Spiking Neural Networks for Gesture Recognition Using Time Domain Radar Data

Ahmed Shaaban[#*1], Wolfgang Furtner[#2], Robert Weigel[*3], Fabian Lurz[*4]
[#]Infineon Technologies AG, Munich, Germany
[*]Institute for Electronics Engineering, University of Erlangen-Nuremberg, Erlangen, Germany
{[1]Ahmed.Shaaban, [2]Wolfgang.Furtner}@infineon.com, {[3]Robert.Weigel, [4]Fabian.Lurz}@fau.de

***Abstract* — Gesture recognition using luminance invariant radar sensors is vital due to its extensive use in human-machine interfaces. However, the necessity for computationally expensive radar data pre-processing steps represented by fast Fourier transforms to get range and Doppler features are regarded as a contemporary concern. In this work, we present a solution for gesture recognition that relies on time-domain radar data applied to an event-driven, sparse, and end-to-end trained spiking neural network architecture. Using the proposed solution, it is possible to discriminate between 10 different gestures in a gesture dataset recorded using a 60 GHz frequency-modulated continuous-wave radar sensor, with a mean test accuracy of 93.1%.**

***Keywords* — Spiking Neural Networks, Radar Gesture Recognition, Convolutional Neural Networks, FMCW Radar, Raw Radar Data, Time Domain Radar Data**

I. Introduction

Gesture recognition has always been interesting since it enables the remote control of a wide range of electronics via human-machine interaction, like smart TVs, virtual reality, audio equipment, etc. Because of their high angular resolution, optical-based camera sensors have been utilized to address human-machine interactions [1]. However, the drawback of these camera sensors is that they are highly affected by lighting conditions. Furthermore, because individuals must be in the camera's line of sight, they constitute a significant loss of privacy. Radar sensors, on the other hand, are insensitive to lighting conditions, have negligible privacy issues, can function through obstacles, and can be easily integrated into equipment. Given the advantages of using radar sensors for gesture recognition, several deep-learning-based approaches for recognizing gestures have been introduced [2]-[4]. Although conventional artificial neural networks (ANNs) can accurately classify radar datasets, they lack the complex characteristics of biological neurons and are energy inefficient. Thus, the third generation of ANNs, known as spiking neural networks (SNNs) [5], has been introduced. SNNs differ from traditional ANNs in that they function with continuous temporal data throughout time and generate a sequence of pulses (i.e., spikes) as an output. When comparing ANNs to SNNs at the neuronal level, spiking neurons do not use a fixed non-linearity weighted sum of inputs. Instead, each spiking neuron accumulates input data over time and fires only when its firing threshold is reached, resulting in a sparse network in which not all neurons fire simultaneously. Therefore, they are energy-efficient and well-suited for low-power embedded devices. SNNs have recently been used for gesture recognition in [6]-[8]. However, all of the previously described works have one commonality: they all rely on conventional radar data pre-processing to extract the essential features for classification, such as range, Doppler, and angles. This necessitates the use of computationally expensive fast Fourier transforms (FFTs) on raw radar data in advance. Finding a way to avoid performing FFTs might dramatically reduce computational overhead, allowing for more energy-efficient deployment on embedded devices, as long as the processing overhead in the neural network does not increase proportionally.

Accordingly, the key contributions of the paper are: We provide a new realistic training approach that skips the FFT's pre-processing steps by utilizing just time-domain radar data. A sparse spiking-based convolutional neural network architecture that can discriminate between ten distinct gesture classes is introduced. Because of its low number of computations and sparsity, the proposed SNN model is highly practical. To the best of the author's knowledge, this is the first work using an SNN model to classify human gestures using time-domain radar data with multiple receive antennas.

II. System and Gesture Dataset

A. Dataset Recording

The gesture dataset was recorded using the FMCW 60 GHz BGT60TR13C radar sensor from Infineon Technologies. The radar sensor comes with three receiving antennas (N_{RX}) and one transmitting antenna (N_{TX}). The FMCW radar's transmit signal is made up of chirps, the frequencies of which are swept linearly from a start frequency (f_{min}) of 58 GHz to a stop frequency (f_{max}) of 63 GHz. The received and transmitted chirps are mixed and low-pass filtered to produce the intermediate frequency (IF) signal. The intermediate frequency signal is then sampled 64 times (N_s) with an analog-to-digital converter (ADC) at a sampling frequency (f_s) of 2 MHz.

The recorded gesture dataset includes eight macro gestures: forward, backward, circle-clockwise, circle-anticlockwise, down-top, top-down, left-right, and right-left, as well as two micro gestures: finger-wave and finger-rub. Fig. 1 depicts an overview of the recorded gestures. Five different persons contributed to the dataset. Each gesture was recorded 200 times, and each record consisted of 60 frames, with 32 chirps and 64 sample points in each frame. Accordingly, the

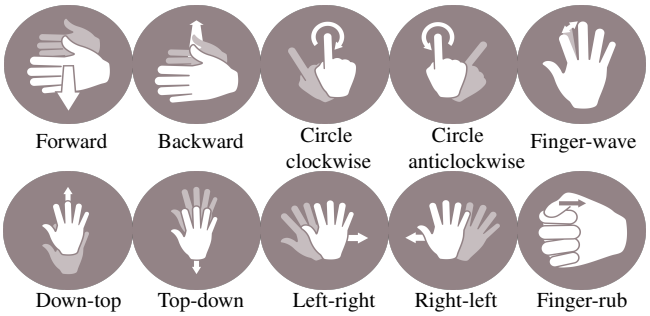

Fig. 1. The ten different recorded gestures.

dimension of the raw data is [recordings, antennas, frames, chirps, samples] with values of [2000, 3, 60, 32, 64].

B. Dataset Preparation

The mean value along the dimension of both the samples and the chirps is removed to overcome transmitter-receiver leakage and the influence of static objects. These procedures are commonly known as DC removal and moving target indication (MTI) [9]. Furthermore, as will be indicated in section IV-C, the suggested training approach is carried out on a frame-by-frame basis. Consequently, a frame-by-frame min-max normalization was performed, where the 60 frames of each record for each antenna were normalized. Thus, the final time-domain data that will be utilized in the rest of the paper is in the form of [recordings, antennas, frames, chirps, samples] with values inside the frame's dimension (chirps, samples) ranging from zero to one, which would be more advantageous to the training approach.

III. Conventional Pre-processing Approach

All of the pre-processing steps are applied to the frame-by-frame normalized recordings as already mentioned in section II-B. The range-Doppler images (RDIs) for each frame are then extracted using 2D FFTs; details on how to generate the RDIs are mentioned in [10]. Since the radar's three receiving antennas are arranged in a triangular pattern, it was possible to estimate the angles in the azimuth and elevation directions with only two antennas. Correspondingly, for the RDI of each frame, using the first and third antennas, it was feasible to retrieve the 32 azimuth-angles corresponding to the range-Doppler bins with the maximum amplitude in that RDI. The same is performed for the elevation angles while utilizing the second and third antennas. Finally, the RDIs of successive frames got concatenated to generate range spectrograms and Doppler spectrograms. Therefore, each record has four available pre-processed features: range spectrograms, Doppler spectrograms, azimuth angles, and elevation angles.

IV. Proposed Solution

SNN's spiking neurons are distinguished by the existence of an internal state (membrane potential) and recurrent feedback connections that update their internal state at each time step, allowing them to exploit the dynamic temporal nature of the sequentially based datasets [11]. Moreover, the radar data includes valuable temporal information of the gestures represented by the number of frames in each record. As a consequence, the idea of directly using the time-domain radar data on an SNN-based architecture emerged.

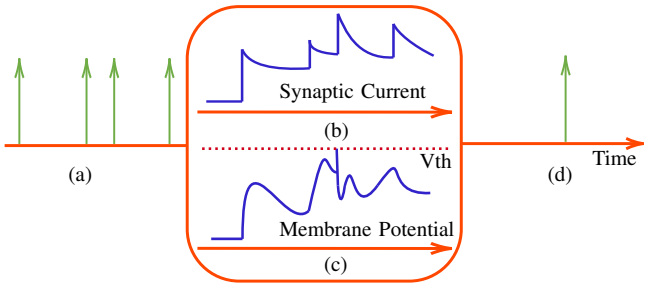

Fig. 2. The synaptic spiking neuron's working principle. (a) Synaptic spiking neuron input spikes over time. (b) The synaptic current integrates input spikes and decays with a decay rate of alpha (α). (c) Membrane potential decays with a beta (β) decay rate as it integrates across the synaptic current. (d) Only when the membrane potential reaches its threshold does an output spike occur. Adapted from [14].

A. Spiking Neural Networks

SNNs use spiking neurons that receive and transmit discrete spikes (either zero or one), which has the advantage of being sparse and energy-efficient in comparison to conventional ANNs. For a realistic, implementable solution, an SNN model that could be trained end-to-end in the spiking domain is desired. However, because discrete spikes are undifferentiable in the spiking domain, the issue of not being able to employ the most successful training algorithm "backpropagation" would arise [12]. Nonetheless, it was possible to train the SNNs end-to-end in the spiking domain using the surrogate gradient descent idea described in [13], in which the gradient of the undifferentiable spikes is replaced in the backward pass by the gradient of another differentiable function. The spiking simulator snnTorch was utilized to realize and train the spiking neural network used in this work [14]. The spiking neuron used is a variant of the leaky integrate-fire (LIF) neuron [15]. The synaptic spiking neuron shown in Fig. 2 was chosen as our spiking neuron because it has the most biological plausibility. It is distinct from the other commonly used LIF neuron variants in that it has two decay rates: α (synaptic current decay rate) and β (membrane potential decay rate). In the synaptic neuron, as a first step,

$$I_{syn}[t] = \alpha I_{syn}[t-1] + WX[t] \tag{1}$$

the synaptic current ($I_{syn}[t]$) relies on the weighted input spike ($WX[t]$) along with a decayed version of the synaptic current in the previous step ($\alpha I_{syn}[t-1]$). Subsequently,

$$U[t] = \beta U[t-1] + I_{syn}[t] \tag{2}$$

the membrane potential ($U[t]$) combines the generated synaptic current ($I_{syn}[t]$) with a decayed form of the preceding membrane potential ($\beta U[t-1]$).

$$S_{out}[t] = \left\{ \begin{array}{ll} 1, & if U[t] > U_{thr} \\ 0, & otherwise \end{array} \right\}, \tag{3}$$

an output voltage spike ($S_{out}[t]$) gets generated when the membrane potential ($U[t]$) surpasses the spiking neuron

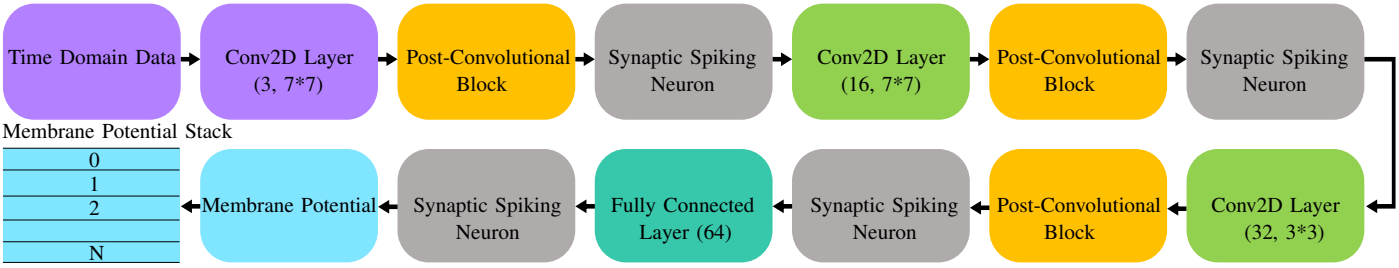

Fig. 3. The architecture of the SCNN model. The numbers on each convolutional layer represent the input channels and the kernel size. The first convolutional layer has three input channels, indicating that the network was fed a single frame (chirps, samples) from a single record at a time, with the input channels being the three antennas. The post-convolutional block is a 2D max-pooling with a stride of 2, followed by 2D batch normalization. The first synaptic spiking neuron is responsible for transforming the output of the first convolutional layer and the first post-convolutional block into spikes, which are subsequently propagated to the network's following layers. The membrane potential from the last synaptic spike neuron is stacked for each frame, where N denotes the number of frames.

threshold (U_{thr}), and the membrane potential will have a threshold value subtracted from it.

B. Proposed Spiking Neural Network Architecture

The introduced spiking convolutional neural network (SCNN) model was constructed using [14]. The model uses the synaptic spiking neuron and approximates the undifferentiability of discrete spikes in the backward pass while training by a gradient of the fast sigmoid function. Fig. 3 presents the architecture of the SCNN model. To gain the most out of the temporal information in the gestures, the first convolutional layer and the first post-convolutional block were used directly on the time-domain input data before their output was converted into spikes via the first synaptic spiking neuron. Thus, allowing spikes to propagate throughout the network. The architecture's first convolutional layer has a padding of one and a stride of two, and the same is true for the second convolutional layer. The first two convolutional layers have a kernel size of seven. The third convolutional layer has padding and stride of two and a kernel size of three.

C. Proposed Training Approach

The proposed approach's procedure shown in Fig. 4 can be summarized as follows: The 60 frames (one record) of radar time-domain data were used as the time steps over which the SCNN iterated in the network forward pass. As a result, and as shown in Fig. 3, one frame (time-step) was sent to the network at a time, and the membrane potential from the architecture's final synaptic neuron was stacked at the end of each frame. After stacking the membrane potentials from each frame, the LogSoftmax function is used for this stack of 60 frames membrane potentials. The negative log-likelihood loss (NLLLoss) function iterates through the LogSoftmax function output, generating a separate loss for each frame, and the total loss is the sum of the losses from each frame. Finally, the gesture with the highest prediction across all frames in a single record is designated as the classified gesture. In comparison to the record-by-record training approach, which requires a fully pre-processed gesture (full stack of frames) to be sent to the network, this approach is more realistic since it can process the frames directly and does not require any additional pre-processing steps.

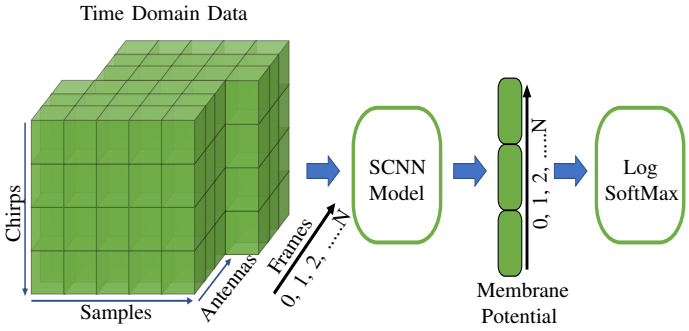

Fig. 4. An overview of the proposed frame-by-frame training approach.

V. Experiments and Results

A. Setup and Optimization

The recorded data, which totaled 2000 recordings for the ten distinct gestures, was split into three datasets: training, validation, and testing. There are 1280, 320, and 400 recordings for training, validation, and testing, respectively.

A conventional-ANN variant of the SCNN has been introduced by replacing ReLUs for synaptic spiking neurons and adding a dropout layer with a probability of 0.25 before transmitting the output of the final convolution layer to the fully connected layer. This ANN variant, which is essentially a convolutional neural network (CNN), was used to assess the conventional pre-processing approach. Therefore, the CNN model was trained record-by-record, with the four ready pre-processed features indicated in section III being input to the network from each record. On the other hand, the SCNN architecture was trained using the frame-by-frame training approach mentioned in section IV-C, on the time-domain data described in section II-B.

During training, an ADAM optimizer has been used. The LogSoftmax and NLLLoss functions were used to compute the loss as stated in section IV-C. All the model's hyperparameters were optimized using Optuna [16]. The SCNN model's hyperparameters were as follows: α = 0.65, β = 0.36, synaptic neuron threshold = 0.5, batch size = 25, optimizer learning rate of 1×10^{-3}, and optimizer weight decay of 6.14×10^{-6}. The CNN model was likewise trained with the same batch size, learning rate, and weight decay as the SCNN model.

B. Classification Accuracies

Both models have been trained for 80 epochs, and early stopping with a patience of 10 epochs was used. The models with the lowest validation loss were chosen as the best during training. These models were then tested on the separately partitioned test dataset to estimate the test accuracy. To verify the results, both experiments were reproduced with the same ten different random seeds to estimate the mean test accuracy. The SCNN model's and CNN model's mean test accuracy were 93.100% and 93.225%, respectively.

C. Computational Complexity

The traditional pre-processing approach primarily relies on the use of 2D FFTs with a computational complexity of $(N_c N_s \log(N_c N_s))$ to get the RDI for each frame, where N_c is the number of chirps and N_s is the number of samples. Although the exact number of multiply-accumulate (MAC) operations for the 2D FFTs relies on implementation specifics along with windowing and slicing operations, it is regarded as a significant number due to its complexity and MACs being done in the complex domain. In addition, the record-by-record trained CNN requires 1.6 million MACs per inference, with all MAC operations being float values. On the contrary, in the time-domain proposed approach, the 2D FFT steps are completely skipped. Moreover, applying this approach to the SCNN has the extra benefit of regarding MACs with spikes as conditional additions. Because, from 1, when the 1-bit spikes $X[t]$ equals one, the synaptic weights W are simply added to the synaptic current. Finally, the SCNN has an upper limit of 85 million real-valued MAC operations for processing and inference, highlighting that the MAC operations following any spiking neuron are essentially just conditional additions.

D. Discussion

As described in section V-C, when comparing the time-domain frame-by-frame-based training approach to the conventional pre-processed record-by-record training approach, the significant benefit is that the expensive FFTs pre-processing steps are bypassed, resulting in reduced processing effort. Furthermore, SNNs are more energy-efficient than ANNs since they communicate using spikes, making them sparse and causing any MAC operations with spikes between their layers to be considered as conditional additions. The experimental results of this work show that the accuracy resulting from combining the time-domain training approach with the SCNN is on par with the conventional computationally expensive pre-processing approach used on a conventional CNN. As a consequence, the proposed solution has the advantage of saving the computations required for pre-processing steps and the network MACs while achieving accuracy comparable to the conventional state-of-the-art model. That is because executing the gesture, then the FFT pre-processing steps, and finally inferring from a CNN network with non-sparse MACs is considered significantly more computationally expensive than the proposed solution, which does not use FFTs and employs sparse MACs. The proposed system's effectiveness is attributed to flowing through the frames in each record, frame-by-frame, along with the synaptic spiking neuron's ability to learn spatio-temporal information; hence, the SCNN accurately learned the temporal information within the gestures.

VI. Conclusion

In this work, a time-domain training approach has been used directly on an event-driven, sparse, and end-to-end trained spiking neural network architecture to classify ten distinct gestures recorded from five different individuals. Classification accuracy as an evaluation metric showed that the proposed solution is on a par with the conventional computationally more expensive approach. These considerations contribute to the proposed system's suitability for usage on edge devices.

References

[1] C. Zimmermann and T. Brox, "Learning to Estimate 3D Hand Pose From Single RGB Images," 2017, pp. 4903–4911.

[2] J. Lien et al., "Soli: ubiquitous gesture sensing with millimeter wave radar," ACM Trans. Graph., vol. 35, no. 4, p. 142:1-142:19, Jul. 2016.

[3] M. G. Amin, Z. Zeng, and T. Shan, "Hand Gesture Recognition based on Radar Micro-Doppler Signature Envelopes," in 2019 IEEE Radar Conference (RadarConf), Apr. 2019, pp. 1–6.

[4] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic Continuous Hand Gesture Recognition Using FMCW Radar Sensor," IEEE Sensors Journal, vol. 18, no. 8, pp. 3278–3289, Apr. 2018.

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

[6] D. Auge, J. Hille, E. Mueller, and A. Knoll, "Hand Gesture Recognition in Range-Doppler Images Using Binary Activated Spiking Neural Networks," in 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021), Dec. 2021.

[7] M. Arsalan, M. Chmurski, A. Santra, M. El-Masry, R. Weigel, and V. Issakov, "Resource Efficient Gesture Sensing Based on FMCW Radar using Spiking Neural Networks," in 2021 IEEE MTT-S International Microwave Symposium (IMS), Jun. 2021, pp. 78–81.

[8] I. J. Tsang, F. Corradi, M. Sifalakis, W. Van Leekwijck, and S. Latré, "Radar-Based Hand Gesture Recognition Using Spiking Neural Networks," Electronics, vol. 10, no. 12, Art. no. 12, Jan. 2021.

[9] M. A. Richards et al., "Principles of Modern Radar Volume I- Basic Principles." 2010.

[10] S. Hazra and A. Santra, "Short-Range Radar-Based Gesture Recognition System Using 3D CNN With Triplet Loss," IEEE Access, vol. 7, pp. 125623–125633, 2019.

[11] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training Deep Spiking Neural Networks Using Backpropagation," Frontiers in Neuroscience, vol. 10, p. 508, 2016.

[12] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[13] E. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks," IEEE Signal Processing Magazine, 2019.

[14] Eshraghian, J.K. et al., "Training spiking neural networks using lessons from deep learning", arXiv preprint arXiv:2109.12894, 2021.

[15] W. Gerstner and W. M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, 2002.

[16] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, Jul. 2019, pp. 2623–2631.