Comparison of Artificial and Spiking Neural Networks for Ambient-Assisted Living

Sven Nitzsche
FZI Research Center for Information Technology
Karlsruhe, Germany
0000-0002-3327-6957

Brian Pachideh
FZI Research Center for Information Technology
Karlsruhe, Germany
0000-0002-6910-6829

Moritz Neher
FZI Research Center for Information Technology
Karlsruhe, Germany

Marius Kreutzer
FZI Research Center for Information Technology
Karlsruhe, Germany
0000-0003-0602-134X

Norbert Link
Inferics GmbH
Karlsruhe, Germany

Lukas Theurer
Inferics GmbH
Karlsruhe, Germany

Jürgen Becker
Karlsruhe Institute of Technology
Karlsruhe, Germany
0000-0002-5082-5487

2022 Smart Systems Integration (SSI) | 978-1-6654-8849-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/SSI56489.2022.9901412

***Abstract*—In assisted living environments, various situations may arise where a person falls or is otherwise injured and is unable to call for help on their own. In such situations, it is necessary to quickly identify the problem and take appropriate action, such as calling for help. This can be supported or even automated by using vision-based AI systems. In this context, we investigated and evaluated different AI solutions for rapid human action recognition. More specifically, we trained and compared artificial neural networks (ANN) in combination with frame-based cameras to a processing pipeline using spiking neural networks (SNN) and event-based cameras. For the SNNs, we further distinguished and compared two models, which we simulated in software and implemented on event-based hardware. The SNNs feature various layer types, e.g. fully-connected, spiking convolutions and recurrent. The implementation on event-based hardware was compared to GPU-based embedded hardware for artificial neural networks. The comparison was made primarily with regard to the highest possible energy efficiency in order to enable battery-powered vision systems in the future that can be used flexibly not only in assisted living, but also in industrial and smart city applications. The networks were constructed in such a way that they achieve a similar classification accuracy, which was measured on our own dataset specifically recorded for the project.**

***Index Terms*—spiking neural network, neuromorphic computing, artificial neural network, machine learning, event-based camera**

I. Introduction

Assisted living, for example for the elderly, requires constant supervision and support from human personnel to quickly identify dangerous situations or provide assistance. These tasks can be supported by AI systems to improve service quality and response time. With this goal in mind, we have investigated how such an AI system can be designed in the future to be deployable as flexible and fast as possible. Our primary goal thereby was to reduce power consumption and the amount of generated data to make the system as mobile as possible and ultimately enable a battery-powered solution. The specific use case is a vision-based system with integrated processing that only outputs detected events without raw video data thus protecting privacy. The original system consists of a frame camera in combination with an NVIDIA Jetson Xavier NX to run artificial neural networks, which perform human action recognition tasks. A description of these networks can be found in Section IV-A, while the specific tasks for which the neural networks were trained can be found in Section III-A. Fitting sufficiently performant ANN on an embedded device such as the above requires low data rates that are manageable by the neural network. In the original system, lowering the data rate is achieved by downsampling the image generated by the camera and dropping frames to lower the frame rate. However, lowering the data rate in this manner leads to a blanket reduction of measured information for the neural network to process. Instead, it is more preferable to filter out information that is insignificant to a particular task, therefore reducing data rate while still maintaining richness of measured information. To this end, we have turned our attention to event-based neuromorphic sensors, of which the event-based camera has recently reached commercial maturity [1, 2, 3]. The output data rate of an event camera becomes negligible when it observes a static scene, while the changes in a moving scene are measured with remarkably high temporal resolution. Combined with other beneficial properties such as low power consumption and high dynamic range, event cameras seem poised to improve our assisted living application. In this work we thus set out to compare the above described standard ANN

This work is part of the EmbeddedNeuroVision project, which was funded by the Ministry of Economy, Labor and Tourism Baden-Württemberg, Germany, as part of the AI innovation competition.

system with two systems that utilize SNNs. For the comparison, we first recorded a gesture recognition dataset with a frame camera and an event camera simultaneously, which is then used to respectively train the ANN and the SNNs. The first SNN targets the event-based hardware accelerator BrainChip Akida, which we use to compare ANN and SNN running on actual hardware. The second SNN is a spiking reimplementation of the deep and recurrent ResNet-18, which we evaluate in software simulation running on GPU.

II. Background

Inspired by biological photoreceptor cells, each sensor pixel in a neuromorphic camera measures for brightness changes in the observed scene. Once the measured brightness in a pixel increases (decreases) to reach a certain threshold, the camera generates a time-stamped event-packet that represents the positive (negative) threshold crossing in that pixel. For the camera as a whole, this results in a sparse stream of spatio-temporal event information, which holds great potential for resource constrained embedded applications [4]. As an example, in a human action recognition application, stationary neuromorphic cameras act as a filter of insignificant information because an inanimate scene, i.e. a scene devoid of human action, is invisible to the neuromorphic image sensor and thus won't generate data that needs to be filtered by the neural network itself. While possible, processing event-streams is largely inefficient on conventional computers (CPU, GPU) and especially undesirable on energy constrained embedded systems [5]. To optimally accommodate for neuromorphic sensors, the whole system pipeline has to be neuromorphic; in other words, the processing of the event stream should itself operate in the domain of events, without buffering events into a rate-based numerical representation. Computation in the event domain is not feasible using classical Artificial Neural Networks (ANN), as these operate on tensors, while buffering and reshaping the event-stream to fit into a tensor greatly reduces the temporal information contained in the data. Feasibility and scalability of event-based processors has already been demonstrated by devices that implement the acceleration of Spiking Neural Networks (SNN) [6, 7], which are distinct from the ANNs used in classical deep learning. With varying degrees of bio-plausibility, SNNs model the temporal spiking behaviour of biological neurons and their connections, while in contrast, the neurons in ANNs represent a more distant abstraction of biological neurons, which summarize temporal spikes in numerical spike rates. Thus, in ANNs any information about individual spike timings is lost, these do however play an essential role in the spatio-temporal spike processing of biological brains [8] and the encoding of sensory stimuli [9, 10]. Furthermore, it has long been shown that, theoretically, SNNs have higher computational capabilities than ANNs do [11]. While the advent of deep learning [12] has led to the prominence ANNs, recent advancements in event-based learning [13, 14, 15] have made it feasible to train deep SNNs and deploy them on a neuromorphic processor, given a suitable dataset is available. Even if methods derived from deep learning represent only an initial set of tools for training SNNs [16], these early methods already allow for the exploration of neuromorphic processing in many productive tasks, such as the ambient-assisted living example presented in this work.

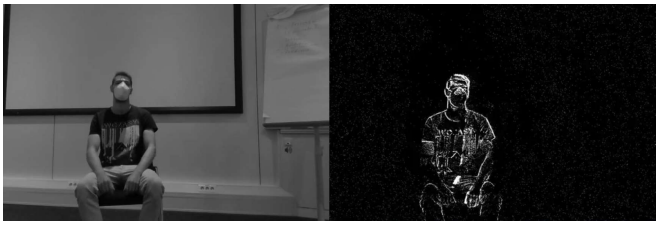

Fig. 1. Exemplary frames from a "sit down" sample recorded simultaneously by a frame camera (left) and an event-based camera (right).

III. Experimental Setup

To ensure an accurate and fair comparison of Artificial and Spiking Neural Networks, we decided to use a symmetric experimental setup, where an event-based camera and a frame camera were placed side by side and aligned in parallel. A Prophesee Gen 4.0 with a resolution of 1280x720 pixels was chosen as the event-based camera. For the frame camera, we used an Intel RealSense D435i with the same resolution. In addition to 2D image information, the RealSense camera also provides depth information, which was, however, not used in the context of this work.

A. Datasets

Using this setup, we recorded 580 samples per camera divided into 12 classes of human actions together with our project partners from Inferics and HS Analysis. Specifically, we had seven persons perform the following actions under varying lighting conditions: sit down, stand up and the ten classes from IBM's DVS Gesture dataset (besides "other gesture") [17]. All samples of the classes "sit down" and "stand up" are individual actions, each approximately 1.5 seconds in length. The remaining classes are repeating actions that we recorded for a total of 10 seconds and then sliced into parts of 1.5 seconds. Fig. 1 shows an exemplary frame from a "sit down" sample; the event-based image was reconstructed by buffering events for 30 milliseconds. Based on these samples we created two datasets, one with only 3 classes ("sit down", "stand up" and "other", where other consists of all samples from the remaining ten classes) and one with all 12 classes.

B. Training Setup

We trained all Spiking Neural Networks on a desktop PC equipped with an AMD Ryzen 9 3900X with 24 threads, an NVIDIA RTX A6000 with 48GB VRAM and 64GB of DDR4 RAM running Ubuntu 20.04 with PyTorch 1.10 and Norse 0.0.6 [18]. The datasets were wrapped using Tonic 1.0.12 [19]. To prevent overfitting, we added multiple augmentations to the training flow and thus effectively doubled the number of samples in a dataset. Specifically, we used dropping of

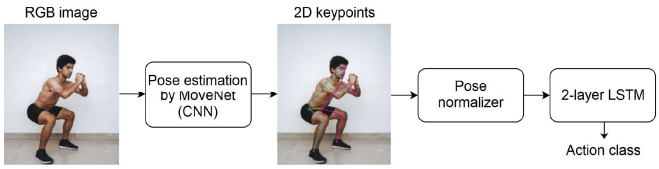

Fig. 2. Data flow for the ANN based solution.

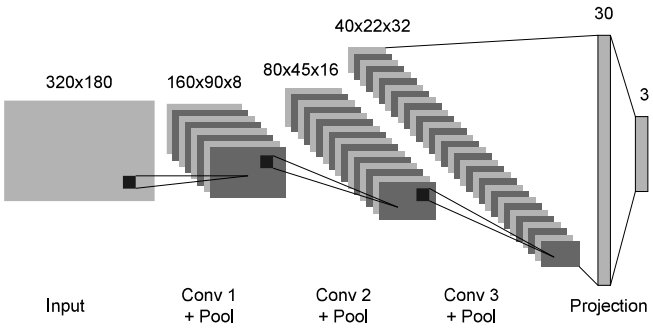

Fig. 3. Visualization of the CSNN architecture for the 3-class dataset.

random events, random cutouts and random cropping as augmentations. Which of these augmentations is actually applied is decided randomly per sample at runtime.

C. Inference Setup

Inference was performed using hardware accelerators and GPU-based simulation. We did most parts of the SNN architecture exploration and fine tuning on the desktop PC described in Section III-B using Norse. However, the given use case requires low power embedded devices, therefore we also tested some models on the event-based hardware accelerator BrainChip Akida using akida framework 2.1.2. For Artificial Neural Networks we used an NVIDIA Xavier Jetson NX with 8GB RAM running in low power (10W) mode.

IV. Neural Network Architectures

A. Artificial Neural Networks

Artificial neural networks are used as a baseline to compare our neuromorphic solution to. The recognition of human activities from RGB video data streams with ANNs was so far tackled mainly with two different approaches: (1) End-to-end approaches, which directly classify the raw data stream to recognize activities captured by the data. (2) Approches, which use pose estimators to first find the coordinate values of person keypoints (joints and facial keypoints) present in each image via a pose estimator and then analyse the stream of pose coordinates to detect the activities. The first approach requires much more training data than the second as it lacks the abstraction from variations of the illumination and the imaging process, which must then be covered by training data. A pre-trained pose estimator can be used, which has already learned these abstraction, so that only variations of a pose sequence within a certain activity have to be covered by training data in the second approach. This is why we chose the second approach for our ANN based action recognition. In particular, our implementation is composed of a pre-trained MoveNet pose estimator [20] with roughly 6.2 million parameters, a pose normalizer and an LSTM recurrent network with two layers and 285 103 parameters, which processes the normalized pose stream. The flow of this solution is shown in Fig. 2, based on work created and shared by Google and used according to terms described in the Creative Commons 4.0 Attribution License. The input image resolution is 640×480 pixels.

B. Spiking Neural Networks

Since we target the BrainChip Akida as hardware platform for our Spiking Neural Networks, we had to take into account some restrictions on the network architecture imposed by the hardware. At the time of our experiments, only fully connected and convolutional layers with certain parameter and kernel sizes were supported. Therefore, our SNN models mainly rely on convolutional layers without biases. However, since there is a huge temporal context in our dataset, we also added recurrent layers to our SNNs and simulated them on a desktop PC. For the Akida implementation this recurrent layer is replaced with a non-recurrent dense layer.

We considered two kinds of convolutional SNNs. The first network is a simple three layer CNN with leaky integrate and fire (LIF) neurons between every layer and its successor, in the following referred to as CSNN. The convolutions use 3×3 kernels with a padding of 1. We also placed an additional dense feed forward layer as projection before the output that uses recurrent LIF activations. Its architecture is depicted in Fig. 3. The output is a layer of leaky integrators matching the number of classes in the dataset. To minimize overfitting, the training was performed using both high probability dropouts on the dense feed forward layers and low probability dropouts on the convolutional layers.

Due to vanishing spike activity, we were however not able to go beyond 4 layers using this simple architecture. Therefore, as a second network we developed a spiking version of the 18-layer ResNet proposed by He et al. [21]. We altered some aspects of the original model to fit the needs of our platform and better suit the problem at hand. For example, the biases for all layers were disabled. Also no global average pooling is done before the output projection layer, since the detail in the activations is needed to produce good results. Just like in the previous model, we used dropout to minimize overfitting. We replaced the originally proposed ReLU activation functions with LIF neurons and, similarly to the CSNN, equipped the layer before the projection with recurrent LIF activations.

For training we used cross-entropy loss and an Adam optimizer with weight decay for additional regularization. We utilized a learning rate scheduler with a multiplicative factor of 0.1 every 5 epochs, starting at an initial learning rate of 0.002. In addition to the connection weights, trainable parameters also include the threshold voltage of the individual neurons in the dense layer. The training was done for a total of 20 epochs

TABLE I
5-FOLD CROSS VALIDATION ACCURACY AND STANDARD DEVIATION OF SNNS TRAINED ON OUR 3-CLASS DATASET.

Model	Accuracy (%)	
	Average	SD
CSNN	93.69	1.87
Spiking ResNet	95.30	1.70

and with a batch size of 4 for each network. To match our hardware constraints, the data was scaled down spatially by a factor of 4, resulting in a resolution of 320 by 180 pixels.

The two networks were designed to be trainable on our hardware given in Section III-B. For that reason, we could not choose an input resolution equal to the 640 by 480 pixels of the ANN as it would overload the 48GB of available VRAM. Besides the input resolution, the chosen SNN architecture with a combination of convolutional and recurrent layers is, however, similar to the CNN+LSTM combination of the ANN. Another design goal was to make both SNNs similar in terms of trainable parameters. The CSNN has 877k parameters and the Spiking ResNet 851k parameters.

V. EVALUATION AND COMPARISON

Using the two datasets described in Section III-A, we trained our models with a 60-20-20 train-val-test split with equally balanced classes for 20 epochs and with 5-fold cross validation. Training took approximately 110 minutes for the shallow CSNN and 260 minutes for the deep Spiking ResNet. The cross validation results are shown in Table I.

Table II compares both SNNs with the baseline ANN in terms of number of parameters and TOP-1 test accuracy. Additionally, the shallow CSNN adapted for the BrainChip Akida, i.e. without recurrent layer, 8-bit quantized weights and with reduced input resolution of 160×100 pixels, is given as 'CSNN-HW'. Considering only the number of parameters and 3-class accuracy, the comparison shows how effectively Spiking Neural Networks can utilize the time-dependent state of their neurons to adapt to temporal context in the data. This leads to an improved accuracy and less required parameters. Furthermore, the deep Spiking ResNet with its 18 layers performs significantly better than the shallow CSNN with only 4 layers, while further reducing the number of parameters. This shows that SNNs can benefit from deep network structures just like ANNs if sufficient measures are taken to combat vanishing spike activity in deeper layers.

When looking at 12-class accuracy, the ANN is more or less on par with both SNNs. Given the fact that CSNN and Spiking ResNet also achieve about the same accuracy, while the Spiking ResNet is clearly better for 3 classes, we assume that the cause is the input resolution of the data rather than the network architecture. With 320×180 pixels (compared to 640×480 pixels for the ANN), it seems to be too low for the Spiking Neural Networks to detect the fine details that distinguish individual actions from each other. This assumption is further supported by the fact that the CSNN-HW

TABLE II
COMPARISON OF ANNS AND SNNS ON OUR SELF RECORDED DATASET FOR HUMAN ACTION RECOGNITION.

Model	Parameters (∼)	Accuracy (%)	
		3-Class	12-Class
CNN + LSTM	6 500 000	95.65	56.51
CSNN	877 000	96.32	57.51
Spiking ResNet	851 000	**99.26**	**57.78**
CSNN-HW	378 000	95.53	30.31

TABLE III
ACCURACY OF SNN MODELS WITH VARYING INPUT RESOLUTION ON THE 12-CLASS DATASET.

Model	Input Resolution	Accuracy (%)
CNN + LSTM	640x480	56.51
CSNN	320x180	57.51
Spiking ResNet	320x180	57.78
Spiking ResNet	427x240	62.32

with even lower resolution (160×100 pixels) yields even worse results, although it is about equal for 3 classes. We investigated this issue by increasing the input resolution step by step up to a resolution of 427×240 pixels, which is the maximum our training hardware allows. The results of this experiment in Table III seem to prove our assumption. By slightly increasing the input resolution to 427×240 pixels the Spiking ResNet already achieves a 12-class accuracy of 62%, outperforming the ANN with even higher resolution.

One important factor that in theory contributes to the efficiency of Spiking Neural Networks is the inherent sparsity of activations. Since spiking neurons are only activated once the internal state crosses a certain threshold, which typically takes multiple input spikes, they are inactive most of the time. We analyzed the spiking behavior for the CSNN and found that the average number of active neurons per timestep is between 0.2% and 0.75% across all convolutional layers. Fig. 4 shows the CSNN's spike behavior for a random input sample at two different timesteps.

In order to compare efficiency not only in theory but also on real hardware, we measured the energy consumption of our 3-class CSNN on a BrainChip Akida. We chose the CSNN over the Spiking ResNet, because the structure of the latter is not out of the box supported by the hardware. We used test samples from our dataset as input and compared it to the ANN running on a NVIDIA Xavier Jetson NX. Results are shown in Table IV. Note that power consumption includes dynamic as well as static power. As stated before, both models yield approximately the same accuracy, but the SNN requires almost 7 times less power and over 20 times less memory due to less total parameters and 8-bit weight quantization.

VI. RELATED WORK

Event-based human action recognition is a popular task in neuromorphic research, with scientists exploring SNNs using different approaches and tools. Massa et al. use SNNToolBox

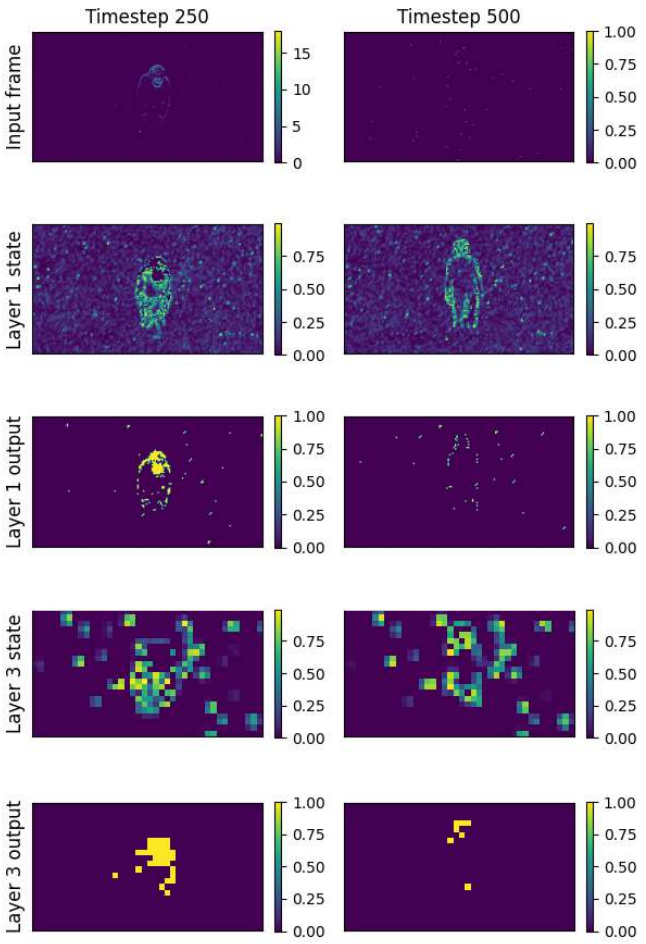

Fig. 4. Spike activity throughout the layers of the CSNN with a random "stand up" sample as input for two timesteps. Timestep 250 shows a high spike activity while the activity at timestep 500 is lower and closer to the average activity we observed across all timesteps and samples. Layer state equals the scaled membrane potential of the neurons, where a value of 1 is the threshold for generating a spike.

TABLE IV
AVERAGE POWER CONSUMPTION OF OUR CSNN RUNNING ON A BRAINCHIP AKIDA AND OUR ANN RUNNING ON A NVIDIA XAVIER JETSON NX WHILE PERFORMING INFERENCE TASKS USING OUR 3-CLASS DATASET.

Model	Power (W)	Parameters (∼)
CNN + LSTM	6.51	6 500 000
CSNN-HW	0.98	378 000

to convert ANNs to SNNs and deploy on Intel Loihi [22]. To train the ANNs, they convert event camera data to frames using multiple methods and evaluate the final accuracy for each [23]. Using the SLAYER algortihm [13], Xing et al. directly train a convolutional and recurrent SNN with LSTM-like cells on the DVSGesture data set [24]. George et al. combine a convolutional SNN with a subsequent reservoir to recognize gestures. They train the network only using bio-plausible learning mechanisms and test on DVS Gesture [25]. Liu et al. propose a hierarchical SNN architecture that extracts both motion and spatial features which are combined for classification in the final layer [26]. Non-neuromorphic approaches to event-based action recognition also exist. Innocenti et al. buffer events and convert them into binary frames for processing with ANNs [27]. Krishnan et al. benchmark conventional frame-based vision models on event-based action recognition. To extract frames, they transform the events using range normalized frequency encoding and extract greyscale images using the surface of active events [28].

VII. CONCLUSION AND OUTLOOK

We were able to show that SNNs can be trained to match or even supersede the accuracy of ANNs for human action recognition tasks with very few training samples when using event-based cameras as input sensors. Applying common deep learning techniques like residual connections enables very deep SNNs that outperformed our ANN in our self recorded dataset by 99.26% to 95.65% while reducing the number of required parameters by a factor of 7 at the same time. We also showed that the power to run inference tasks can be reduced by a factor of 7 with SNNs if executed on appropriate event-based hardware. With the commercial availability of event-based sensors and hardware accelerators, Spiking Neural Networks might be the better choice in future if the task to solve has a temporal context. The biggest hurdle that remains is the high price of the event-based processing pipeline including sensor and hardware platform, which at the time of writing is much higher at almost 3000€ to about 600€ for the classic processing. Nevertheless, especially the deep Spiking ResNet architecture seems to be very promising. Therefore, we plan to analyze it in more detail and also compare it to other SNN architectures on public datasets like IBM's DVS Gesture [17]. Furthermore, we will perform a more comprehensive evaluation of our models on the BrainChip Akida and investigate workarounds for the Spiking ResNet so it can be mapped to hardware.

REFERENCES

[1] Thomas Finateu et al. "5.10 A 1280× 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 GEPS readout, programmable event-rate controller and compressive data-formatting pipeline". In: *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE. 2020, pp. 112–114.

[2] Christian Brandli et al. "A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor". In: *IEEE Journal of Solid-State Circuits* 49.10 (2014), pp. 2333–2341.

[3] Shoushun Chen and Menghan Guo. "Live demonstration: CeleX-V: A 1M pixel multi-mode event-based sensor". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. 2019, pp. 1682–1683.

[4] Guillermo Gallego et al. “Event-based vision: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 44.1 (2020), pp. 154–180.

[5] Charlotte Frenkel, David Bol, and Giacomo Indiveri. “Bottom-up and top-down neural processing systems design: Neuromorphic intelligence as the convergence of natural and artificial intelligence”. In: *arXiv preprint arXiv:2106.01288* (2021).

[6] Mike Davies et al. “Loihi: A neuromorphic manycore processor with on-chip learning”. In: *Ieee Micro* 38.1 (2018), pp. 82–99.

[7] Filipp Akopyan et al. “Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip”. In: *IEEE transactions on computer-aided design of integrated circuits and systems* 34.10 (2015), pp. 1537–1557.

[8] LF Abbott and Wade G Regehr. “Synaptic computation”. In: *Nature* 431.7010 (2004), pp. 796–803.

[9] Sander M Bohte. “The evidence for neural information processing with precise spike-times: A survey”. In: *Natural Computing* 3.2 (2004), pp. 195–206.

[10] Rufin VanRullen, Rudy Guyonneau, and Simon J Thorpe. “Spike times make sense”. In: *Trends in neurosciences* 28.1 (2005), pp. 1–4.

[11] Wolfgang Maass. “Networks of spiking neurons: the third generation of neural network models”. In: *Neural networks* 10.9 (1997), pp. 1659–1671.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[13] Sumit B Shrestha and Garrick Orchard. “Slayer: Spike layer error reassignment in time”. In: *Advances in neural information processing systems* 31 (2018).

[14] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks”. In: *IEEE Signal Processing Magazine* 36.6 (2019), pp. 51–63.

[15] Timo C Wunderlich and Christian Pehle. “Eventprop: Backpropagation for exact gradients in spiking neural networks”. In: *arXiv preprint arXiv:2009.08378* 15 (2020), pp. 16–86.

[16] Mike Davies et al. “Advancing neuromorphic computing with loihi: A survey of results and outlook”. In: *Proceedings of the IEEE* 109.5 (2021), pp. 911–934.

[17] Arnon 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.

[18] Christian Pehle and Jens Egholm Pedersen. *Norse - A deep learning library for spiking neural networks*. Version 0.0.7. Documentation: https://norse.ai/docs/. Jan. 2021. DOI: 10.5281/zenodo.4422025. URL: https://doi.org/10.5281/zenodo.4422025.

[19] Gregor Lenz et al. *Tonic: event-based datasets and transformations.* Version 0.4.0. Documentation available under https://tonic.readthedocs.io. July 2021. DOI: 10.5281/zenodo.5079802. URL: https://doi.org/10.5281/zenodo.5079802.

[20] Google. *MoveNet - TensorFlow Hub*. 2022. URL: https://tfhub.dev/google/movenet/singlepose/thunder/4 (visited on 03/03/2022).

[21] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[22] Bodo Rueckauer et al. “Conversion of continuous-valued deep networks to efficient event-driven networks for image classification”. In: *Frontiers in neuroscience* 11 (2017), p. 682.

[23] Riccardo Massa et al. “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.

[24] Yannan Xing, Gaetano Di Caterina, and John Soraghan. “A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition”. In: *Frontiers in neuroscience* 14 (2020), p. 1143.

[25] Arun M George et al. “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.

[26] Qianhui Liu et al. “Event-based Action Recognition Using Motion Information and Spiking Neural Networks”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization*. Vol. 8. 2021, pp. 1743–1749.

[27] Simone Undri Innocenti et al. “Temporal binary representation for event-based action recognition”. In: *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE. 2021, pp. 10426–10432.

[28] Karthik Sivarama Krishnan and Koushik Sivarama Krishnan. “Benchmarking Conventional Vision Models on Neuromorphic Fall Detection and Action Recognition Dataset”. In: *arXiv preprint arXiv:2201.12285* (2022).