Signal, Image and Video Processing (2023) 17:3505–3513
https://doi.org/10.1007/s11760-023-02574-3

ORIGINAL PAPER

Spike representation of depth image sequences and its application to hand gesture recognition with spiking neural network

Daisuke Miki[1] · **Kento Kamitsuma**[1] · **Taiga Matsunaga**[1]

Received: 18 November 2022 / Revised: 27 January 2023 / Accepted: 23 March 2023 / Published online: 24 April 2023

Abstract
Hand gestures play an important role in expressing the emotions of people and communicating their intentions. Therefore, various methods have been studied to clearly capture and understand them. Artificial neural networks (ANNs) are widely used for gesture recognition owing to their expressive power and ease of implementation. However, this task remains challenging because it requires abundant data and energy for computation. Recently, low-power neuromorphic devices that use spiking neural networks (SNNs), which can process temporal information and require lower power consumption for computing, have attracted significant research interest. In this study, we present a method for the spike representation of human hand gestures and analyzing them using SNNs. An SNN comprises multiple convolutional layers; when a sequence of spike trains corresponding to a hand gesture is inputted, the spiking neurons in the output layer corresponding to each gesture fire, and the gesture is classified based on its firing frequency. Using a sequence of depth images of hand gestures, a method to generate spike trains from the training image data was investigated. The gestures could be classified by training the SNN using surrogate gradient (SG) learning. Additionally, by converting the depth image data into spike trains, 68% of the training data volume could be reduced without significantly reducing the classification accuracy, compared to the classification accuracy under ANNs.

Keywords Spiking neural network · Hand gesture recognition · Depth image · Surrogate gradient learning

1 Introduction

Hand gestures play an important role in expressing emotions and communicating intentions, and various methods using cameras and sensors have been proposed to understand them. In recent years, depth sensors [1] and video analysis [2, 3] techniques have enabled easy acquisition of hand poses, and various methods have been proposed to recognize gestures by analyzing temporal changes in the 2D or 3D shapes of fingers [4–8]. Using depth images is an effective way to represent these changes and examining depth images for hand gesture analysis is an active research area in computer vision because of the potential applications of hand gesture analysis in human–computer interaction. Using convolutional neural networks (CNNs) is a method for recognizing hand gestures from depth images using information regarding the appearance of finger shapes by handling sequences of spatial information of hand motions.

In conjunction with CNNs, there exist methods that successfully handle temporal information using recurrent neural networks (RNNs) and long short-term memory (LSTM). Although these deep learning-based methods have been reported to produce better results not only in analyzing human motion data but also in various tasks including image processing, language processing, and robot control, they require a large amount of computational power for training and execution [9]. Furthermore, the human brain consumes only approximately 12–20 W of electric energy and is capable of performing not only specific tasks but also various other life-sustaining processes, such as processing inputs

Kento Kamitsuma, Taiga Matsunaga have contributed equally to this work.

✉ Daisuke Miki
miki.daisuke@p.chibakoudai.jp

Kento Kamitsuma
s1931052MQ@s.chibakoudai.jp

Taiga Matsunaga
s1931142XJ@s.chibakoudai.jp

[1] Department of Computer Science, Chiba Institute of Technology, 2-17-1, Tsudanuma, Narashino, Chiba 2750016, Japan

to multiple sensory organs simultaneously [10]. Recently, low-power neuromorphic devices [11] that mimic biological brains have been developed, for which applications such as video recognition [12], robotic control [13], and chemosensor data analysis [14] have been proposed. According to the Gartner hype cycle [15], neuromorphic hardware is currently in the innovation trigger phase and is expected to reach a plateau of productivity in the next 5–10 years. Neuromorphic devices are expected to achieve low-power consumption by replacing the information transfer between neurons in artificial neural networks (ANNs) with spiking neural networks (SNN), which are realized through spike trains. The sparsity of the spike representation helps reduce the large amount of data storage required for deep learning; and from a signal-processing perspective, it is also considered advantageous in terms of noise immunity and hardware implementation. Furthermore, an SNN can be treated as a neural network with an internal closed circuit, similar to RNN and LSTM, and is characterized by its superiority in handling time-series data. In this paper, we propose a sparse data representation method for hand gestures that reduces the volume of training data and analyze it using an SNN. To extract the spatial-temporal features related to hand motions necessary for gesture recognition and enable their treatment through SNNs, we present a method for converting a sequence of depth images into spike trains. In addition, a combined convolutional and recurrent structure is applied to the connections between neurons in the SNN to enable gesture recognition that handles the spatial-temporal characteristics of human motion. In the experiment, the SNN was trained on spike data generated using the proposed spike transformation method, and the hand gesture classification accuracy was evaluated. We also evaluated the effectiveness of the proposed method in reducing the data volume (Fig. 1).

2 Related work

2.1 Conventional ANNs for gesture recognition

In earlier research, several ANN-based human gesture recognition methods have been proposed, including CNN-based ones that use spatial information about human motion in videos [4, 5, 16–18]; those that successfully handle temporal information using RNNs or LSTM [6, 19–23]; and those using graph convolutional networks (GCNs) [7, 24–28]. One commonly adopted method is utilizing skeleton data. To explicitly exploit the graphical structure of hand joints, Yan et al. [24] and Li et al. [25] reported a human motion analysis method using spatial-temporal (ST)-GCNs. The ST-GCN-based method can achieve a better classification accuracy by treating temporal changes in human joint locations as a graph structure, and various techniques have been proposed to improve this method. Furthermore, Li et al. [7] treated the hand skeleton as a graph structure and used an ST-GCN to recognize hand gestures. However, there are several challenges to using skeletal information in gesture recognition. For example, localizing hand joints in a depth image is a time-consuming process and only works for high-resolution images. Furthermore, the predicted locations of these joints are not always reliable, and this prediction may fail in the presence of self-occlusion.

An effective method for improving the efficiency of hand gesture recognition is using depth images. Depth images not only help separate the hand-finger region from the background, but also help understand the motion of the hand in the depth direction. Hand-crafted feature-based methods [8, 29–31] and ANN-based methods [4, 32–35] have been proposed for recognizing dynamic hand gestures using depth images. Regarding the hand-crafted feature-based method,

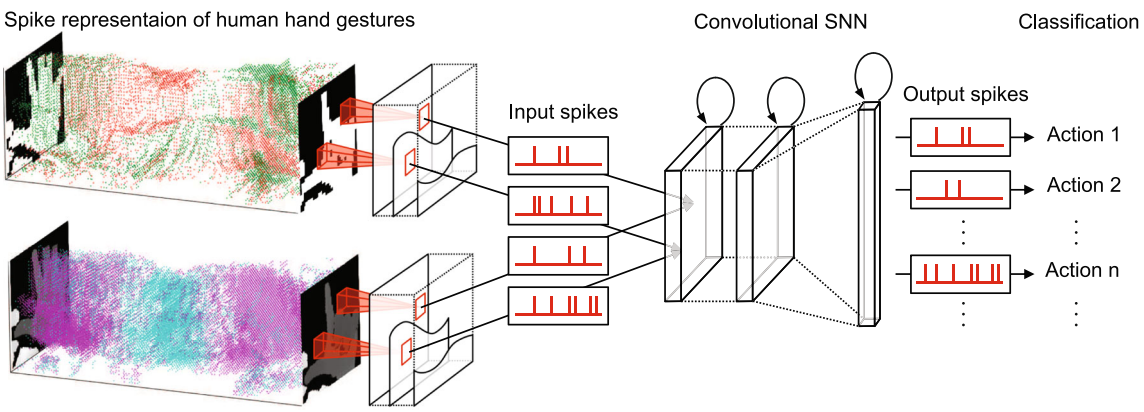

Fig. 1 Principle of proposed method

features related to the spatial and temporal volume of the hand have been utilized. Yang et al. [29] proposed depth motion maps (DMMs) that project point clouds of the human body onto three orthogonal Cartesian views; the spatial-temporal activity of the entire video sequence is accumulated on these planes. Additionally, the histogram of oriented gradients (HOG) and local binary patterns (LBP) descriptors were used to form the final descriptor. Oreifej proposed the histogram of oriented 4D normals (HON4D) descriptor [30], which is based on the distribution of 4D normal vectors in several spatial-temporal cells of action. Using unsupervised kernel Principal Component Analysis (kPCA), Kong et al. [32] proposed a 3D kernel descriptor that learns compact descriptors from the spatial-temporal gradients of depth data. Regarding ANN-based methods, 2D and 3D CNN- and RNN-based methods have been proposed. Wang et al [33] used DMMs and constructed improved depth motion maps (IDMMs) for each individual gesture; these maps were handled using a 2D-CNN. Another method used a 3D-CNN to handle spatial and temporal information simultaneously; Molchanov et al. [4] generated saliency videos from RGB data and used a 3D-CNN to focus on salient objects in the video. Additionally, exploiting the effective ability of ANNs in handling time-series data (such as RNNs or LSTMs) as a means to process input sequences, Molchanov et al. [4] used both 3D-CNNs and RNNs to recognize hand gestures. In their method, a short video clip was fed to the 3D-CNN, and then the output was used as input to the RNN. Additionally, gesture recognition was determined from the estimated scores of the gestures in each frame.

In this study, we employed SNNs instead of RNNs and LSTMs in ANN-based methods that dealt with temporal information. The spatial features of hand gestures were processed in a convolutional structure, and temporal information was handled using the dynamics of spiking neurons in SNNs. We discussed the data spiking representation and other methods for using SNNs in handling depth image data.

2.2 SNNs for classification task

Similar to ANNs, SNNs and their training methods that are applicable to classification tasks have been proposed. Diehl et al. [36] reported that a two-layer SNN model can be trained using the spike-time-dependent plasticity (STDP) as an unsupervised learning method for SNN without using backpropagation (BP) to classify digit images in the modified national institute of standards and technology (MNIST) dataset. SNNs can be configured in a variety of network structures by modifying the connections between neurons and can handle various types of data, such as videos and time-series signals. However, until a BP-based learning rule for multi-layer SNNs was established, there was a significant difference in the classification accuracy compared to ANNs. Recently, the application of BP in SNN training has been studied. Shrestha et al. [37] proposed spike layer error reassignment (SLAYER), a learning rule for sending BP errors to the previous SNN layer. By applying the SLAYER training algorithm, Xing et al. [38] proposed a spiking convolutional recurrent neural network (SCRNN) and confirmed that gesture recognition is possible by analyzing spike sequences acquired from an event camera. Recently, Neftci et al. [39] demonstrated the robustness of surrogate gradient (SG) methods and showed that SNNs trained with SG methods can achieve a performance that is competitive with ANNs. SG utilizes surrogate derivatives to define the derivative of the threshold-triggered firing mechanism, and the SNNs can be trained using gradient descent algorithms as ANNs. Fang et al. [40] proposed incorporating the learnable membrane time constants of a leaky integrate and fire (LIF) neuron model in SNN and its SG-based training method, and demonstrated state-of-the-art accuracy on an event-camera-based classification task. Because this method uses an event camera, it is limited to changes in brightness and does not represent depth-directional movements of the hand.

3 Methods

3.1 Spiking neuron model

The LIF neuron model is the most common model that represents the membrane potential $V_i(t)$ as the internal state and the input current $I_j(t)$ as the input, and its membrane potential is changed via the corresponding weights between connected neurons. A neuron fires when its membrane potential exceeds a certain threshold ϑ and propagates a spike to an external neuron. The time variation of the membrane potential is expressed by the following differential equation:

$$\tau \frac{dV_i(t)}{dt} = -V_i(t) + \sum_t w_{ij} I_j(t). \tag{1}$$

where $V_i(t)$ is the membrane potential of the ith neuron at time t, $I_j(t)$ is the current owing to the spike output from the jth neuron, and τ is the time constant. Additionally, when the membrane potential exceeds ϑ, it is set to the potential $V_i - \vartheta$:

$$V_i \leftarrow V_i - \vartheta \text{ when } V_i \geq \vartheta. \tag{2}$$

By using the Euler method to approximate the general solution of Eq.(1), the membrane potential can be expressed as

$$V[t] = \beta V[t-1] + (1-\beta) I[t], \tag{3}$$

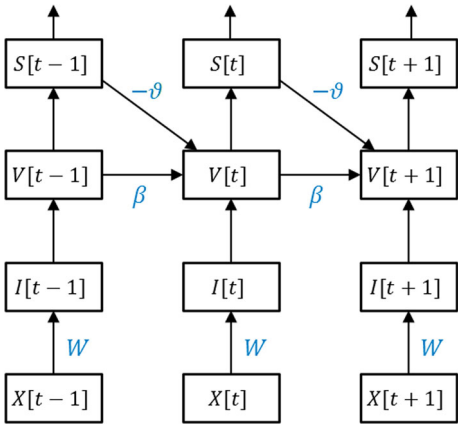

Fig. 2 Computational graphs of the SNN model

where $\beta = e^{-1/\tau}$is the decay rate of $V[t]$. For simplicity, $(1-\beta)$ is replaced by the learnable weight W, and the input current $I[t]$ is expressed as $I[t] = WX[t]$, where $X[t]$ is a single input spike. Furthermore, considering the resetting of the membrane potential because of spike firing, we obtain

$$V[t] = \beta V[t-1] + X[t] - \vartheta S[t-1], \tag{4}$$

$$S[t] = \Theta\left(V[t] - \vartheta\right), \tag{5}$$

where $S[t] \in \{0, 1\}$ is the output spike generated by the spiking neuron using step function Θ. Figure 2 shows a schematic representation of the SNN's computational graph, wherein LIF neurons can be treated in the same manner as RNN. In this study, for gesture recognition, the features of spatial information were extracted through 2D convolution between the LIF neurons and by handling temporal information through the temporal dynamics of the SNN model.

3.2 Spiking data representation

In previous studies, the spike sequence handled by the SNN was generated using information about the frame-to-frame difference in the brightness of RGB images [41–43]. In this study, depth images were used in conjunction with RGB images, and information about changes in image depth information was used to generate spikes, facilitating a better understanding of hand gestures.

To realize hand gesture recognition using depth images with the SNN model, the depth image $\mathbf{D}(t)$ was first binarized to separate the hand region and background, as shown in Fig. 3. From the binarized image $\mathbf{B}(t)$, the value $b(t)$ at coordinate $\{p, q\}$ and the change in value from the previous frame's value $b(t-1)$, the value at coordinate $\{p, q\}$ of the spike image $\mathbf{S}_b^+$ is written as

$$s_b^+(t) = \begin{cases} 1 & b(t) - b(t-1) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Furthermore, the value of the spike image $\mathbf{S}_d^+$ at coordinates $\{p, q\}$ is based on the change in depth information that exceeds the depth change threshold d_θ between the values $d(t)$ and $d(t-1)$ of coordinates $\{p, q\}$ with respect to depth image $\mathbf{D}(t)$ and is written as

$$s_d^+(t) = \begin{cases} 1 & d(t) - d(t-1) > d_\theta \text{ and } s_b^+(t) \neq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Similarly, $\mathbf{S}_b^-$ and $\mathbf{S}_d^-$ were generated in the case of negative changes in the binarized and depth images, and a spike sequence $\{\mathbf{S}_b^+, \mathbf{S}_b^-, \mathbf{S}_d^+, \mathbf{S}_d^-\}$ was generated. Although the original depth image had 32 bits of information at each pixel, each pixel in the spike had 1 bit of information. As shown in Fig. 3g and h, the fewer the number of pixels that exceed the threshold d_θ, the sparser the spike sequence; saving this as a sparse matrix will help reduce the training data volume.

3.3 SNN training

The SNN parameters were trained using the SG method, which is a BP-based method for training SNNs. When using supervised training of the ANN including the BP method, the weight parameter W is updated to minimize the loss function L. As shown in Fig. 2, the gradient with respect to the weight parameters of the loss function is written as

$$\frac{\partial L}{\partial W} = \sum_{t=0}^{T-1} \frac{\partial L}{\partial S_t} \frac{\partial S_t}{\partial V_t} \frac{\partial V_t}{\partial X_t} I_t, \tag{8}$$

and here, from Eq.(5) we can obtain

$$\frac{\partial S_t}{\partial V_t} = \delta(V_t - \vartheta). \tag{9}$$

In this case, owing to the inclusion of Dirac's delta function δ, $\partial S/\partial V$ becomes zero except when the value of $V(t)$ is the threshold ϑ, and the BP cannot be applied. Therefore, in this study, we employed the gradient descent method with SG to train the SNN model. In this method, a fast sigmoid function $\tilde{S}$ is used to approximate the step function Θ and its derivative is written as

$$\tilde{S} = \frac{V - \vartheta}{1/k + \|V - \vartheta\|}, \tag{10}$$

$$\frac{\partial \tilde{S}}{\partial V} = \frac{1}{(k\|V - \vartheta\|)^2}. \tag{11}$$

Here, $\partial \tilde{S}/\partial V$ does not become zero, even when the membrane potential V is outside the threshold value ϑ, which

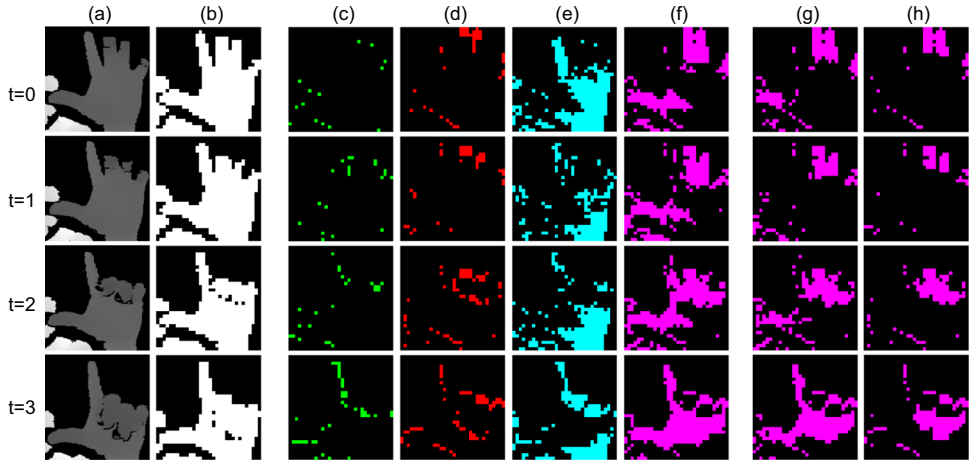

Fig. 3 Depth images representing hand gestures and spike images generated using the proposed method. **a** Depth image **D**. **b** Binarized depth image **B**. **c**, **d** Spike images $\mathbf{S}_b^+$ and $\mathbf{S}_b^-$ generated by increase and decrease in brightness, respectively, of binarized depth image. **e**, **f** Spike images $\mathbf{S}_d^+$ and $\mathbf{S}_d^-$ generated by increase and decrease in brightness, respectively, of raw depth images. **g** Spike images $\mathbf{S}_d^-$ generated by changing $d_\theta = 0.01$, **h** $d_\theta = 0.05$

enables the BP method. In this implementation, the step function in Eq.(5) was used for forward propagation, and SG in Eq.(11) was used for the BP.

4 Experimental results

4.1 Datasets

The dynamic hand gesture-14/28 dataset (DHG14/28) [44] was utilized for quantitative evaluation of the proposed method for gesture recognition. This dataset contains 2800 gesture sequences comprising 14 gestures performed by 20 subjects from depth image data captured by a depth sensor device (Intel RealSense depth camera). Each depth image has a height of 480 pixels and a width of 640 pixels. Additionally, gesture labels and bounding boxes that indicate hand areas based on the depth images are also included. To utilize a depth image with the proposed SNN, it was converted into spike images, as described in Section 3.3. The spike images were resized to 32 pixels both in height and width. In the experiment, 70% (1960) of the total data were used for training and 30% (840) for evaluation. The length of the spike trains was set to 150 frames, which was the longest gesture in the dataset, and zero padding was applied to each data point if the data length was insufficient.

4.2 SNN implementation

The network structure of SNN comprises four two-dimensional convolution layers. The number of channels in each convolutional layer is 32 and in the last layer it is the number of gestures to be classified (14 or 28). We set the kernel size = 3 and stride = 2 for each layer. Post-activation, using the log of the softmax function for the number of output spikes associated with each gesture, loss L was determined through the negative log likelihood. The Adam optimizer was used to optimize the network. After updating the weight parameters 1000 times with a learning rate of 10^{-3}, the learning rate decayed to 10^{-4}, and the weight parameters were updated another 1000 times. Additionally, an ANN with four ConvLSTM [45] layers was prepared for comparison with the proposed SNN. Each ConvLSTM layer has the same channel as the SNN. The difference is that the max pooling operation is introduced after the first three layers instead of a stride operation in the 2D convolution layer. These SNN and ANN were implemented in the PyTorch [46] and snnTorch [9] libraries. The training time for the proposed method was approximately 3 h and 30 min, and that for the ANN was 8 h (AMD EPYC Milan 7443P 2.85 GHz, NVIDIA RTX A6000).

4.3 Gesture classification

Figure 4a shows a raster plot of the input and output spikes of the trained SNN against the evaluation data of the DHG14/28 dataset and 14 gestures and 28 gestures. The classification accuracies for the 14 and 28 gestures are also shown in Tables 1 and 2. The estimation results were obtained by analyzing the data belonging to the "Swipe X (2)" class in the dataset. The numbers in parentheses indicate the shape of the hand (using one finger and the whole hand). The results show that the most frequently fired spikes were those belonging to the relevant class, indicating that the estimation was performed appropriately. The results also show that high values were estimated for similar actions such as "Swipe Left (2)," "Swipe

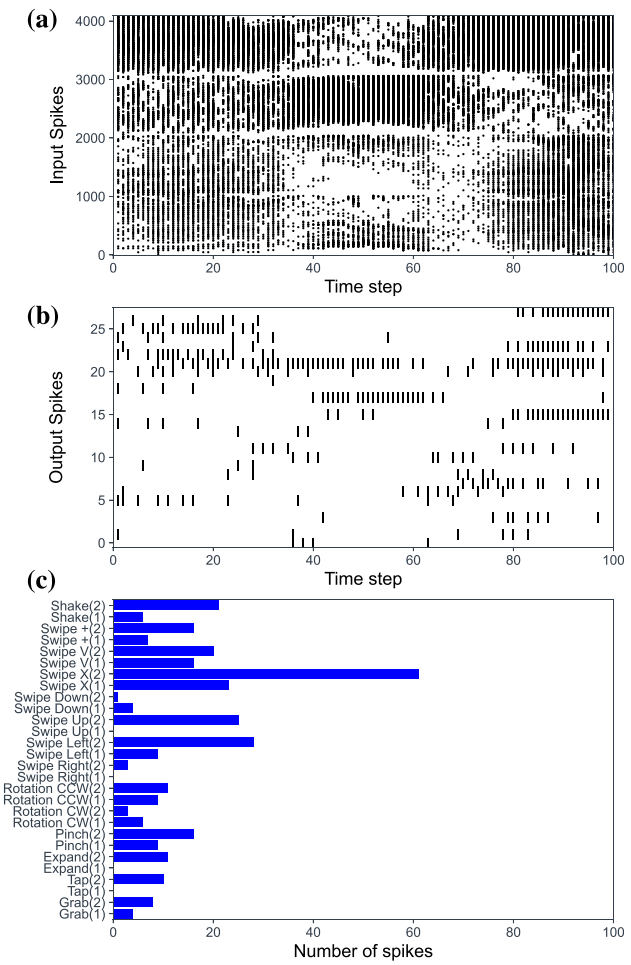

Fig. 4 **a** Raster plot of the input spikes **b** output spikes of the trained SNN and **c** Number of output neuron firings corresponding to each gesture

Up (2)," and "Rotation CCW(Counterclockwise)," while low values were estimated for the other classes. Optimizations were performed for the membrane potential decay rate β, gradient descent slope k, and threshold of spiking neuron ϑ based on the tree-structured Parzen estimator algorithm [47] using Optuna [48]. The resulting membrane potential decay rates for each layer were β_0=0.22, β_1=0.55, β_2=0.68, β_3=0.76, the gradient descent slope k=6.2, and the threshold ϑ=1.5. Figure 5 and Table 3 show the optimization of β in each layer improving the classification accuracy. First, the ConvLSTM-based method yielded a classification accuracy of 93.0%, whereas the SNN-based method yielded accuracies of 86.6%(without depth information) and 92.9% (with depth information). Compared with the method wherein only the spike sequence generated from the information on the brightness change of the binarized depth image was used, under the proposed method, the spike sequences generated from the information on both the depth change and brightness change improved the classification accuracy. Additionally, the SNN-based method achieved a reduction of 87% in the number of trainable parameters compared with that done under the ANN-based method.

Table 1 Comparison of classification accuracy on DHG-14 datasets with depth spike train

Methods	Type	Params	Accuracy (%)
ConvLSTM	ANN	186.3k	93.0
w/o depth spike train	SNN	23.2k	86.6
Ours (d_θ=0)	SNN	23.8k	92.9

Table 2 Comparison of classification accuracy on DHG-28 datasets with depth spike train

Methods	Type	Params	Accuracy (%)
ConvLSTM	ANN	186.8k	89.0
w/o depth spike train	SNN	27.3k	81.5
Ours (d_θ=0)	SNN	27.8k	91.6

4.4 Data volume reduction

Table 4 shows a comparison of different threshold values d_θ for generating spike images $\mathbf{S}_d^+$ and $\mathbf{S}_d^-$. As the value of d_θ increases, the data volume can be reduced, and when d_θ=0, the accuracy in 14 gesture classification tasks is inferior by 0.1 points, but 68% of the data volume can be reduced. When d_θ=0.05, the classification accuracy was inferior by 2.5 points, but the data volume was reduced by 88%. Although the classification accuracy was slightly inferior, the training data volume could be reduced without a significant decrease in the classification accuracy (Fig. 6).

5 Discussion

In this study, we used information on the brightness change and depth information of depth images as a spike sequence and showed that gesture recognition by SNNs was possible. We also investigated how to adjust the parameters to achieve the same level of accuracy under ConvLSTM and how to set the threshold d_θ to control the firing frequency of the spike sequence. By converting depth image data into spike trains, 68% of the training data volume could be reduced without a significant drop in classification accuracy compared to that under the ConvLSTM. On the contrary, it was difficult to build a deep structure like that of a CNN, and it was difficult to obtain a high classification accuracy by freely devising the structure of SNNs. In this study, we conducted experiments within the limitation of SNN and optimized the hyperparameters; the results showed that the classification performance

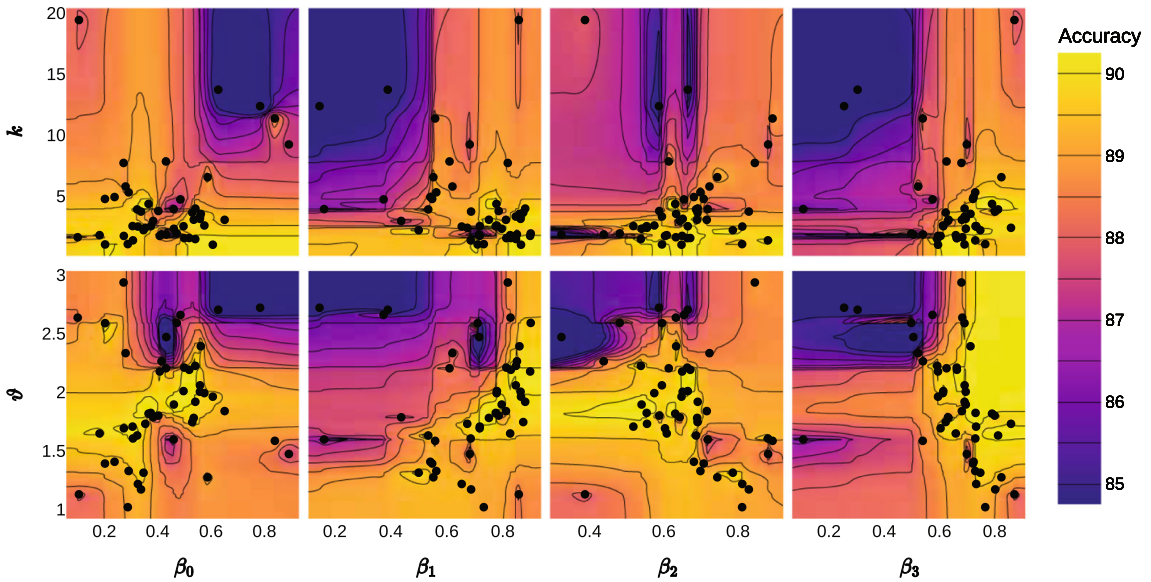

Fig. 5 Optimization of the hyperparameters. **a** Membrane potential decay rate versus surrogate gradient slope. **b** Membrane potential decay rate versus threshold

Table 3 Comparison of classification accuracy on DHG datasets according to membrane potential decay rate of each layer

Membrane potential decay rate				Accuracy (%)	
β_0	β_1	β_2	β_3	14 gestures	28 gestures
0.20	0.20	0.20	0.20	88.3	89.5
0.50	0.50	0.50	0.50	92.5	88.7
0.80	0.80	0.80	0.80	92.1	90.1
0.22	0.55	0.68	0.76	92.9	91.6

The values of the gradient descent slope, the threshold of the spiking neuron, and the threshold of the depth change were set to k=6.2, ϑ=1.5,and d_θ=0, respectively

Table 4 Comparison of classification accuracy (%) on DHG 14/28 datasets according to threshold d_θ

Methods	Type	Data size	Accuracy (%)	
			14 gestures	28 gestures
ConvLSTM	ANN	135.1 MB	93.0	89.0
ours (d_θ=0)	SNN	43.7 MB	92.9	91.6
ours (d_θ=0.01)	SNN	31.5 MB	92.7	89.4
ours (d_θ=0.05)	SNN	16.1 MB	90.6	87.5

of the proposed method was comparable to that of ConvLSTMs. The optimization of the membrane potential decay rate individually in each layer was effective in improving the classification accuracy, which may have resulted in a mechanism wherein the first half of the network is used to extract features and the second is used to learn the handling of temporal information.

6 Conclusions

In this paper, we proposed a sparse data representation method for hand gestures that reduces the volume of training data and its analysis method using an SNN. To extract spatial-temporal features related to hand motions necessary for gesture recognition and to enable them to be treated by SNNs, we present a method for generating a sequence of depth images into spike trains. The experimental results show that after transforming the depth image into a sequence of spikes using the proposed conversion method, the hand gesture can be classified by the SNN. Compared with methods that use ANN, our method requires fewer parameters to be trained. Additionally, by using sparsity to convert depth image data into a spike sequence, the amount of training data can be significantly reduced without a significant loss of classification accuracy compared to the methods using ANNs. Furthermore, it could potentially run on a neuromorphic device in the future, which would reduce the power consumption.

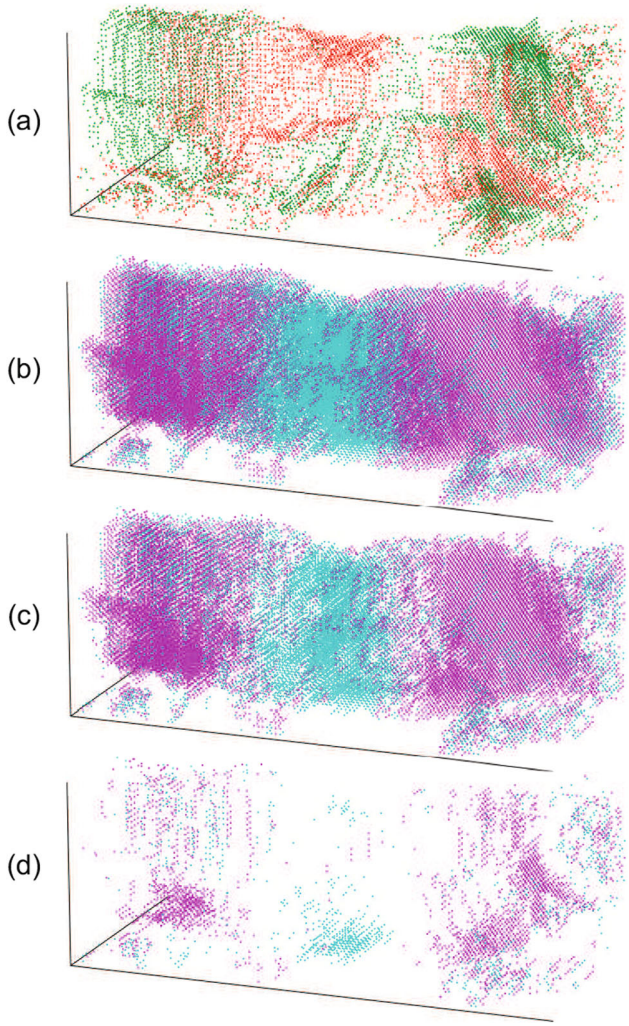

Fig. 6 Visualization of spike image sequences **a** $\mathbf{S}_b^+$ and $\mathbf{S}_b^-$; and **b** $\mathbf{S}_d^+$ and $\mathbf{S}_d^-$, where d_θ=0, **c** d_θ=0.01, and **d** d_θ=0.05

Author Contributions Daisuke Miki wrote the main manuscript text and Kento Kamitsuma and Taiga Matsunaga prepared Tables 1, 2, 3, 4. All authors reviewed the manuscript.

Funding This work was supported by Japan Society for the Promotion of Science (JSPS) Grants-in-Aid for Scientific Research Grant Numbers 22K17937.

Data Availability The data that support the findings of this study are not openly available. Data may be available (http://www-rech.telecom-lille.fr/DHGdataset) upon reasonable request.

Declarations

Conflict of interest The authors declare no competing interests.

Ethical Approval Not applicable.

References

1. Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., Sodnik, J.: An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. Sensors **14**(2), 3702–3720 (2014)
2. Zimmermann, C., Brox, T.: Learning to estimate 3d hand pose from single rgb images. In: Proceedings of the IEEE International Conference on Computer Vision. 4903–4911 (2017)
3. Simon, T., Joo, H., Matthews, I., Sheikh, Y.: Hand keypoint detection in single images using multiview bootstrapping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1145–1153 (2017)
4. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., Kautz, J.: Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 4207–4215 (2016)
5. Liu, Z., Chai, X., Liu, Z., Chen, X.: Continuous gesture recognition with hand-oriented spatiotemporal feature. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. 3056–3064 (2017)
6. Ma, C., Wang, A., Chen, G., Xu, C.: Hand joints-based gesture recognition for noisy dataset using nested interval unscented kalman filter with lstm network. Vis. Comput. **34**(6), 1053–1063 (2018)
7. Li, Y., He, Z., Ye, X., He, Z., Han, K.: Spatial temporal graph convolutional networks for skeleton-based dynamic hand gesture recognition. EURASIP J. Image Video Process. **78**, 1–7 (2019)
8. Verma, B., Choudhary, A.: Grassmann manifold based dynamic hand gesture recognition using depth data. Multimed. Tools Appl. **79**(3), 2213–2237 (2020)
9. Eshraghian, J.K., Ward, M., Neftci, E., Wang, X., Lenz, G., Dwivedi, G., Bennamoun, M., Jeong, D.S., Lu, W.D.: Training spiking neural networks using lessons from deep learning. arXiv preprint arXiv:2109.12894 (2021)
10. Levy, W.B., Calvert, V.G.: Computation in the human cerebral cortex uses less than 0.2 watts yet this great expense is optimal when considering communication costs. BioRxiv (2020)
11. Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G.A.F., Joshi, P., Plank, P., Risbud, S.R.: Advancing neuromorphic computing with loihi: a survey of results and outlook. Proc. IEEE **109**(5), 911–934 (2021)
12. Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., Nayak, T., Andreopoulos, A., Garreau, G., Mendoza, M., et al.: A low power, fully event-based gesture recognition system. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 7243–7252 (2017)
13. DeWolf, T., Jaworski, P., Eliasmith, C.: Nengo and low-power ai hardware for robust, embedded neurorobotics. Front. Neurorobot. **14**, 568359 (2020)
14. Imam, N., Cleland, T.A.: Rapid online learning and robust recall in a neuromorphic olfactory circuit. Nat. Mach. Intel. **2**(3), 181–191 (2020)
15. The Gartner hype cycle (2022) https://www.gartner.com/en/articles/what-s-new-in-artificial-intelligence-from-the-2022-gartner-hype-cycle. Accessed 18 Nov 2022
16. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn. **68**, 346–362 (2017)
17. Verma, B., Choudhary, A.: Dynamic hand gesture recognition using convolutional neural network with rgb-d fusion. In: Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing. 1–8 (2018)

18. Bhaumik, G., Verma, M., Govil, M.C., Vipparthi, S.K.: Extridenet: an intensive feature extrication deep network for hand gesture recognition. The Visual Computer 1–14 (2021)
19. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1110–1118 (2015)
20. Liu, J., Wang, G., Duan, L.Y., Abdiyeva, K., Kot, A.C.: Skeleton-based human action recognition with global context-aware attention lstm networks. IEEE Trans. Image Process. **27**(4), 1586–1599 (2017)
21. Liu, J., Shahroudy, A., Xu, D., Kot, A.C., Wang, G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 3007–3021 (2017)
22. Nguyen, X.S., Brun, L., Lézoray, O., Bougleux, S.: Learning recurrent high-order statistics for skeleton-based hand gesture recognition. In: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE 975–982 (2021)
23. Verma, B.: A two stream convolutional neural network with bi-directional gru model to classify dynamic hand gesture. J. Vis. Commun. Image Represent. **87**, 103554 (2022)
24. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018)
25. Li, B., Li, X., Zhang, Z., Wu, F.: Spatio-temporal graph routing for skeleton-based action recognition. Proc. AAAI Conf. Artif. Intell. **33**, 8561–8568 (2019)
26. Si, C., Chen, W., Wang, W., Wang, L., Tan, T.: An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1227–1236 (2019)
27. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 12026–12035 (2019)
28. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7912–7921 (2019)
29. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM International Conference on Multimedia. 1057–1060 (2012)
30. Oreifej, O., Liu, Z.: Hon4d: histogram of oriented 4d normals for activity recognition from depth sequences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 716–723 (2013)
31. Verma, B., Choudhary, A.: Framework for dynamic hand gesture recognition using grassmann manifold for intelligent vehicles. IET Intel. Transp. Syst. **12**(7), 721–729 (2018)
32. Kong, Y., Satarboroujeni, B., Fu, Y.: Learning hierarchical 3d kernel descriptors for rgb-d action recognition. Comput. Vis. Image Underst. **144**, 14–23 (2016)
33. Wang, P., Li, W., Liu, S., Zhang, Y., Gao, Z., Ogunbona, P.: Large-scale continuous gesture recognition using convolutional neural networks. In: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE 13–18 (2016)
34. Wu, J., Ishwar, P., Konrad, J.: Two-stream cnns for gesture-based verification and identification: Learning user style. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 42–50 (2016)
35. Jain, R., Karsh, R.K., Barbhuiya, A.A.: Encoded motion image-based dynamic hand gesture recognition. Vis. Comput. **38**(6), 1957–1974 (2022)
36. Diehl, P.U., Cook, M.: Unsupervised learning of digit recognition using spike-timing dependent plasticity. Front. Comput. Neurosci. **9**, 99 (2015)
37. Shrestha, S.B., Orchard, G.: Slayer: Spike layer error reassignment in time. Adv. Neural Inf. Process. Syst. **31**, (2018)
38. Xing, Y., Di Caterina, G., Soraghan, J.: A new spiking convolutional recurrent neural network (scrnn) with applications to event-based hand gesture recognition. Front. Neurosci. **14**, 1143 (2020)
39. Neftci, E.O., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. IEEE Signal Process. Mag. **36**(6), 51–63 (2019)
40. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2661–2671 (2021)
41. Kaiser, J., Tieck, V., Hubschneider, C., Wolf, P., Weber, M., Hoff, M., Friedrich, A. Wojtasik, K., Roennau, A., Kohlhaas, R., Dillmann, R., Zöllener, M.:Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks, In: 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), 127–134 (2016)
42. Bi, Y., Andreopoulos, Y.: PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams, In: 2017 IEEE International Conference on Image Processing (ICIP) 1990–1994 (2017)
43. Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., Scaramuzza, D.: Video to events: Recycling video datasets for event cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 3586–3595 (2020)
44. De Smedt, Q., Wannous, H., Vandeborre, J.P.: Skeleton-based dynamic hand gesture recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 1–9, (2016)
45. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. Adv. Neural Inf. Process. Syst. **28**, (2015)
46. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: an imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. **32**, 8024–8035 (2019)
47. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. Adv. Neural Inf. Process. Syst. **24**, (2011)
48. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge discovery and data mining. 2623–2631 (2019)