1, 0, 1\}$ , which are generated by the Vector Field Unit, are defined as follows:

$$s_V = \begin{cases} 1 & \text{if } P \geq P_h(V, U) \\ 0 & \text{otherwise} \end{cases} \quad s_U = \begin{cases} 1 & \text{if } Q \geq Q_h(V, U) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$\delta_V = D_V(V, U), \delta_U = D_U(V, U) \tag{2}$$

$$F(V, U) = N(\gamma_1(V/N - \gamma_2)^2 + \gamma_3 - U/M)/\lambda, \tag{3}$$

$$G(V, U) = \mu M(\gamma_4(V/N - \gamma_2) + (\gamma_3 + \gamma_5) - U/M)\lambda, \tag{4}$$

$$P_h(V, U) = \lfloor |F^{-1}(V, U)| \rfloor - 1, \tag{5}$$

$$Q_h(V, U) = \lfloor |G^{-1}(V, U)| \rfloor - 1, \tag{6}$$

$$D_V(V, U) = \text{sgn}(F(V, U)), \tag{7}$$

$$D_U(V, U) = \text{sgn}(G(V, U)) \tag{8}$$

The rest value unit generates two signals ( $A, B$ ) that determine the reset values of the ( $V, U$ ) states after a reset is triggered.

$$A = \lfloor \rho_1 N \rfloor, \tag{9}$$

$$B(U) = U + \lfloor \rho_2 M \rfloor \tag{10}$$

where ( $\rho_1, \rho_2$ ) are parameters.

The ACAN's dynamics are characterized by nine hyper-parameters, namely,  $M, N, K, J, \Gamma = (\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \lambda, \mu, \rho_1, \rho_2)$  [22,24]. The aforementioned parameters are pivotal for the variety of spiking patterns that are achievable by the ACAN. These patterns are described in detail in Table 2 and are documented in the study by Matsubara et al. [31]. The ACAN configuration types are commonly utilized in the ensuing parametric analysis.

In a nutshell, the ACAN model functions in the following manner: Input is received by the ACAN in the form of binary spikes, which are weighted by synaptic weight. After considering the input and its previous state, the unit carries out calculations based on Equations (1)–(10), altering its internal state and generating a new output. In this study, we focus on monitoring the basic parameter  $V$ , which represents the membrane potential of a physical neuron.

**Table 2.** ACAN types based on spiking activity and corresponding parameter values.

Type	$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\lambda$	$\mu$	$\rho_1$	$\rho_2$
a	7	0.3	0.2	2.8	0.06	R	0.7	0.3	0
b	7	0.3	0.2	2.8	0.06	R	0.7	0.3	0
c	7	0.3	0.2	2.8	0.06	R	0.7	0.55	−0.2
d	7	0.3	0.2	2.8	0.06	R	0.7	0.55	−0.2
e	7	0.3	0.2	2.8	0.06	R	0.7	0.55	−0.2
f	7	0.3	0.2	1.1	0.03	R	0.01	0.2	0.15
g	7	0.3	0.2	−0.5	0.05	R	4	0.25	0.4
h	7	0.3	0.2	3	−0.09	R	0.5	0.3	0
i	7	0.3	0.2	−0.5	0.05	R	4	0.25	0.4
j	7	0.3	0.2	3	−0.09	R	0.5	0.3	0
k	7	0.3	0.2	3	−0.09	R	0.5	0.3	0
l	7	0.3	0.2	−0.5	0.05	R	4	0.25	0.4
m	7	0.3	0.2	3	−0.1	R	0.5	0.3	0
n	7	0.3	0.2	3	−0.1	R	0.5	0.48	−0.42
o	7	0.3	0.2	3	0.1	R	0.5	0.3	0
p	7	0.3	0.2	3	−0.11	R	0.5	0.3	0

Table 2. Cont.

Type	$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\lambda$	$\mu$	$\rho_1$	$\rho_2$
q	7	0.3	0.2	−0.5	0.15	R	0.5	0.2	−0.3
r	7	0.3	0.2	2.8	0.06	R	0.7	0.3	0
s	7	0.3	0.5	−5	0	R	−0.1	0.4	0.3
t	7	0.3	0.5	−5	0	R	−0.1	0.55	−0.1

### 3.2. Remote Supervised Method (ReSuMe)

The remote supervised method (*ReSuMe*) [32], which was used for SNN training, is described along with its adaptation for digital settings. The *ReSuMe* is a learning method that relies on the synaptic plasticity rule introduced by Hebb in his publication [33]. In order to successfully modify *ReSuMe* for a digital environment, we aimed to closely replicate the procedure in a specifically digital format. As a result, it was necessary to deviate from the continuous-time formulation described in [34] and modify it to suit digital implementation. We compute the difference between the values of  $S^d(t)$  and  $S^l(t)$  by using discrete time intervals. This involves transforming the output (post-synaptic) and desired spike trains,  $S^l(t)$  and  $S^d(t)$ , into their discrete time-series counterparts,  $S^l[n]$  and  $S^d[n]$ . At each time step  $n$ , there is either a spike (1) or no spike (0) in this format. The output at time  $n_i$  is determined based on the occurrence of spikes in  $S^l[n_i]$  and  $S^d[n_i]$ . If a spike occurs in  $S^l[n_i]$  but not in  $S^d[n_i]$ , the output is 1. If a spike occurs in  $S^d[n_i]$  but not in  $S^l[n_i]$ , the output is −1. If spikes occur in both  $S^l[n_i]$  and  $S^d[n_i]$ , the output is 0. This digital adaptation enables concurrent processing across the full frame, making it more suitable for our system’s real-time operation, where inputs are received as distinct binary spikes rather than spike timing.

In particular, *ReSuMe* modifies the synaptic weight ( $w$ ) between a pre-synaptic neuron ( $n^m$ ) and a post-synaptic neuron ( $n^l$ ) based on a target spike train, a pre-synaptic spike train, and a post-synaptic spike train, i.e.,  $S^d(t)$ ,  $S^{in}(t)$ , and  $S^l(t)$ , respectively, with the following rule:

$$\frac{d}{dt}w(t) = [S^d(t) - S^l(t)] \left[ a + \int_0^\infty W(s)S^{in}(t - s)ds \right] \tag{11}$$

The aforementioned equation computes the alteration in each synaptic weight ( $w$ ). The learning rate ( $lr$ ) controls the ultimate adjustment of weights. In this instance, the variable  $a$  denotes the magnitude of the non-correlation component’s influence on the total weight alteration. The following convolution represents the alterations of  $w$  through the Hebbian-like process. The learning window, denoted by the integral kernel  $W(s)$ , represents the convolution’s kernel. It is defined based on the time delay ( $s$ ) between the spiking events occurring among neurons. The form of the learning window ( $W(s)$ ) bears resemblance to the one described in spike-timing-dependent plasticity (STDP) [34]. In our implementation of the *ReSuMe* algorithm, we computed the exponential window,  $W$ , in advance in order to optimize the speed of our learning process, thus achieving faster processing times. The convolution in the equation above,  $LW = e^{-\Delta t/\tau}$ , was computed for each spike in the input spike trains ( $S^{in}(t)$ ). Here,  $\Delta t$  represents the time difference between a prior spike and the current time ( $t$ ) in the spike train.

The final form of the learning rule employed is the following one:

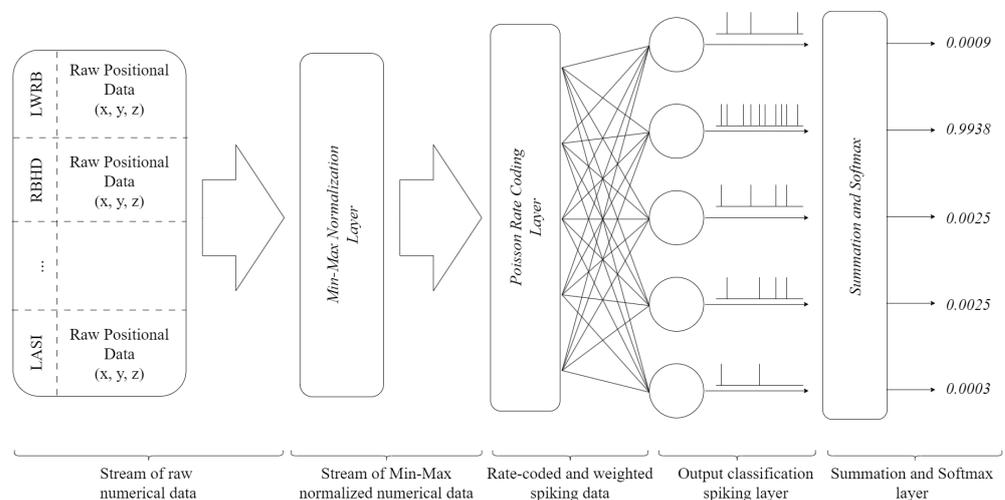
$$w' = w + lr(S^d - S^l) \cdot (\alpha + LW) \tag{12}$$

where  $w$  is the synaptic weight matrix of each connection,  $lr$  is the learning rate,  $S^d$  and  $S^l$  are the desired and the post-synaptic spike trains matrices for all available neurons,  $\alpha$  is the non-correlation amplitude term, and  $LW$  is the pre-calculated learning window. Equation (12) achieves parallelization by performing vector–matrix multiplication between the output matrix,  $(S^d - S^l)$ , and the pre-calculated learning window matrix,  $LW$ , both of which have

the same dimension representing time step  $S$ . This subtle and sophisticated technique enhances the effectiveness and adaptability of learning, enabling seamless integration of the *ReSuMe* method into digital neuromorphic systems.

### 3.3. Network Architecture

The architecture of the SNN, encompassing its layers and the procedure of converting data into spikes, is elucidated. The proposed SNN consists of three main layers, as illustrated in Figure 3: an input layer, where data are fed into the network; a rate-coding layer, where data are transformed into a format suitable for neurons (i.e., spikes); and a classification layer composed of ACANs with uniform registers of size  $R$ , which also functions as the output layer. This layer conducts a comparison between the real outputs and the predicted outputs of the neurons. During both training and testing, performance is evaluated by using a summation and softmax layer, which is implemented as described in [35,36]. The data are displayed as time intervals of positional  $x$ ,  $y$ , and  $z$  data for thirty-eight (38) distinct markers positioned on the individuals' bodies. This particular dataset is unsuitable for utilization with SNNs, hence necessitating its conversion into spike-based representations.



**Figure 3.** ACAN network architecture.

To accomplish this, we employ the inhomogeneous Poisson process for spike generation [37]. This procedure utilizes normalized data within the range of  $[0, 1]$  as time-dependent spike rates. Every feature within the frame is allocated a distinct random number from a uniform distribution, which is subsequently compared to that particular feature. A spike is recorded at the specific time if the drawn number does not exceed the characteristic. The spike trains are next fed into five (5) ACANs, which categorize them based on one of five movement classifications (Figure 1). After each training sample, the network's training accuracy is assessed by comparing its pre-training output to the ground truth. This is feasible because the network utilizes the complete output for a specific input sample during training, guaranteeing that the data are essentially unknown, particularly during the first epoch.

After the training process, a test phase is conducted when the network identifies the appropriate label by considering the neuron with the highest spiking rate, which is the most active one. In the last layer, the output of every neuron is added up over the whole sample time period. This is then processed through a softmax layer to find the output with the most spikes, which is then assumed to be the network's output. Subsequently, each instance is juxtaposed with the ground truth, and every accurate prediction increases a counter. Following each training period, the network's overall accuracy is computed in the following manner:

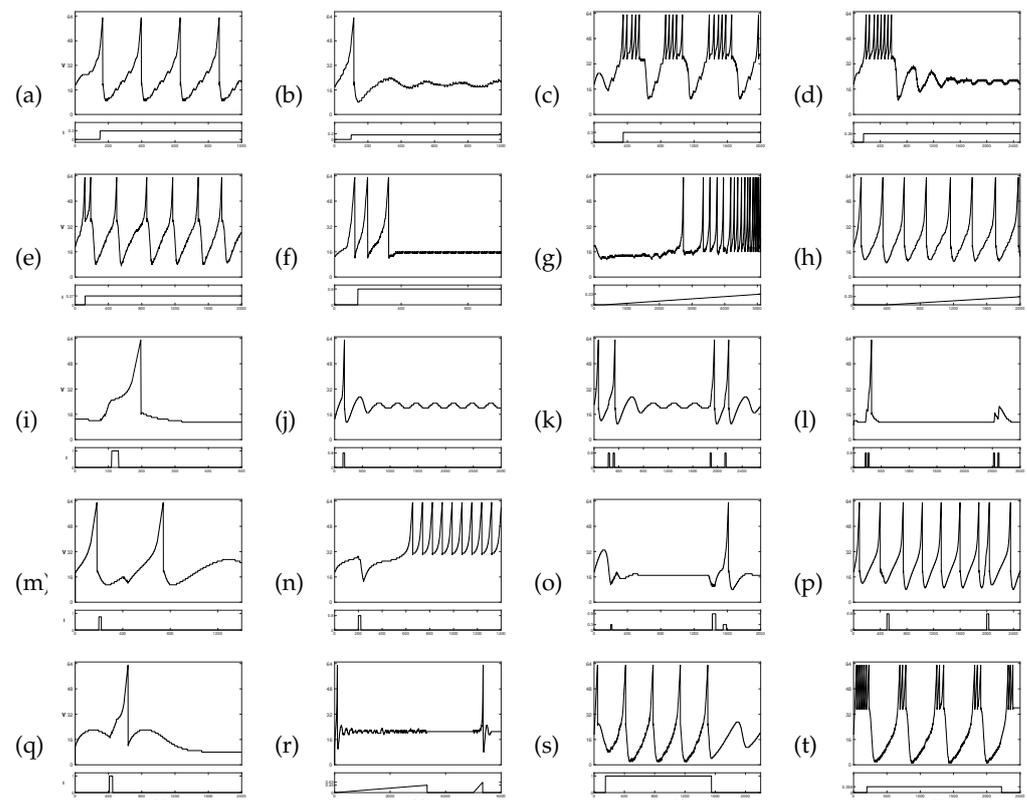
$$accuracy = correct / total \ samples$$

The suggested architecture offers a comprehensive framework for efficiently converting and processing data by using the proposed SNN, enabling precise movement classification in active and assisted living (AAL) applications, as explained in the following section.

## 4. Experiments and Results

### 4.1. ACAN Spiking Activity Reproduction on an FPGA

We assessed the feasibility of the ACAN model by implementing it and replicating its cortical spiking processes by using the ModelSim environment. The VHDL-coded ACAN model was developed to empirically showcase its capacity to accurately replicate a range of neuronal responses, as depicted in Figure 4. The model's adaptability was demonstrated by reproducing various patterns of spiking and bursting activities, which were based on the ACAN types listed in Table 2. These spiking and bursting activities are as follows: (a) tonic spiking, (b) phasic spiking, (c) tonic bursting, (d) phasic bursting, (e) mixed-mode spiking, (f) spike frequency adaptation, (g) class 1 excitation, (h) class 2 excitation, (i) spike latency, (j) sub-threshold oscillation, (k) resonator, (l) integrator, (m) rebound spike, (n) rebound burst, (o) threshold variability, (p) bistability, (q) depolarizing after-potential, (r) accommodation, (s) inhibition-induced spiking, and (t) inhibition-induced bursting.



**Figure 4.** The various spiking patterns reproduced by using our VHDL-implemented ACAN model in the ModelSim environment. The patterns are organized from left to right and from top to bottom: (a) tonic spiking, (b) phasic spiking, (c) tonic bursting, (d) phasic bursting, (e) mixed-mode spiking, (f) spike frequency adaptation, (g) class 1 excitation, (h) class 2 excitation, (i) spike latency, (j) sub-threshold oscillation, (k) resonator, (l) integrator, (m) rebound spike, (n) rebound burst, (o) threshold variability, (p) bistability, (q) depolarizing after-potential, (r) accommodation, (s) inhibition-induced spiking, and (t) inhibition-induced bursting.

#### 4.2. MNIST Hand-Written Digit Dataset

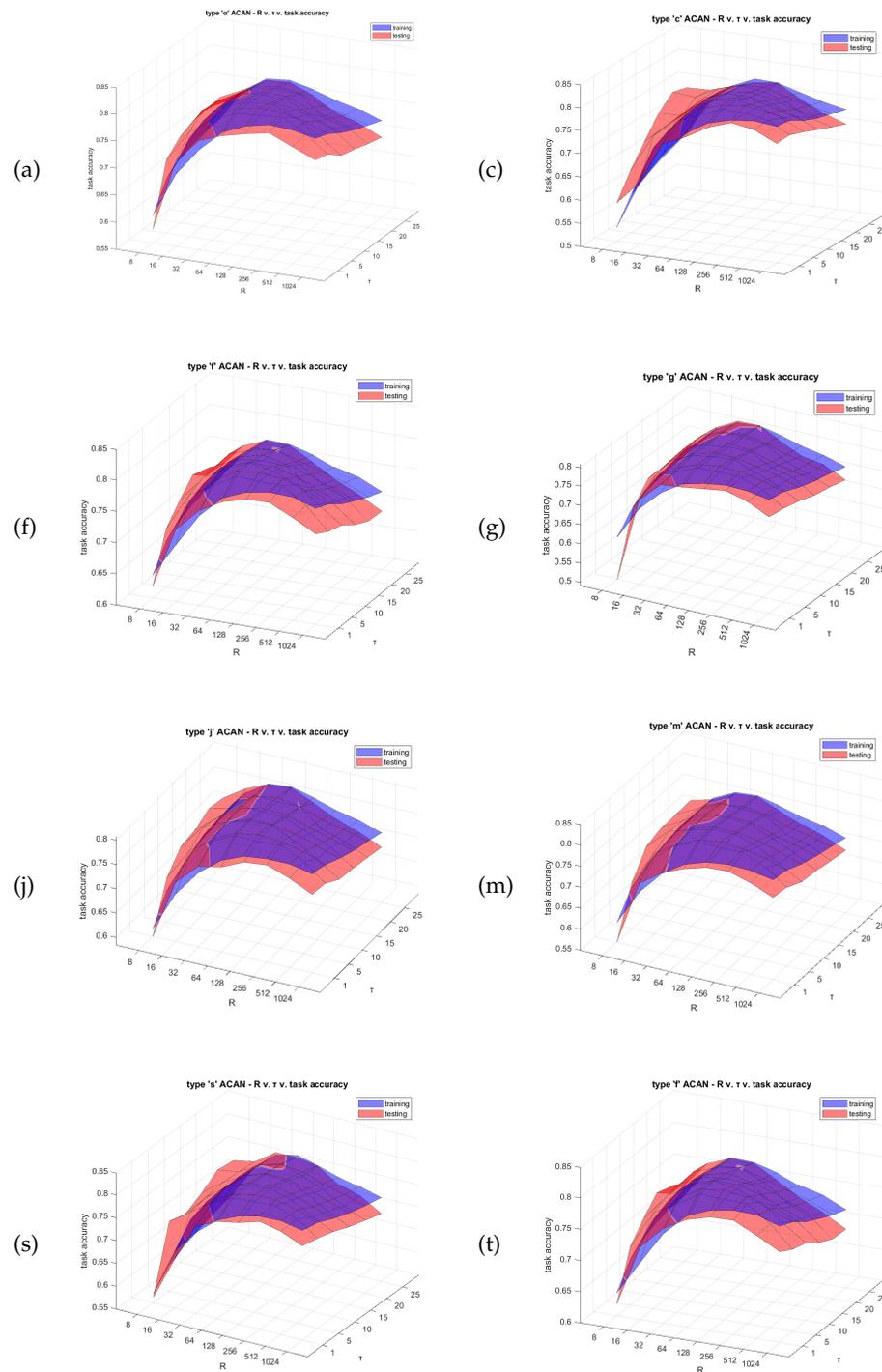
In order to evaluate the capabilities of the SNN, we performed an initial experiment by utilizing the MNIST hand-written digit dataset [38]. Our ACAN-based SNN architecture’s fundamental learning capabilities were initially validated by using the MNIST dataset as a proof of concept. Next, we concentrated on a new collection of movement data acquired by using the Vicon motion capture device. This dataset showcases the system’s capability to transition from recognizing patterns in images to analyzing intricate time-series data that represent human movements. We utilized MATLAB to train the network by using a dataset consisting of 60,000 training photos and 10,000 test images. The 28-pixel-by-28-pixel input images were transformed into spike trains of 30 time steps apiece, with the pixel values determining the spiking rates. This resulted in  $784 \times 30$  samples. The ACAN parameters that were adjusted for maximum accuracy are as follows: Vector  $\Gamma$  assumed the values  $(7, 0.3, 0.2, 2.8, 0.06, 1, 0.7, 0.3, 0)$ , and the register size was  $R = 64$  for all registers. The network was trained by using *ReSuMe*, with a learning rate of 0.0001 and  $\tau$  of 15. Following training, the network was tested by using new images and achieved an accuracy of 89%, which is comparable to prior research in the field of SNNs [26], demonstrating the network’s capacity to understand spatio-temporal correlations within the data.

#### 4.3. Network and Training Optimization

An in-depth parametric analysis was carried out to fine-tune the network’s hyperparameters, including the register size ( $R$ ) of the ACAN model neuron, the time constant of *ReSuMe*, and the overall type of the ACAN model (please refer to Table 2), and to evaluate the impact of spiking activity patterns on learning effectiveness. The analysis was performed in MATLAB, evaluating several ACAN models by systematically varying each parameter, such as register sizes and time constants, within a specified range. The type of ACAN includes the set of the following parameters:  $\{a, c, f, g, j, m, s, t\}$ . For the size of the register ( $R$ ), we tested the values  $\{8, 16, 32, 64, 128, 256, 512, 1024\}$ . Finally, for the time constant ( $\tau$ ), we tested the values  $\{1, 5, 10, 15, 20, 25\}$ . The data shown in Figure 5 and Table 3 indicate that the ACAN parameters play a crucial role in determining the system’s learning capability. Figure 5 illustrates the substantial impact of various ACAN parameters on the learning process of the system. This refers to how the neuron’s different spiking patterns affect the learning process. The sub-figures of Figure 5 are distinctively unique from each other, especially in terms of their visual characteristics. Table 3 displays the optimal performance achieved in both training and testing by the parametric analysis. The ideal setup consisted of the  $t$  type ACAN with a register size ( $R$ ) of 128 and a time constant ( $\tau$ ) of 10. This setup achieved an accuracy of 80.12% on the test set. The data shown in Table 3 help us choose the optimal settings for our system considering the limitations of the particular application we are dealing with.

**Table 3.** Accuracy results for various ACAN neuron types on the novel movement dataset. Each type has an accuracy (%) score, as well as values for the register size ( $R$ ) and time constant ( $\tau$ ).

ACAN Model Types	Training			Testing		
	R	$\tau$	Accuracy %	R	$\tau$	Accuracy %
a	128	5	80.75	128	10	79.7
c	256	5	80.87	128	10	79.62
f	128	5	80.77	64	10	79.53
g	128	5	80.87	128	10	79.53
j	128	5	80.87	128	10	79.86
m	128	5	80.81	256	5	79.53
s	256	5	80.79	256,128	5, 10	79.87
<b>t</b>	256	5	80.86	<b>128</b>	<b>10</b>	<b>80.12</b>

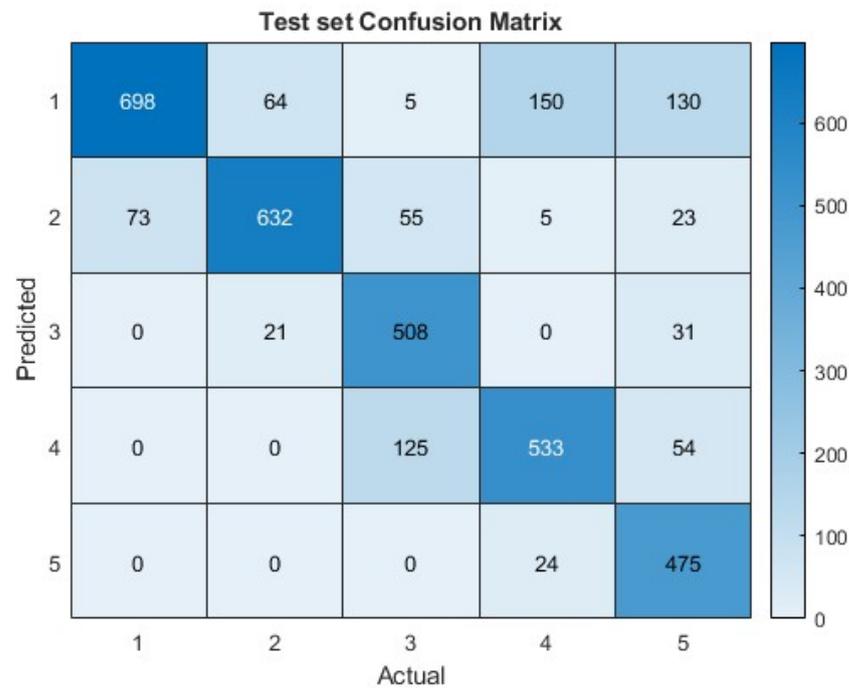


**Figure 5.** Parametric analysis of different ACAN neuron types for the MNIST hand-written digit dataset. The analysis was carried out by adjusting the register range ( $R$ ), time constant ( $\tau$ ), and ACAN neuron configuration parameters. Left to right, top to bottom: (a,c,f,g,j,m,s,t) ACAN model parameters were used as presented in Table 2. The figures illustrate the influence of varying the size of the ACAN’s register ( $R$ ) and the time constant ( $\tau$ ) of *ReSuMe* on the training accuracy (in blue) and the test accuracy (in red).

#### 4.4. Novel Movement Dataset

We applied the ACAN spiking network to the new movement dataset, following the preprocessing steps described in Section 2. The samples were set to be  $S = 100$  frames long, resulting in the creation of matrices sized  $114 \times 100$ . The normalized data were converted

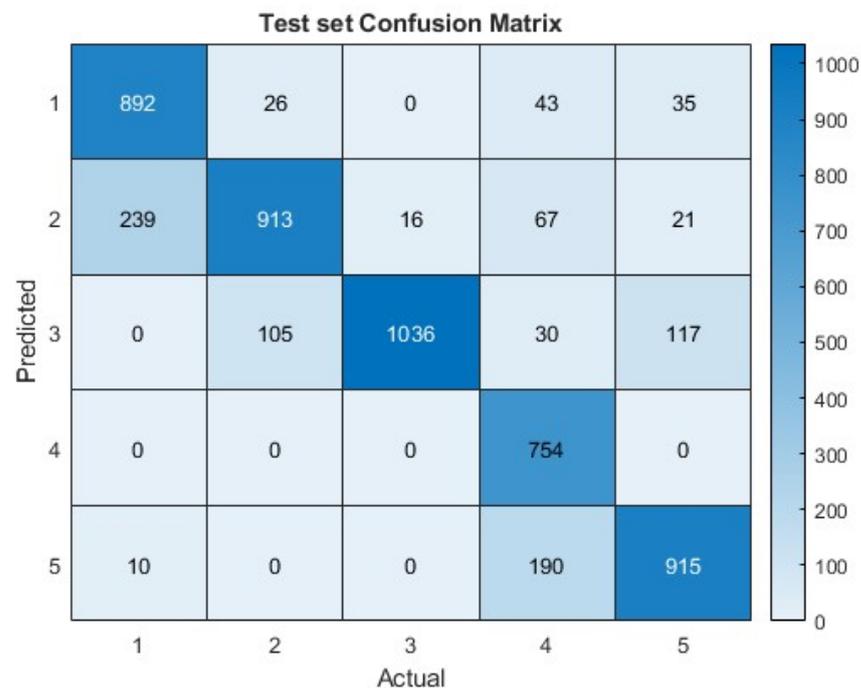
into spike trains and utilized to train the SNN by using a specified 90/10 training/test split. Incorporating velocity data enhanced performance and led to input matrices of dimensions  $228 \times 100$ . The MNIST experiment utilized certain settings for training by using *ReSuMe*:  $\tau = 15$ ,  $\alpha = 1$ , and  $lr = 0.0001$ , respectively. The training lasted for 75 epochs. The SNN accurately identified new data points with an accuracy of 78.9%, as shown in Figure 6.



**Figure 6.** Confusion matrix of SNN performance on the test set of the first experiment with a prediction accuracy of 78.9%.

The accuracy scores achieved were satisfactory, as demonstrated in comparable research cited in [39]; nonetheless, several factors might impact performance. The number of data is a crucial factor, since a larger dataset can lead to a more effective classifier. The network's single-layer architecture may restrict its performance; multi-layered spiking neural networks often achieve higher accuracy. The model's ability to learn may be restricted by the training method. Sequential data in a real-world scenario are likely to be highly correlated due to sharing most frames and belonging to the same sample class, perhaps resulting in more accurate results. The specific moment when a significant movement starts and the moments around it are not dependable indicators of the type of movement occurring might be crucial to enhancing classification accuracy.

Another approach to enhancing the network's performance was attempted by considering input features as distinct and autonomous, similar to how the image's pixels were handled in the MNIST hand-written digit classification experiment. With the identical conditions as the MNIST experiment, but with 10 time step units per input instead of 30, the input size increased to an  $11,400 \times 10$  matrix, which is a thousand times larger than that in the previous experiment. This has a substantial impact on execution times (about  $\sim 2800$  s each epoch). By using an 85/15 training/test split, the data setup achieved a test accuracy of 83.3%, indicating a 4.5% improvement, as shown in Figure 7.



**Figure 7.** Confusion matrix of SNN performance on the test set of the second experiment with a prediction accuracy of 83.3%.

## 5. Conclusions and Future Work

The study has successfully demonstrated the effectiveness of the ACAN model within the domain of SNNs for AAL applications, emphasizing its precision and versatility in elderly activity recognition. This research focuses on experimentally validating the ACAN model's capability to reproduce several types of known neural spiking activity, showcasing its promise for diverse neuromorphic applications. The study incorporates a thorough parametric analysis to pinpoint essential settings for enhancing learning speed and accuracy in various circumstances. The experiment we conducted on human movement categorization by using SNNs showcased the model's robust capabilities and offered valuable insights into sample processing and training techniques. The experiment obtained an accuracy of 83.3% in classifying five unique types of motions. This accuracy is regarded satisfactory considering the limitations of FPGA-based implementation and the novelty of the dataset. It is also in line with findings from comparable research as found in the literature [39]. Moreover, the results show the promise of neuromorphic computing in the field of AAL.

Our method shows great promise for real-time applications in fields that require low-energy, real-time processing, such as wearable technology, edge computing, and robotics. Although the reported accuracy limits are notable, they offer valuable guidance for enhancing future model versions. This study significantly contributes to the area of SNNs and human-based movement classification, establishing a foundation for future advancements in real-time, energy-efficient computing systems.

In the future, we aim to improve the ACAN model's performance and relevance. Future research will focus on integrating continuous learning algorithms to tackle the discrete aspects of the training process and enhance logical continuity between subsequent samples. In addition, attempts will be made to expand the model's relevance to a broader spectrum of motions and situations, encompassing complex, unstructured environments seen in actual AAL settings. Investigating the compatibility with various neuromorphic hardware and sensors, along with assessing the possibilities for scaling and personalization to meet particular user requirements, will be essential. Our goal is to close the gap between the present constraints and the extensive capabilities of SNNs in real-time, adaptive, and energy-efficient applications.

**Author Contributions:** Conceptualization, A.P., K.-A.T., I.K. and G.C.S.; Methodology, A.P., K.-A.T., I.K. and G.C.S.; Software, A.P.; Validation, A.P., K.-A.T. and I.K.; Formal analysis, A.P., K.-A.T., I.K., A.M.K., A.G. (Athanasios Gkrekidis) and D.M.; Investigation, A.P., K.-A.T., I.K., A.M.K., A.G. (Athanasios Gkrekidis) and D.M.; Resources, N.A., M.M., A.G. (Antonios Gasteratos) and G.C.S.; Data curation, A.P., K.-A.T., I.K., A.M.K., A.G. (Athanasios Gkrekidis) and D.M.; Writing—original draft preparation, A.P., K.-A.T. and I.K.; Writing—review and editing, N.A., M.M., A.G. (Antonios Gasteratos) and G.C.S.; Visualization, A.P. and K.-A.T.; Supervision, G.C.S.; Project administration, G.C.S.; Funding acquisition, A.P., K.-A.T., I.K., A.M.K., A.G. (Athanasios Gkrekidis), D.M., N.A., M.M., A.G. (Antonios Gasteratos) and G.C.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the project “Study, Design, Development and Implementation of a Holistic System for Upgrading the Quality of Life and Activity of the Elderly” (MIS 5047294), which is implemented under the Action “Support for Regional Excellence”, funded by the Operational Programme “Competitiveness, Entrepreneurship and Innovation” (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The datasets presented in this article are not readily available because the data are part of an ongoing study. Requests to access the datasets should be directed to G.C.S.

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

## Abbreviations

The following abbreviations are used in this manuscript:

AAL	ambient assisted living
SNN	spiking neural network
DNN	deep neural network
ACAN	asynchronous cellular automata-based neuron
ReSuMe	remote supervised learning
RC	Reservoir Computing
MCG	Magnetocardiogram
EEG	Electroencephalogram
FPGA	field-programmable gate array
STDP	spike-timing-dependent plasticity
VHDL	VHSIC Hardware Description Language
VHSIC	Very-High-Speed Integrated Circuit
MNIST	Modified National Institutes of Standards and Technology
NN	neural network

## References

1. Mois, G.; Beer, J.M. Chapter 3—Robotics to support aging in place. In *Living with Robots*; Pak, R., de Visser, E.J., Rovira, E., Eds.; Academic Press: Cambridge, MA, USA, 2020; pp. 49–74.
2. Keroglou, C.; Kansizoglou, I.; Michailidis, P.; Oikonomou, K.M.; Papapetros, I.T.; Dragkola, P.; Michailidis, I.T.; Gasteratos, A.; Kosmatopoulos, E.B.; Sirakoulis, G.C. A Survey on Technical Challenges of Assistive Robotics for Elder People in Domestic Environments: The ASPiDA Concept. *IEEE Trans. Med. Robot. Bionics* **2023**, *5*, 196–205. [[CrossRef](#)]
3. Sabater, A.; Santos, L.; Santos-Victor, J.; Bernardino, A.; Montesano, L.; Murillo, A.C. One-shot action recognition towards novel assistive therapies. *arXiv* **2021**, arXiv: 2102.08997.
4. Moschetti, A.; Fiorini, L.; Esposito, D.; Dario, P.; Cavallo, F. Toward an unsupervised approach for daily gesture recognition in assisted living applications. *IEEE Sens. J.* **2017**, *17*, 8395–8403. [[CrossRef](#)]
5. Kansizoglou, I.; Bampis, L.; Gasteratos, A. An active learning paradigm for online audio-visual emotion recognition. *IEEE Trans. Affect. Comput.* **2019**, *13*, 756–768. [[CrossRef](#)]
6. Chandra, I.; Sivakumar, N.; Gokulnath, C.B.; Parthasarathy, P. IoT based fall detection and ambient assisted system for the elderly. *Clust. Comput.* **2019**, *22*, 2517–2525. [[CrossRef](#)]
7. Bao, Y.; Sun, F.; Hua, X.; Wang, B.; Yin, J. Operation action recognition using wearable devices with inertial sensors. In Proceedings of the 2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Daegu, Republic of Korea, 16–18 November 2017; pp. 536–541.

8. Giannakopoulos, T.; Konstantopoulos, S. Daily Activity Recognition based on Meta-classification of Low-level Audio Events. In Proceedings of the ICT4AgeingWell, Porto, Portugal, 28–29 April 2017; pp. 220–227.
9. Laraba, S.; Brahimi, M.; Tilmanne, J.; Dutoit, T. 3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images. *Comput. Animat. Virtual Worlds* **2017**, *28*, e1782. [[CrossRef](#)]
10. Oikonomou, K.M.; Kansizoglou, I.; Manaveli, P.; Grekidis, A.; Menychtas, D.; Aggelousis, N.; Sirakoulis, G.C.; Gasteratos, A. Joint-Aware Action Recognition for Ambient Assisted Living. In Proceedings of the 2022 IEEE International Conference on Imaging Systems and Techniques (IST), Kaohsiung, Taiwan, 21–23 June 2022; pp. 1–6.
11. Kansizoglou, I.; Bampis, L.; Gasteratos, A. Do neural network weights account for classes centers? *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 8815–8824. [[CrossRef](#)] [[PubMed](#)]
12. Swanson, L.W. *Brain Architecture: Understanding the Basic Plan*; Oxford University Press: Oxford, UK, 2012.
13. Pfeiffer, M.; Pfeil, T. Deep learning with spiking neurons: Opportunities and challenges. *Front. Neurosci.* **2018**, *12*, 774. [[CrossRef](#)] [[PubMed](#)]
14. Yang, Y.S.; Kim, Y. Recent trend of neuromorphic computing hardware: Intel’s neuromorphic system perspective. In Proceedings of the 2020 International SoC Design Conference (ISOCC), Yeosu, Republic of Korea, 21–24 October 2020; pp. 218–219.
15. Tang, G.; Kumar, N.; Michmizos, K.P. Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October–24 January 2020; pp. 6090–6097.
16. Takeda, K.; Torikai, H. A novel hardware-efficient CPG model for a hexapod robot based on nonlinear dynamics of coupled asynchronous cellular automaton oscillators. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
17. Takeda, K.; Torikai, H. A novel hardware-oriented recurrent network of asynchronous CA neurons for a neural integrator. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 2972–2976. [[CrossRef](#)]
18. Horie, N.; Torikai, H. A novel hardware-efficient asynchronous cellular automaton model of tumor immunotherapy and its FPGA implementation. In Proceedings of the 2021 17th International Workshop on Cellular Nanoscale Networks and Their Applications (CNNA), Catania, Italy, 29 September–1 October 2021; pp. 1–4.
19. Suzuki, H.; Torikai, H. A Novel Hardware-Efficient Network of Ergodic Cellular Automaton Neuron Models and its On-FPGA Learning. In Proceedings of the 2022 IEEE International Symposium on Circuits and Systems (ISCAS), Austin, TX, USA, 27 May–1 June 2022; pp. 2266–2270.
20. Nakata, K.; Torikai, H. Analysis of time series classification of a multi-layer reservoir neural network based on asynchronous cellular automaton neurons with transmission delays. In Proceedings of the 2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA), Catania, Italy, 29–30 September 2021; pp. 1–4.
21. Matsubara, T.; Torikai, H. A novel reservoir network of asynchronous cellular automaton based neurons for MIMO neural system reproduction. In Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–7.
22. Tsakalos, K.A.; Dragkola, P.; Karamani, R.E.; Tsompanas, M.A.; Provata, A.; Dimitrakis, P.; Adamatzky, A.I.; Sirakoulis, G.C. Chimera states in neuro-inspired area-efficient asynchronous cellular automata networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2022**, *69*, 4128–4140. [[CrossRef](#)]
23. Chatzipaschalis, I.K.; Tsakalos, K.A.; Sirakoulis, G.C.; Rubio, A. Parkinson’s Treatment Emulation Using Asynchronous Cellular Neural Networks. In Proceedings of the 2023 IEEE 14th Latin America Symposium on Circuits and Systems (LASCAS), Quito, Ecuador, 28 February–3 March 2023; pp. 1–4.
24. Matsubara, T.; Torikai, H. Asynchronous cellular automaton-based neuron: Theoretical analysis and on-FPGA learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 736–748. [[CrossRef](#)] [[PubMed](#)]
25. Siddique, A.; Vai, M.I.; Pun, S.H. *A Low-Cost, High-Throughput Neuromorphic Computer for Online Snn Learning*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–18.
26. Valencia, D.; Alimohammad, A. A generalized hardware architecture for real-time spiking neural networks. *Neural Comput. Appl.* **2023**, *35*, 17821–17835. [[CrossRef](#)]
27. Barré, A.; Armand, A. Biomechanical ToolKit: Open-Source Framework to Visualize and Process Biomechanical Data. *Comput. Methods Programs Biomed.* **2014**, *114*, 80–87. Available online: <https://biomechanical-toolkit.github.io/mokka/> (accessed on 20 December 2023). [[CrossRef](#)] [[PubMed](#)]
28. Menychtas, D.; Petrou, N.; Kansizoglou, I.; Giannakou, E.; Grekidis, A.; Gasteratos, A.; Gourgoulis, V.; Douda, E.; Smilios, I.; Michalopoulou, M.; et al. Gait analysis comparison between manual marking, 2D pose estimation algorithms, and 3D marker-based system. *Front. Rehabil. Sci.* **2023**, *4*, 1238134. [[CrossRef](#)] [[PubMed](#)]
29. Izhikevich, E.M. Simple model of spiking neurons. *IEEE Trans. Neural Netw.* **2003**, *14*, 1569–1572. [[CrossRef](#)] [[PubMed](#)]
30. Izhikevich, E.M. *Dynamical Systems in Neuroscience*; MIT Press: Cambridge, MA, USA, 2007.
31. Matsubara, T.; Torikai, H. Bifurcation-based synthesis of asynchronous cellular automaton based neuron. *Nonlinear Theory Its Appl. IEICE* **2013**, *4*, 111–126. [[CrossRef](#)]
32. Ponulak, F.; Kasinski, A. ReSuMe learning method for Spiking Neural Networks dedicated to neuroprostheses control. In Proceedings of the EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices, Citeseer, Lausanne, Switzerland, 8–10 March 2006; pp. 119–120.

33. Hebb, D.O. *The Organization of Behavior: A Neuropsychological Theory*; Psychology Press: London, UK, 2005.
34. Gerstner, W.; Kistler, W.M. *Spiking Neuron Models: Single Neurons, Populations, Plasticity*; Cambridge University Press: Cambridge, UK, 2002.
35. Tsakalos, K.A.; Sirakoulis, G.C.; Adamatzky, A. Unsupervised Learning Approach Using Reinforcement Techniques on Bio-inspired Topologies. In *Handbook of Unconventional Computing*; WSPC Book Series in Unconventional Computing; World Scientific: Singapore, 2021; Chapter 17, pp. 507–533.
36. Tsakalos, K.A.; Sirakoulis, G.C.; Adamatzky, A.; Smith, J. Protein structured reservoir computing for spike-based pattern recognition. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 322–331. [[CrossRef](#)]
37. Heeger, D. Poisson model of spike generation. *Handout Univ. Stanford* **2000**, *5*, 76.
38. LeCun, Y.; Cortes, C. The MNIST Database of Handwritten Digits. 2005. Available online: <http://yann.lecun.com/exdb/mnist/> (accessed on 22 December 2023).
39. Khokhlova, M.; Migniot, C.; Morozov, A.; Sushkova, O.; Dipanda, A. Normal and pathological gait classification LSTM model. *Artif. Intell. Med.* **2019**, *94*, 54–66. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.