RT-SCNNs: real-time spiking convolutional neural networks for a novel hand gesture recognition using time-domain mm-wave radar data

Ahmed Shaaban[1,2] (iD), Maximilian Strobel[2], Wolfgang Furtner[2], Robert Weigel[1] (iD) and Fabian Lurz[1,3] (iD)

[1]Institute for Electronics Engineering, University of Erlangen-Nuremberg, Erlangen, Germany; [2]Infineon Technologies AG, Munich, Germany and [3]Chair of Integrated Electronic Systems, Otto-von-Guericke-University Magdeburg, Magdeburg, Germany

Abstract

This study introduces a novel approach to radar-based hand gesture recognition (HGR), addressing the challenges of energy efficiency and reliability by employing real-time gesture recognition at the frame level. Our solution bypasses the computationally expensive preprocessing steps, such as 2D fast Fourier transforms (FFTs), traditionally employed for range-Doppler information generation. Instead, we capitalize on time-domain radar data and harness the energy-efficient capabilities of spiking neural networks (SNNs) models, recognized for their sparsity and spikes-based communication, thus optimizing the overall energy efficiency of our proposed solution. Experimental results affirm the effectiveness of our approach, showcasing significant classification accuracy on the test dataset, with peak performance achieving a mean accuracy of 99.75%. To further validate the reliability of our solution, individuals who have not participated in the dataset collection conduct real-time live testing, demonstrating the consistency of our theoretical findings. Real-time inference reveals a substantial degree of spikes sparsity, ranging from 75% to 97%, depending on the presence or absence of a performed gesture. By eliminating the computational burden of preprocessing steps and leveraging the power of (SNNs), our solution presents a promising alternative that enhances the performance and usability of radar-based (HGR) systems.

CAMBRIDGE
UNIVERSITY PRESS

Introduction

Hand gesture recognition (HGR) is a rapidly growing field with potential applications in a wide range of domains, including smart TVs, automotive systems, and virtual reality [1]. Camera-based (HGR) systems are widely used, but they suffer from privacy concerns and performance issues in challenging environments [2, 3]. Non-vision solutions such as wearable sensors have been proposed to address these limitations, but they can be uncomfortable to wear [4, 5]. Radar-based (HGR) solutions offer distinct advantages, including privacy preservation, immunity to illumination variations, and seamless integration into various operating environments [6]. As a result, extensive research has been conducted to explore the use of conventional artificial neural networks (ANNs) for gesture recognition using radar technology [7–10]. However, the utilization of ANNs, such as convolutional neural networks (CNNs), for inference requires extensive non-sparse multiply-accumulate (MAC) operations between network layers, making them unsuitable for AI applications that require great computational efficiency and resource optimization [11].

Spiking neural networks (SNNs) [12] represent a type of ANN designed to emulate the behavior of biological neurons in the human brain. Communication within SNNs is achieved through discrete electrical pulses called spikes, in contrast to the continuous-valued activation functions employed by conventional ANNs. SNNs have several advantages over conventional ANNs. They exhibit prominent energy efficiency since energy consumption is limited to spike generation, rendering them particularly suitable for low-power devices and applications. Additionally, SNNs can exploit the dynamic temporal nature of data, as they can track the timing of spikes to learn temporal patterns, making them highly suitable for analyzing sequential datasets, including HGR data.

SNNs have been used for their inherent benefits in gesture recognition. In [13], the authors explored gesture classification on the Soli [14] and Dop-NET [15] datasets using a spiking liquid state machine along with traditional machine learning classifiers. The Soli and DopNET datasets are provided in a preprocessed format (non-time-domain data). Therefore, an

additional conversion step is employed to transform the preprocessed range-Doppler maps (RDMs) and micro-Doppler radar signals into spike representations. Also, their approach utilizes a single receiving antenna output signal. Authors in [16] performed gesture recognition on datasets from [17] and Soli [14]. The [17] dataset, range profiles are obtained through discrete Fourier transform (DFT) and short-time Fourier transform (STFT), followed by conversion into spikes using a time-to-first-spike scheme [18] and classification using a convolutional-based SNN. The Soli range-Doppler signals are directly encoded into spikes using thresholding and then classified through the SNN model. In [19], a self-recorded dataset of three gestures (Push, SwipeLeft, SwipeRight) and a Background class is classified using a fully connected SNN. They preprocess the raw data with 2D fast Fourier transforms (FFTs) to obtain RDMs and calculate the azimuth angle information using two out of three receiving antennas. The maps and angles are then encoded into spikes using binary encoding. Their results showed improved accuracy by incorporating both RDMs and angles, surpassing a recurrent neural network (RNN)-based baseline performance while being computationally efficient.

The existing solutions, including ANNs and sparse SNNs, typically depend on the conventional radar preprocessing chain to extract range, Doppler, or angular information as the initial step in the network input procedure. Nevertheless, this preprocessing chain is computationally expensive and introduces substantial complexity. The solution proposed in [20] addresses these challenges using only raw time-domain radar data with simplified processing. They skip the 2D FFTs for RDMs and emulate DFT operations in the initial layer of their SNN network. Also, they exclude Doppler information and angular data, achieving 98.1% accuracy in distinguishing four self-recorded gestures. The time-domain to spikes conversion is integrated within the SNN network, eliminating the need for a separate step.

In our earlier work, presented at the EUMW2022 conference and published in its proceedings [1], we propose a gesture recognition solution that combines time-domain radar data with a spiking convolutional neural network (SCNN) architecture. Our approach achieves comparable performance to a solution that incorporates 2D FFTs, azimuth, elevation angle extraction, and classification using a conventional CNN. In a subsequent work [21], we demonstrated that our proposed solution outperforms ANNs when applied to SNNs. Unlike [13, 16, 19], our approach in [1] avoids the complexity and computational cost of radar preprocessing 2D FFT steps and eliminates the need for encoding data into spikes by directly processing the time-domain data through the SNN network, preserving all information without loss. In contrast to [20], we utilized all 32 chirps to capture the entire Doppler information from the time-domain data, and unlike [13, 16, 19, 20], we leverage signals from all three receiving antennas to capture complete angular information. This enables effective differentiation between gestures with similar range-Doppler characteristics, such as distinguishing between SwipeLeft and SwipeRight gestures.

Our current work presents the following contributions in comparison to the conference paper [1]:

(1) **Dataset Enhancement:** We utilize a more complex dataset with a larger size and diverse recording specifications, surpassing the previous work [1].
(2) **Gesture Frame Detection:** We introduce and utilize this process to effectively discriminate between specific gesture frames and non-gesture frames.

(3) **Enhanced Time-Domain Processing:** Our proposed time-domain processing approach in [1] is modified to enable the direct prediction of gestures on a frame basis, aligning with the new labeling format based on the gesture frame detection process.
(4) **Simplified SCNN Architecture:** We streamline the proposed (SCNN) architecture in [1], by replacing complex synaptic spiking neurons with simpler leaky integrate and fire (LIF) spiking neurons [22] and reducing the number of layers.
(5) **Live Testing and Evaluation:** We conduct live testing to validate our results and comprehensively assess the capabilities of our solution, including real-time control of presentation slides and engaging in online gaming activities.
(6) **Computational Complexity Analysis:** We provide a detailed estimation of the computational complexity associated with our solution, offering insights into its computational efficiency.

To the best of our knowledge, this study, in contrast to existing works [1, 13, 16, 19–21], introduces and evaluates frame-based HGR prediction, enabling real-time utilization for the first time. Consequently, it also demonstrates live testing of this approach, thereby validating the presented results and directly assessing the computational complexity and sparsity of the solution in real-time scenarios.

The remainder of this article is structured as follows: Section "Radar System Design" outlines the radar hardware and configuration utilized in this study and provides comprehensive insights into the gesture dataset. Section "Gesture Frame Detection: A Key Step in Data Preprocessing for Training Enhancement" focuses on the gesture frame detection process. In "Hand Gesture Recognition Proposed System" section, the utilization of SNNs, along with the methods employed for time-domain data preparation and the time-domain processing approach, are thoroughly discussed. Section "Data Preparation and Model Evaluation" discusses dataset splitting and presents an overview of the training and testing procedures. Section "Experimental Results" is dedicated to presenting the experimental results. The "Discussion" section discusses the results. Finally, the "Conclusion" Section provides several concluding remarks.

Radar system design

System hardware

This work employs the BGT60TR13C radar chipset, a frequency-modulated continuous-wave (FMCW) radar chipset developed by Infineon Technologies [23, 24]. FMCW radars offer the advantage of a compact form factor, facilitating their efficient deployment in various applications. These radar systems provide valuable insights into target characteristics such as range and velocity through the analysis of the signals they produce [25, 26].

Figure 1(a) depicts a simplified block diagram of the BGT60TR13C chipset, which retains only a single transmitter and receiver. On the transmitter side, a phase-locked loop (PLL) governs the linear frequency sweeping process. A reference oscillator at 80 MHz clocks the loop, and a finite state machine (FSM) controlled by the same reference clock generates a linear voltage ramp. This voltage ramp is applied to a voltage-controlled oscillator (VCO), which generates continuous signals known as chirps. These chirps exhibit linear frequency-modulated (LFM) characteristics, spanning a frequency range from 58.5 to 62.5 GHz for the proposed application. The transmit output power of the
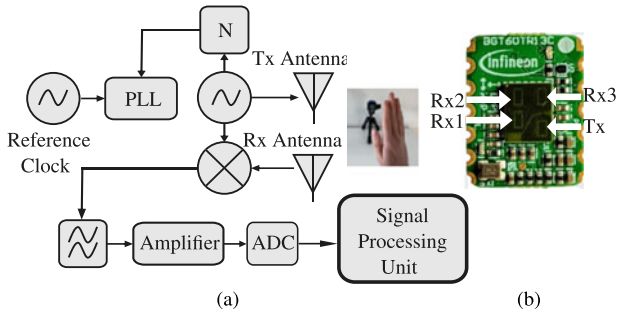
Figure 1. (a) FMCW radar block diagram. (b) Infineon's FMCW radar chipset with a transmitter antenna and three L-shaped receiving antennas.

LFM signal at the transmitter antenna port is approximately 5 dBm. Following transmission, radar-receiving antennas capture the backscattered signal from the target. Upon reception, the received signal undergoes a low-noise amplification of about 12 dB. It is then time-domain mixed with the transmitter signal, followed by a high-pass filter to eliminate frequencies below 100 kHz and an anti-aliasing filter (AAF) to eliminate frequencies above 600 kHz. As a result, the intermediate frequency (IF) signal is significantly narrower than the originally transmitted signal [23, 24]. This narrow bandwidth greatly enhances subsequent processing efficiency. As part of the successive stages of processing, the IF signal is directed to an analog-to-digital converter (ADC) with a sampling rate of 2 MHz and a resolution of 12 bits. For more information on the chipset hardware, the reader can refer to [23, 24].

Further, as shown in Fig. 1(b), the radar system incorporates three receiving antennas arranged in an L-shaped configuration. As a result, the received IF digital signal from all three receiving antennas encompasses valuable target information, such as range, Doppler, and angles. These values can be estimated through additional utilization of digital signal processing algorithms, as will be elaborated in "Gesture Frame Detection: A Key Step in Data Preprocessing for Training Enhancement" section.

System parameters

In the experimental setup, the radar chip is configured to generate 32 chirps per frame. Each chirp undergoes frequency modulation spanning from 58.5 to 62.5 GHz, effectively covering a bandwidth of 4 GHz. A single gesture recording comprises 100 frames, with each frame lasting 30 ms. Thereby, the cumulative duration of a complete gesture recording amounts to approximately 3 seconds. The output shape of the recording is represented as (frames, chirps, samples), and given the presence of three receiving antennas, the final output shape is denoted as (frames, antennas, chirps, samples), with dimensions of (100, 3, 32, 64), respectively. Table 1 provides an overview of the radar operating parameters adopted throughout the work.

Based on the aforementioned operating parameters, several key metrics can be derived. The maximum measurable Doppler velocity, i.e.,

$$V_{\max} = \frac{c_o}{4 \cdot f_{\text{center}} \cdot T_c}, \tag{1}$$

is estimated to be 4.13 m/s, with c_o representing the speed of light. In the context of Doppler resolution,

$$V_{\text{res}} = \frac{2V_{\max}}{N_c}, \tag{2}$$

Table 1. Radar operating parameters

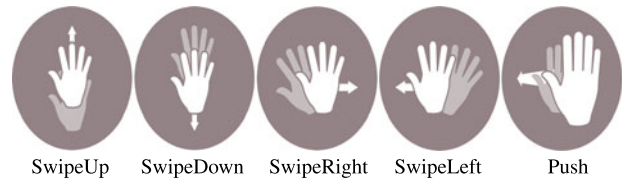| Parameter	Symbol	Value
Start frequency	$f_{\min}$	58.5 *GHz*
Stop frequency	$f_{\max}$	62.5 *GHz*
Center frequency	f_{center}	60.5 *GHz*
Bandwidth	B	4 *GHz*
Number of samples per chirp	N_s	64
Number of chirps	N_c	32
Chirp repetition time	T_c	0.3 *ms*
Frame repetition time	T_f	30 *ms*
Number of frames	N_f	100
Number of transmit antennas	N_{TX}	1
Number of receive antennas	N_{RX}	3
Sampling frequency	F_s	2 *MHz*

Figure 2. A visual representation showcasing the execution of the recorded gesture.

an approximate value of 0.26 m/s is obtained. The range resolution:

$$R_{\text{res}} = \frac{c_o}{2B}, \tag{3}$$

stands at 0.038 m. Finally, the maximum unambiguous range,

$$R_{\max} = R_{\text{res}} \times \frac{N_s}{2}, \tag{4}$$

is identified as 1.2 m [25, 26].

Gesture dataset acquisition

This paper presents a significant expansion to the previous dataset in [1], with an increase in size and a broader range of specifications. The new dataset contains 19,400 recordings for five macro gestures (Push, SwipeRight, SwipeLeft, SwipeDown, and SwipeUp), representing approximately a tenfold increase over the previous dataset, which had 2000 recordings for eight macro and two micro gestures. We have chosen to focus on these five gestures due to their direct relevance to task execution and system control, as well as their user-friendliness and robustness against variations in user execution. Additionally, the dataset incorporates a Background class to indicate the absence of any specific gesture.

Figure 2 presents a visual representation of the execution of these gestures. To ensure comprehensive scenarios and introduce variability, the executed gestures incorporated deliberate variations in angles, distances, and hand dominance (right and left), while concurrently adjusting the radar holder height. The gestures were performed in a room with only static objects (walls, chairs, and tables) and no other moving objects. A single individual performed the gestures with both hands while standing at discrete distances of 0.6, 0.8, and 1.0 m, maintaining three distinct angles relative to
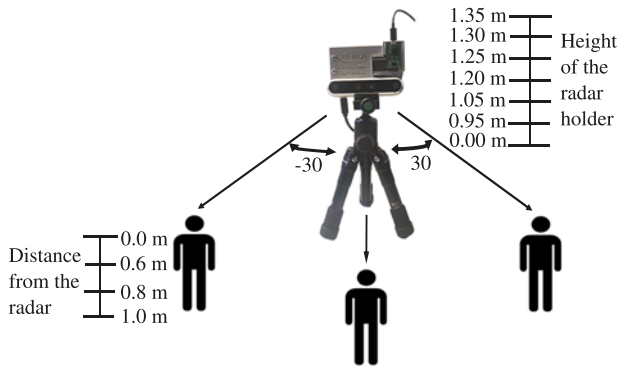
Figure 3. Configuration of the radar recording setup, illustrating the range of recording positions, angles, and height adjustments of the radar holder.

the radar at each recording distance: directly facing (0°), slightly turned to the left (-30°), and slightly turned to the right (30°). Similarly, the radar holder height was adjusted while recording within a range of 0.95–1.35 m to capture data from various vertical positions. Figure 3 illustrates the different recording scenarios to clarify the experimental setup.

The adopted recording criterion and environment yielded a clean, non-noise-limited gesture dataset with minimal intra-class variability between the different gestures while simultaneously incorporating diverse, realistic variations in gesture execution. Given that the dataset comprises a collection of 19,400 recordings, and based on the information provided in "System Parameters" section, the final shape of the dataset, including all available recordings, is presented as (recordings, frames, antennas, chirps, and samples), with respective values of (19,400, 100, 3, 32, 64).

Gesture frame detection: a key step in data preprocessing for training enhancement

As demonstrated in the "System Parameters" section, a gesture comprises a sequential series of 100 frames. However, real-world gestures are performed infrequently and within a few hundred milliseconds. This implies that among the 100 frames recorded for

a gesture, only a few capture the actual gesture movement, while the majority capture Background noise. To develop a robust solution for real-world scenarios, differentiating between gesture frames and non-gesture frames is crucial for minimizing false alarms. Therefore, this section introduces a novel approach to precisely identify the exact frame at which a gesture occurs [27].

It is worth noting that while this approach incorporates conventional radar preprocessing procedures involving two FFT steps, its main objective revolves around detecting the frame index where the hand is closest to the radar. Thereupon, a windowing technique is applied around this frame index to label the entire frames where the gesture is occurring. This approach enhances the effectiveness of SNN training detailed in the "Time-Domain Processing Approach" section by enabling the spiking neurons to extract essential features from the temporal information in the time-domain gesture frames.

It should be noted that these steps are specifically carried out during the preparation of the time-domain dataset for training, as clarified in the "Refining Time-Domain Gesture Data" section. Therefore, they are only implemented in software and are not required for live testing, as detailed in the "Live Testing" section. This guarantees that live testing is conducted exclusively with time-domain data without conventional preprocessing steps.

Figure 4 provides a comprehensive overview of the gesture frame detection process.

Range information

Initially, the ADC time-domain raw data within each frame undergoes a preprocessing operation to alleviate the potential impact of transmitter-receiver antenna leakage. This operation entails subtracting the mean along the fast-time (a.k.a. samples) dimension, thereby effectively removing the DC component. The presence of static objects in the surrounding environment of the radar system can impede the accurate detection of gesture reflections. To address this issue, a moving target indication (MTI) removal step is employed. By subtracting the mean across the slow-time (a.k.a. chirps) dimension, signals arising from static objects are effectively eliminated, thus enhancing the clarity of gesture reflections.
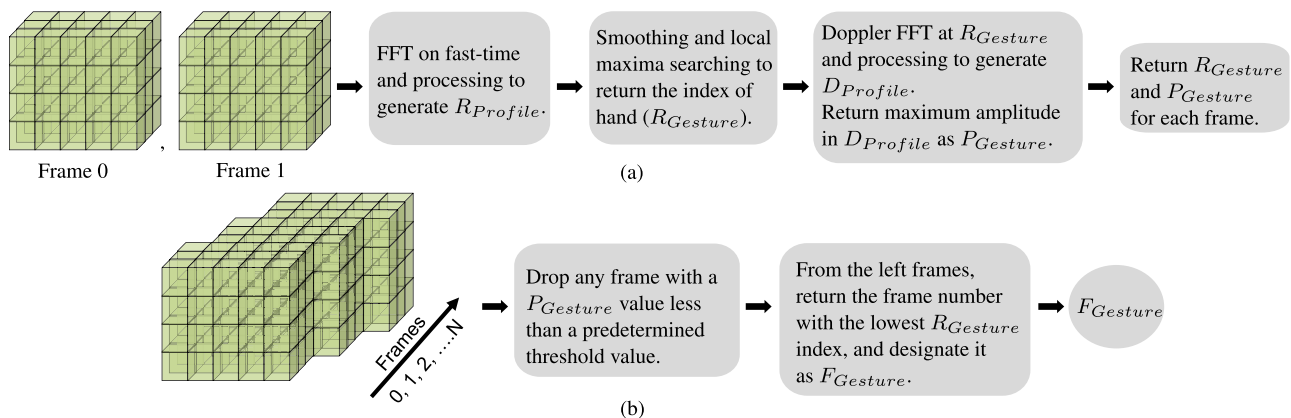
Figure 4. Outline of the gesture frame detection process: (a) Preprocessing on each frame involves a first FFT on the fast-time dimension to generate the range profile ($R_{Profile}$). Smoothing and refinement locate the first local maxima as $R_{Gesture}$, representing the range bin of the hand. A Doppler FFT on $R_{Gesture}$ produces the Doppler profile ($D_{Profile}$). The peak signal amplitude in $D_{Profile}$ is designated as $P_{Gesture}$. (b) Frame refinement: Using $R_{Gesture}$ and $P_{Gesture}$ values for each frame, a refinement process is performed across all frames. Frames with a $P_{Gesture}$ value below the predetermined threshold are discarded as they are considered not to contain any gesture. From the remaining frames, the frame closest to the radar, determined by the nearest $R_{Gesture}$ index, is identified as $F_{Gesture}$, indicating the frame where the hand performed the gesture and was closest to the radar.

To extract range information, an FFT is applied to the preprocessed and filtered data along the fast-time dimension [13]. This allows for the identification of specific frequency components associated with target reflections. Following these steps, the range information (R) is subjected to extra analysis. In this regard, the absolute values of the mean along both antennas and slow-time dimensions are computed, yielding the range profile ($R_{Profile}$). This profile provides valuable insights into the spatial distribution and intensity of target reflections.

Range profile processing

After obtaining the $R_{Profile}$ for each frame, the next step involves identifying the bin with the highest value in the $R_{Profile}$, as the global maximum range bin. Further refinement procedures are then conducted to determine the presence of local maxima within the $R_{Profile}$. The filtering process applied to the $R_{Profile}$ includes the following sequential steps:

(1) To eliminate insignificant local maxima and mitigate noisy targets in the near field, especially when background responses are of high-magnitude, the $R_{Profile}$ is smoothed with a 1D Gaussian filter of standard deviation one and subject to dynamic thresholding. Dynamic thresholding assigns a value of zero to any range value below the maximum of 0.1 times the range profile value at the global maximum range bin in the frame under processing and a fixed threshold of 1×10^{-4}. The fixed threshold is selected based on extensive analysis of numerous recordings, which revealed that the noise level associated with a user's approach to the radar system is typically around this value.
(2) In cases where multiple local maxima are detected, precedence is given to the first local maximum identified. This particular bin ($R_{Gesture}$) is recognized as the point of interest within the $R_{Profile}$, signifying the presence of hand movement.
(3) If no local maxima are detected, the bin containing the global maximum is designated as $R_{Gesture}$ within the $R_{Profile}$.

The distinction between the bin containing the global maximum value and the bin associated with the first local maximum is crucial for effectively discerning the hand from the body, given that the hand typically exhibits closer proximity to the radar than the rest of the body.

Doppler profile processing

In order to obtain the Doppler information, a second FFT is applied exclusively to the range information (R) at only the final $R_{Gesture}$ derived from the range profile processing step. This procedure ensures that only one Doppler FFT [13] is performed per frame. Afterward, the absolute values of the mean along the antenna's dimensions are computed to generate the Doppler profile ($D_{Profile}$). The peak value of the signal amplitude in the $D_{Profile}$ is returned and designated as the $P_{Gesture}$ in the current frame under processing.

Enhancing gesture onset detection through frame filtering

After processing all frames, the "Range Profile Processing" section determines the $R_{Gesture}$ and the "Doppler Profile Processing" section generates the corresponding $P_{Gesture}$ as output. To identify the precise frame in which the gesture occurs, a threshold value of 5×10^{-5} is established. This threshold was selected after examining
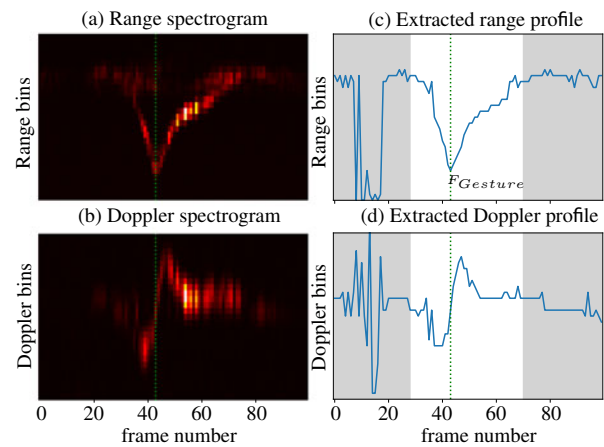

Figure 5. Illustration of gesture frame detection for a SwipeLeft gesture. (a) Conventional range spectrogram of a SwipeLeft gesture. (b) Conventional Doppler spectrogram. (c) and (d) $R_{Profile}$ and $D_{Profile}$ for all 100 frames within the SwipeLeft gesture, respectively. Frames in the gray areas are discarded due to thresholding during $F_{Gesture}$ estimation. These frames are not clearly visible in panels (a) and (b), confirming that they represent frames with gesture-accompanying noise rather than those with actual gesture execution. The green dotted line highlights the estimated $F_{Gesture}$, indicating the hand's closest position to the radar during the gesture. The results from panels (c) and (d) are corroborated by the conventional spectrograms in panels (a) and (b).

several gesture recordings and determining that a frame with a signal amplitude below this value is most likely devoid of any gesture. Consequently, any frame with a $P_{Gesture}$ value below this threshold is disregarded. Subsequently, the frame with the lowest range bin index among the remaining frames is identified, indicating its closest proximity to the radar. For ease of reference, this frame is denoted as the "Gesture Frame ($F_{Gesture}$)." In the context of the gestures, the $F_{Gesture}$ corresponds to the midpoint of Swipe gestures or approximately the endpoint of Push gestures. This distinction arises from the fact that in Swipe gestures, the hand is closest to the radar when it is in the middle of the gesture, whereas in Push gestures, the hand is nearly at the end of the gesture. As a result of the gesture frame detection process, the index of the $F_{Gesture}$ for each gesture is returned and stored for later utilization in the upcoming processing of the gesture data, as discussed in the "Refining Time-Domain Gesture Data" section.

Figure 5 shows the $R_{Profile}$ and $D_{Profile}$ of the full frames of a SwipeLeft gesture, emphasizing the accurate detection of the $F_{Gesture}$ where the hand was executing the gesture at its closest proximity to the radar.

Hand gesture recognition proposed system

Spiking neural network

In this work, the utilization of the SCNN proposed in [1] is adopted with certain modifications to streamline the model and enhance its computational efficiency. The architecture of the modified SCNN model is illustrated in Fig. 6. Specifically, the three convolutional layers maintain identical kernel sizes, padding, and stride, preserving their characteristics from the original SCNN model. Equally, the fully connected (FC) layer with an input size of 64 is retained. To maintain consistent model performance during both training and inference, batch normalization layers are omitted after each convolutional layer due to their different behaviors in these two processes. Likewise, in place of the synaptic spiking neurons layer,
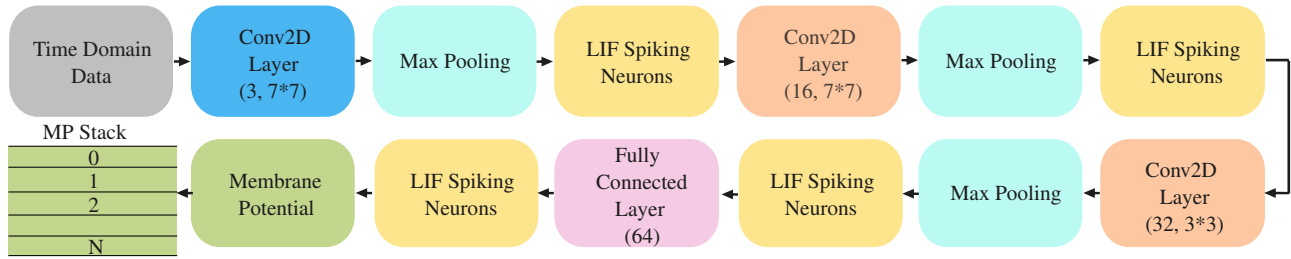
Figure 6. The architecture of the SCNN model is presented, with each convolutional layer annotated with its respective input channels and kernel size. The initial convolutional layer comprises three input channels, indicating that the network receives a single frame (composed of chirps or samples) from a single gesture at a time, with the input channels corresponding to the three antennas. Max-pooling layers with a stride of 2 are utilized. The first layer of LIF spiking neurons converts the output of the initial max-pooled convolutional layer into spike representations, which are then propagated to the subsequent layers of the network. The membrane potential of the final LIF spiking neurons layer is stacked for each frame, denoted by N, indicating the number of frames in the processed gesture.

a layer comprising LIF neurons [22] is introduced after the convolutional and max pooling layers. The LIF neurons exhibit simplified internal dynamics, thereby reducing the overall computational complexity of the model. Finally, considering the five gestures and Background class present in the current dataset, the output of the fully connected layer is constrained to six neurons, aligning with the class labels. The selection of a SCNN is motivated by its ability to process time-domain radar data directly. As illustrated in Fig. 6, the SCNN receives one frame of the recorded gesture at a time, preserving temporal information. Additionally, the input channels of the first convolutional layer are set to 3, corresponding to the three receiving antennas, ensuring the preservation of angular information. Convolution operations are performed on the (chirps, samples) information for each gesture, preserving (Doppler, range) information.

The LIF neuron, commonly used in computational neuroscience, resembles a resistor-capacitor (RC) circuit at its core. The membrane potential of the LIF neuron undergoes charging and discharging, similar to a capacitor in an RC circuit, as current flows through a resistor. This behavior is characterized by the potential increasing in response to incoming inputs and gradually decreasing in the absence of input due to a leaking mechanism. The LIF neuron, together with the SCNN, are implemented using the snnTorch spiking simulator [28]. As defined in [28], the inner dynamics of the LIF neuron is governed by:

$$U_t = \beta U_{t-1} + WI_{\text{in}}[t], \tag{5}$$

where each neuron possesses an intrinsic membrane potential (U_t) that evolves with incoming synaptic inputs ($I_{\text{in}}[t]$) modulated by the synaptic connection weight (W). The membrane potential integrates over time with a leakage term β, commonly represented by an exponential decay function. Similarly, as defined in [28]:

$$U_t > U_{\text{thr}} \Rightarrow S_t = 1, \tag{6}$$

when the membrane potential (U_t) surpasses a predetermined threshold (U_{thr}), the neuron generates an output spike (S_t), followed by a reset of its membrane potential by subtracting a threshold value. The LIF neuron's dynamic behavior allows for effective exploitation of the sequential nature of time-dependent datasets. Figure 7 provides an illustration depicting the operational principle of the LIF neuron.

To address the challenges posed by non-differentiable spikes during model training within the spiking domain and enable end-to-end optimization, the surrogate gradient descent method is employed [29]. This approach involves substituting the gradient
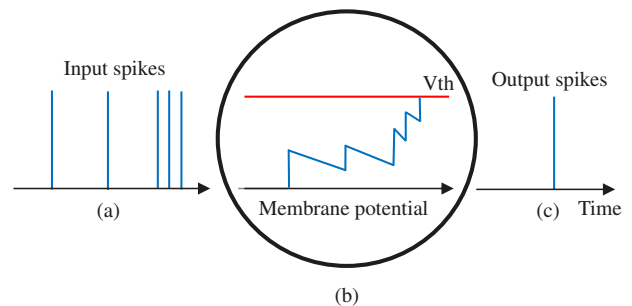

Figure 7. The LIF spiking neuron operating principle. (a) Spikes as inputs to the LIF neuron over time. (b) The membrane potential integrates over input spikes over time with a decay rate of beta. (c) An output spike is generated only when the membrane potential exceeds the spiking threshold (V_{th}).

of non-differentiable spikes in the backward pass with the gradient of a differentiable function, accordingly facilitating effective optimization of the SCNN model. In this study, we utilize the state-of-the-art surrogate gradient shifted arc-tan (ATan) function [30], representing an advancement over our previous work [1]. During the forward pass of training, spikes are generated in accordance with (5) and (6). Nevertheless, during the backward pass, the gradient of the spikes is substituted with the gradient of the ATan function as defined in [30]:

$$\frac{\delta S}{\delta U} = \frac{1}{\pi} \cdot \frac{1}{(1 + (\pi U \frac{\alpha}{2})^2)}, \tag{7}$$

where U denotes the membrane potential and α denotes the surrogate slope. It is worth noting, that the surrogate slope (α) plays a pivotal role in determining the steepness of the ATan function. Larger values result in a narrower surrogate gradient and a more pronounced transition from the function's minimum to the maximum value, while a smaller value means a wider surrogate gradient and a gradual transition from the minimum to the maximum value of the function.

Refining time-domain gesture data

As discussed in the "Gesture Dataset Acquisition" section, the dataset comprises a total of 19, 400 recorded gestures. All these gestures undergo the gesture frame detection process outlined in the "Gesture Frame Detection: A Key Step in Data Preprocessing for Training Enhancement" section. As a result of this process, the

$F_{Gesture}$ in which the gesture is clearly presented and in close proximity to the radar is detected. It is important to note that the $F_{Gesture}$ alone does not capture the entirety of the gesture but provides an indication that the gesture occurred in the vicinity. Windowing around this $F_{Gesture}$ is performed as the next step. To determine the optimal windowing size, various sizes are proposed and evaluated through comprehensive training and testing processes, as described in the "Training Process Overview" and "Testing Process Overview" sections. A thorough analysis of the results reveals that a window size of 8 around the $F_{Gesture}$ produces the most favorable outcomes. For the sake of simplicity, only the windowing size yielding the best performance is mentioned here and is subsequently used to generate the results in the "Experimental Results" section. Although, it is worth mentioning that the interpretation of the $F_{Gesture}$ differs for the Swipe gestures and the Push gesture. For Swipes, four frames before and after the $F_{Gesture}$ are labeled to accurately represent the gesture, while for the Push gesture, six frames before the $F_{Gesture}$ and two frames after are labeled to capture the correct execution of the gesture. Accordingly, the overall sequence of processing steps required to transform the ADC time-domain raw radar data into a suitable final time-domain format for model training and inference can be outlined as follows:

(1) Conventional Radar Preprocessing: The ADC time-domain raw radar data undergoes standard radar preprocessing steps, including the removal of the DC component and the (MTI) technique. This involves subtracting the mean along the fast-time and slow-time dimensions, respectively.
(2) Min-Max Normalization: Within each gesture, the 100 frames undergo a min-max normalization step. This normalization ensures that the values of the chirps and samples within each frame are scaled between zero and one.
(3) Gesture Labeling: The index of the $F_{Gesture}$ obtained from the gesture frame detection process for the current gesture is retrieved. Subsequently, windowing is applied to label eight frames out of the 100 frames as gesture frames, as described in the earlier discussion. The remaining 92 frames are labeled with the Background class label.
(4) Saving Processed Gesture Data: The processed data from step 2 is saved as the gesture data file, preserving the processed time-domain representation of the gestures.
(5) Saving Gesture Labels: The labeling of the frames from step 3 is saved as the gesture label file.

At the conclusion of this process, the entire dataset comprising 19,400 gestures is transformed into the processed time-domain format. Moreover, refinement of all gestures is performed to assign appropriate labels to the frames capturing the occurrences of the gestures, while simultaneously designating the remaining frames as Background. As a consequence, the dataset is fully prepared for utilization in the tasks of training, validation, and testing.

Time-domain processing approach

The frame-by-frame time-domain processing approach, introduced in [1], has undergone modifications to incorporate the newly introduced labeling structure. Unlike the previous implementation [1], which assigned a uniform label to all frames within a gesture, the current approach distinguishes between frames that exhibit active gesture performance and those that do not, as discussed in the "Refining Time-Domain Gesture Data" section. The modified processing approach can be summarized as follows:

(1) The processed time-domain data obtained from the "Refining Time-Domain Gesture Data" section consists of a sequence of 100 frames, representing a single gesture. These frames serve as the temporal input steps for the SCNN during its forward pass.
(2) The membrane potentials of the six LIF neurons in the final layer following the FC layer in the SCNN model are stacked for each frame, as illustrated in Fig. 6.
(3) The stacked membrane potentials of size $(100, 6)$ undergo processing through the LogSoftMax function [31], resulting in predicted log probabilities for each frame within the gesture. LogSoftMax is a common method for transforming unnormalized outputs into log probabilities, a prerequisite for the negative log-likelihood loss (NLLLoss) function [32].
(4) Utilizing the NLLLoss function, each frame's predicted log probability is compared with its corresponding label, derived from the saved label gesture in the "Refining Time-Domain Gesture Data" section. NLLLoss proves appropriate for this task as it operates on log probabilities produced by LogSoftMax. Moreover, combining LogSoftMax with NLLLoss is a standard practice in multi-class classification tasks, such as gesture recognition. Despite that, it is essential to emphasize that the overall loss for the entire gesture, which is vital for backpropagation and updating the trainable parameters of the network, is computed by summing the losses of each frame within the gesture.

Remarkably, unlike the previous work's approach [1], this procedure provides classification per frame, negating the need to aggregate predictions from all frames within a gesture to determine the performed gesture. For this reason, this approach demonstrates its suitability for real-time gesture detection where prediction is needed per frame received from the radar.

Data preparation and model evaluation

Dataset splitting

The dataset of 19,400 time-domain recordings, as defined in the 'Refining Time-Domain Gesture Data" section, was divided into training, validation, and testing subsets following standard practices. Specifically, 75% of the recordings (14,550) were allocated for training to ensure effective model training. The validation set, which is typically around 25% of the training set, was adjusted to approximately 22% of the training set, resulting in 3201 recordings to maintain a reasonable validation set size. The remaining 4850 recordings were reserved for independent testing, enabling a robust evaluation of the model's performance. This division aligns with established machine learning practices, taking into account our dataset's size and ensuring robust model evaluation.

Training process overview

The training process of the SCNN model involves utilizing the training dataset and utilizing the modified time-domain processing approach detailed in the "Time-Domain Processing Approach" section. During training, the adaptive moment estimation (Adam) optimization algorithm is employed to update the model's trainable parameters [33]. Hyperparameter tuning is performed to optimize the training performance of the model before full training. This is done using the Optuna library [34] to fine-tune various

Table 2. Optuna hyperparameter search range

Hyperparameter	Optimization range
Adam learning rate	1×10^{-4} to 1×10^{-3}
Adam weight decay	1×10^{-6} to 1×10^{-3}
LIF neuron decay rate (Beta)	0.5–1.0
LIF neuron threshold	0.2–0.7
Surrogate gradient slope	0–10

Table 3. Hyperparameters overview

General hyperparameters	
Batch size	32
Epochs	120
Adam learning rate	4.9×10^{-4}
Adam weight decay	1.27×10^{-5}
SCNN specific hyperparameters	
Surrogate gradient slope	4.0
Beta	0.5
LIF neuron threshold	0.25

hyperparameters associated with the model and the spiking LIF neuron.

An Optuna-based hyperparameter optimization environment is set up, where a random value is selected for the Adam optimizer learning rate, weight decay, spiking LIF neuron decay rate, threshold, and surrogate gradient slope for each optimization trial. Table 2 shows the hyperparameter search range, selected based on best-known common practices. The SCNN is then trained for only 20 epochs using the suggested values, and the validation accuracy is calculated for each trial. Through a rigorous hyperparameter tuning process involving more than 200 optimization trials, the optimal values for the learning rate and weight decay of the Adam optimizer are determined to be 4.9×10^{-4} and 1.27×10^{-5}, respectively. Similarly, the beta decay rates for the membrane potential, threshold, and surrogate gradient slope of the LIF neuron are designated as 0.5, 0.25, and 4.0, respectively. It is worth noting that these values resulted in the highest validation accuracy, thus ensuring that the selected values maximize model performance.

Table 3 provides a comprehensive overview of the optimized and non-optimized hyperparameter values utilized in the training process.

Following hyperparameter optimization, the SCNN model undergoes a 120-epoch training phase. To address the common challenge of overfitting and enhance the model's ability to generalize to unseen data, the early-stopping approach [35] is employed. This method is a widely used regularization technique in machine learning, primarily employed to prevent overfitting. It operates by continuously monitoring the model's performance on the separate validation dataset during training. If no noticeable improvement in the model's performance on the validation data is observed over a consecutive span of 10 epochs, the training process is halted. In essence, early stopping prevents the model from becoming overly specialized to the training data. This, in turn, improves the model's capability to generalize to new, unseen data, ultimately enhancing its overall performance.

Testing process overview

Throughout the training process, attention is given to identifying and preserving the model with the lowest validation loss, referred to as the "best model." In the following testing phase, this best model is retrieved and subjected to testing using the independent and previously unseen test dataset. The testing phase is critical in detecting potential biases and reliably assessing the model's generalization capabilities since it leverages the best model, which demonstrates reduced sensitivity to overfitting. The model selection process serves as an additional safeguard against overfitting, ensuring that the model's performance is carefully evaluated based on its true potential.

Experimental results

In this section, we describe the experimental results of the study. To account for the inherent randomness in the optimization process of neural networks, multiple training iterations were conducted for the SCNN model, as described in the "Training Process Overview" section. In particular, the training procedure was executed five times, each time employing a new random seed for initialization. This approach enabled the acquisition and retention of the five top-best models, as outlined in the "Testing Process Overview" section. By averaging the performance of these five models, the results presented in this section provide a more reliable and representative assessment of the model's performance compared to a single model.

Effectiveness of the time-domain frame-based prediction approach

This section compares the proposed frame-based prediction approach with the previously introduced record-based prediction approach in [1]. Unlike the frame-based approach, the record-based approach cannot distinguish between frames that contain the gesture and non-gesture frames, resulting in assigning the same label to all gesture frames. Consequently, the network performs a single prediction for the entire 100-frame gesture, aiming to detect the presence of the gesture within this duration, rather than making predictions on individual frames. To obtain experimental results, the modified SCNN architecture introduced in this study is trained on the newly introduced dataset using the record-based prediction approach for five separate training runs with different seeds. The mean test accuracy, derived from the best (lowest validation loss) among the five SCNN models, is observed to be 95.40%.

In the newly proposed frame-based prediction solution, during the training process, the SCNN model leverages the time-domain processing approach described in the "Time-Domain Processing Approach" section, enabling it to generate predictions for each frame individually. Likewise, in the testing phase, the accuracy function assesses the model's performance by evaluating the accuracy of its frame-level predictions for each gesture. The SCNN model is trained on five different random seeds using the frame-based prediction approach. Thereupon, the mean test accuracy is determined by evaluating the performance of the five best models, resulting in an accuracy of 98.45%.

Table 4 confirms that the frame-based prediction approach outperforms the record-based prediction approach in terms of mean test accuracy. Also noteworthy is that the models employed in both approaches were trained on the same dataset and evaluated using

Table 4. Comparison of time-domain frame-based and record-based approaches

	Frame-based prediction	Record-based prediction
Mean test accuracy	98.45%	95.40%

the same criteria, highlighting the usefulness of the frame-based approach for reliably predicting gestures.

Further analysis

The precision and recall scores obtained from testing the five SCNN models are examined to gain deeper insights into the frame-based prediction approach. The average precision, calculated as 0.90, highlights the model's ability to accurately identify positive instances. Likewise, the average recall, calculated as 0.89, indicates the model's effectiveness in capturing relevant information and minimizing false negatives. These metrics serve as valuable indicators of the model's overall performance and their ability to strike a balance between precision and recall in classification tasks [36].

Table 5 presents the consolidated confusion matrix, providing a comprehensive overview of the model's classification outcomes across the different gestures. This matrix aids in identifying specific areas of improvement and understanding the patterns of misclassifications exhibited by the models. It is noteworthy that the values observed in Table 5 provide insights into the frame distribution within the test data, considering the current frame-based prediction approach. Furthermore, there is a noticeable difference in the number of frames predicted as Background compared to the other classes. This distinction arises from the labeling scheme, where each gesture comprises eight frames explicitly designated as gesture frames, while the remaining 92 frames are assigned the background label, as already noted in the "Gesture Frame Detection: A Key Step in Data Preprocessing for Training Enhancement" and "Refining Time-Domain Gesture Data" sections.

Modified evaluation protocol

Upon closer examination of the model's outcomes and the evaluation of their respective confusion matrix, it becomes apparent that the models consistently demonstrate proficiency in identifying frames in which gestures are performed correctly. Even so, occasional temporal deviations exist, where the models predict the occurrence of a gesture slightly before or after the labeled frames. Considering that each frame spans 30 ms, such marginal deviations in temporal detection do not have a significant impact on

real-world practical applications. Figure 8 provides a clear depiction of this behavior. In the investigation, a sequence of three gestures is provided as input to one of the SCNN networks, revealing the infrequent occurrence of incorrect predictions. Nonetheless, it is possible to observe a subtle temporal shift in the predictions. In light of this observation, a decision is made to refine the accuracy measurement approach to account for this characteristic behavior. The refined accuracy comprises the following steps:

(1) Initially, the predicted classification labels for each set of 100 frames within a gesture are examined. It is essential to ensure that among these 100 frames, only one non-zero label is present. This criterion guarantees accurate prediction of the Background frames and ensures that each gesture is uniquely identified without any overlapping predictions with other gestures in the same frame sequence.

(2) Subsequently, the correspondence between the ground truth non-zero labels and the non-zero predicted values within these 100 frames is validated. This process confirms that the predicted frames accurately capture the presence of the same gesture as indicated by the ground truth labels.

(3) To still refine the accuracy, a windowing technique is employed. A window of size 10, centered around the frames containing the 8 non-zero ground truth labels, is utilized. This window encompasses 5 frames preceding and 5 frames succeeding the non-zero labels.

(4) Afterward, analysis is conducted to determine if the non-zero predicted frames, signifying the predicted gesture, fall within the designated window. On top of that, a condition is set that the predicted non-zero frames must exhibit a repetition of at least 5 frames compared to the 8 non-zero ground truth frames. Meeting these requirements indicates that the network correctly predicted the gesture, although with a minor temporal variation. Therefore, these predictions are deemed accurate in their entirety.

By incorporating these refined accuracy measurement procedures, the precision and recall values derived from the five SCNN models reach more reasonable levels, with precision achieving 0.980 and recall attaining 0.975. The SCNN models demonstrate an average testing accuracy of 99.75%. Table 6 presents a concise summary of the average confusion matrix obtained from the five SCNN models, offering a comprehensive evaluation of their collective performance. To further validate the effectiveness of the proposed time-domain frame-based solution, a comparison is made to a recent work [27] utilizing an augmented version of the gesture dataset employed in this work. The authors applied a slim radar conventional preprocessing pipeline to the time-domain data, extracting five features: radial distance, radial velocity,

Table 5. Average confusion matrix

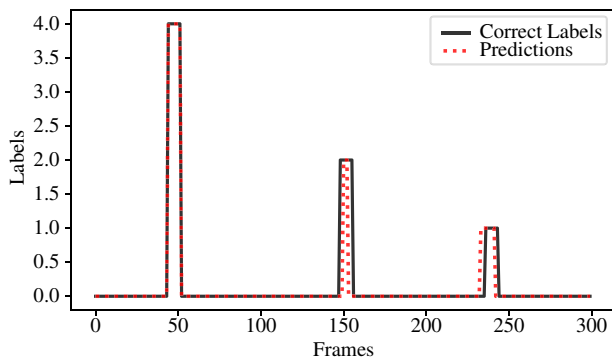|	Background	Push	SwipeRight	SwipeLeft	SwipeDown	SwipeUp
Background	**447,949**	868	595	670	654	606
Push	928	**5356**	26	16	7	2
SwipeRight	666	44	**5492**	69	32	31
SwipeLeft	838	19	76	**5383**	12	30
SwipeDown	546	9	19	16	**5780**	0
SwipeUp	634	2	24	23	0	**5732**

Figure 8. Prediction results out of a SCNN model for three gestures (300 frames). The results exhibit a high level of agreement between the predicted labels and ground truth. Minute shifts between predictions and labels are noticed in certain cases, thus explaining the model's behavior.

azimuth angles, elevation angles, and signal amplitude per frame. These five features are then fed into an RNN to generate a prediction per frame. An average F1 score of 98.4% is reported. Given the average precision of 0.980 and an average recall of 0.975 achieved by the proposed SCNN model, the average F1 score is 97.7%. This demonstrates that the proposed solution is comparable to solutions based on conventional radar preprocessing pipelines and conventional ANN-based networks with strong temporal learning capabilities.

Live testing

Live testing plays a critical role in evaluating trained models in real-time scenarios, ensuring their consistency with the results obtained during the testing phase. Real-time retrieval of frames from the radar is achieved using a dedicated Python script. The acquired ADC time-domain frames undergo the time-domain refinement process described in the "Refining Time-Domain Gesture Data" section. The processed time-domain frame is then fed into the SCNN, where predictions for each gesture class are generated based on the membrane potential values of the last layer of LIF spiking neurons, as explained in the "Time-Domain Processing Approach" section. The gesture class with the highest predicted probability is displayed in real-time through the Python script, providing immediate feedback on the detected gesture. During live testing, the radar's interactive control capabilities are demonstrated by incorporating corresponding movement actions based on the predicted gestures. If no gesture or a Background class is detected, the system remains responsive without triggering any action. Gestures, including SwipeDown, SwipeUp, SwipeLeft, and SwipeRight, are mapped to keyboard actions that simulate pressing

the "Down," "Up," "Left," and "Right" keys, respectively. Likewise, the Push gesture corresponds to pressing the "PageUp" key. These interactive controls exemplify the practicality of the radar system, enabling remote control in various scenarios. For instance, it allows seamless control over presentation slides and facilitates engagement in online gaming solely through radar-predicted gestures.

The effectiveness of the proposed real-time solution is validated by live testing from five individuals (three males and two females, aged 25–30 years) uninvolved in the dataset collection process. These individuals performed a set of 20 gestures in a randomized sequence, using both the right and left hands at the same distances and angles used in the gesture recording, as detailed in the "Gesture Dataset Acquisition" section. The solution accurately predicted 96 of the 100 performed gestures, confirming the accuracy, reliability, and suitability of the trained SCNN models for real-time gesture recognition.

Computational complexity

The SCNN model, with a top-level metric of 1.4 million multiply-accumulate (MMAC) operations per frame inference, assumes that LIF spiking neurons display full spiking activity, with all spikes equal to one. The central processing unit (CPU), by default, does not account for single-bit event spikes, treating their multiplication with the single-bit event as a standard multiplication, similar to the ANN. However, this operation should be less costly as it is a multiplication between the synaptic weight and a single-bit event (spike). Therefore the SNN in the worst-case scenario still demonstrates computational efficiency compared to an equivalent ANN. Additionally, since SNNs generate spikes only when they exceed the internal threshold (as discussed in the "Spiking Neural Network" section), the resulting spikes are primarily zeroes with a few occurrences of 1-bit spikes. This eliminates the need for multiplication between synaptic weights and zero spikes, simplifying the MAC operation to just an addition. And, when the spike is a 1-bit, it results in a simpler multiplication than the conventional case, underscoring the computational efficiency and potential of the SCNN model. In the context of hardware implementation, SNNs offer a distinct advantage over conventional ANNs. By leveraging 1-bit spikes, SNNs streamline the computational process, replacing the conventional multiplier with a simpler logical AND operation. This efficient approach reduces the computational complexity associated with synaptic weight multiplication, making SNNs a promising solution for hardware implementations, particularly on neuromorphic hardware where the inherent sparsity of SNNs can be fully leveraged [37].

The sparsity of SCNN models is evaluated via live testing, as detailed in the "Live Testing" section. In this approach, one of the

Table 6. Average refined confusion matrix

	Background	Push	SwipeRight	SwipeLeft	SwipeDown	SwipeUp
Background	451,062	81	62	79	23	32
Push	107	**6208**	26	16	6	3
SwipeRight	142	43	**6056**	70	32	31
SwipeLeft	123	19	64	**6090**	12	18
SwipeDown	34	9	21	18	**6290**	0
SwipeUp	76	2	25	24	0	**6241**

SCNN models is executed in real-time, enabling the monitoring of the number of non-zero spikes and zero spikes produced during each frame prediction. Through the calculation of the ratio between zero spikes and the total number of spikes, the sparsity level for each frame prediction is determined. The investigations continually demonstrate a pattern: the sparsity level is consistently high during non-gesture times, ranging from 90% to 97%. Conversely, when a gesture is detected, the sparsity level slightly decreases to around 75%. The consistent generation of sparse predictions by the SCNN model reaffirms the inherent advantage of SNNs over ANNs.

Discussion

In contrast to the SNN works in [13, 16, 19], our approach focuses solely on utilizing time-domain data, bypassing the computationally expensive FFT preprocessing steps. Our approach also does not require an encoding step to convert input data into the spiking format. Unlike previous SNN-based gesture recognition solutions [1, 13, 16, 19–21], our solution introduces a gesture frame detection process outlined in the "Gesture Frame Detection: A Key Step in Data Preprocessing for Training Enhancement" section. This process enables us to accurately differentiate between frames that capture genuine gesture execution and those characterized by the absence of gestures. This distinction allows for a modified form of gesture labeling, assigning labels to individual frames and enabling frame-based predictions by the model. By incorporating these modifications during model training, the model becomes adept at extracting crucial features from accurately executed gestures, resulting in improved accuracy in recognizing and classifying gestures.

Despite the challenges posed by the dataset's recording characteristics, the combination of the modified time-domain processing approach and the proposed SCNN architecture results in successful classification among the five different gestures.

The comparative analysis between the proposed frame-based prediction approach and our previous approach [1] discussed in the "Effectiveness of the Time-Domain Frame-Based Prediction Approach" section demonstrates a higher mean test accuracy, as presented in Table 4. Additionally, the frame-based approach exhibits enhanced suitability for real-time testing and implementation.

The evaluation of the proposed solution in this work demonstrates its effectiveness through the average precision-recall measurements and average testing accuracies, as discussed in the "Further Analysis" and "Modified Evaluation Protocol" sections.

The average confusion matrix in Table 5 reveals noteworthy findings. It indicates that the majority of misclassifications occur between ground-truth gesture frames and the Background class, without any specific pattern of misclassification among pairs of gestures. To further investigate why this is the case, we demonstrate the temporal aspect of the model's predictions in Fig. 8, revealing slight variations in timing when compared to the ground truth labels. However, given that each frame corresponds to 30 ms, these temporal differences have minimal practical significance. Based on this observation, a refined accuracy evaluation approach is proposed, focusing on classifying gestures within a temporal window. The refined confusion matrix in Table 6, along with the precision and recall values in the "Modified Evaluation Protocol" section, confirm the correct classification of gestures and a significant reduction in misclassifications associated with the Background class. These findings validate the effectiveness of the enhanced evaluation methodology in accurately assessing the model's performance in gesture recognition tasks.

Our proposed solution distinguishes itself from previous works on SNNs [1, 13, 16, 19–21] by showcasing robust classification capabilities on a frame-by-frame basis, making it highly suitable for real-time implementation. In contrast to the approach presented in [20], where it is mentioned that their SNN network requires multiple time steps to accurately predict a gesture during testing, which hinders real-time implementation, our solution generates predictions for each frame, ensuring real-time performance. Unlike [27], which employs an augmented version of the dataset used in this study and provides real-time classification from an RNN per frame, the proposed solution directly utilizes just time-domain data, bypassing any radar preprocessing pipeline and achieving on-par accuracy while taking advantage of the sparsity of the SCNN. The effectiveness of our solution is reinforced by comprehensive live testing of the radar system involving five individuals who were not involved in the dataset acquisition, as described in the "Live Testing" section. Results obtained from this testing further support the effectiveness of our proposed approach and confirm the evaluation results. Additionally, the radar system empowers users with versatile functionalities, such as slide switching in presentations and enjoyable participation in online games like "Play Snake" [38]. These practical applications serve as tangible demonstrations of the solution's versatility and practicality in real-world scenarios.

The "Computational Complexity" section discusses the computational complexity per frame inference demonstrated by the SCNN model. Leveraging SNNs provides inherent sparsity and improved computational efficiency compared to ANNs. To assess this behavior, the SCNN model undergoes live testing to measure sparsity per inference. Notably, during periods of no gesture execution, the SCNN exhibits an average sparsity rate ranging from 90% to 97%, which decreases to approximately 75% during gesture execution. As a result, the sparsity characteristic of the SCNN significantly reduces the top-level computations per frame inference, resulting in a notable decrease in the 1.4 MMAC.

Conclusion

This article presented a modified approach for gesture recognition in time-domain radar data processing. This approach eliminates the computationally expensive preprocessing FFT steps and introduces a novel gesture frame detection process for model training. This process accurately identifies frames containing gestures and distinguishes them from frames with Background noise. By focusing on frames where gestures occur, the model can effectively learn and extract relevant features. The proposed solution, in conjunction with a lightweight SCNN, achieves significant results in gesture recognition for five different gestures. The frame-based prediction capability of our solution enables real-time testing, allowing users to interact, for example, with PowerPoint presentation slides using radar gestures. Moreover, an evaluation of the sparsity of the SCNN in real-time demonstrates the advantages of our SNN-based approach over conventional ANNs. Overall, the modified time-domain processing approach, combined with the lightweight SCNN, yields notable gesture recognition outcomes. This combination provides practical benefits in real-time applications and motivates further promising implementation on neuromorphic hardware to fully harness the SCNN's sparsity.

References

1. **Shaaban A, Furtner W, Weigel R and Lurz F** (2022) Spiking neural networks for gesture recognition using time domain radar data. In *2022 19th European Radar Conference (EuRAD)*. Milan, Italy, 33–36.

2. **Tran D-S, Ho N-H, Yang H-J, Baek E-T, Kim S-H and Lee G** (2020) Real-time hand gesture spotting and recognition using RGB-D camera and 3D convolutional neural network. *Applied Sciences* **10**(2), 722.

3. **Rogez G, Supancic JS and Ramanan D** (2015) Understanding everyday hands in action from RGB-D images. In *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, 3889–3897.

4. **Ramalingame R** (2021) Wearable smart band for American sign language recognition with polymer carbon nanocomposite-based pressure sensors. *IEEE Sensors Letters* **5**(6), 1–4.

5. **Fan T** (2016) Wireless hand gesture recognition based on continuous-wave doppler radar sensors. *IEEE Transactions on Microwave Theory and Techniques* **64**(11), 4012–4020.

6. **Ahuja K, Jiang Y, Goel M and Harrison C** (2021) Vid2Doppler: synthesizing doppler radar data from videos for training privacy-preserving activity recognition. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan.

7. **Scherer M, Magno M, Erb J, Mayer P, Eggimann M and Benini L** (2021) Tinyradarnn: combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars. *IEEE Internet of Things Journal* **8**(13), 10336–10346.

8. **Choi J-W, Ryu S-J and Kim J-H** (2019) Short-range radar based real-time hand gesture recognition using LSTM encoder. *IEEE Access* **7**, 33610–33618.

9. **Wu Q and Zhao D** (2018) Dynamic hand gesture recognition using FMCW radar sensor for driving assistance. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. Hangzhou, China, IEEE, 1–6.

10. **Yang Z and Zheng X** (2021) Hand gesture recognition based on trajectories features and computation-efficient reused LSTM network. *IEEE Sensors Journal* **21**(15), 16945–16960.

11. **Sze V, Chen Y-H, Yang T-J and Emer JS** (2017) Efficient processing of deep neural networks: a tutorial and survey. *Proceedings of the IEEE* **105**(12), 2295–2329.

12. **Maass W** (1997) Networks of spiking neurons: the third generation of neural network models. *Neural Networks* **10**(9), 1659–1671.

13. **Tsang IJ, Corradi F, Sifalakis M, Van Leekwijck W and Latré S** (2021) Radar-based hand gesture recognition using spiking neural networks. *Electronics* **10**(12), 1405.

14. **Wang S, Song J, Lien J, Poupyrev I and Hilliges O** (2016) Interacting with soli: exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. Tokyo, Japan, 851–860.

15. **Ritchie M, Capraru R and Fioranelli F** (2020) Dop-NET: a micro-Doppler radar data challenge. *Electronics Letters* **56**(11), 568–570.

16. **Safa A, Bourdoux A, Ocket I, Catthoor F and Gielen GG** (2021) On the use of spiking neural networks for ultralow-power radar gesture recognition. *IEEE Microwave and Wireless Components Letters* **32**(3), 222–225.

17. **Stuijt J, Sifalakis M, Yousefzadeh A and Corradi F** (2021) μbrain: an event-driven and fully synthesizable architecture for spiking neural networks. *Frontiers in Neuroscience* **15**.

18. **Rueckauer B and Liu S-C** (2018) Conversion of analog to spiking neural networks using sparse temporal coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence, Italy, 1–5.

19. **Gerhards P, Kreutz F, Knobloch K and Mayr CG** (2022) Radar-based gesture recognition with spiking neural networks. In *2022 7th International Conference on Frontiers of Signal Processing (ICFSP)*. Paris, France, IEEE, 40–44.

20. **Arsalan M, Santra A and Issakov V** (2022) Spiking neural network-based radar gesture recognition system using raw ADC data. *IEEE Sensors Letters* **6**(6), 1–4.

21. **Shaaban A, Furtner W, Weigel R and Lurz F** (2022) Evaluation of spiking neural networks for time domain-based radar hand gesture recognition. In *2022 23rd International Radar Symposium (IRS)*. Gdansk, Poland, 474–479.

22. **Gerstner W and Kistler WM** (2002) *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press.

23. **Infineon Technologies**. BGT60TR13C 60GHz radar sensor for advanced sensing. https://www.infineon.com/cms/en/product/sensor/radar-sensors/radar-sensors-for-iot/60ghz-radar/bgt60tr13c/.

24. **Trotta S, Weber D, Jungmaier RW, Baheti A, Lien J, Noppeney D, Tabesh M, Rumpler C, Aichner M, Albel C and Bal JS** (2021) 2.3 SOLI: a tiny device for a new human machine interface. In *2021 IEEE International Solid-State Circuits Conference (ISSCC)*. San Francisco, CA, USA, 42–44.

25. **Jankiraman M** (2018) *FMCW Radar Design*. Artech House.

26. **Gurbuz SZ** (2020) *Deep Neural Network Design for Radar Applications*. Institution of Engineering and Technology.

27. **Strobel M, Schoenfeldt S and Daugalas J** (2023) Gesture recognition for FMCW radar on the edge. arXiv, Oct. 13.

28. **Eshraghian JK** (2023) Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE* **111**(9), 1016–1054.

29. **Neftci EO, Mostafa H and Zenke F** (2019) Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine* **36**(6), 51–63.

30. **Fang W, Yu Z, Chen Y, Masquelier T, Huang T and Tian Y** (2021) Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Montreal, QC, Canada, 2661–2671.

31. **Bridle J** (1989) Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *In Proceedings of the 2nd International Conference on Neural Information Processing Systems (NIPS'89), MIT Press, Cambridge, MA, USA,* 211–217.

32. **Yao H, Zhu D, Jiang B and Yu P** (2020) Negative log likelihood ratio loss for deep neural network classification. In Arai K, Bhatia R, Kapoor S, (eds), *Proceedings of the Future Technologies Conference (FTC) 2019 in Advances in Intelligent Systems and Computing*, Springer International Publishing, Cham, 276–282.

33. **Kingma D and Ba J** (2014) Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*. San Diego, CA, USA.

34. **Akiba T, Sano S, Yanase T, Ohta T and Koyama M** (2019) Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD'19*. Association for Computing Machinery, New York, NY, 2623–2631

35. **Prechelt L** 1998 *Early stopping – but when?* In Orr GB and Müller K-R (eds)), *Lecture Notes in Computer Science.*. Berlin, Heidelberg: Springer, pp.55–69.

36. **Buckland M and Gey F** (1994) The relationship between recall and precision. *Journal of the American Society for Information Science* **45**(1), 12–19.

37. **Schuman CD, Potok TE, Patton RM, Birdwell JD, Dean ME, Rose GS and Plank JS Schuman CD** (2017) A Survey of Neuromorphic Computing and Neural Networks in Hardware. arXiv, May 19.

38. **Interactive N**, *Snake – Play the Retro Snake Game Online for Free*, Snake, https://playsnake.org/.

Ahmed Shaaban received the B.Sc. degree in Electronics and Communication Systems Engineering in 2017 from Ain Shams University (ASU) in Cairo, Egypt, and the M.Sc. degree in Computational Engineering in 2020 from Friedrich-Alexander University Erlangen-Nuremberg (FAU) in Erlangen, Germany. In 2021, he joined Infineon Technologies AG, where he works at the intersection of machine learning and radar systems. His research interests include using spiking neural networks (SNNs) for mm-Wave radar applications.

Maximilian Strobel received the B.Eng. degree in Electrical Engineering and Information Technology from the Munich University of Applied Sciences, Germany, in 2017 and the M.Sc. degree in Robotics, Cognition, and Intelligence from the Technical University of Munich, Germany, in 2019. In 2019, he joined Infineon Technologies AG as a System Architect in the field of Machine Learning for smart sensor applications with a focus on edge computing. He is currently involved in the design and implementation of AI/ML algorithms for mm-Wave and vision sensors.

Wolfgang Furtner is a Distinguished Engineer for SoC Architecture at Infineon Technologies AG. He received his degree in Electrical Engineering from the University of Applied Sciences in Munich, Germany. He started his career working 4 years in a startup developing Graphics Processors (GPUs), followed by 11 years architecting Graphics and Video Processing ICs at Philips Semiconductors. Since 2006 he is with Infineon and heading System Concept Engineering for power and sensors. His interests are Embedded Architectures for Artificial Intelligence and Machine Learning, Smart Sensors, and System Architectures for Quantum Computing.

Robert Weigel is the director of the Institute for Electronics Engineering at FAU, Erlangen, Germany. He has co-founded several companies like DICE in Linz, Austria (meanwhile split into Infineon Technologies and Apple with over 400 staff members) or EESY-IC in Erlangen (now a Bosch company with over 60 staff members). Dr Weigel has been engaged in the design of circuits and systems. He received several awards like the IEEE Microwave Application Award, the IEEE Distinguished Microwave Educator Award, the IEEE Microwave Career Award, and the Cross of Merit First Class of Order of Merit of Germany. He is a Life Fellow of the IEEE and a Fellow of the German ITG, an Elected Member of the German National Academy of Science and Engineering. He served in many roles for IEEE and EuMA. He was General Chair of the 2013 European Microwave Week in Nuremberg, Germany, and the 2014 IEEE MTT-S President.

Fabian Lurz received the B.Sc. and M.Sc. degrees in information and communication technology and the Dr.-Ing. degree from the Friedrich-Alexander University (FAU), Erlangen, Germany, in 2010, 2013, and 2019, respectively. In 2013, he joined the Institute of Electronics Engineering, FAU, as a Research Assistant, and from 2017 to 2020, he was a Research Group Leader of the Circuits, Systems and Hardware Test Group. In June 2020, he joined the Institute of High-Frequency Technology at the Hamburg University of Technology, Hamburg, Germany, as a senior engineer and research group leader. Since October 2023, he has been a full professor and head of the Chair of Integrated Electronic Systems at the Otto-von-Guericke University Magdeburg, Magdeburg, Germany. He is a member of the IEEE Microwave Theory and Techniques Society (IEEE MTT-S) and the IEEE Instrumentation and Measurement Society (IEEE IMS). He was a recipient of the First Prize in the High Sensitivity Radar Student Design Competition of the IEEE International Microwave Symposium in 2014, 2017, and 2018, respectively, and the IEEE Microwave Theory and Techniques Society Graduate Fellowship Award in 2016.