Application of Spiking Neural Networks for Action Recognition from Radar Data

Dighanchal Banerjee*, Smriti Rani†, Arun M. George‡, Arijit Chowdhury§,
Sounak Dey¶, Arijit Mukherjee‖, Tapas Chakravarty**, Arpan Pal††
TCS Research & Innovation, Kolkata, India
Email: *dighanchal.b@tcs.com, †smriti.rani@tcs.com, ‡arunm.george@tcs.com, §arijit.chowdhury2@tcs.com,
¶sounak.d@tcs.com, ‖mukherjee.arijit@tcs.com, **tapas.chakravarty@tcs.com, ††arpan.pal@tcs.com

***Abstract*—In the past two decades, radar-based human sensing has become a topic of intense research. Unlike vision-based techniques which require the use of camera, radars are unobtrusive and privacy preserving in nature. Further, radars are agnostic of the lighting conditions and can be used for through-the-wall imaging thereby making them hugely effective in many situations. Compact, affordable radars have been designed that can be easily integrated with remote monitoring systems. However, the classical machine learning techniques currently used for learning and inferring human actions from radar images are compute intensive, and require large volume of training data, making them unsuitable for deployment on the network edge. In this paper, we propose to use the concepts of neuromorphic computing and Spiking Neural Networks (SNN) to learn human actions from data captured by the radar. To the best our knowledge, this is the first attempt of using SNNs on micro-Doppler data from radars. Our SNN model is capable of learning spatial as well as temporal features from the data and our experiments have resulted in 85% accuracy which is comparable with the classical machine learning approaches that are typically used on similar data. Further, the use of neuromorphic and SNN concepts make our model deployable over evolving neuromorphic edge devices thereby making the entire approach more efficient in terms of data, computation and energy consumption.**

I. Introduction

Radar-based human sensing has become a topic of intense research in last two decades. Special radar designs are being investigated for unobtrusive detection of human physiology [1] as well as recognizing gestures/activities [2]. These radars are compact in size, affordable and can be easily integrated to remote monitoring systems. Using radar for human sensing has certain advantages over vision technologies in that this is privacy preserving, independent of lighting conditions, usually does not require background subtraction (static background is defined as 'Clutter' in radar terminology) and can be used for through-the-wall imaging. Amongst the radar technologies, 'Continuous Wave' (CW) or 'Frequency Modulated Continuous Wave' (FMCW) radars are preferred for short range (upto 15 metres) and indoor applications like elderly care [3]. For CW radar, one measures motion directly in terms of Doppler frequency while for FMCW or Pulse radars, Doppler frequency is derived through pre-processing. However the disadvantage of CW radar is that it can not measure the distance of the target from the radar. Therefore, FMCW radars are usually considered for such applications. It is to be noted that human movements constitute articulated motion vide linkages through flexible joints. When a rigid body moves in the field of view of the radar, the return signal displays doppler frequency by taking the difference between transmitted and received frequencies. But when a person performs some action, even while standing still, the radar return displays time varying doppler frequency patterns. This is known as micro-Doppler effect. Thus, the radar signal is usually processed by generating spectrogram or joint time-frequency plots. The classification and detection of different actions from such radar signals is complex and we observe a shift of research from radar system design to designing new signal processing algorithms. The signatures due to human motion displays complex pattern. While attempts have been made to create simulators through approximate modeling approach [4], machine learning techniques are usually applied for action detection. But the investigations are continuing and new insights are required for accurate and reliable detection of human gestures from radar returns.

Insofar, in the state of the art research works, classical machine learning techniques, including Artificial Neural Networks (ANN) and Deep Learning models have been used on data from vision sensors for identifying actions. However, apart from the privacy concern, the classical approaches suffer from another great disadvantage in that the methods are not tailored for end-to-end execution on *edge devices*. In various industrial domains, such as Internet of Things (IoT), robotics, healthcare, retail etc., an abundance of low powered devices exist at the edge of the network and there is a drive to utilise the available compute cycles on such devices. The advantage this approach has over the prevailing methodology is that the data need not be sent upstream to the computing infrastructure over the network, thereby reducing the latency and communication cost. However, the classical approaches mentioned above require a large volume of data for training and are highly compute/memory intensive making them too heavy-weight for edge devices. Pre-trained compressed models can however be deployed on constrained devices - but that does not avoid the cost incurred during training, the requirement of a large volume of training data, and being compressed, they often sacrifice accuracy.

At the same time, the concept of neuromorphic computing [5], [6] has evolved, that, unlike classical von Neumann architectures, mimicks mammalian sensing and data processing

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

mechanism. In this form of computing architecture, memory is collocated with processor and the data flow is in the form of inherently sparse spike trains thus reducing computation cost and time. The architecture typically runs networks designed using spiking neurons which are energy efficient due to their sparse and asynchronous communication in form of spikes [7]–[9]. Being energy efficient, the neuromorphic architecture is highly suitable for large-scale deployment as edge devices and hence, the neural network models designed to take advantage of the underlying neuromorphic hardware, have become prime candidates to run at the edge. During the last few years, the Spiking Neural Network (SNN) approach has increased in popularity with their computational capabilities being formally proven [10].

In this paper, we propose to leverage the paradigm of neuromorphic computing and apply it over radar data for learning and identifying human actions. The neuromorphic and SNN paradigm have been successfully used on image, speech and video data; but to the best of our knowledge, this is the first application of SNN on micro-Doppler data from radars, which in our opinion has a high degree of similarity with event-based data typically used within the SNN paradigm. We have implemented a novel convolution-based spiking neural network that is capable of learning both spatial and temporal features of the actions. We have created a dataset of 8 action classes performed by 5 persons in front of a radar and then have trained the SNN network with a training set from therein and tested the performance. We have observed 85% accuracy on test data which is comparable to the results obtained by applying classical techniques that are used in similar scenarios. The benefit of our approach lies in the use of the neuromorphic concept and SNN, making it deployable over a neuromorphic edge attached to the radar so that the data need not be sent over the network for inferencing, thereby decreasing the inference latency. Also, as mentioned above, the system is efficient in terms of computation and energy usage.

The paper is organised as follows: Section II provides details of existing works in the area of action recognition while Section III gives a brief background on the functionality of radars as well as SNN and the associated learning mechanism. Section IV explains components and functionalities of our proposed spiking network and Section V presents the implementation and discusses results. We conclude with our future plan of work in Section VI.

II. Related works

A. Traditional Action recognition techniques

The multi-disciplinary research of action recognition has been attempted by vision and pattern-recognition experts, using cameras and surveillance videos [11]. Wearables such as inertial sensors [12] are used to indirectly measure activities or gestures. The need for presence and action recognition in vision impaired, off-body detection scenarios such as defense, disaster and rescue operations has propelled interest in other sensors such as PIR sensors [13], piezo sensors [14], radars and many more.

B. Action recognition using micro-Doppler

Due to the non-intrusive nature of radar sensors, they are being explored for action recognition. Detailed analysis of the micro-Doppler phenomenon was done by Chen et al. in [15]. Following this, in [16], expanded understanding was provided, with applications ranging from rigid bodies like pendulum motion and rotating helicopter blades to non-rigid bodies like humans walking, bird flapping its wings, quadrupedal animal motion, etc. The work contains simulation study along with mathematical conceptualization. Consequently, researchers started working on utilizing this to detect human activity and falls. For instance, authors of [17] use a deep learning network on the data collected from two pulse-Doppler RCRs to detect falls in elderly. Unaided and aided activity recognition using radars and deep convolutional autoencoder was attempted by Mehmet et al. in [18]. Google went ahead to develop 'Project Soli', which identifies various finger based gestures [19]. Despite these developments, the research community is yet to agree on a benchmark technique to handle radar signals for action recognition.

C. Action recognition using DNN

In the last few decades, different types of deep learning techniques have been applied to learn the human activities. The ability to learn visual patterns directly from the pixels without any pre-processing step makes the Convolutional Neural Network (CNN) suitable for learning human actions. Ji et al. [20] proposed a 3D CNN architecture that applies multiple convolution at one pixel to identify different features at that position and using multiple channels to learn from different video frames. The final feature is generated as a combination of all those extracted features. Tran et al. [21] proposed a deep 3-D convolutional network (ConvNet) that tries to learn the spatio-temporal activity from a video. Simonyan et al. [22] improved the methodology by training a temporal ConvNet on optical flow instead of raw frames of a video. Another group of researchers used the recurrent neural network for classifying the action sequences. Baccouche et al. [23] exploited the capability long-short term memory (LSTM) [24] cells to learn dynamics of the spatial features extracted by a convolutional neural network (CNN). Shi et al. [25] used a 3D-ConvNet to capture 3D features and attached an LSTM network to capture the temporal pattern of those 3D filter features. Later works [26], [27] showed improvements by fusing different streams of features along with the above techniques. As learning methods and inference frameworks of the conventional deep networks need large amount of training data and are typically computation intensive, these models are not the most efficient solutions.

D. Action recognition using SNN

The task of recognizing human actions using SNNs has been explored by a group of researchers. Escobar et al. [28] proposed a bio-inspired feed-forward spiking network for action recognition using mean firing rate of every neuron and synchrony between neuronal firing. However, this model does not take into account the property of action-selective neurons, which is essential for the decoding the observed pattern. A variation of the feed-forward network is also showed in [29], [30] which is a recent work by the present authors. Yang et al. [31] used a two layer spiking neural network to learn human body movement using a gradient descent based learning mechanism by encoding the trajectories of the joints as spike trains. This inherently brings in the question of biological plausibility. Wang et al. [32] proposed a novel Temporal Spiking Recurrent Neural Network (TSRNN) to perform robust action recognition from a video. The SNN model provides reliable and sparse frames to the recurrent units using a temporal pooling mechanism. Also a continuous message passing from spiking signals to RNN helps the recurrent unit to retain its long term memory.

The other idea explored in the literature is to capture the temporal features of the input that are extracted by a recurrently connected network of spiking neurons, called the *"liquid"* or *"reservoir"*, the output of which is trained to produce certain desired activity based on some learning rule. Using this idea of reservoir computing Panda et al. [33] applied a *"Driven/Autonomous"* approach for reservoir creation that can learn video activity with limited examples. We observed that driven/autonomous models are good for temporal dependency modelling of a single-dimensional pre-known time series but it cannot learn spatio-temporal features together needed for action recognition. Soures et al. [34] proposed a deep architecture of a reservoir connected to an unsupervised Winner Take All (WTA) layer, that captures input in a higher dimensional space (by the reservoir) and encodes that to a low dimensional representation (by the WTA layer). All the information from the layers in the deep network are selectively processed by *"attention based neural mechanism"*. They have used ANN-based spatial feature extraction using ResNet but it is compute intensive.

The following section provides a background of related radar physics and SNN learning mechanisms which are important for this work.

III. Background knowledge: Radar and SNN

A. Radar Physics - Doppler and Micro-Doppler Effect

Human motion shows complex patterns. When a person walks, there are micro-motions like arm swing associated with the movement of the body. When electromagnetic wave is scattered from human bodies (under motion), the resultant signal displays both Doppler effect as well as modulation of Doppler frequency. While Doppler frequency arises due to the target, i.e human body moving towards (or away) from the radar, micro-Doppler signatures are seen due to the micro-motions. Doppler frequency is visible in the frequency domain of a signal. Distinct micro-Doppler effect for different kinds of movements is examined in the joint time and Doppler frequency domain of a signal. Spectrogram plots, which are intensity plots of STFT (Short Time Fourier Transform), are used to visualise spectro-temporal plots from radar returns. These plots help in analysing the change of frequency with time and thus characterize the unique micro-Doppler signatures of different activities performed by a human.

The STFT of a time domain signal x(t) is given by equation 1.

$$X(t,\omega)=\int_{-\infty}^{\infty} x(t+\tau)w(\tau)exp(-j\omega\tau)d\tau dx \qquad (1)$$

where $w(\tau)$ is the selected time window. Magnitude squared of the STFT gives the spectrogram (SP), shown by equation 2.

$$X_{SP}(t,\omega)=\mid X(t,\omega)\mid^{2} \qquad (2)$$

A narrow time window results in better resolution in time axis, a poor one in frequency domain and vice versa. Thus, a unique trade-off point has to be achieved between time-frequency resolution as both these information are important for the analysis of time-frequency plots. Figure 1 shows a spectrogram for *bow* action. Zero Doppler frequency is observed when the person is still.

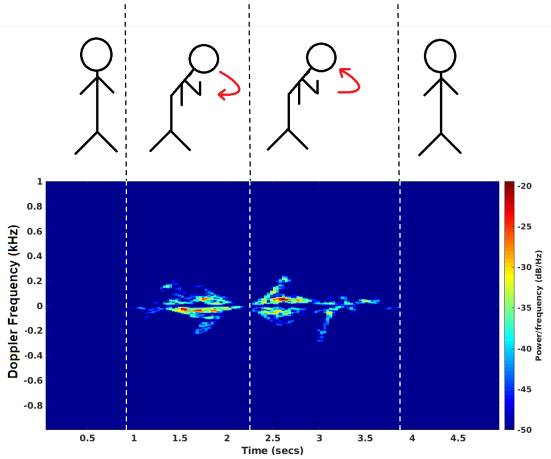


Fig. 1: Spectrogram of bow action

Positive and negative Doppler frequencies are observed when different body parts move towards or away from the radar. Together, all these frequencies constitute the micro-Doppler signatures for a particular action. Since different body parts move at different frequencies for multiple actions, their micro-Doppler signatures are different in time frequency domain.

B. SNN and its Learning Mechanism

Mammalian brains are composed of hugely connected neurons and synapses which maintain a stability via mutual excitation and inhibition unless external stimuli affect the status-quo. When a neuron receives a stimulus, the membrane potential rises due to intra-cellular activity, and if a threshold is breached, the neuron generates a *spike* which is carried forward to the subsequent neuron via the synapse. Spikes can assume the form bursts (repeated spikes within a short duration) or a single time event depending on the stimuli and the nature of the receiving neuron. Further, the biological features like composition of the synapse, the conductance of the cell body, and related chemical reactions play important roles in generation and processing of spikes. For the computation aspects and its adaptation in neural networks, the two most important factors are the rate at which spikes occur and the temporal relations of spike response between the pre- and post-synaptic neurons, i.e. whether the post-synaptic neuron fired after the pre-synaptic neuron, or after it, the latter affecting the synaptic bond between the neurons by making it stronger or weaker. In the language of Neuroscience, especially Hebbian learning, this is called *”fire together, wire together.”*

Unlike classical ANNs, the SNNs use biologically plausible neuron models and are thus closer to mammalian brains. Spikes offer inherent sparsity and massively parallel asynchronous communication [7]–[9], and resulting in spiking neuron mmodels being energy efficient. However, ANNs operate on continuous valued input, whereas SNNs require the input data to be encoded in spike format for subsequent processing. SNNs are considered as the third generation of neural networks with formally proven computational capabilities comparable to that of regular ANNs [10].

1) Spiking Neuron Model: There are various mathematically modelled spiking neurons with different levels of complexity and granularity with the Hodgkin-Huxley model [35] being the most detailed one. However, for our purposes, we use the simplest and most popular Leaky Integrate and Fire (LIF) model [36]. An LIF, with a membrane potential V at any point in time, can be described by the differential equation 3.

$$\tau \frac{dV}{dt} = (V_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inh} - V) \quad (3)$$

To achieve stability, the membrane potential always tend to evolve towards the resting potential, V_{rest}. Hence, in the absence of any stimulus from pre-synaptic neurons, the membrane potential of a particular neuron remains at V_{rest}. Similarly, the equilibrium potentials of the excitatory and inhibitory synapses are represented by E_{exc} and E_{inh}. Synapses are modelled as conductance values, namely, g_e, the excitatory conductance, and g_i, the inhibitory conductance. Excitatory pre-synaptic neurons increase the membrane potential, whereas, inhibitory pre-synaptic neurons tend to decrease it. As mentioned before, a spike is generated when the membrane potential breaches a threshold (V_{thresh}). A spike in the pre-synaptic neuron increases the conductance of the synapse in magnitude. The dynamics of excitatory and inhibitory conductance are modelled as per equations 4 and 5 respectively.

$$\tau_e \frac{dg_e}{dt} = -g_e \quad (4)$$

$$\tau_i \frac{dg_i}{dt} = -g_i \quad (5)$$

2) Learning Rule: The mathematical function used to model a spike is the well known *Dirac Delta function*[1]. As this model is non-differentiable (which is logical for a spike which occurs at a time instance only), the gradient based learning algorithms popular in ANNs, cannot be applied in case of SNN. Learning and memory in SNNs are thus modelled using Spike Time Dependent Plasticity (STDP) [**?**] which takes into account the strengthening of synaptic bonds due to positive temporal correlation between pre- and post-synaptic spiking neurons. The STDP protocol modifies classical Hebbian learning rule [37] by improving it with temporal asymmetry. It has been proven that a spiking neuron with STDP can learn a linear dynamical system with minimum least square error [38]. A pre-synaptic trace, x_{pre}, for each synapse keeps track of the activity of the pre-synaptic neuron, and likewise a post-synaptic trace x_{post}, keeps track of the activity of the post-synaptic neuron. Each trace decays exponentially with time as shown in the equations 6 and 7 with synaptic trace decay constants τ_{pre} and τ_{post}.

$$\tau_{pre} \frac{dx_{pre}}{dt} = -x_{pre} \quad (6)$$

$$\tau_{post} \frac{dx_{post}}{dt} = -x_{post} \quad (7)$$

At the occurence of a spike at a pre- or post-synaptic neuron, the trace is incremented by a constant value a. For each pre-synaptic firing, the synaptic weight is reduced with a value proportional to the post-synaptic trace (the phenomenon is called *depression*) and for each post-synaptic firing, it is increased with a value proportional to the pre-synaptic trace (the phenomenon is called *potentiation*. The learning process of an arbitrary synapse is shown in Figure 2.

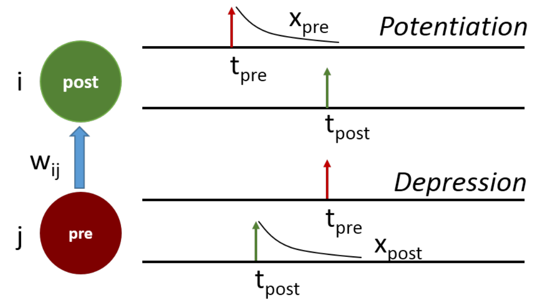


Fig. 2: Synaptic depression and potentiation in STDP

The complete learning rule can be described by equations 8 and 9.

$$\Delta w_{dep} = \eta_{dep}(x_{post} \times s_{pre}) \quad (8)$$

$$\Delta w_{pot} = \eta_{pot}(x_{pre} \times s_{post}) \quad (9)$$

s_{pre} and s_{post} represent spike of the pre- and post-synaptic neurons. In practice, equations 8 and 9 are vector equations where s_{pre} and s_{post} denote the spike vectors of a population of neurons and × is an outer product operation.

[1] https://en.wikipedia.org/wiki/Dirac_delta_function

3) Lateral Inhibition and Homeostasis: A popular biologically plausible approach adopted in neural networks in order to enhance competition between neurons is called Lateral Inhibition or Winner-Take-All [39], [40]. The first excited neuron to produce a spike attempts to stimulate other neurons or directly inhibits one or more of them. In a learning scenario, a pattern to be learnt excites one or more neurons, which in turn try to deactivate other neurons with the help of lateral inhibition, preventing them from learning the same pattern. In SNN world, this mechanism helps multiple neurons to compete and learn different patterns. In our network, we use a softer form of Lateral Inhibition like that of k-WTA, which is proven to be computationally less power intensive than a hard Lateral Inhibition [41] and leads to better shared feature selectivity in cortical pyramidal cells [42].

The process of maintaining a stable internal state, prevalent in many biological systems (e.g. maintaining body temperature, pressure, blood sugar etc.) is known as homeostatis. In the context of SNNs, homeostasis of neuronal firing rate is meant to prevent the dominating effect of any particular neuron. We employ a rate homeostasis similar to that used in Diehl et al. [43], where threshold of neuronal firing is adapted so that continuous firing by the same neuron can be discouraged. Our membrane threshold, V_{thresh} is a combination of a static threshold value, $V_{thresh-static}$ and a dynamic memory based component, θ which increases with each firing by a constant value and decays exponentially with time. The complete spiking mechanism is described by equations

$$S(t) = \begin{cases} 1, & V(t) > V_{thresh} \\ 0, & V(t) \geq V_{thresh} \end{cases} \tag{10}$$

$$V_{thresh} = V_{thresh-static} + \theta(t) \tag{11}$$

$$\theta(t+1) = \begin{cases} \theta(t) + C, & S(t) = 1 \\ \theta(t), & S(t) = 0 \end{cases} \tag{12}$$

$$\tau_\theta \frac{d\theta}{dt} = -\theta \tag{13}$$

In the following section, we describe the network that we have designed and implemented for action recognition.

IV. Network Architecture

The proposed spiking network architecture for detecting human actions from radar data (refer Figure 4) consists of mainly three main components: *(i)* Data pre-processing layer *(ii)* Convolutional Spiking layer (CSNN) *(iii)* Classifier layer

The first component performs compression and encoding on radar data in order to make the computation faster, while the second component, whose design and action is inspired from CNN, contains multiple spiking layers and they extract the spatial features from the input spiking data. A special technique as detailed below is used to capture the temporal signature of action while the data is being processed in this layer. The spatial feature extraction is hierarchical in nature, with first layers capturing low level features like edges with complexity keep on increasing till the last layer. The convolutional features of a layer along with it's temporal spiking signature become an enriched feature set and is then passed to a classifier for finally recognising the actions. Each of the components are described below in details.

A. Data pre-processing layer

A pre-processing layer as shown in Figure 3 processes the radar data to allow its use with SNN.

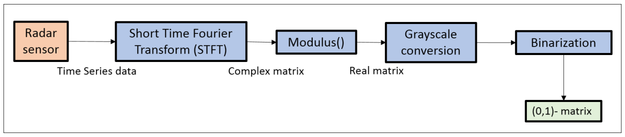


Fig. 3: Block diagram of data pre-processing module

For the current work, a 24 GHz Continuous Wave (CW) radar is used with I(in-phase) and Q(quadrature i.e. shifted by 90 degrees) channels at a sampling frequency of 2 KHz. Data is collected for 5 seconds for all actions. Thus, for each activity we have quadrature time domain data of length 10000. From the dataset, spectrogram for each action is computed, which is a time-frequency domain representation of the time series data obtained from radar. We have used 1024 point Fast Fourier Transform (FFT) with a 256 length Kaiser window with 75% overlap to compute the spectrogram [4]. The number of time bins for a spectrogram is calculated by the formula in equation14

$$T(No.\ of\ time\ bins) = \frac{N - W_{overlap}}{W_{length} - W_{overlap}} \tag{14}$$

where N is the total sample data(5×2000), W_{length} is the window length used for STFT computation(256) and $W_{overlap}$ is the overlapping no of data points(75 % of 256 = 192). Thus, we obtain 153 time bins (T = 153). (±1 KHz) 2000 Hz data is represented by 1024 data points(owing to 1024 point FFT). Hence

$$Frequency\ Resolution = 2000/1024\ Hz = 1.953Hz$$

A 5 second data have 153 samples leading to time resolution,

$$(5/153)\ s = 32.68\ ms$$

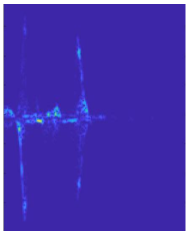
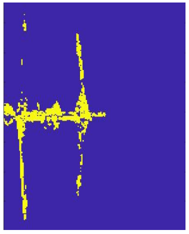
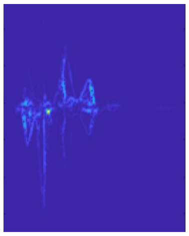
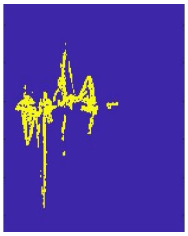
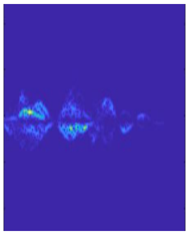
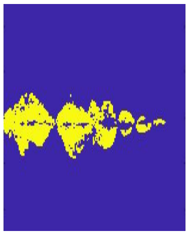
.

Thus, from 2-D radar data (I and Q channel), we obtained a time frequency data in -1 KHz to 1 KHz range for 5 seconds and this led to time-frequency dataset of 1024×153 size matrix (representing spectrogram).

As shown in Figure 3, Time domain data from the two channels is converted into Time-Frequency domain using STFT(eq 1). STFT provides a complex matrix. Modulus operation is performed on this STFT matrix to obtain a real valued matrix. A binary matrix is obtained by consecutively converting this real matrix into a grayscale image and later into a binary matrix using appropriate threshold. This binary matrix is the input to the spiking neural network where each column becomes an input at i-th instance of time, $i = 1...T$. Thus input to the network is a 1-D binary image (aka vector).

Spectrogram and the corresponding binarized images for three different activities - *Throw, Kick* and *Bow* respectively are depicted in three rows of Table I.

TABLE I: Visualisation of data at pre-processing stage (a) Throw action (b) Kick action (c) Bow action

Raw Input	Spectrogram of radar data	Binarised spectrogram
		
		
		

B. Convolutional Spiking Layer (CSNN)

The overall architecture of the network is shown in Figure 4. A set of class-wise filter blocks with lateral inhibition with competition mechanism make up the the Convolutional Spiking layer which takes in pre-processed data as input. The network may consist of multiple such CSNN layers following a CNN-like connectivity between consecutive layers. Every spike-frame of an action sequence is connected to the convolutional layers via a sliding window of an initial dimension of $w \times h$ pixels (w=1 in this case), with each pixel being connected to a single neuron of the filter-block of the first convolutional layer. The window is slided vertically by a *stride (s)* to connect each pixel of the new window to the second neuron. The process is repeated till every pixel within the spike-frame is connected to the neurons in the filter. Similar connections are made for further input spike-frames to the neurons within the same filter. Once the input frames are connected to the first CSNN layer, consecutive layers can be connected in a similar fashion. The number of layers depends on the complexity of the spatial features of the dataset and hence remains a design choice.

In order to enable the CSNN layers to capture spatially collocated patterns within the same spike frame of a single action class, multiple filters are created within each filter block which are connected via a *switcher* node, which in fact is a special LIF neuron. This lets us avoid learning 3D spatio-temporal filters from consecutive spike frames by activating only one filter at a given time. The switcher applies inhibition to force all but one filter in the block to inactive state, the duration of which depends on the strength of inhibition, which is a configurable parameter. After the period of inactivity, all filters start competing again and the one which causes the maximum spike is considered as the *winner* - which is an effective way of utilising the *winner takes all* concept explained in Section III-B3. The process repeats depending on the *decay time constant* during the training time of the convolutional filters. That all filters get a chance during the training phase is ensured by the switching mechanism, and this also ensures that spatially collocated but temporally separable features appear on different filters.

To guarantee activation of only one filter block at a given point of time for a given action frame sequence, we apply another level of long-term inhibition which additionally ensures that multiple filter blocks are not trying to learn the same redundant pattern. Instead, the lateral inhibition among filter blocks allows them to compete for classes. We initialise the weights in the filter blocks randomly and one block wins for the first time for a particular action class. This win ensures that the filter block will provide the maximum spike only for that particular class during further training. Once a filter block wins due to maximum initial spiking, an inhibition signal of higher strength is sent to other filter blocks preventing them form being activated.

This filter-block-wise inhibition mechanism provides two distinct advantages:

(i) Since all the filter blocks are not active at a given time, the number of active convolutional neurons of a CSNN layer during training time for each action is reduced.

(ii) It allows us to set different periodicity (i.e. different decay time constant) for switcher nodes of different filter blocks according to its associated action. Switching periodicity is dependent on the total duration of the action and different spatial patterns present therein. If multiple repetitive patterns occur within a short duration, switching periodicity for that particular filter block can be set to a small value.

During testing time, both long term inhibition between filter block as well as switching of filters within a block are removed as they are useful during training only.

Temporal features are extremely important for action recognition as these enable the system to capture the sequence of events by which a particular action is performed. This is especially useful for the cases where actions are spatially overlapping (for e.g. doing sit-up and jumping, right hand clockwise rotation and anticlockwise rotation etc.) but temporal sequence of events within the actions are different. The radar signature of whole action will look very similar for those spatially overlapping actions and spatial features as extracted by above CSNN layer would not be sufficient to accurately classify them.

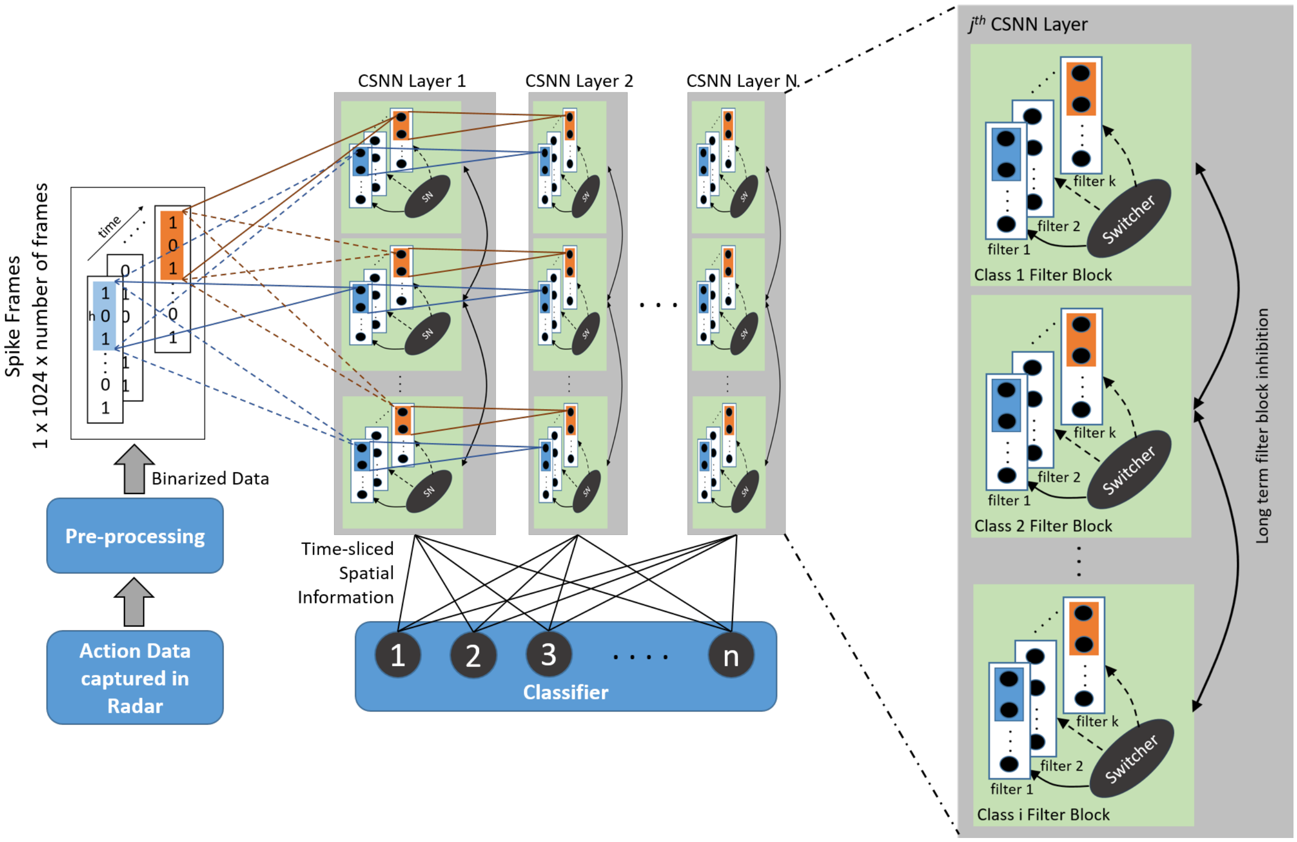


Fig. 4: Network architecture

In spiking domain, events are usually characterised by the timing of spike and by the number of spikes caused by an event. For spatially overlapping actions, the total number of spikes for two such actions will be nearly same and hence cannot be used for identifying the classes distinctly. Instead, if the entire action sequence is divided into multiple equal sized time-windows and if one logs the individual spike-count during each such time-window then chances are more that the count will be different for same time-window of two spatially overlapping actions. For example, *Sit up* and *Jump* are spatially overlapping actions and when their binarized spectrograms are fed into the network, they create almost same number spikes (6253 & 6479). As shown in Figure 5), if binarized spectrograms of *Sit up* (top) and *Jump* (bottom) are divided into 4 equal time windows, then spike count of each time window for those two classes are found to be different. In our case, all the action classes are of same time duration (i.e. 5 seconds) and we sliced the duration of each action into equal sized time-windows. These time-window-wise spike counts which capture the temporal signature of the action were used (along with spatial features) for classification in next layer.

C. Classifier layer

The spatial features and temporal features (in the form of time-window-wise spike counts) from CSNN layer corresponding to respective actions are input to the classifier layer. A simple logistic regression based classifier is used here.

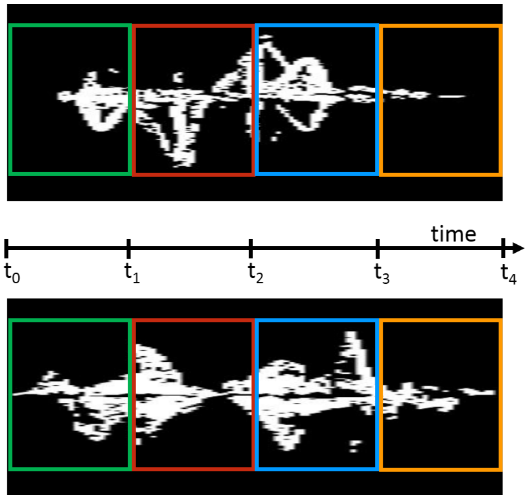


Fig. 5: Time window based temporal feature extraction.

V. Data Collection, Implementation and Results

A. Datasets

5 people were asked to perform 8 actions, each 10 times. Subjects stood at 2 meter away from the radar sensor. The experimentation was done on 3 males and 2 females. 24 GHz Quadrature channel CW radar [44], along with NI DAQ USB 6001 and LabView 2016 were used to collect data.

Actions performed in front of radar are - *1) Bow with upper torso, 2) Slide right hand horizontally, 3) Kick with right leg,* *4) Front foot batting with a cricket bat (right handed), 5) Ball throwing, 6) Wearing & removing glasses, 7) Sit up, and 8) Jump .*

Data processing: The data captured by the system was cleaned using the pre-processing module algorithm in Matlab, as discussed in Section IV-A and fed into SNN.

B. Implementation

The network described in Section-IV is implemented using *BindsNet 0.2.4* [45], a GPU based open source SNN simulator in Python. It had some bugs (as it is still in development stage) but those have been handled. BindsNet is used because it supports parallel computing unlike other available simulators like Brian, Nest etc.

Table II summarizes the parameters used for implementation of the neuron model and learning mechanism. Many of these parameters are consistent with the values of their biological counterparts. Also, the learning rate parameters η_{dep} & η_{pot} (as in Eqns. 8 & 9) are set to those values so that CSNN layer can learn features best. Value of E_{exc} & E_{inh} (as in Eqn. 3) are kept same as V_{rest}. Also value of τ_e and τ_i (as in Eqns. 4 & 5) have been kept same as that of τ.

TABLE II: Parameters for neuron model & learning mechanism

Parameter	Value	Parameter	Value
$V_{thresh-static}$	-55.0 mV	V_{rest}	-65.0 mV
τ	100 ms	τ_θ	10^7 ms
τ_{pre}	3 ms	τ_{post}	5 ms
η_{dep}	10^{-3}	η_{pot}	5×10^{-3}

C. Results & Discussion

Entire pipeline is used to train and classify on binarized radar spectrograms of aforesaid dataset. The dataset is split into a ratio of 4:1 for training and testing. As mentioned in Section IV-A, all the 1-D binary images (aka vectors), also referred as spike-frames, corresponding to an action are fed into the CSNN layer sequentially. By varying the stride length of the sliding window (of size w=1, h=5) on the input spike-frame, three different experiments E1, E2, and E3 were conducted with stride length(s) being 3, 5 and 7 respectively. Width of sliding window is taken as 1 so that we do not loose time resolution. These experiments were performed in order to find the highest classification accuracy that can be achieved by processing the least amount of data. Lesser amount of data will excite lesser number of spiking neurons and consequently lesser computation effort will be required thus reducing power consumption. Detailed results of the experiments are provided in Table III. It can be observed that, E1 and E2 have the same classification accuracy (85%) while it decreases to 81.25% for E3. For the stride length 3, the system processes almost 66% more data per spike-frame compared to the case with stride length 5, however the accuracy remains same. When stride length is 7, 27% less data is processed but accuracy drops. Thus, it can be concluded that, for binarized spectrograms, one cannot afford to loose further information than is already lost during the pre-processing stage. For action recognition, Precision and Recall values for each action are important to look at. As can be seen from Table III, action specific recall values & precision shows slight variation for different stride lengths - highest values obtained for both being 1, lowest being 0.6 (precision) for *Sit Up* and 0.64 (Recall) for *Bow*.

As we cannot afford loosing data in frequency domain (Y-axis of spectrogram), it is to be investigated whether we can afford to do same in time domain (X-axis of spectrogram). Based on experiment E2 mentioned above, we investigated further on the effect of downsampling of data in time domain on accuracy. Downsampling of data essentially means reducing simulation time (aka training time) for the spiking network thus reducing time & data to learn. As shown in Table V, downsampling of data results in quick loss of accuracy. Hence, we proceeded with experimental set up of E2 without any downsampling for arriving at the final results.

The final results are presented in the form of a confusion matrix (refer Table IV). While the action classes *Bow, Bat & Jump* are correctly classified with precision 1, highest recall values have been obtained for action classes *Bat & Kick*. Average precision obtained for all classes in 0.85 with a standard deviation of 0.15 while those values for recall is 0.86 and 0.1 respectively. It is to be noted that instances of *Hand slide & Throw* have been misclassified between themselves: 2 instances of *Throw* were classified as *Hand slide* and 1 instance of *Hand slide* as *Throw*. This can be explained as a result of overlapping in these two actions with respect to radar owing to the fact that CW radar recognises object speeds towards and away from it, making their spatial signature partially similar. Worst result is obtained for *Sit up* with precision and recall values being 0.6 and 0.75 respectively. 2 instances of *Sit up*s have been classified as *Bow* and another 2 as *Jump*. These misclassifications are the results of very similar upper and lower torso movements in all the three actions.

Overall, we can conclude that using spiking neural network (with experimental set up of E2), we can distinctly classify 8 human action actions performed in front of radar with acceptable accuracy of 85%. In a comparative analysis, the accuracy obtained on radar spectrogram data using computationally intensive deep learning techniques like auto-encoder [46], [47] is $\sim 90\%$. If logistic regression [48] technique is tried on same binarized image data, an accuracy of 81.25% is achieved but computation cost and training time of logistic regression is higher compared to SNN. Thus SNN appears to be a more suitable candidate for learning and classifying radar data and can exploit the evolving neuromorphic edge devices.

VI. Conclusion

In this paper we have discussed a novel way of using spiking neural network to classify human actions as captured by using a single CW radar. The network has been tested on a varied action dataset and is found to be capable of classifying the actions with acceptable high accuracy. In a continuation of this work, we plan to replace the logistic regression classifier in the

TABLE III: Effect of stride length (s) on Precision & Recall

Experiment	Metrices	Bow	Hand slide	Kick	Bat	Throw	Wear glasses	Sit up	Jump	Accuracy
E1 (h=5, s=3)	Precision	0.9	0.8	0.9	1.0	0.7	1.0	0.6	0.9	85%
	Recall	0.82	0.73	1.0	1.0	0.78	0.91	0.75	0.82	
E2 (h=5, s=5)	Precision	1	0.8	0.8	1.0	0.7	0.9	0.6	1.0	85%
	Recall	0.83	0.73	1.0	1.0	0.88	0.82	0.75	0.83	
E3 (h=5, s=7)	Precision	0.9	0.7	0.9	1.0	0.7	0.9	0.6	0.8	81.25%
	Recall	0.64	0.78	1.0	1.0	0.78	0.82	0.75	0.80	

TABLE IV: Test results: Confusion matrix

	Action	Predicted								Precision	Recall
	Class	Bow	Hand slide	Kick	Bat	Throw	Wear glasses	Sit Up	Jump		
Actual	Bow	**10**	0	0	0	0	0	0	0	1.0	0.83
	Hand slide	0	**8**	0	0	1	1	0	0	0.8	0.73
	Kick	0	0	**8**	0	0	1	1	0	0.8	1.0
	Bat	0	0	0	**10**	0	0	0	0	1.0	1.0
	Throw	0	2	0	0	**7**	0	1	0	0.7	0.88
	Wear glasses	0	1	0	0	0	**9**	0	0	0.9	0.82
	Sit Up	2	0	0	0	0	0	**6**	2	0.6	0.75
	Jump	0	0	0	0	0	0	0	**10**	1.0	0.83

TABLE V: Effect of down-sampling the radar data on classification

Down sampling Factor	**Accuracy**
1/2	71.25%
5/8	77.5%
3/4	75
7/8	77.5
1	85%

last layer of the architecture with a SNN based classifier so that the whole architecture becomes compatible with neuromorphic paradigm. We also intend to use a heterogeneous multi radar setup to collect data and train with SNN.

References

[1] A. Mishra, W. McDonnell, J. Wang, D. Rodriguez, and C. Li, "Intermodulation-based nonlinear smart health sensing of human vital signs and location," *IEEE Access*, vol. 7, pp. 158 284–158 295, 2019.

[2] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 6414–6417.

[3] M. Amin, *Radar for indoor monitoring: Detection, classification, and assessment*. CRC Press, 2017.

[4] A. Gigie, S. Rani, A. Chowdhury, T. Chakravarty, and A. Pal, "An agile approach for human gesture detection using synthetic radar data," in *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, 2019, pp. 558–564.

[5] S. Furber, "Large-scale neuromorphic computing systems," *Journal of neural engineering*, vol. 13, no. 5, p. 051001, 2016.

[6] M. A. Zidan, J. P. Strachan, and W. D. Lu, "The future of electronics based on memristive systems," *Nature Electronics*, vol. 1, no. 1, p. 22, 2018.

[7] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 699–716, 2014.

[8] J. Park, S. Ha, T. Yu, E. Neftci, and G. Cauwenberghs, "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," in *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, 2014, pp. 675–678.

[9] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[10] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[11] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[12] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[13] P. Wojtczuk, A. Armitage, T. D. Binnie, and T. Chamberlain, "Pir sensor array for hand motion recognition," in *Proc. 2nd Int. Conf. on Sensor Device Technologies and Applications*, 2011, pp. 99–102.

[14] E. Jeong, J. Lee, and D. Kim, "Finger-gesture recognition glove using velostat (iccas 2011)," in *2011 11th International Conference on Control, Automation and Systems*. IEEE, 2011, pp. 206–210.

[15] V. C. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-doppler effect in radar: phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and electronic systems*, vol. 42, no. 1, pp. 2–21, 2006.

[16] V. C. Chen, *The micro-Doppler effect in radar*. Artech House, 2019.

[17] L. Liu, M. Popescu, M. Skubic, M. Rantz, T. Yardibi, and P. Cuddihy, "Automatic fall detection based on doppler radar motion signature," in *2011 5th International Conference on*

Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops. IEEE, 2011, pp. 222–225.

[18] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.

[19] "Project Soli," https://atap.google.com/soli/, accessed: 2020-01-29.

[20] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[22] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[23] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International workshop on human behavior understanding*. Springer, 2011, pp. 29–39.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 802–810.

[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1933–1941.

[27] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[28] M.-J. Escobar, G. S. Masson, T. Vieville, and P. Kornprobst, "Action recognition using a bio-inspired feedforward spiking network," *International journal of computer vision*, vol. 82, no. 3, p. 284, 2009.

[29] S. Dey, A. Mukherjee, G. Bzard, and D. McLelland, "Demo: Human gesture recognition using spiking input on akida neuromorphic platform," *Neural Information Processing Systems (NeurIPS)*, 2019.

[30] "Human gesture recognition using spiking input on akida neuromorphic platform," https://ir.brainchipinc.com/press-releases/detail/90/brainchip-and-tata-consultancy-services-tcs-jointly, BrainChip Inc., and Tata Consultancy Services, 2019.

[31] J. Yang, Q. Wu, M. Huang, and T. Luo, "Real time human motion recognition via spiking neural network," in *IOP Conference Series: Materials Science and Engineering*, vol. 366, no. 1. IOP Publishing, 2018, p. 012042.

[32] W. Wang, S. Hao, Y. Wei, S. Xiao, J. Feng, and N. Sebe, "Temporal spiking recurrent neural network for action recognition," *IEEE Access*, vol. 7, pp. 117 165–117 175, 2019.

[33] P. Panda and N. Srinivasa, "Learning to recognize actions from limited training examples using a recurrent spiking neural model," *Frontiers in neuroscience*, vol. 12, p. 126, 2018.

[34] N. Soures and D. Kudithipudi, "Deep liquid state machines with neural plasticity for video activity recognition," *Frontiers in neuroscience*, vol. 13, p. 686, 2019.

[35] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[36] L. Lapicque, "Recherches quantitatives sur lexcitation electrique des nerfs traite comme une polarization," *J. Physiol. Pathol. Gen.*, vol. 9, pp. 620–635, 1907.

[37] H. D.0, *The Organization of Behavior*. Wiley and Sons, 1949.

[38] R. E. Suri, "A computational framework for cortical learning," *Biological Cybernetics*, vol. 90, no. 9, pp. 400–409, 2004.

[39] S. Grossberg, "Contour enhancement, short term memory, and constancies in reverberating neural networks," *Studies in Applied Mathematics*, vol. 52, no. 3, pp. 213–257, 1973.

[40] M. Oster, R. Douglas, and S.-C. Liu, "Computation with spikes in a winner-take-all network," *Neural Computation*, vol. 21, no. 9, pp. 2437–2465, 2009.

[41] W. Maass, "On the computational power of winner-take-all," *Neural Computation*, vol. 12, no. 11, pp. 2519–2535, 2000.

[42] Z. Jonke, R. Legenstein, S. Habenschuss, and W. Maass, "Feedback inhibition shapes emergent computational properties of cortical microcircuit motifs," *Journal of Neuroscience*, vol. 37, no. 35, pp. 8511–8523, 2017.

[43] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.

[44] "Infineon sense2go website," https://www.infineon.com/cms/media/PMM_3dmodels/sense2gol.html, accessed: 2020-01-30.

[45] H. Hazan, D. J. Saunders, H. Khan, D. T. Sanghavi, H. T. Siegelmann, and R. Kozma, "Bindsnet: A machine learning-oriented spiking neural networks library in python," *Frontiers in neuroinformatics*, vol. 12, p. 89, 2018.

[46] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[48] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.