# Spiking Neural Network Enhanced Hand Gesture Recognition Using Low-Cost Single-Photon Avalanche Diode Array

Zhenya Zang, Xingda Li, Xi Chen, David Day Uei Li

## Abstract

We present a compact spiking convolutional neural network (SCNN) and spiking multilayer perceptron (SMLP) to recognize ten types of gestures in dark and high-ambient light environment, using a 9.6 USD single-photon avalanche diode (SPAD) array. In our hand gesture recognition (HGR) system, photon intensity data from the sensor is leveraged to train and test the network. A vanilla convolutional neural network (CNN) is also implemented to compare the performance of the SCNN with the same network topologies and training strategies. The SCNN is trained from scratch instead of being converted from the CNN. The test of the networks is conducted in both dark and ambient light (AL)-corrupted environments. The result indicates that SCNN achieves comparable accuracy (90.8%) to CNN (92.9%) and exhibits much lower computational complexity with only 8 timesteps. The code and dataset are available at

<https://github.com/zzy666666zzy/TinyLiDAR_NET_SNN>.

## Introduction

Spiking neural network (SNN), known as a typical implementation of neuromorphic computing, is an emerging algorithm for a wide range of applications in vision [1][2][3][4], due to its high computational energy efficiency and accuracy. It exhibits biological plausible as it emulates biological nerves where information is transferred via spike streams. Differentiating with conventional artificial neural network (ANN) using multiply-accumulate (MAC) as basic operators, SNN includes accumulation and dot-product, consuming less hardware and performing higher computational efficiency. Also, SNN does not need pruning mechanism to explore the sparsity in neuron connections because spikes between layers are naturally sparse.

Hardware communities leverage the high efficiency to design neuromorphic chips, such as Tianjic [5], TrueNorth [6], Liohi [7], etc. SNN usually processes data from event-based cameras that only record moving objectives and generate event sequences, represented by spike streams. SNN has been adopted by dynamic vision sensors (DVS) for hand gesture recognition (HGR) [8], [2]. IBM published dynamic gesture datasets [9] that enable various research for HGR using SNN [8], [10], [11], [12]. However, this dataset is more suitable for dynamic gestures, and existing DVS is expensive (normally thousands of USD). Other datasets with conventional high-resolution images used ANN [13], [14] or SNN [15] for static and dynamic HGR. Although background masking and box annotation discard unnecessary data, the preprocessing still costs computational overhead. And high-resolution input data directly inflates models. In this work, we propose a lightweight solution employing a low-cost (9.6 USD) single-photon avalanche diode (SPAD) array and two lightweight SNN models for accurate static HGR.

The contribution of this work is three-fold:

1. We created and released a dataset of gestures captured by a low-cost SPAD array. Using this dataset, we trained spiking convolutional (SCNN) and spiking multiplayer perceptron (SMLP) models from scratch, avoiding conversion from an ANN for guaranteeing accuracy and computational efficiency.
2. We assessed the classification accuracy of SCNN and SMLP during the inference phase in two distinct environments. The first environment involved a dark setting, where the sensor exclusively received reflected photons from the integrated VSCEL laser. In the second case, we tested the models in a noisy environment with ambient light (AL) originating from an LED bulb.
3. We discuss the computational complexity of SCNN and SMLP and demonstrate the efficacy of hardware implementation.

## Related Work

### ST’s SPAD Using Machine Learning

The series of low-cost, portable sensor from ST become increasingly popular in the research of computer vision and pattern recognitions, with the synergy of machine learning. A sensor embedding a VSCEL module and a SPAD array, VL53L1X, was used to classify five types of objects by using the on-chip histogram method to capture reflected photon from the objectives [16]. The same sensor was employed to capture low-resolution depth images and generate high-resolution and skeleton images using DL [17]. Also, five VL53L1X were mounted on a small drone to realize obstacle avoiding and maze-solving [18]. ST also developed its integrated software tool to realize dynamic gesture recognition using time-of-flight technology [19]. The sensor was also successfully integrated with a costume RISC-V processor on a drone to measure its motion and distance from the ground, thereby assisting detecting human poses using DL [20].

### SPAD Sensors Emitting and Processing Spikes

A monolithic chip [21] integrating a SPAD array a SNN processor was designed to process spike trains encoded from 2-D intensity, 3-D depth, and dim-vision imaging. A monolithic chip [22] embedding an analog SNN processor was proposed to process raw photon events from the SPAD sensor on the same chip.

## Data Acquisition and Processing

### Sensor Configuration

### The SPAD sensor we are using is VL53L8CH, which is mounted on an ST’s evaluation board, NUCLEO-F401RE. The key feature of the SPAD sensor that we use is that it supports 15 fps for 8×8 spatial resolution. Due to the limited memory of the processor, the number of time bins (no. bin) of in-pixel histograms should be smaller than 18, with a time resolution (bins’ width in histograms) of ca. 123.3 ps (equivalent to 37 mm). Therefore, background father than 18×33=666 mm is not detectable, presenting a filtering behavior. The integration time is 5 ms. The ranging frequency is 60 Hz. All the mentioned parameters were configured using the CPU on the evaluation board via its firmware. The sensor has a wide, non-configurable diagonal field of view 65°. The reason we used intensity (photon counts) rather than depth images is because the pixel-wise depth data is reconstructed by ST’s fitting algorithm, which slightly introduces errors to images. The compiled code of configuration will be transferred to the SPAD sensor through an I2C interface. As the sensor outputs photon counts, depth data, and histograms concurrently, we further configured the firmware, making the sensor only output photon counts. A Python script receives the number of photo counts through an UART interface.

A group of images of different colors

Description automatically generated

Figure 1. Images in the first row are normalized photon counts images. Ones in the second row are encoded images using Poisson encoder.

### Neural Network Details

The training datasets include 5,100 images of 10 kinds of gestures, including 5,000 gesture images and 1,100 images without gestures. With the same ratio, the test datasets have 1,100 images, with 100 images for each gesture and 100 images for the no-gesture condition. We implemented the networks using PyTorch. SpikingJelley [23] was imported to train the SCNN and SMLP. As spiking neurons generate Dirac delta-alike spikes that are non-differentiable during back-propagation, we used Sigmoid surrogate function to calculate the surrogate gradient, where is a parameter controlling the slope of the function. Due to the small spatial resolution (4×4 or 8×8) of the SPAD sensor, it is arduous to recognize multiple stationary gestures. We utilized bicubic interpolation [24] from OpenCV to enhance the spatial resolution from 8×8 to 25×25, yet managed to keep high fidelity. During the training phase, normalized images () in each batch were encoded as Poisson events with *T* (configure to 8 here) timesteps. We used a Poisson encoder [23] with time-step index to encode the images

Encoded images are shown in Fig. Notably, the input to spiking networks is not the encoded images but spikes at each timestep. Architectures of SCNN and SMLP are depicted in Fig. Due to the simplicity of topologies, three networks were trained using Intel (R) Core i5 CPU @ 3.1 GHz. We used early stopping strategy with 20 epoch patients to avoid over-fitting. Adam is the optimizer. The learning rate is 10­-3. Cross-entropy is the loss function.

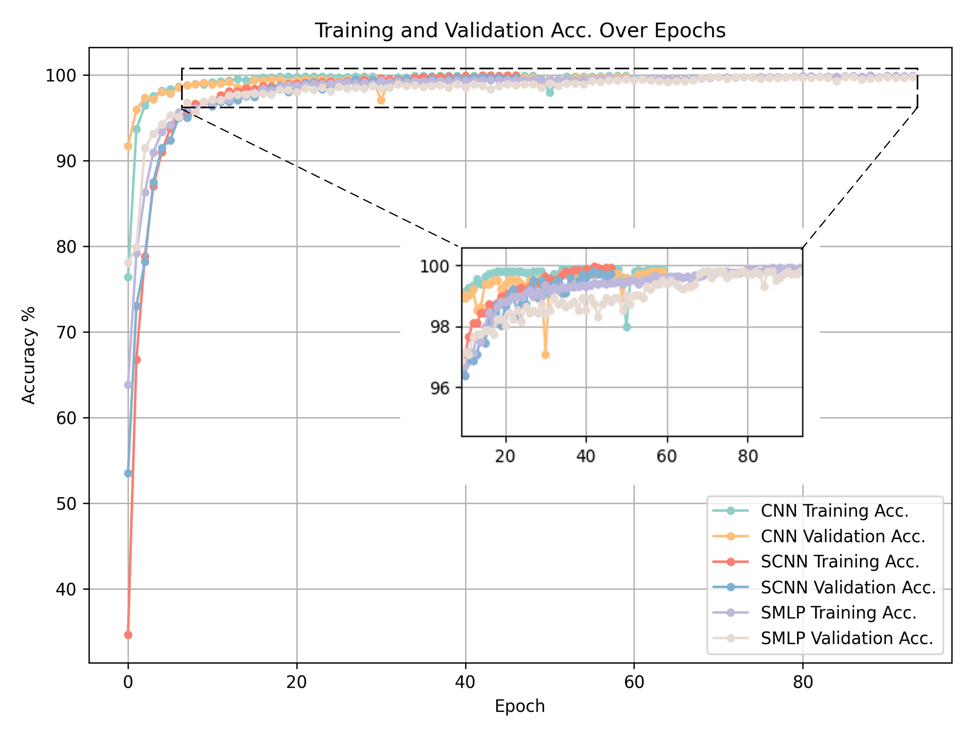


Figure 1. Training and validation accuracy over epochs. CNN, SCNN, and SMLP terminate training at 58, 48, and 93 epochs, respectively.

Two types of SNNs, SCNN and SMLP were constructed to investigate their differentiation in term of performance. In SCNN, batch normalization follows each convolutional layer to accelerate the training convergence. Integrate-and-Fire (LF) neuro was adopted as the activation function. Besides, SMLP includes two fully connected layer followed by drop-off modules to alleviate overfitting. Details of the architectures are depicted in Fig. We also designed a vanilla CNN with the same training strategies and network topology to compare the training performance versus the SNNs. As shown in Fig., the vanilla CNN shows faster convergence than SCNN and SMLP. And the final accuracy of three networks is nearly same.

## Evaluation on Test Datasets

We collected two sets of test datasets with and without AL to validate the robustness of our SNN. The AL is a 60-Watt LED bulb that illuminate the environment and is placed ca. 1 meter away from the sensor. When AL is applied on the sensor, background photons from AL randomly spread over all the time bin in histograms all pixels. We used confusion matrix to evaluate the accuracy of classification of three networks. As shown in Fig, all models testing clear datasets yield higher accuracy than the noisy dataset corrupted by the AL. In Fig. () SCNN and SMLP perform comparable accuracy to CNN. SCNN exhibits high accuracy yet slightly lower accuracy than CNN.

A group of blue squares

Description automatically generated

Figure 2. Confusion matrix of CNN, SCNN, and SMLP with and without AL, where accuracy in each figure indicates the performance of classification from models.

We evaluated the computational workload for inference, targeting the design of lightweight models suitable for low-power devices. Three models were tested on the CPU, and the results are presented in Table I. SCNN and SMLP exhibited longer training times but similar inference times (for 1,100 images in the test dataset) compared to CNN. However, the sparsity of spiking networks shows significant potential in accelerating inference on customized hardware implementations. In contrast to floating-point operations per neuron (FLOP/neuron) in CNN, which involves both multiplication and accumulation, FLOP/neuron in SNN requires only accumulation. The details of calculating FLOPs for each network are provided in the footnote of Table I. SCNN incurs longer inference times due to the regularized hardware architecture on the CPU, which cannot fully leverage the potential of sparsity. Nevertheless, the actual operations in SCNN are reduced by 20.5% compared to CNN with the same topology. Generally, SCNN's accumulations demonstrate lower latency and reduced hardware consumption compared to the MAC operation in CNN. The extent of improvement depends on the specific hardware implementation of adders and multipliers.

Table 1. Computational Evaluation of CNN, SCNN, and SMLP.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Model Size | Parameters | Training time | Inference time | No. operations1 |
| CNN | 0.34 MB | 0.042 MB | 203.52 s | 0.043 ms | 745,750 FLOPs2 |
| SCNN | 0.34 MB | 0.042 MB | 939.21 s | 0.41 ms | 592,617FLOPs3 (reduced by 20.5%) |
| SMLP | 1.02 MB | 0.12 MB | 376.22 s | 0.064 ms | 482,620 FLOPs4 |

1. Inference one image

2. No. operations of CNN is defined by FLOPs. CNN’s FLOPs = , where and are amount of convolutional and fully connected layers. In each convolutional layer , where and are numbers of input and output channel, and indicate output feature size, and indicate the kernel size. In fully connected layer , where and are numbers of nodes of input and output.

3. SCNN’s FLOPs = , where is the spiking rate: .

4. SCMLP’s FLOPs =

## Conclusion

## Reference

[1] S. Afshar, T. J. Hamilton, L. Davis, A. Van Schaik, and D. Delic, ‘Event-Based Processing of Single Photon Avalanche Diode Sensors’, *IEEE Sens. J.*, vol. 20, no. 14, pp. 7677–7691, 2020, doi: 10.1109/JSEN.2020.2979761.

[2] S. Singh, A. Sarma, S. Lu, A. Sengupta, V. Narayanan, and C. R. Das, ‘Gesture-SNN: Co-optimizing accuracy, latency and energy of SNNs for neuromorphic vision sensors’, in *2021 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2021, pp. 1–6. doi: 10.1109/ISLPED52811.2021.9502506.

[3] X. She and S. Mukhopadhyay, ‘Speed: Spiking neural network with event-driven unsupervised learning and near-real-time inference for event-based vision’, *IEEE Sens. J.*, vol. 21, no. 18, pp. 20578–20588, 2021.

[4] Y. Lin and E. Charbon, ‘Spiking Neural Networks for Active Time-Resolved SPAD Imaging’, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 8147–8156.

[5] J. Pei *et al.*, ‘Towards artificial general intelligence with hybrid Tianjic chip architecture’, *Nature*, vol. 572, no. 7767, pp. 106–111, 2019.

[6] F. Akopyan *et al.*, ‘Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip’, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, 2015.

[7] M. Davies *et al.*, ‘Loihi: A neuromorphic manycore processor with on-chip learning’, *Ieee Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[8] A. M. George, D. Banerjee, S. Dey, A. Mukherjee, and P. Balamurali, ‘A reservoir-based convolutional spiking neural network for gesture recognition from dvs input’, in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–9.

[9] A. Amir *et al.*, ‘A low power, fully event-based gesture recognition system’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7243–7252.

[10] R. Massa, A. Marchisio, M. Martina, and M. Shafique, ‘An efficient spiking neural network for recognizing gestures with a dvs camera on the loihi neuromorphic processor’, in *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2020, pp. 1–9.

[11] A. K. Kosta, M. P. E. Apolinario, and K. Roy, ‘Live Demonstration: ANN vs SNN vs Hybrid Architectures for Event-Based Real-Time Gesture Recognition and Optical Flow Estimation’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4147–4148.

[12] S. U. Innocenti, F. Becattini, F. Pernici, and A. Del Bimbo, ‘Temporal binary representation for event-based action recognition’, in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 10426–10432.

[13] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, and A. Makhliarchuk, ‘HaGRID–HAnd Gesture Recognition Image Dataset’, in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4572–4581.

[14] T.-H. Tsai, Y.-J. Luo, and W.-C. Wan, ‘A Skeleton-based Dynamic Hand Gesture Recognition for Home Appliance Control System’, in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022, pp. 3265–3268. doi: 10.1109/ISCAS48785.2022.9937780.

[15] D. Miki, K. Kamitsuma, and T. Matsunaga, ‘Spike representation of depth image sequences and its application to hand gesture recognition with spiking neural network’, *Signal Image Video Process.*, pp. 1–9, 2023.

[16] C. Callenberg, Z. Shi, F. Heide, and M. B. Hullin, ‘Low-cost SPAD sensing for non-line-of-sight tracking, material classification and depth imaging’, *ACM Trans. Graph. TOG*, vol. 40, no. 4, pp. 1–12, 2021.

[17] A. Ruget *et al.*, ‘Pixels2Pose: Super-resolution time-of-flight imaging for 3D pose estimation’, *Sci. Adv.*, vol. 8, no. 48, p. eade0123, 2022.

[18] S. Pikalov, E. Azaria, S. Sonnenberg, B. Ben-Moshe, and A. Azaria, ‘Vision-Less Sensing for Autonomous Micro-Drones’, *Sensors*, vol. 21, no. 16, p. 5293, 2021.

[19] STMicroelectronics, ‘Turnkey gesture recognition solution based on VL53L5CX, VL53L7CX and VL53L8CX multizone Time-of-Flight ranging sensors’. STMicroelectronics, 2023. [Online]. Available: https://www.st.com/en/embedded-software/stsw-img035.html#documentation

[20] D. Palossi *et al.*, ‘Fully onboard ai-powered human-drone pose estimation on ultralow-power autonomous flying nano-uavs’, *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1913–1929, 2021.

[21] X. Yang *et al.*, ‘A Bio-Inspired Spiking Vision Chip Based on SPAD Imaging and Direct Spike Computing for Versatile Edge Vision’, *IEEE J. Solid-State Circuits*, 2023.

[22] M. S. A. Shawkat, M. M. Adnan, R. D. Febbo, J. J. Murray, and G. S. Rose, ‘A Single Chip SPAD Based Vision Sensing System With Integrated Memristive Spiking Neuromorphic Processing’, *IEEE Access*, vol. 11, pp. 19441–19457, 2023.

[23] W. Fang *et al.*, ‘SpikingJelly: An open-source machine learning infrastructure platform for spike-based intelligence’, *Sci. Adv.*, vol. 9, no. 40, p. eadi1480, 2023.

[24] ‘OpenCV: Geometric Image Transformations’. [Online]. Available: https://docs.opencv.org/4.x/da/d6e/tutorial\_py\_geometric\_transformations.html