RadarSNN: A Resource Efficient Gesture Sensing System Based on mm-Wave Radar

Muhammad Arsalan®, Graduate Student Member, IEEE, Avik Santra®, Senior Member, IEEE, and Vadim Issakov®, Senior Member, IEEE

Abstract—Radar offers a promising modality for enabling gesture recognition, which is a simple and intuitive alternative to click and touch-based human–computer interface. In this article, we propose a spiking neural network (SNN)-based hand gesture recognition with frequency-modulated continuous-wave 60-GHz radar. As preprocessing, the 2-D fast Fourier transform (FFT) is performed across fast time and slow time to generate a video of range–Doppler maps, which are then processed to generate range spectrograms, Doppler spectrograms, and angle spectrograms. The spike trains are fed into an SNN to classify the gesture that has been performed. We demonstrate that with few neurons, SNNs can achieve recognition accuracies close to 99.50% comparable to their deep learning counterparts for eight dynamic gestures. Moreover, the proposed model size is 75 kB, which is substantially smaller compared to the state-of-the-art models making it memory efficient. We also demonstrate using tSNE plots that SNNs can operate with lower embedding dimensions, implying that we can realize SNN with a small compute and memory footprint.

Index Terms—Gesture sensing, human–computer interface, Nengo, NengoDL, neuromorphic hardware, spiking neural networks (SNNs).

I. INTRODUCTION

GESTURE sensing offers an intuitive and easy interface to machines [2] compared to click and touch. Gesture sensing enables several applications in TVs, homes, automotive, and virtual reality. This field of research is primarily dominated by camera-based systems that leverage the high data content and computer vision techniques [3]–[6]. However, since camera-based systems work on images and videos, they cause privacy concerns. Furthermore, they require proper illumination conditions, which limits the use of these kinds of systems. To avoid these limitations, nonvision-based techniques have been introduced. Nonvision-based systems commonly employ sensors attached to the human hands in the form of gloves or bands, and the identification of the gestures is done by analyzing the signals acquired with these sensors [7]–[11]. Although these systems do not suffer from the limitations of camera-based systems due to their wearable nature, they provide a cumbersome user experience.

With recent advancements in sensor technology, contactless nonvision-based systems are getting popular. In this regard, radar-based solutions offer the advantage of being insensitive to illumination conditions, invariant to occlusion of hand, and using a simpler signal processing pipeline [12]–[14]. Radars have garnered a lot of attention as a sensor of choice due to their privacy-preserving features, ability to work within the enclosure, and their sensitivity to fine-grained gestures. There are two key aspects to radar-based gesture system solution: one being efficient miniature hardware that is capable of generating high-fidelity target data [15]–[22] and the other being the algorithm pipeline, propelled by deep learning, that parses target data to extract meaningful information of the user’s intent [23]–[32].

Recent approaches of radar-based gesture sensing aim toward resource-efficient solutions for the Internet-of-Things (IoT) devices where energy consumption is vital [33]–[35]. RadarNet [36] demonstrates recognition of hand swipe gestures using radar signals with minimal computational resources for mobile devices. Similarly, TinyRadarNN [37] utilizes the range–frequency Doppler features using a 2-D convolutional neural network (CNN) combined with a temporal convolutional neural network (TCN) for low-power gesture sensing. In [38], instead of only feeding the range–Doppler, to improve the robustness, a CNN connected recurrent neural networks (RNNs) is fed with a comprehensive hand profile, including range, Doppler, azimuth, and elevation. In [39], the hand gesture recognition with 802.11ad 60 GHz [millimeter-wave (mm-Wave)] technology in smartphones has been proposed. A tiny CNN for an embedded solution has been developed by minimizing the power consumption and computation cost. The radar-based gesture sensing is not limited only to hand swipes, and its applications have been shown in air-writing where characters or words drawn in the air in front of radar are detected and classified [40]–[42].

Although the conventional deep neural network (deepNet) approaches show quite promising results in terms of detection and recognition, energy efficiency is still an issue. The majority of the energy is consumed by the
Deviating from the traditional deep learning-based solutions, we have taken a different approach for radar-based gesture recognition systems by using spiking neural networks (SNNs). In SNNs, the information is transmitted by spike timing, including latencies and spike rates. Furthermore, the transmission is sparse and information is transmitted only when the membrane potential reaches a specific threshold. Similarly, the accumulation only occurs when information in terms of spike arrives at the node. The sparse nature (1-bit activity) reduces the volume of the data communicated between nodes. Moreover, the multipliers in MAC arrays are replaced with adders making the SNNs energy-efficient. Although the SNNs are energy-efficient [44]–[47], training the SNNs is challenging due to the nondifferentiable transfer function preventing backpropagation. Therefore, a suitable learning mechanism is required for training the SNNs. Among the existing spiking neuron models, the most notable spiking neuron model is the leaky integrate-and-fire (LIF) model. We have used the LIF model that is a modification to the integrate-and-fire (IF) model, including a leaky term [48]. The leaky term introduces neuron potential decay over time, which makes it more biologically plausible. Furthermore, the LIF is quite simple and easy to implement requiring less computation (floating-point operations) bearing the neurocomputational properties and thus making it a suitable choice for building SNN models.

This article is the extension of our IMS2021 work [1]. Compared to the conference paper, in this article, we have added the following contributions.

1) We have added more complex gestures for a comprehensive evaluation of the proposed solution.
2) A detailed assessment of additional spiking neuron models and their behavior is conducted on the augmented dataset.
3) The preprocessing pipeline is improved to generate better representable features for the predictive network to learn.
4) A modified and enhanced SNN architecture to detect and identify the eight gestures with high accuracy is proposed.
5) Compared to range–Doppler images (RDIs), to improve the robustness of the system comprehensive hand profiles, namely, range–time, velocity–time, and angle–time images are used as features.
6) In addition, the simulator is described in more detail and how the neuron models can be simulated to improve the reproducibility of the results.

The rest of this article is organized as follows. Section II presents the radar hardware used in the gesture solution along with the preprocessing pipeline. Section III presents the SNN. In Section IV, we introduce the proposed SNN architecture and the learning hyperparameters used. Finally, we present the results and discussions in Section V and conclude in Section VI.

### II. System

#### A. Hardware

We have used the frequency-modulated continuous-wave (FMCW) radar chipset BGT60TR13C [49] by Infineon Technologies, as shown in Fig. 1(a). The simplified block diagram of the chipset is shown in Fig. 1(b). The chipset embodies a transmitter, receiver, mixer, and analog-to-digital converter (ADC) with an external phase-locked loop that controls the linear frequency sweep. The loop is controlled by the frequency divider output pin with an 80-MHz reference oscillator. The reference clock running at 80 MHz is used to clock the finite-state machine (FSM) [50]. To generate the linear frequency sweeps from 57 to 63 GHz, the voltage-controlled oscillator (VCO) is enabled by varying the tune voltage $V_{\text{tuned}}$ from 1 to 4.5 V. The chipset comprises one transmit (TX) path antenna and three receive (RX) paths and we have used all the channels in our experiment. The chip comes with serial peripheral interface (SPI) and queued serial peripheral interface (QSPI) for memory readout with a maximum data transfer goes up to 200 Mb/s ($4 \times 50$ Mb/s).

For streaming out of data, as soon as the threshold set by the host is achieved in the memory, the FSM issues an interrupt (IRQ) flag. The mixer propels an active high-pass filter that is followed by a VGA, AAF, and an ADC driver. The multichannel ADC (MADC) is based on a 4-Mb/s 12-b successive approximation register (SAR) topology. SRAM of 196-kbit capacity is utilized to store the raw data and a sensor ADC (SADC) is used to provide temperature and transmit power readout [50]. The transmitted signal by TX and the received signals by RX are multiplied in the time domain and passed on for further processing. The chipset can transmit the signal up to 6-GHz bandwidth providing the resolution $\Delta r$ and a Doppler resolution $\Delta v$ of

$$\Delta r = \frac{c}{2B} = 2.5 \text{ cm}$$

$$\Delta v = \frac{1}{2f_c N_c T_c} = \frac{1}{122 \text{ cm/s}}$$

where $c$ is the speed of light, $B$ is the bandwidth, $f_c$ is the center frequency set to 60 GHz, $N_c$ is the number of chirps set to 64, and the chirp duration $T_c$ is set to 32 $\mu$s.

#### B. System Parameters

Fig. 2 shows our experimental setup where a hand gesture is made in front of the radar. The chipset is configured with the system parameters and derived parameters shown in Table I.

The radar is connected to a PC via a USB interface. ADC data from the radar are recorded and stored in the PC for further processing, as shown in Fig. 2. The distance between the radar and the hand is around 0.2–1 m. The hand detection is done automatically. As soon as the hand enters the field of view of the radar, it is detected and the system marks it as a starting point for the gesture and the frame count starts. The data are recorded for 32 consecutive frames for each gesture. The number of frames is chosen as 32 because the
slowest gesture took 32 frames. Each gesture is recorded and labeled with a large interclass variance. The gestures with a shorter number of frames (less than 32) are appended with zeros. Moreover, to improve the latency of the gesture sensing solution, apart from the start of gesture, also, the end of gesture could also be detected (based on the same proposed MSE approach). Once the end of gesture is detected, i.e., at 12 frames, the spectrograms can be appended with zeros before feeding into the SNN for classification. This would improve the latency of the real-time solution; however, the detection and classification performance would be the same.

### C. Signal Processing Pipeline

1) **Preprocessing:** The raw ADC data across a chirp (fast time) are collected, stacked, and arranged in rows along the frame (slow time). The ADC matrix is preprocessed as range fast Fourier transform (FFT) with coherent integration to increase the signal-to-noise ratio (SNR), followed by a moving target indicator to remove static targets in the field, followed by target detection and selection of the target range bin.

   a) **Coherent pulse integration:** To improve the strength of the signal for target detection, we applied pulse integration to combine fast time FFT data for each chirp in a frame to improve the SNR. The coherent pulse integration combines the phase and magnitude of the range FFT data coherently over all chirps of a frame. Coherent integration at frame $k$ is based on the mean $R_{ci}(k)$ of the range FFT data $\{R_i(k)\}_{i=1}^{NC}$ over $NC$ chirps

$$R_{ci}(k) = \frac{1}{N_C} \sum_{i=1}^{N_C} R_i(k)$$

where $N_C = 32$ in our experiment. The frame time was chosen carefully to ensure that there was no range walk for the top-bottom gesture under the worst case, i.e., fastest performed top-down gesture by any individual.

   b) **Moving target indication filtering:** In FMCW radar, the TX-to-RX leakage limits the usability of the first few FFT range bins. Furthermore, the reflection from the hand can be subdued by the reflection from the stationary objects present around. We applied the moving target indication (MTI) to suppress the reflection from these stationary objects and leakage. At each frame, a running average is applied to the coherently integrated range FFT spectrum $R_{ci(k)}$ at the $k$th frame, which is mathematically given as

$$R_{ci}(k) = R_{ci}(k) - S(k - 1)$$

$$S(k) = \alpha \cdot R_{ci}(k) + (1 - \alpha) \cdot S(k - 1)$$

where $\alpha$ denotes the forget factor set to 0.01. This filtered range FFT spectrum is fed to the subsequent target detection block.

It is to be noted that once the start of the gesture is detected, the spectrograms can be constructed without the MTI operation. However, the MTI operation being a high-pass filter helps in improving the SNR by deemphasizing/removing any component due to static hand reflection between subsequent frames in the spectrogram features, which are inconsequential for the gesture classification application while emphasizing the change in hand or movement, which are crucial for gesture classification.

In addition, since the radar setup focuses on top-to-bottom illumination, e.g., smartphone lying on the table, it avoids the vital or human body reflections being manifested in radar spectrograms. However, other configurations such as radar being mounted close to the web camera of a laptop lid would have reflections from the human body and also the vital signals. In such a configuration, the signal processing pipeline needs to be adapted to rely on the first detected target on the range bin, thus constructing the range, Doppler, and angle spectrograms based on the first detected target (possibly using a bandpass filter around the detected target).

2) **Target detection:** The target detection and selection on the filtered range FFT spectrum is performed through simple thresholding. The threshold is obtained by the mean value of the range FFT spectrum with some scaling factor.
The threshold $\Gamma_k$ at frame $k$ is given by

$$
\Gamma_k = \beta \sum_{n=1}^{N_r} R_{ci}^n(k)
$$

where $\beta$ is the scaling set to 3 in our case, $n$ is the index along the range bins, and $N_r$ is the total number of range bins. $\beta = 3$ was chosen empirically for tradeoff between false positive and probability of detection.

2) Range–Doppler Image: The frequency shift in the reflected signal due to the target range and velocity is mapped simultaneously to RDI. The energy of the reflected signal from the target is encoded as a blob pixel intensity in the RDI. A single range–Doppler per frame is obtained by first applying the first-order FFT to raw data along the fast time direction to bring the data in the range domain. Afterward, 1-D FFT is applied to data in the range–Doppler domain. This process is performed for multiple frames, and then, they are stacked together to form a 3-D cube called an RDI. The whole process is summarized in Fig. 3.

3) Angle-of-Arrival Estimation: The angle of arrival is one of the key parameters used for gesture recognition. We have used the minimum variance distortion-less response (MVDR) or Capon beamformer [51] for estimating the angle or direction of arrival (DOA). In the proposed system, we have used a Capon beamformer for azimuth angle estimation represented in form of a range-angle image (RAI) per frame.

4) Data Generation for the Model: In our case, instead of using the 3-D cube that would be computationally costly for the model, we projected the RDI in a 2-D image by extracting the row corresponding to the maximum pixel intensity from each frame and then stacked them together to form a 2-D image (range–time image), as shown in Fig. 3. Analogously, the same procedure is performed for generating velocity–time and angle–time images. Thus, each gesture is represented by three images, i.e., range–time, velocity-time, and angle–time images.

III. SPIKING NEURAL NETWORKS

Although inspired by the biological nervous systems, artificial neural networks (ANNs) still are unable to capture the complex neurocomputational properties of the biological neurons. To fill in this gap, the neuromorphic community has introduced the third generation of ANNs known as SNNs. SNNs, in contrast to the ANNs, capture more closely the functionality of the nervous system by considering not only the spatial but also temporal aspects of the input data for the construction of the computational model. The sparse and asynchronous communication in SNNs allows data processing in a massively parallel fashion [52]. Moreover, the low power consumption, fast inference, and event-driven information processing of SNNs make them a suitable candidate for efficient implementation of deepNets/machine learning tasks where energy efficiency is a prime requirement. The following describes the experiment and methods applied in this work for the analysis of SNNs.

A. Neural Engineering Object (Nengo) Simulator

Neural engineering object (Nengo) is a neural network tool for simulating large-scale neural systems with applications in cognitive science, psychology, artificial intelligence, and neuroscience [53]. Nengo offers a deep learning simulator NengoDL [54] that allows easy integration of TensorFlow library providing access to rich features such as convolution connections. Nengo uses a neural engineering framework (NEF) to design spiking neuron models for applications in machine learning and deep learning, for example, image classification [55], inductive reasoning [56], action selection [56], speech production [57], motor control [58], and planning with problem-solving [59].
1) NengoDL: We have used NengoDL [54] for our spiking simulations. NengoDL uses NEF for building neuron models for building biologically plausible neural networks. NEF provides the principles of representation, transformation, and dynamics to construct a neural model.

2) Neuron Models: The SNN neuron models are described on different abstraction levels starting from the most realistic and complex model Hodgkin–Huxley (HH) model, to the LIF model that is the simplest and most computationally efficient. Among them, the LIF model is equitably popular because of its simplest implementation, making it a computationally efficient model bearing neurocomputational properties.

a) LIF: The LIF introduces a leaky term to the IF model that causes neuron potential decay over time making it more biologically plausible. Mathematically, the LIF model is given as [48]

\[ C_m \frac{db}{dt} = g_L(V(t) - E_L) + I(t) \]  (7)

where \( C_m \) stands for the membrane capacitance, \( g_L \) denotes the conductance of the leakage channels, \( E_L \) denotes the equilibrium potential of the leakage channels, \( V \) represents the voltage, \( t \) denotes the time, and \( I \) denotes the total amount of the input current. The LIF generates synchronous spikes to asynchronous inputs in certain firing patterns imitating the neurons in the human brain.

b) Spiking version of ANN neurons: Besides the biologically plausible LIF neuron, we have also tested other rate-based neurons such as LIFRate, rectified linear, sigmoid, and tanh by converting them to spiking versions of themselves. NengoDL offers three different ways or wrappers for this conversion. These wrappers [54] accept some functions and provide an instantaneous firing rate. The firing rate is the number of spikes generated by a neuron per time step.

1) Regular Spiking: This wrapper turns a rate neuron into spiking one with regular interspike intervals. It takes the instantaneous firing rate and integrates it multiplied by a time step.

2) Poisson Spiking: Given an instantaneous rate, this wrapper draws a sample from a Poisson distribution with \( \lambda \) as the instantaneous firing rate.

3) Stochastic Spiking: This wrapper is a kind of a mix between the two, regular spiking and Poisson spiking, where a neuron can spike more than once per time step.

Some of the examples of conversion are shown in Fig. 4. These neurons are obtained by converting the rate-based neurons to their spiking version using regular spiking. For example, for the tanh, Fig. 4(a) is created with \( \tau_{ref} = 0.0025 \), which means that the firing will saturate at 400 Hz. The neuron starts with the blank state (i.e., 0 input current and 0 membrane current), which means that the neurons are doing nothing when the simulations start and take a few time steps for the neuron to get going. Toward the mid at \( y = 0 \), the curve gets a bit noisy, since around that region, the neuron has low firing rates and therefore few spikes in that region. It can be seen that the neuron has two kinds of spikes: positive and negative. This kind of spiking behavior is not biologically plausible and, therefore, will not work on most of the neuromorphic hardware. Similarly, Fig. 4(b)–(d) shows the spiking version of sigmoid, rectified linear, and LIFRate-based neurons. It must be noted that the slope of the curve depends on the gain of the neuron. In these examples, the gain of the neurons is adjusted to have not so noisy curves.

IV. ARCHITECTURE AND LEARNING

The main objective of the proposed model is to keep the model biologically plausible and keep the computation low. The LIF has been adopted in this model due to its biologically plausible nature because it requires less computation.

A. Architecture

We have modified the architecture that we proposed in our previous paper [1]. The SNN architecture proposed in this work is shown in Fig. 5. The network is constructed in NengoDL [54], which offers a differential approximation of the firing rate of the LIF neurons. The input to the model is a \( 14 \times 14 \times 3 \) image. The input layer is followed by a convolutional layer of filter size 3 and a total number of 16 filters with stride 2. The convolutional layer is followed by LIF as a nonlinearity function that converts the output into spikes. It is followed by a dense layer with 32 neurons appended with LIF. The last layer of the network is a dense layer that acts as the output layer of the network. The network is trained with a SoftLIF [60] activation (an approximation to LIF) with a multiclass cross-entropy function as an objective function.

B. Loss Function

We have used softmax classifier to obtain the classification probabilities. The softmax uses the cross-entropy as a loss function for \( N \) number of training examples belonging to \( M \) number of classes given by

\[ L = -\frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N} y_n^m \log(h_\theta(x_n, m)) \]  (8)
where $y_m^n$ are true labels for training example $n$ for class $m$ and $h$ is the model with weights $\theta$ taking $x$ as input training example.

### C. Learning Schedule and Weight Initialization

We have used adaptive moment estimation (Adam) optimizer that leverages the potential of adaptive learning rates techniques to obtain optimal learning rates for each parameter. In our training, we set the learning rate ($\alpha$) to 0.002 and the exponential decay rate for the first ($\beta_1$) and second ($\beta_2$) moment estimates were set to 0.9 and 0.999, respectively. The epsilon is set to 1e$^{-8}$ for numerical stability.

The weights for dense layers were drawn randomly from a normal distribution $\mathcal{N}(0, 0.01)$, and for biases, normal distribution with a mean 0.5 has been used. For convolutional layers, the weights $w$ were drawn randomly from between $[-(6/((N_{in} + N_{out}))^{1/2}, (6/((N_{in} + N_{out}))^{1/2}]$, where $N_{in}$ and $N_{out}$ are the number of input and output units, respectively.

### D. Model Testing

During the testing phase, the trained network is rebuilt by using LIF neurons to make the network an SNN. The weights and biases are extracted from the trained model and used to connect spiking LIF neurons that act as a testing model. For testing, the test inputs or samples are modified so that they are presented multiple times or steps to the network to get the accurate measure of the spiking neuron output over time.

### E. Model Hyperparameters

Model hyperparameters are defined as parameters that cannot be learned or calculated from data. They are referred to as tuning parameters because their optimal value cannot be obtained analytically. Optimal settings of these hyperparameters would allow models to learn faster and achieve better performance. The hyperparameters we used for network training are shown in Table II. Apart from the hyperparameters of the network, there are some important parameters for the LIF neuron, which are given as follows [54].

1) $\tau_{re}^c$: It is the membrane RC time constant (in seconds). It determines how quickly the membrane voltage decays to zero in the absence of input.
2) $\tau_{ref}$: It is the absolute refractory period given in seconds. It determines how long the membrane voltage is held at zero after a spike.
3) **Minimum Voltage**: It is the minimum membrane voltage.
4) **Amplitude**: It is the scaling factor on the neuron output.

For our experiments, we have chosen the values shown in Table III.

### V. Results and Discussion

#### A. Dataset

A total of 4800 hand gesture swipes has been collected with a total of five people. The dataset contains eight different gestures that are given as follows:

1) down-up (moving the hand from down to up);
2) up-down (moving the hand from up to down);
3) left-right (swiping the hand from left to right);
4) rubbing (rubbing two fingers together);
5) right-left (swiping the hand from right to left);
6) diagonal-southwest-northeast (swiping the hand from the bottom-left corner to the top-right corner);
7) diagonal-southeast-northwest (swiping the hand from the bottom-right corner to the top-left corner);
8) clapping (clapping two hands).

Each gesture has 600 samples in the dataset. The dataset was collected by offering minimal prior supervision to the users while collecting the dataset both during training and validation. In addition, the dataset was collected in different environmental conditions, such as different locations and surroundings. Moreover, in order to introduce the diversity in the dataset, an example for each gesture was shown to two out of five subjects and they performed the gestures with their own interpretation. Their gestures were no longer as rational as the ones performed by the three minimal supervised subjects. We selected randomly 80% of the dataset for training and the 20% has been used for testing. This is done for several trials. Each gesture is represented by range spectrogram, velocity spectrogram, and angle spectrogram as shown for some gestures in Fig. 8(a)–(c), respectively.
TABLE IV  
COMPARISON OF THE CLASSIFICATION ACCURACY OF THE PROPOSED SNN WITH OTHER MODELS

| Approaches           | Accuracy | Model Size (kB) |
|----------------------|----------|-----------------|
| CNN3D [61]           | 99.63 %  | 12586.58        |
| CNN2D [61]           | 86.25 %  | 375.89          |
| MobileNetV2 - 1 bottleneck [61] | 98.88 %  | 1770.96         |
| MobileNetV2 - 2 bottleneck [61] | 99.00 %  | 2028.85         |
| MobileNetV2 - 3 bottleneck [61] | 97.13 %  | 2287.06         |
| MobileNetV2 - 4 bottleneck [61] | 98.50 %  | 2545.35         |
| MobileNetV2 - 5 bottleneck [61] | 97.75 %  | 2804.27         |
| MobileNetV2 - 6 bottleneck [61] | 98.00 %  | 3063.25         |
| Custom model v_1 [61] | 98.00 %  | 624.92          |
| Custom model v_2 [61] | 97.50 %  | 999.00          |
| Custom model v_3 [61] | 98.13 %  | 1543.89         |
| Custom model v_4 [61] | 97.63 %  | 2233.44         |
| **Proposed SNN**     | **99.50 %** | **75**          |

Fig. 6. Effect of the input presentation time on accuracy.

B. Results

We have used accuracy as a measure to evaluate the classification performance of the proposed system. The system achieves similar average accuracy as is achieved by state-of-the-art methods over random trials, as shown in Table IV. The performance of the different neuron models is compared empirically in Table IV, whereas the prediction of firing choice of the SNN model for some of the examples is shown in Fig. 8(d). The accuracy versus input representation tradeoff is shown in Fig. 6. The feature space visualization for system performance is shown in Fig. 7.

C. Discussion

We have demonstrated the application of SNN in radar-based gesture sensing. One of the key advantages of SNN is energy efficiency, which comes from the temporal processing of the SNN. Based on SNN, we have proposed a gesture recognition system that is resource-efficient in terms of power consumption. Moreover, SNNs are fast, scalable, and easy to implement on hardware, which makes them cost-effective.

Fig. 8(d) shows the firing pattern over time of the SNN for eight gesture classes. It can be observed that the SNN fires correctly for the given example image after a few time steps. This is because for a longer period of time steps, the spikes get integrated over a longer time enabling higher accuracy in predicting the correct class.

Among the various neuron models, we have chosen the LIF model for our proposed SNN because of their biological plausibility and simple implementation requiring less computation. For better performance of the SNN, we tuned the hyperparameters of the network by performing a grid search allowing the network to overfit and underfit. We observed that the SNN performs better with the hyperparameters shown in Table II. We observe that the neuron parameter settings shown in Table II aid the performance of the SNN at the neuron level. With the aforementioned parameters shown in Table II, our proposed system is able to classify eight gestures with a similar level of accuracy 99.50% as conventional deepNets in Table IV. This performance of the proposed system is attributed to the use of spatiotemporal information encoding by SNN, exploiting the network dynamics for learning. Moreover, the size of the proposed model is 75 kB, which is significantly smaller than its counterpart deepNets making it more memory efficient.

To better illustrate the advantages of various neuron types, training neurons, neuron models, and optimization techniques, we empirically compared them with the same network architecture against accuracy, simulation time, and training time for the same epoch runs. We chose various neuron models, such as LIF, and spiking versions of rate-based neurons (LIF rate, rectified linear, sigmoid, and tanh), such as spiking LIF, spiking rectified linear, spiking sigmoid, and spiking tanh, and tested for two optimization techniques, RMSProp and Adam. The results from the empirical study are presented in Table V, which demonstrates that the LIF neuron type...
Fig. 8. Each row represents a sample from each class and choice of firing by SNN. Starting from top row, the samples are from class—0: down up, 1: up-down, 2: left-right, 3: rubbing, 4: right-left, 5: diagonal southwest to northeast, 6: diagonal southeast to northwest, and 7: clapping. (a)–(c) Range spectrogram, velocity spectrogram, and angle spectrogram, respectively. (d) Firing choice of the SNN model.
with Soft LIF achieves the best classification accuracy with comparable simulation and training times to other neuron types and training neurons. Furthermore, it can be noted that for LIF neuron type and training neuron, both optimizers give similar accuracy results, and this is not the case for other neuron types or training neurons.

One of the important factors affecting the accuracy is the number of times the input is presented to the network. Since we are working with static images, therefore, we have to choose a suitable presentation time for which we present the images. This is done to allow the network to accumulate the current for a particular image to make the correct prediction/firing. There is a tradeoff between accuracy and latency when presentation time is taken into consideration. Presenting each image for a longer time allows integration of spikes over a longer time period, thus resulting in better accuracy, and having a shorter presentation time results in higher throughput and less latency. Fig. 6 shows the effect of the presentation time on accuracy. It can be seen that the accuracy is increasing with the increasing presentation time or steps and toward the end, and the increase in accuracy becomes saturated. In contrast to other models, LIF requires fewer steps to achieve its peak accuracy. We have chosen 25 steps as a suitable presentation time for our comparisons as it achieves comparable results to the state of the art for the LIF neuron.

We visualize the high-dimensional feature space to further investigate our system performance. The convolution layers extract the features from the incoming input. To see how these feature space is discriminating enough for each class, we fed the features from the last layer (before classification layer) into the t-distributed stochastic neighbor embedding (t-SNE) algorithm with the corresponding labels. Just like principal component analysis, t-SNE is one of the popular dimensionality reduction methods particularly well suited for the visualization of high-dimensional datasets. However, t-SNE is a probabilistic technique, whereas PCA is a mathematical technique. t-SNE measures the pairwise neighboring similarities first in high-dimensional space and then in corresponding low-dimensional space using the $\ell_2$-norm metric. Once the pairwise similarities are measured, it then scales the distance between two probability distributions using the KL divergence. Fig. 7 shows the low-dimensional (2-D) feature visualization for eight gestures using t-SNE. The tSNE plot shows that the SNN network is able to learn both separable and discriminative features and is capable of generating close-knit clusters for classifying the eight gestures already at four dimensions of embedding space. It depicts that even a low-dimensional SNN is capable of classifying the eight gesture classes reliably.

To evaluate the energy efficiency of the proposed system, we evaluate the energy consumption per classification. For this purpose, we used the hardware metrics of the $\mu$Brain chip defined in [62] and mathematically given as

$$E_c = N_{\text{spikes}} \times E_{\text{spikes}} + \delta T \times P_{\text{leakage}}$$

where $E_c$ is the energy consumed per classification, $N_{\text{spikes}}$ is the maximum number of spikes during classification, $E_{\text{spikes}} = 2.1 \text{ pJ}$ is the energy per spike, $P_{\text{leakage}} = 73 \mu\text{W}$ is the static leakage power, and $\delta T$ is the inference time. Assume that

| Neuron Type          | Accuracy | Optimizer | Training Neuron | Simulation Time | Training Time | No. of Epochs |
|---------------------|----------|-----------|-----------------|-----------------|--------------|--------------|
| LIF                 | 99.37    | RMSprop   | Soft LIF        | 0.5             | 9.29         | 10           |
|                     | 99.50    | Adam      | LIF Rate        | 0.5             | 12.58        | 10           |
| Spiking LIF         | 22.0     | RMSprop   | LIF Rate        | 0.5             | 12.03        | 10           |
| Spiking Rectified Linear | 24.5    | RMSprop   | Rectified Linear| 0.48            | 10.7         | 10           |
| Spiking Sigmoid     | 16.75    | RMSprop   | Sigmoid         | 0.47            | 10.55        | 10           |
| Spiking Tanh        | 16.88    | Adam      | Sigmoid         | 0.8             | 10.5         | 10           |
|                     | 56.5     | RMSprop   | Tanh            | 0.46            | 11.07        | 10           |
|                     | 29.62    | Adam      | Tanh            | 0.5             | 10.69        | 10           |

**TABLE V**

**COMPARISON OF THE CLASSIFICATION ACCURACY OF THE BEST PERFORMING MODELS TO ACHIEVE BETTER ACCURACY**

**Regular Spike Interval**

**Stochastic Spiking**

**Poisson Spiking**
\( \delta T = 28 \text{ ms}. \) The energy consumption per classification of the proposed system is \( E_c = 2.04 \mu \text{J}. \)

The proposed prototype SNN solution is simple but capable of recognizing the performed hand gestures with high accuracy in real time and with low latency that is comparable to their deepNNs counterparts. Furthermore, SNNs are energy-efficient since they consume power only when the corresponding neuron fires, which is sparse by design, making them a low-power hardware-friendly solution ideal for consumer applications.

VI. CONCLUSION

In this article, we presented an SNN-based gesture recognition system using the FMCW radar. We propose to extract range spectrogram, Doppler spectrogram, and angle spectrogram from video of RDIs to be fed as feature images into our proposed SNN architecture. SNNs are conducive for low-latency and low-power embedded implementations and thus are attractive for such human–machine interface applications. A key challenge in training SNNs is optimizing the learning rules to learn spatiotemporal spike trains. In this article, we demonstrate that the proposed spiking network with optimized learning rules and significantly small in size is capable of achieving a similar level of classification accuracy compared to their deep learning counterparts for eight gestures.

REFERENCES

[1] M. Arsalan, M. Chmurski, A. Santra, M. El-Masyr, R. Weigel, and V. Issakov, “Resource-efficient gesture sensing based on FMCW radar using spiking neural networks,” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2021, pp. 78–81.
[2] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, “Multi-sensor system for driver’s hand-gesture recognition,” in Proc. 11th IEEE FG, vol. 1, May 2015, pp. 1–8.
[3] X. Zabulis et al., “Vision-based hand gesture recognition for human-computer interaction,” in The Universal Access Handbook. Springer, Jun. 2009.
[4] X. Ma and J. Peng, “Kinect sensor-based long-distance hand gesture recognition and fingertip detection with depth information,” J. Sensors, vol. 2018, pp. 1–9, Mar. 2018.
[5] A. Malima, E. Ozgur, and M. Cetin, “A fast algorithm for vision-based hand gesture recognition for robot control,” in Proc. 14th IEEE SIV, May 2006, pp. 1–4.
[6] D.-S. Tran, N.-H. Ho, H.-J. Yang, E.-T. Baek, S.-H. Kim, and G. Lee, “Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network,” Appl. Sci., vol. 10, no. 2, p. 722, Jan. 2020.
[7] R. Ramalingam et al., “Wearable smart band for American sign language recognition with polymer carbon nanocomposite-based pressure sensors,” IEEE Sensors Lett., vol. 5, no. 6, pp. 1–4, Jun. 2021.
[8] D. Jiang, Y. Wu, and A. Demosthenous, “Hand gesture recognition using three-dimensional electrical impedance tomography,” IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 9, pp. 1554–1558, Sep. 2020.
[9] S.-W. Byun and S.-P. Lee, “Implementation of hand gesture recognition device applicable to smart watch based on flexible epidermal tactile sensor array,” Micromachines, vol. 10, no. 6, p. 692, Oct. 2019.
[10] M. Georgi, C. Amma, and T. Schultz, “Recognizing hand and finger gestures with IMU based motion and EMG based muscle activity sensing,” in Proc. Int. Joint Conf. BIOSIG, vol. 4, Jun. 2015, pp. 99–108.
[11] A. Ferrone et al., “Wearable band for hand gesture recognition based on strain sensors,” in Proc. 6th IEEE BioRob, Jun. 2016, pp. 1319–1322.
[12] F.-K. Wang, M.-C. Tang, Y.-C. Chiu, and T.-S. Horng, “Gesture sensing using retransmitted wireless communication signals based on Doppler radar technology,” IEEE Trans. Microw. Theory Techn., vol. 63, no. 12, pp. 4592–4602, Dec. 2015.
[13] T. Fan et al., “Wireless hand gesture recognition based on continuous-wave Doppler radar sensors,” IEEE Trans. Microw. Theory Techn., vol. 64, no. 11, pp. 4012–4020, Nov. 2016.
[14] Y. Zhang, S. Dong, C. Zhu, M. Balle, B. Zhang, and L. Ran, “Hand gesture recognition for smart devices by classifying deterministic Doppler signals,” IEEE Trans. Microw. Theory Techn., vol. 69, no. 1, pp. 365–377, Jan. 2021.
[15] V. Lamert et al., “A 122 GHz ISM-band FMCW radar transceiver,” in Proc. 13th GeMIC, Mar. 2020, pp. 96–99.
[16] V. Issakov, A. Bilato, V. Kurz, D. Englisch, and A. Geiselbrechtinger, “A highly-integrated D-band multi-chip transceiver for radar applications,” in Proc. IEEE BCICS, Nov. 2019, pp. 1–4.
[17] J. Rimmelspacher, R. Ciocoveanu, G. Steffan, M. Bassi, and V. Issakov, “Low power low phase noise 60 GHz multichannel transceiver in 28 nm CMOS for radar applications,” in Proc. IEEE RFIC, Aug. 2020, pp. 19–22.
[18] A. Bilato, V. Issakov, A. Mazzanti, and A. Bevilacqua, “A multichannel D-band radar receiver with optimized LO distribution,” IEEE Solid-State Circuits Lett., vol. 4, pp. 144–145, 2021.
[19] E. Aguilar, V. Issakov, and R. Weigel, “A fundamental-frequency 122 GHz radar transceiver with 5.3 dBm single-ended output power in a 130 nm SiGe technology,” in IEEE MIT-S Int. Microw. Symp. Dig., Aug. 2020, pp. 1215–1218.
[20] E. Aguilar et al., “Highly-integrated scalable D-band receiver front-end modules in a 130 nm SiGe technology for imaging and radar applications,” in Proc. GeMIC, Mar. 2020, pp. 68–71.
[21] E. Aguilar, V. Issakov, and R. Weigel, “A 130 GHz fully-integrated fundamental-frequency D-Band transmitter module with > 4 dBm single-ended output power,” IEEE Trans. Circuits Syst. I, Exp. Briefs, vol. 67, no. 5, pp. 906–910, May 2020.
[22] V. Issakov, R. Ciocoveanu, R. Weigel, A. Geiselbrechtinger, and J. Rimmelspacher, “Highly-integrated low-power 60 GHz multichannel transceiver for radar applications in 28 nm CMOS,” in IEEE MIT-S Int. Microw. Symp. Dig., Jun. 2019, pp. 650–653.
[23] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, “Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks,” in Proc. IEEE CVPR, Jun. 2016, pp. 4207–4215.
[24] S. S. Rautaray and A. Agrawal, “Vision based hand gesture recognition for human computer interaction: A survey,” J. Artificial. Intel. Res., vol. 43, no. 1, pp. 1–54, Jan. 2015.
[25] J. Lien et al., “Soli: Ubiquitous gesture sensing with millimeter wave radar,” ACM Trans. Graph., vol. 35, no. 4, Jul. 2016, Art. no. 142.
[26] S. Hazra and A. Santra, “Robust gesture recognition using millimetric-wave radar system,” IEEE Sensors Lett., vol. 2, no. 4, pp. 1–4, Dec. 2018.
[27] A. Santra and S. Hazra, Deep Learning Applications of Short-Range Radars. Norwood, MA, USA: Artech House, Sep. 2020.
[28] Z. Zhang, Z. Tian, and M. Zhou, “Laterm: Dynamic continuous hand gesture recognition using FMCW radar sensor,” IEEE Sensors J., vol. 18, no. 8, pp. 3278–3289, Apr. 2018.
[29] Y. Sun, T. Fei, X. Li, A. Warnecke, E. Warsitz, and N. Pohl, “Multi-feature encoder for radar-based gesture recognition,” in Proc. IEEE GEMAR, Apr. 2020, pp. 1–5.
[30] N. Kern, M. Steiner, R. Lorenzin, and C. Waldschmidt, “Robust Doppler-based gesture recognition with incoherent automotive radar sensor networks,” IEEE Sensors Lett., vol. 4, no. 11, pp. 1–4, Nov. 2020.
[31] M. Altmann, P. Ott, N. C. Stache, and C. Waldschmidt, “Multi-modal cross learning for an FMCW radar assisted by thermal and RGB cameras to monitor gestures and cooking processes,” IEEE Access, vol. 9, pp. 22298–22303, 2021.
[32] K. Ishak, N. Appenrodt, J. Dickmann, and C. Waldschmidt, “Human gesture classification for autonomous driving applications using radars,” in IEEE MIT-S Int. Microw. Symp. Dig., Nov. 2020, pp. 1–4.
[33] M. Q. Nguyen, A. Flores-Nigaglioni, and C. Li, “Range-gating technology for millimeter-wave radar remote gesture control in IoT applications,” in IEEE MIT-S Int. Microw. Symp. Dig., May 2018, pp. 1–4.
[34] C. Gu, J. Wang, and J. Lien, “Motion sensing using radar: Gesture interaction and beyond,” IEEE Microw. Mag., vol. 20, no. 8, pp. 44–57, Aug. 2019.
[35] X. Cai et al., “One-shot radar-based gesture recognizer for fast prototyping,” in Proc. IEEE Sensors, Oct./Nov. 2021, pp. 1–4.
[36] E. Hayashi et al., “RadarNet: Efficient gesture recognition technique utilizing a miniature radar sensor,” in Proc. CHI, May 2021, pp. 1–14.
[37] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, “Thresholding, merging, and temporal fusion for convolutional neural networks for embedded gesture recognition with short range radars,” IEEE Internet Things J., vol. 8, no. 13, pp. 10336–10346, Jul. 2021.
ARSALAN et al.: RadarSNN: RESOURCE-EFFICIENT GESTURE SENSING SYSTEM BASED ON mm-WAVE RADAR

Muhammad Arsalan (Graduate Student Member, IEEE) received the M.Sc. degree in computation engineering from the University of Rostock, Rostock, Germany, in 2013, and the M.Sc. degree in biomedical computing from the Technical University of Munich, Munich, Germany, in 2020. He is currently pursuing the Ph.D. degree at Infineon Technologies AG, Neuiberg, Germany, in collaboration with Technische Universität Braunschweig, Braunschweig, Germany. He has filed four patents and has published several research articles related to various topics of radar machine/deep learning topics, computational photography, and digital watermarking. His research areas include radar signal processing, deep learning, image processing, computer vision, machine learning, and digital watermarking.

Avik Santra (Senior Member, IEEE) received the B.E. degree from the West Bengal University of Technology, Kolkata, India, in 2008, and the M.E. degree (Hons.) in signal processing from the Indian Institute of Science, Bengaluru, India, in 2010. He is currently the Head of the Advanced Artificial Intelligence Team developing signal processing and machine learning algorithms for industrial and consumer radars and depth sensors at Infineon Technologies AG, Neuiberg, Germany. Earlier in his career, he has worked as a System Engineer for LTE/4G modem at Broadcom Communications. He has worked as a Research Engineer with TI-Nexus Group, where he was developing fully adaptive phased array and multi-input–multi output (MIMO) radars. He is the coauthor of the book titled Deep Learning Applications of Short-Range Radars (Arttech House). He has filed over 50 patents and published 40 research articles related to various topics of radar waveform design, radar signal processing, and radar machine/deep learning topics.

Dr. Santra is a reviewer for various IEEE and Elsevier journals. He was a recipient of several outstanding reviewer awards.

Vadim Issakov (Senior Member, IEEE) received the M.Sc. degree in microwave engineering from the Technical University of Munich, Munich, Germany, in 2006, and the Ph.D. degree (summa cum laude) from the University of Paderborn, Paderborn, Germany, in 2010. In March 2010, he joined Infineon Technologies AG, Neuiberg, Germany. Afterward, he worked at imec, Leuven, Belgium, and Intel Corporation, before he came back to Infineon Technologies AG in August 2015 as an mm-wave Design Lead and Principal Engineer leading a research group working on predevelopment of millimeter-wave (mm-Wave) radar and communication products. Since 2014, he has been a Guest Lecturer with Ruhr-University Bochum, Bochum, Germany, and the University of Erlangen, Erlangen, Germany. In September 2019, he joined the University of Magdeburg, Magdeburg, Germany, as a Full Professor holding the Chair for Electronics. Since April 2021, he has been working as a Full Professor with Technische Universität Braunschweig, Braunschweig, Germany. His research interests include mm-Wave circuits, RF systems, mm-Wave measurement techniques, and RF-ESD. He has authored or coauthored over 100 papers in international journals and conference proceedings and 11 patent applications and has published a book on mm-Wave circuits for radar applications.

Dr. Issakov received an award for the outstanding dissertation from the VDI/VDE in Germany and the Best Dissertation Award from the University of Paderborn. His work has been recognized by the IEEE MTT Outstanding Young Engineer Award.

[38] Y. Ren et al., “Hand gesture recognition using 802.11ad mmWave sensor in the mobile device,” in Proc. IEEE WCNCW, Mar. 2021, pp. 1–6.

[40] M. Arsalan and A. Santra, “Character recognition in air-writing based on network of radars for human-machine interface,” IEEE Sensors J., vol. 19, no. 19, pp. 8855–8864, Oct. 2019.

[43] Y. Ren, T. Fei, X. Li, A. Warnecke, E. Warsitz, and N. Pohl, “Real-time ARSALAN et al. RadarSNN: RESOURCE-EFFICIENT GESTURE SENSING SYSTEM BASED ON mm-WAVE RADAR

Muhammad Arsalan (Graduate Student Member, IEEE) received the M.Sc. degree in computation engineering from the University of Rostock, Rostock, Germany, in 2013, and the M.Sc. degree in biomedical computing from the Technical University of Munich, Munich, Germany, in 2020. He is currently pursuing the Ph.D. degree at Infineon Technologies AG, Neuiberg, Germany, in collaboration with Technische Universität Braunschweig, Braunschweig, Germany. He has filed four patents and has published several research articles related to various topics of radar machine/deep learning topics, computational photography, and digital watermarking. His research areas include radar signal processing, deep learning, image processing, computer vision, machine learning, and digital watermarking.

Avik Santra (Senior Member, IEEE) received the B.E. degree from the West Bengal University of Technology, Kolkata, India, in 2008, and the M.E. degree (Hons.) in signal processing from the Indian Institute of Science, Bengaluru, India, in 2010. He is currently the Head of the Advanced Artificial Intelligence Team developing signal processing and machine learning algorithms for industrial and consumer radars and depth sensors at Infineon Technologies AG, Neuiberg, Germany. Earlier in his career, he has worked as a System Engineer for LTE/4G modem at Broadcom Communications. He has worked as a Research Engineer with TI-Nexus Group, where he was developing fully adaptive phased array and multi-input–multi output (MIMO) radars. He is the coauthor of the book titled Deep Learning Applications of Short-Range Radars (Arttech House). He has filed over 50 patents and published 40 research articles related to various topics of radar waveform design, radar signal processing, and radar machine/deep learning topics.

Dr. Santra is a reviewer for various IEEE and Elsevier journals. He was a recipient of several outstanding reviewer awards.

Vadim Issakov (Senior Member, IEEE) received the M.Sc. degree in microwave engineering from the Technical University of Munich, Munich, Germany, in 2006, and the Ph.D. degree (summa cum laude) from the University of Paderborn, Paderborn, Germany, in 2010. In March 2010, he joined Infineon Technologies AG, Neuiberg, Germany. Afterward, he worked at imec, Leuven, Belgium, and Intel Corporation, before he came back to Infineon Technologies AG in August 2015 as an mm-wave Design Lead and Principal Engineer leading a research group working on predevelopment of millimeter-wave (mm-Wave) radar and communication products. Since 2014, he has been a Guest Lecturer with Ruhr-University Bochum, Bochum, Germany, and the University of Erlangen, Erlangen, Germany. In September 2019, he joined the University of Magdeburg, Magdeburg, Germany, as a Full Professor holding the Chair for Electronics. Since April 2021, he has been working as a Full Professor with Technische Universität Braunschweig, Braunschweig, Germany. His research interests include mm-Wave circuits, RF systems, mm-Wave measurement techniques, and RF-ESD. He has authored or coauthored over 100 papers in international journals and conference proceedings and 11 patent applications and has published a book on mm-Wave circuits for radar applications.

Dr. Issakov received an award for the outstanding dissertation from the VDI/VDE in Germany and the Best Dissertation Award from the University of Paderborn. His work has been recognized by the IEEE MTT Outstanding Young Engineer Award.

[38] Y. Sun, T. Fei, X. Li, A. Warnecke, E. Warsitz, and N. Pohl, “Real-time radar-based gesture detection and recognition built in an edge-computing platform,” IEEE Sensors J., vol. 20, no. 18, pp. 10706–10716, Sep. 2020.

[40] M. Arsalan and A. Santra, “Character recognition in air-writing based on network of radars for human-machine interface,” IEEE Sensors J., vol. 19, no. 19, pp. 8855–8864, Oct. 2019.

[43] Y. Ren, T. Fei, X. Li, A. Warnecke, E. Warsitz, and N. Pohl, “Real-time