Hand Gesture Recognition in Range-Doppler Images Using Binary Activated Spiking Neural Networks

Daniel Auge[1], Julian Hille[1,2], Etienne Mueller[1], Alois Knoll[1]
[1] Department of Informatics, Technical University of Munich, Munich, Germany
[2] Infineon Technologies AG, Munich, Germany

2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021) | 978-1-6654-3176-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/FG52635.2021.9666988

Abstract— **Many hand gesture recognition systems use radar to sense the motion of the hand due to its independence of lighting and its inherent privacy. As in the case of cameras, complex signal processing chains consisting of classical algorithms and neural network-base approaches are necessary to evaluate the incoming data stream. Especially on mobile devices, the reduction of the total energy consumption of the recognition system is crucial as it would lead to an increased battery life. Spiking neural networks have been shown to consume much less energy than current networks by operating event-driven and using time as the main information carrier. However, practical applications in which they are on par with classical approaches are rare. In this paper we utilize spiking neural networks to perform hand gesture recognition in radar data. We show that the temporal affinity of spiking networks and the possibility to binarize the radar-generated range-Doppler images without large loss of information introduces a promising synergy. Using simple networks consisting of 75 recurrently connected spiking neurons, we are able to reach current state-of-the-art performance on two public datasets. With this approach, gesture recognition systems can operate much more energy-efficient, making spiking neural networks viable alternatives to current solutions.**

I. Introduction

In many fields, human-computer interaction evolves towards dynamic interactions which are more intuitive for humans. One possible natural method for humans are hand gestures, which are sensed and recognized by machines. The applications for this type of interaction are various: smart home, car entertainment, mobile phones, robot control, or even interactive display panels in smart cities.

The gestures are commonly detected using cameras [19], [28], however, multiple other approaches have been proposed. Human attached sensors, such as an armband to detect muscle activity, showed successful classification of 6 different classes [4]. Gesture recognition can also be extended to full body movements as shown by [6], where a neural network (NN) classifies the micro-Doppler signatures of sonar sensors.

Radar-based gesture recognition provides many advantages compared to other solutions like the independence of lighting, atmospheric conditions, and the inherent privacy because it neglects the object's optical properties. Radar systems can detect range, velocity, and angle of arrival of nearby targets independent of environmental conditions. Therefore, a variety of classification algorithms based on NNs have been proposed [1], [9], [13], [21], [25], [27].

For all applications, a high classification accuracy as well as a low energy consumption are of major importance. Especially for mobile devices, a low energy consumption of the sensor itself and the attached signal processing is crucial for a long battery life.

Taking the next step in energy-efficiency of neural network-based processing and following recent research of biological inspired NNs, spiking neural networks (SNNs) gained traction in the research community. SNNs communicate via short all-or-nothing pulses and leverage time as the main information carrier. Compared to today's standard neural networks based on continuous-valued activation functions, SNNs have shown superior computational power [14].

The potential energy efficiency of SNNs can be reached particularly on specialized hardware, called neuromorphic hardware. Through optimized communication paths and compute units or analog implementations, neuromorphic solutions can be 100 times more efficient than comparable solutions [5], [18]. Reconfigurable neuromorphic multi-purpose chips like TrueNorth [16], Loihi [8], or SpiNNaker [12], [15] are constantly being improved for future applications.

In this work we introduce the utilization of SNNs for radar-based hand gesture recognition. The event-driven and energy-efficient nature of SNNs make them a perfect fit for the analysis of sequence data, especially in mobile environments. Radar data is particularly suited as input since we assume that only the nearest recognized objects are relevant for the recognition, which makes a binarization of the input feasible. We therefore propose a network architecture comprising spiking neurons for radar-based gesture recognition. We additionally analyze different input encoding approaches, which introduce the trade-off between accuracy and network activity. We evaluate our architecture on two public datasets [21], [25] which are based on the same set of 11 gestures performed in front of a radar sensor. There, our approach shows superior classification performance while additionally being potentially much more energy efficient. The code of our simulations is available on GitHub*.

We thank Infineon Technologies AG for supporting this research.

978-1-6654-3176-7/21/$31.00 ©2021 IEEE

*`https://github.com/GustavEye/spiking-radar-gestures`

II. Background

A. Radar-Based Gesture Sensing

Frequency modulated continuous wave (FMCW) radars emit continuous electromagnetic waves which are reflected by nearby objects [26]. The reflected signal contains information about the distance and relative velocity between the sensor and the targets. During a measurement, the frequency of the emitted signal is swept with a very large slope. Accordingly, the frequencies of the transmitted and received signal are different. Since the Doppler effect is negligible at these high rates, the frequency difference is directly proportional to the time of flight of the signal and with that to the range of the target. Multiple of these measurements, chirps, can be used to obtain the velocity of the target by evaluating the phase change of the respective complex range components. The resulting two-dimensional matrix is called Range-Doppler map (RDM) which contains the received signal strength at each combination of ranges and velocities.

If the used radar system has multiple receive channels, multiple RDMs are generated and can be used to additionally determine the angle of the perceived targets. This is enabled by the physical positioning of different antennas. The positions result in phase differences of the received signals due to slight variations – in the order of fractions to multiple wavelengths – in the distance to the targets.

B. Spiking Neural Networks

The leaky integrate-and-fire (LIF) neuron model [10] used in this work, accumulates the weighted inputs in a stateful hidden variable. When the hidden variable reaches the firing threshold, the binary activation function is activated and the hidden state variable is reset. An exponential decay (leakage) of the hidden variable over time adds the fading of information which is not frequently updated. Despite the biological evidence, this simple behavior results in several technical characteristics:

- the computationally cheaper summation of the inputs instead of multiply-accumulate (MAC) operations due to the binary activation function [7],
- event-based processing, since neurons only need to be updated if non-zero inputs are present [11], and
- pattern detection in data sequences due to the inherent memorization property of the hidden state [10].

Learning algorithms for spiking networks, however, have not shown comparable performance to backpropagation in artificial neural networks (ANNs) for a long time [23]. A promising approach has been to train ANNs with backpropagation and use the weights in spiking networks resulting in a conversion of ANNs to SNNs [20]. Recent developments introduced pseudo gradients to overcome the problem of the non-existent derivatives of the discrete spike events, which makes the direct training of SNNs using backpropagation a viable option [3], [17].

C. Related Radar-Based Gesture Recognition Solutions

Most gesture recognition systems use classical signal processing algorithms to generate RDMs and use subsequent neural networks to classify the sensed gesture. Often, the networks are separated in two parts: spatial feature generation and temporal sequence recognition. The former part is achieved using convolutional layer structures to extract meaningful features from the incoming range-Doppler images [9], [25], [27]. The latter part then analyzes sequences of extracted features to recognize the actual movement of the hand and fingers. These temporal dependencies are evaluated using either temporal convolutions [21] or long-short term memory (LSTM) layers [25], [27]. Alternative solutions based on convolutional NNs omit the second Fourier transform to generate the RDMs and instead evaluate the temporal development of the distance metric using multiple convolutional layers [22].

SNN-based radar processing has been shown to perform well on the recognition of whole body gestures [2]. There, the authors use convolutional structures to extract features from the spectral input data. Additionally, they use the bio-inspired spike timing dependent plasticity learning rule to adapt the network weights based on the relative timing between spikes. However, they use comparatively large convolutional NNs to classify large body movements in a small dataset. Despite the similar domain and neuron type, the approaches vary in many aspects. Our approach differs in the network size, layer connectivity schemes, the learning algorithm, and the complexity of the task to be solved.

III. Network Architecture

A. Range-Doppler Map Encoding

The input to the network consists of RDMs, which represent the received signal strength in a range-velocity system (Fig. 1b). If the used radar system has multiple receive channels, multiple RDMs are generated and can be used to additionally determine the angle of the perceived targets. The maps can either be fed directly into the network by converting the values of the RDMs directly to the input currents of the spiking neurons, sometimes called current injection, or be encoded into spikes to match the information exchange format of the binary activated networks (Fig. 1c). In this work, we use and compare two different schemes to encode the data: (i) the activation of the highest valued entries of the RDMs, and (ii) the activation of every value which is larger than the average. In both cases, the binarization is performed for each input channel individually to prevent the dominance of a single channel due to different characteristics of the channels. The resulting channel-wise binary encoded RDMs are stacked and form a 4-dimensional tensor of the format [time, range, velocity, channel]. Accordingly, the encoding scheme converts the amplitude values in the RDMs along the range and Doppler axis $v_{r,d}$ for each time step and channel into a binary form $\delta_{\text{input},r,d}$.

In the first encoding scheme, the highest valued entries of the RDMs are activated. The activation threshold α defines

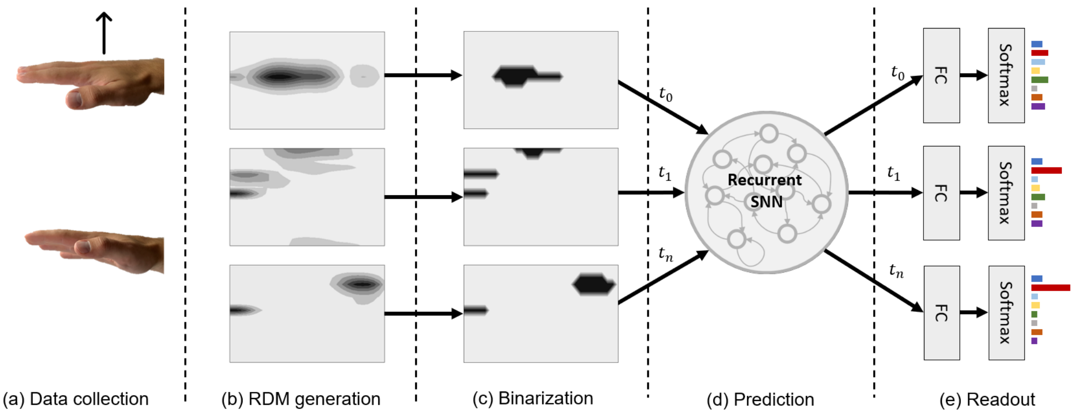


Fig. 1: Examined network architecture consisting of an encoding layer, a fully recurrently connected population of binary activated neurons, and an integrating output layer with softmax activation.

the fraction of bins which are active at any time

$$\delta_{\text{input},r,d} = \begin{cases} 1, & \text{if } v_{r,d} \in \{x \mid x \text{ in } \alpha \text{ largest values of } v\} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The second scheme does not require hyperparameters. For each frame and channel the RDMs are averaged and each bin which has a value larger than the average is activated:

$$\delta_{\text{input},r,d} = \begin{cases} 1, & \text{if } v_{r,d} > \text{mean}(v) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

B. Hidden Binary Activated Layers

The hidden layers consist of populations of LIF neurons. There, the hidden variable V_i of each neuron is updated in every time step:

$$V_i[t+1] = e^{-1/\tau_i} V_i[t] + I_{i,\text{charge}}[t] - I_{i,\text{reset}}[t] \tag{3}$$

$$\delta_i[t+1] = \begin{cases} 1, & \text{if } V[t+1] > \theta \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$I_{i,\text{charge}}[t] = \sum w_{i,j}\delta_j[t] \tag{5}$$

$$I_{i,\text{reset}}[t] = \delta_i[t]V_i[t]. \tag{6}$$

Subsequently, the non-differentiable binary activation function is applied to compute the neuron's output δ_i. The two currents $I_{i,\text{charge}}$ and $I_{i,\text{reset}}$, in accordance to an equivalent electrical circuit, comprise the weighted inputs of other neurons and the reset of the hidden variable after an activation, respectively.

Different to classical network architectures, binary activated spiking networks allow simple but rich lateral communication between neurons within the same layer using recurrent connections. The hidden layers of the proposed architecture (Fig. 1d) therefore consist of populations of neurons with full connectivity to the preceding as well as within the same layer. Each neuron therefore has $N = N_{\text{pre}} + N_{\text{rec}}$ trainable weights with N_{pre} being the number of afferent neurons (inputs from the feedforward connections) and N_{rec} being the number of neurons within the same population. To incorporate the recurrent connections, equation 5 extends to

$$I_{i,\text{charge}}[t] = \sum w_{i,j}\delta_j[t] + \sum r_{i,k}\delta_k[t]. \tag{7}$$

The matrices w and r hold the feedforward and recurrent connection weights, respectively.

The fully recurrently connected populations can be stacked as any other layer in a neural network. However, in our experiments networks consisting of one single hidden population led to the best results.

The pseudo gradient ψ used in this work is linearly dependent on the current value of the hidden variable V of each neuron:

$$\psi_i = \max\left(1 - \left|\frac{V_i}{\theta} - 1\right|, 0\right). \tag{8}$$

The resulting gradients are used to optimize the connection weights $w_{i,j}$ and $r_{i,k}$ between the neurons. Additionally, the time constant τ and the firing threshold θ can be optimized for each neuron individually or jointly for the whole layer. We use the layer-wise joint optimization to prevent overfitting but enable the population to adapt to the characteristics of the dataset.

C. Output Layer

The output neurons are modeled by simple integrators without any leakage or firing characteristics. The latent variable of each output neuron corresponds to the probability of its represented class to be present at the input. A softmax activation function is applied to normalize the output across the layer (Fig. 1e). By that, a meaningful loss can be computed between the true and the inferred class label.

IV. Experiments

A. Setup

The datasets used in the experiments consist of 11 different gestures which are specifically designed to assess the performance of gesture recognition systems [25]. The gestures comprise small finger movements (pinch pinky, pinch index finger), larger movements of the whole hand (push, pull),

TABLE I: Dataset parameters from the papers Interfacing Soli [25] and TinyRadarNN [21].

Parameter	Interfacing Soli	TinyRadarNN
# of channels	4	2
Chirp frequency	$32 \cdot 40$ Hz	160 Hz
Range bins	32	492
Velocity bins	32	32
Recording length	≈ 1 s	≤ 3 s
# of persons	10	26
Recordings per gesture	$25 \cdot 10$	$35 \cdot 26$
Total recordings	2750	10010

and gestures with different speed of movement (slow and fast swipe).

The initial version of the dataset [25] uses Google's project Soli sensor [13]. A second, independent version uses Acconeer's A1 RADAR sensors to record the same set of gestures [21]. Although both datasets consist of the same set of gestures, the classification accuracies reached during the evaluation can not be compared directly. The two sensors produce quite different data streams due to their different designs. Additionally, the datasets differ significantly in the size of the available training data. The relevant differences of the datasets are summarized in table I. We apply the proposed networks to both datasets to achieve the best comparability of the specific algorithms.

B. Preprocessing

For both datasets we use zero padding to reach a constant number of frames for each gesture. Therefore, empty RDMs are added at the end of each recording. In contrast, [13] uses temporal interpolation to achieve constant sequence lengths. However, as this stretches or compresses the sequence, temporal relationships are altered.

The TinyRadarNN dataset is available in a raw format, leaving the freedom to choose the parameters to generate the RDMs. The chirps in the dataset provide information about nearly 500 range bins with a resolution of below a millimeter. To reduce the data rate and to prevent overfitting, each chirp is subsampled using average pooling with a kernel size of four. Subsequently, the Fourier transform is applied to generate the velocity axis of the RDMs. The window size of the Fourier transform is chosen to be 16 chirps to avoid the smearing on the range axis due to too long evaluation time windows. To reduce the size of the input data further and to increase the generalization, an additional max pooling is performed on the range dimension of the generated RDMs.

C. Evaluation Metrics

In [25], the authors differentiate between the per-sequence and the per-frame accuracy. In the former accuracy metric, the information of the completed gesture is present at the time of evaluation and classification. In the latter case, only the instantaneous RDM is presented without the specific information of the start and the end of the sequence. Since the definition of a frame is ambiguous [21], we report the per-sequence accuracy only. The reported accuracies are based on the multi-user leave-one-out (LOO) cross-validation tests. There, the networks are trained using the data of all but one subject and are evaluated on the unseen data. This is repeated for every subject and the resulting accuracy values are averaged, hence the cross-validation. This is the most realistic scenario since it is very unlikely that each user's gesture data is also part of the training set.

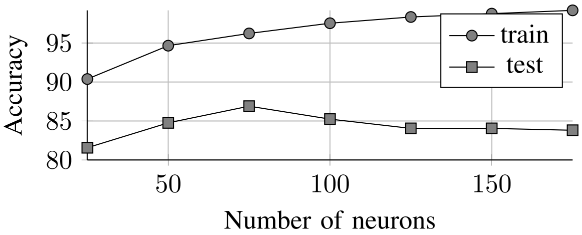


Fig. 2: Relation between the classification accuracy and the number of neurons of the population in the hidden layer on the TinyRadarNN dataset.

V. Results

A. Classification Performance

The resulting classification accuracies are shown in table II. The binary activated stateful network is thus able to outperform previous works [21], [25], on both datasets. Our results show that on both datasets, our binarized inputs lead to higher classification accuracies. The two reported accuracies per network correspond to the test accuracy of the final trained network and the best intermediate performance, respectively. With a discrepancy of two to three percentage points between those values, there might still be room for improvements. Because the best intermediate accuracy is reached at different stages during the cross-validation, however, these values can not be used for comparison with other publications.

The final network's neuron population in the hidden layer consists of 75 recurrently connected neurons. On both datasets, this size marks the optimum whereas larger networks tend to overfit with growing number of neurons while reaching lower test accuracies. Fig. 2 depicts this relation.

The network activity during the inference of the gesture "finger slider" is shown in fig. 3. The differences in the spike activity between the binarization approaches is clearly visible. The average spike activity of the hidden layer is not affected much by the chosen binarization method. We therefore depict only the activity of a population with the input of one of the possible schemes. However, as table II shows, the final classification accuracy varies based on the binarization.

The confusion matrix shown in Fig. 4 illustrates the links between the false classifications of one exemplary cross-validation test. Especially small finger movements – pinch index and pinch pinky – are easily mistaken. Also a steady hand is often confused with gestures involving finger

TABLE II: Comparsion between the Interfacing Soli [25], TinyRadarNN [21], and the proposed spiking architectures for the corresponding dataset. We benchmark the accuracy by using multi-user leave-one-out cross-validation.

	Interfacing Soli		**TinyRadarNN**	
Wang et al. (2016) [25]	88.27%		-	
Scherer et al. (2020) [21]	-		78.85%	
current injection	86.24%	best: 88.58%	79.15%	best: 82.22%
α=0.05 max binarization	87.4%	best: 89.53%	76.43%	best: 79.43%
α=0.10 max binarization	86.71%	best: 88.98%	79.02%	best: 81.47%
mean binarization	88.2%	best: 89.82%	80.31%	best: 82.98%

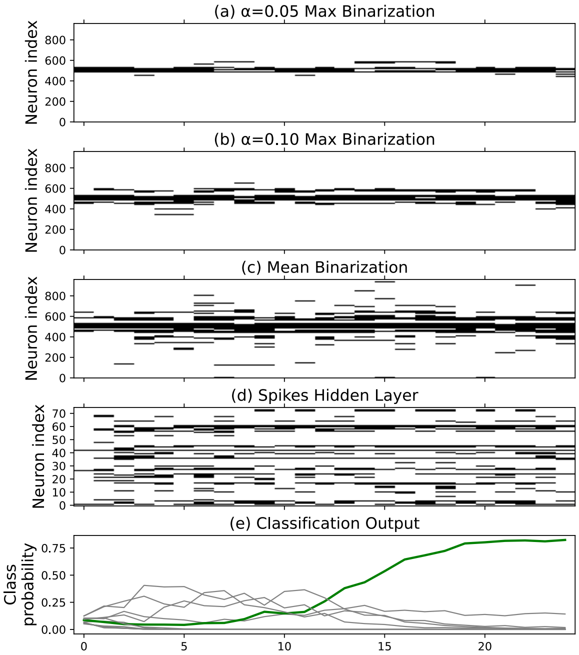
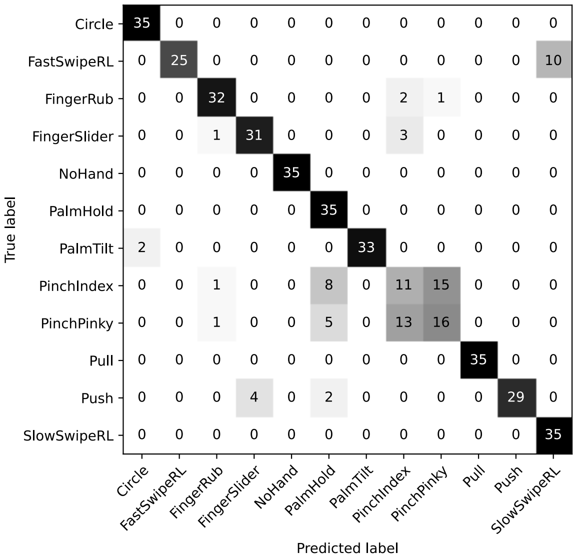


Fig. 3: Exemplary network evaluation of the gestures “finger slider”. (a) Input encoded by $\alpha = 0.05$ max binarization. (b) Input encoded by $\alpha = 0.10$ max binarization. (c) Input encoded by mean binarization. (d) Spikes emitted by the hidden layer. (e) Class probability provided by the output of the network.

movements as in both cases, most of the hand does not move at all. The distinction between a fast and slow swipe gesture is also often misclassified, suggesting a too coarse time resolution or too vague data in the training set due to different interpretations of “fast” and “slow” by the recorded persons. Most gestures, however, can be recognized with low to no error.

B. Input Encoding

t-distributed Stochastic Neighbor Embedding (t-SNE) visualization plots [24] are handy tools to graphically analyze nonlinear relationships within multi-dimensional data where the low-dimensional representation projects the neighboring probability of the higher dimensional data clusters. In our case, we use it to get an understanding of the separability of the raw RDMs as well as the spike train representations of the two input binarization methods and the activity of the hidden layer. Fig. 5 shows the clustering of all classes with color indications where each dot corresponds to a gesture in the test set.



	Circle	FastSwipeRL	FingerRub	FingerSlider	NoHand	PalmHold	PalmTilt	PinchIndex	PinchPinky	Pull	Push	SlowSwipeRL
Circle	35	0	0	0	0	0	0	0	0	0	0	0
FastSwipeRL	0	25	0	0	0	0	0	0	0	0	0	10
FingerRub	0	0	32	0	0	0	0	2	1	0	0	0
FingerSlider	0	0	1	31	0	0	0	3	0	0	0	0
NoHand	0	0	0	0	35	0	0	0	0	0	0	0
PalmHold	0	0	0	0	0	35	0	0	0	0	0	0
PalmTilt	2	0	0	0	0	0	33	0	0	0	0	0
PinchIndex	0	0	1	0	0	8	0	11	15	0	0	0
PinchPinky	0	0	1	0	0	5	0	13	16	0	0	0
Pull	0	0	0	0	0	0	0	0	0	35	0	0
Push	0	0	0	4	0	2	0	0	0	0	29	0
SlowSwipeRL	0	0	0	0	0	0	0	0	0	0	0	35

Fig. 4: Confusion matrix of one LOO cross-validation test (accuracy: 84.67%) using the TinyRadarNN dataset. Misclassifications mainly concern small finger movements and gestures of different speeds.

The visualization shows, that using a non-linear unsupervised clustering algorithm, the different gestures performed by a single person can be separated. Though, a perfect separability of the raw input data is not necessarily given. The binarization encoding does not affect the clustering much, thus suggesting that no relevant information is lost. Some clusters are hard to distinguish in both, the raw and the binarized format. Note the distance between data points in t-SNE is only a representation of clusters, it is not suited to make quantitative statements. The hidden layer shows a clear separation between most of the clusters, highlighting the capabilities of the recurrently connected neuron population to extract temporal patterns from the input signal. The two blue shaded clusters which are not separable in all four t-SNE plots correspond to the gestures “pinch index” and “pinch pinky”. As already seen in the confusion matrix, the distinction between those two gestures is in most cases not

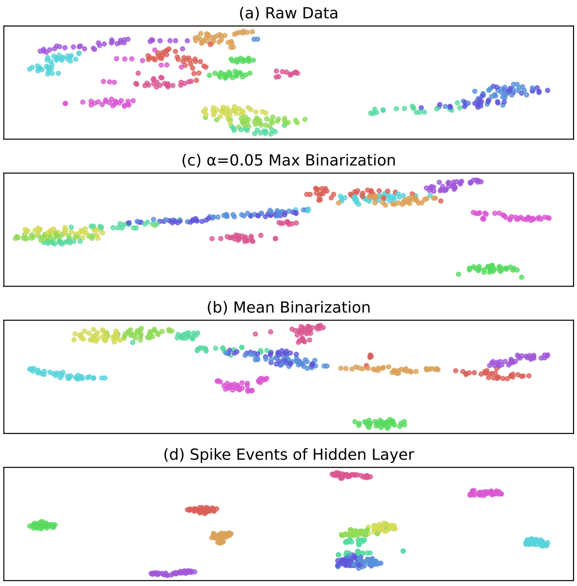


Fig. 5: t-SNE visualization plots for raw data, $\alpha = 0.05$ max binarization, mean binarization, and spike events of the hidden layer. Based on arbitrary LOO test set from the TinyRadarNN dataset.

possible for the network.

C. Analysis of the Computational Complexity

One major motivation for the utilization of spiking neural networks is based on their energy-saving potential [8]. Reasons for that are among others the event-based processing of amplitude-less spike events (see section II-B). This enables the network to only compute the parts of the network which received a non-zero input and to perform additions instead of MAC operations [7]. The latter is possible as the input of a neuron does not have to be multiplied with the connection's weight, but the weight can directly be added to the hidden variable. To exploit these properties already at the input of the network, we binarize the RDMs as shown in section III.

During the processing of one gesture encoded by $\alpha = 0.05$ max binarization, 890 spikes are exchanged on average. 570 spikes are thereby emitted by the encoding layer and 320 by the neurons in the hidden layer. If n $\alpha = 0.10$ max binarization or mean binarization is used, the average number of emitted spikes increases to 1500 and 3000, respectively. In a network with one hidden layer consisting of 75 recurrently connected neurons, each spike emitted by the encoding layer leads to 75 add operations to adapt the hidden variables of the neurons of the hidden layer. Spikes emitted by the neurons in the hidden layer lead to 75 additions at their recurrently connected neurons within the population and 12 additions in the output layer.

Accordingly, using the $\alpha = 0.05$ max binarization approximately 70,500 addition operations are required to classify a gesture, not taking the encoding itself and the activation functions of the neurons into account. This number however already shows the energy saving potential of these type of networks compared to classical approaches. In comparison, [21] reports a total of $20 \cdot 10^6$ multiply-accumulate operations for a convolutional neural network-based solution to the same dataset with a slightly worse classification performance.

VI. Conclusion

In this work, we demonstrate the successful utilization of binary activated spiking neural networks for the recognition of hand gestures in radar data. The proposed network architecture thereby outperforms the results achieved by classical ANN-based approaches on two public datasets. We showed that for radar-based gesture recognition, higher classification accuracies can be reached if the input range-Doppler images are binarized before being analyzed by the SNN. The unique characteristics of SNNs make it possible to replace the huge amount of MAC operations required by large classical networks with an order of magnitude smaller number of additions. This promises a large potential for energy saving while maintaining or even increasing the network's classification performance when executed on neuromorphic hardware.

References

[1] M. G. Amin, Z. Zeng, and T. Shan. Hand gesture recognition based on radar micro-doppler signature envelopes. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2019.
[2] D. Banerjee, S. Rani, A. M. George, A. Chowdhury, S. Dey, A. Mukherjee, T. Chakravarty, and A. Pal. Application of spiking neural networks for action recognition from radar data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2020.
[3] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein, and W. Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. In *Advances in Neural Information Processing Systems*, pages 787–797, 2018.
[4] M. E. Benalcázar, C. Motoche, J. A. Zea, A. G. Jaramillo, C. E. Anchundia, P. Zambrano, M. Segura, F. B. Palacios, and M. Pérez. Real-time hand gesture recognition using the myo armband and muscle activity detection. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, pages 1–6. IEEE, 2017.
[5] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith. Benchmarking keyword spotting efficiency on neuromorphic hardware. In *Proceedings of the 7th Annual Neuro-inspired Computational Elements Workshop*, pages 1–8, 2019.
[6] J. Craley, T. S. Murray, D. R. Mendat, and A. G. Andreou. Action recognition using micro-doppler signatures and a recurrent neural network. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pages 1–5. IEEE, 2017.
[7] S. Davidson and S. B. Furber. Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, 15:345, 2021.
[8] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018.
[9] B. Dekker, S. Jacobs, A. Kossen, M. Kruithof, A. Huizing, and M. Geurts. Gesture recognition with a low power fmcw radar and a deep convolutional neural network. In *2017 European Radar Conference (EURAD)*, pages 163–166. IEEE, 2017.
[10] W. Gerstner and W. M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
[11] C. Graßmann. Simulation pulsverarbeitender neuronaler netze. 2003.
[12] M. M. Khan, D. R. Lester, L. A. Plana, A. Rast, X. Jin, E. Painkras, and S. B. Furber. Spinnaker: mapping neural networks onto a massively-parallel chip multiprocessor. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2849–2856. Ieee, 2008.
[13] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Transactions on Graphics (TOG)*, 35(4):1–19, 2016.

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

[15] C. Mayr, S. Hoeppner, and S. Furber. Spinnaker 2: A 10 million core processor system for brain simulation and machine learning. *arXiv preprint arXiv:1911.02385*, 2019.

[16] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[17] E. O. Neftci, H. Mostafa, and F. Zenke. Surrogate gradient learning in spiking neural networks. *IEEE Signal Processing Magazine*, 36:61–63, 2019.

[18] C.-S. Poon and K. Zhou. Neuromorphic silicon neurons and large-scale neural networks: challenges and opportunities. *Frontiers in neuroscience*, 5:108, 2011.

[19] G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from rgb-d images. In *Proceedings of the IEEE international conference on computer vision*, pages 3889–3897, 2015.

[20] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

[21] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini. Tinyradarnn: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars. *arXiv preprint arXiv:2006.16281*, 2020.

[22] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans. Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks. *IEEE Sensors Journal*, 19(8):3041–3048, 2019.

[23] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida. Deep learning in spiking neural networks. *Neural Networks*, 111:47–63, 2019.

[24] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[25] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges. 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*, pages 851–860. ACM, 2016.

[26] H. Winner, S. Hakuli, F. Lotz, and C. Singer. *Handbook of driver assistance systems*. Springer International Publishing Amsterdam, The Netherlands:, 2014.

[27] Z. Zhang, Z. Tian, and M. Zhou. Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. *IEEE Sensors Journal*, 18(8):3278–3289, 2018.

[28] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*, pages 4903–4911, 2017.