2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Efficient Gesture Recognition on Spiking Convolutional Networks Through Sensor Fusion of Event-Based and Depth Data

Lea Steffen[1,2], Thomas Trapp[1], Arne Roennau[1] and Rüdiger Dillmann[1]

***Abstract*— As intelligent systems become increasingly important in our daily lives, new ways of interaction are needed. Classical user interfaces pose issues for the physically impaired and are partially not practical or convenient. Gesture recognition is an alternative, but often not reactive enough when conventional cameras are used. This work proposes a Spiking Convolutional Neural Network, processing event- and depth data for gesture recognition. The network is simulated using the open-source neuromorphic computing framework LAVA for offline training and evaluation on an embedded system. For the evaluation three open source data sets are used. Since these do not represent the applied bi-modality, a new data set with synchronized event- and depth data was recorded. The results show the viability of temporal encoding on depth information and modality fusion, even on differently encoded data, to be beneficial to network performance and generalization capabilities.**

I. INTRODUCTION

To allow efficient human-robot collaboration, a meaningful and uncomplicated means of interaction between the human and the machine is necessary. Historically, interfaces such as buttons, keyboards and joysticks were mainly used here. However, there is now a growing interest in novel interaction schemes. Gesture recognition is a versatile way to allow communication despite situational difficulties. This may be due to physical impairments, noisy environments, or simply convenience.

Event cameras [1] are a very interesting technology for this use case, due to their high temporal resolution. Hereby, the image acquisition is not transmitted synchronously, but events are generated independently for each pixel if a change in illumination exceeds a threshold. This technique automatically filters out static objects and creates a sparse representation of the scene. Since only movements are of interest for gesture recognition, this functionality is ideally suited. In addition, there is virtually no motion blur with these sensors, which has great advantages for gesture recognition. A Spiking Neural Network (SNN), inspired by biological neurons and neural behavior, processes temporal information instead of analog signals [2], [3]. They are a subspecies of Artificial Neural Network (ANN), but they differ greatly in some respects, require significantly less energy and are capable of much faster inference; they also perform particularly well on sparse data. However, due to their principle of operation, they are not as accurate as ANN. A simple neuron model, commonly used for SNN, is the Leaky Integrate and Fire (LIF) [4]. This model does not consider the shape of biological action potentials, instead a uniform event is used. Information is encoded in spike patterns, thus, the time of a spike's appearance carries its essential data content. The choice of encoding plays a major role for SNN and has a decisive influence on their performance. Rate-based encoding is sub-optimal, as the output it produces is essentially the same as that of an ANN and the theoretically possible efficiency increase of SNN cannot be achieved [5], [6]. Better suited are encoding schemes that take the exact timing of spikes into account, such as time-to-first-spike (TTFS), which enables fast responses within milliseconds [7]. TTFS encodes the amplitude as the relative distance in timing between the first spike and a global reference pulse. Thereby, shorter relative distances are interpreted as more intense signals, than longer ones. The majority of deep learning applications rely on gradient descent using the backpropagation of error [8]. However, SNN are very difficult to train by this method as described in more depth in [9]. A central problem is that spikes cannot be differentiated due to the hard threshold of spiking neurons for spike emission. However, this issue was solved with surrogate gradients [10], [11]. A Spiking Convolutional Neural Network (SCNN) uses spiking neurons to perform convolutions. Thus, a multi-layer SCNN is comprised of alternating convolutional and pooling layers followed by fully-connected layers [12].

In this paper, an SCNN is used on temporally encoded

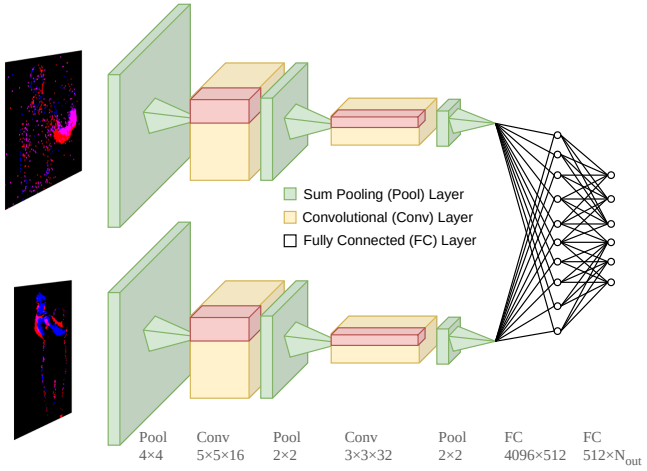


Fig. 1: The network architecture includes feature extraction and sensor fusion. Event streams and temporally encoded depth data are used as input for the feature extraction. The high-level features are then used together for classification.

[1]These authors are with FZI Research Center for Information Technology, 76131 Karlsruhe, Germany steffen@fzi.de
[2]These authors are with Karlsruhe Institute of Technology (KIT), 76131 Karlsruhe, Germany

depth data and event-based data. For learning, the gradient descent algorithm is used with the help of surrogate gradient functions. The proposed method is a new direction in gesture recognition using SNN, not only through the viability of encoding depth data but also through the use of modality fusion to remedy the lack of widely available training data.

II. RELATED WORK

Methods for gesture recognition on event streams are already presented in [13], [14], [15], [16], [17]. In [13] a stereo vision method is applied. The work in [14] focuses on the application of the method on neuromorphic hardware. Similar to our approach, in [15], [16] a convolutional network architecture is used, however, it contains conventional neurons with no temporal dynamics. In [17] gesture recognition is solely a case study for a novel training method, whereby the training process is divided. First, in an offline phase, an SNN is trained on GPU using SLAYER and afterward, the network is deployed on a neuromorphic chip, Intel's Loihi [18]. To allow gradient descent for SNN, the last layer is retrained with a surrogate gradient.

Sensor fusion for gesture recognition is proposed in [19], [20], [21]. In [20] hand gesture recognition is achieved by fusing electromyographic and event data. This is realized by a combination of both in a higher-level feature space. In contrast, in [19] and [21] event data is combined with frame-based.

Regarding temporal coding, to enable processing of depth data with SNN, previous work is presented in [12]. The target application in [12] is object detection on LiDAR point cloud data. Temporal encoding is realized using TTFS which allows sparse processing with a SCNN.

III. SIGNAL ENCODING

The processing is done by an SCNN, as depicted in section IV. SNN work on temporal data, referred to as spike trains. Thus, the sensor signal of both modalities must conform to this format, which is already true for event streams (see Figure 2(a)). The recorded depth data, however, needs conversion, as visualized in Figure 2(b). Thus, a pre-processing step is necessary to encode depth data into spike trains. The encoding must contain enough information to allow for classification inference while being fast enough for a real-time setting. Additionally, sparse encoding is very desirable, as it enables efficient processing. Based on this reasoning, the encoding scheme TTFS is used. The frame rate of the depth sensor is used as a global reference time and a time window is calculated as the inverse of the frames per second (FPS): $T_s = \frac{1}{f_{fps}}$. This allows depth data, which represents the distance to the sensor, to be encoded as a fraction of this time window. Thus, the spike timing is defined relative to a global pulse and either one or no spike is generated in each time window. The encoding scheme is formalized as

$$t_s = (1 - \frac{I_{in}}{I_{max}}) * T_s \Rightarrow t_s = (1 - \frac{d(x,y)}{d_{max}}) * \frac{1}{f_{fps}}, \quad (1)$$

t_s	time of a specific spike occurrence
T_s	time window between two spikes
I_{in}	intensity of the current input
I_{max}	maximum intesity across all relevant inputs
$d(x,y)$	the depth value of the pixel (x,y)
d_{max}	the maximum depth value of the frame
f_{fps}	the frame rate of the depth sensor
$\hat{\boldsymbol{r}}$	the ground truth rate of spikes
r_{true}	the rate of spikes required for prediction
r_{false}	base rate of spikes signifying no prediction
$\mathbf{1}[]$	one hot encoded vector
L	the loss function

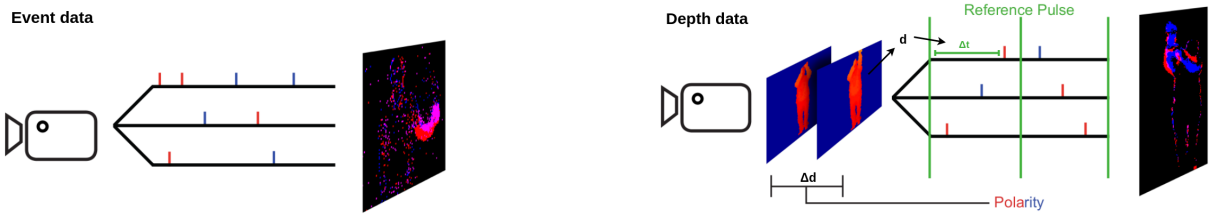
TABLE I: Parameter definition for Equation 1, 3 and 4.

and was developed considering the following aspects (for parameter definition see Table I):

1) *Ordering:* Depth values represent an object's distance to the sensor, with large values representing distant and small values close objects. Encoding this information in a typical TTFS fashion means that distance is conveyed by a temporal delay to the global reference pulse. Thereby, an inverse ordering is applied, which represents large distances in shorter spike times than small distances.
2) *Relativity:* To calculate relative values, either a linear or a logarithmic transform can be used. The linear one directly calculates spike times so the delay grows linearly for the depth value. The direct benefit is its computational simplicity and equal distribution of values along the time frame given by two reference pulses. The logarithmic one, can emphasize differences in depth and result in more differentiated spike timing. However, it leads to clustering of spike times, numerical instabilities and overflow of spike timings into later pulses. Therefore, the linear transform is realized in Equation 1.
3) *Polarity:* In SNN synaptic connections are either excitatory or inhibitory [2]. This is realized for event cameras by the 1-bit polarity, whereby an ON-event represents an increase in brightness and an OFF-event a decrease [1]. This dual-channel approach permits easier perception of motion in the scene and allows for more computational possibilities. Depth data does not contain polarity, thus, it was calculated using the differences in pixels between two frames. An increase in depth excites the neurons, equivalent to an ON-event, and a decrease inhibits the neurons, equivalent to an OFF-event

IV. NETWORK ARCHITECTURE

The network topology is a typical Convolutional Neural Network (CNN) structure. To increase performance, the input is first put through a pooling layer to reduce its size. Then the input is further processed through a combination of pooling and convolutional layers. The architecture, as visualized



(a) (b)

Fig. 2: Preprocessing is only required for (b) the depth data, as (a) the event stream of the ATIS can be used directly as input for SNN. The depth data are encoded with TTFS to be available in spike trains as well.

in Figure 1 is sub-dived in three parts. The subnets for feature extraction are shown in color and the network for sensor fusion is shown in black. As a first training step, the two network instances for feature extraction are trained on the two different modalities. The same network topology is chosen for both networks, the one receiving encoded depth data and the other one taking event data as input. This enables the option for transfer learning between different modalities, which could further improve the generalization capabilities of the network, but is not intended for this approach. The networks used for this step, contain two spiking convolutional layers which alternate with two spiking pooling layers. This is completed by two spiking fully connected layers for classification. To merge the features of the two modalities, the weights of the two classification networks after pre-training are used to initialize the fusion network. This is done by concatenating data before the final two fully connected layers, allowing the final layers to make use of both features for gesture classification. Hence, network fusion is realized by concatenating the two classification networks into a single one. CNN have spatial dimensions and depth channels. The convolution is done by concatenating the depth channels.

A. Training Spiking Networks

As common for classification tasks, the supervised training approach backpropagation, realizing gradient descent, is used. Regarding the application on SNN, two challenges need to be addressed [9]:

1) the non-differentiable activation function
2) the credit assignment problem

To overcome these issues an approach based on the surrogate gradient methodology [10] is used. A LIF model realizes temporal dynamics through a kernel activation function α, which governs the leaky integration dynamics, and the refractory kernel ν, which governs spike generation and resetting the cell state. This is formalized as:

$$u(t) = \sum_{i=0}^{n} w_i(\alpha * s_i)(t) + (\nu * s)(t), \tag{2}$$

with $*$ being the convolution operation. A complete parameter definition is provided in Table I. This formulation as kernel applications, allows for the reversal of this dynamic through element-wise correlation.

The loss function for the training procedure is formalized by:

$$\hat{\boldsymbol{r}} = r_{true}\mathbf{1}[\text{label}] + r_{false}(1 - \mathbf{1}[\text{label}]) \tag{3}$$

and

$$L = \frac{1}{2}\int_{T_s} (\boldsymbol{r}(t) - \hat{\boldsymbol{r}}(t))^\top \mathbf{1}[\text{label}]\, dt. \tag{4}$$

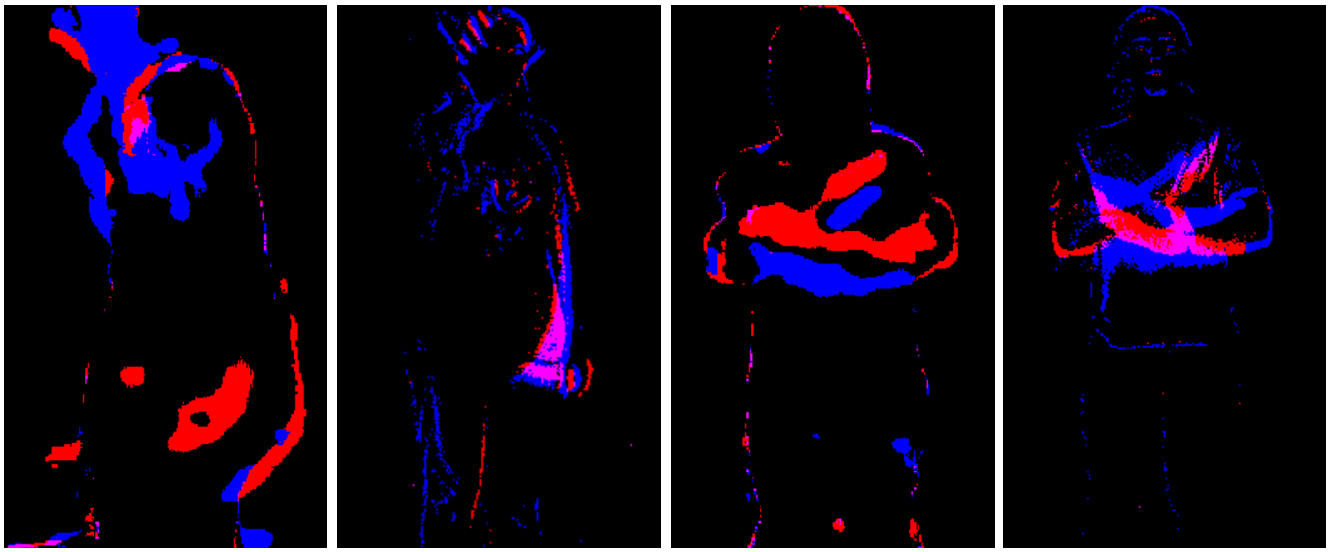
Thereby, $\mathbf{1}[\text{label}]$ represents a hot encoding of the classification target, meaning a vector where only the index of the corresponding class is 1 while all other entries are 0. The target spike rate $\hat{\boldsymbol{r}}$ is governed by two hyperparameters, where r_{true} defines how often spikes should be generated in a time window when the correct class is detected. In addition, the idea is not to force the network not to spike at all when the wrong class is detected. Instead, the second hyperparameter r_{false} defines an allowed maximal spike rate for wrong classifications, if this is surpassed it will be penalized by increasing the error. The loss is consequently expressed as the distance of the actual spike rate to the target rate.

V. EXPERIMENTS

For the SCNN implementation a spiking-focused Pytorch extension is used, the Lava software framework [22]. It is an open-source neuromorphic framework, developed and maintained by Intel's Neuromorphic Computing Lab. Its goal is to exploit the principles of neural computation while also mapping them to neuromorphic hardware. Lava is well suited to implement the learning process, as described in subsection IV-A, because it contains an implementation of the SLAYER algorithm [10].

A. Datasets

Several datasets for gesture recognition exist, however, mainly for conventional sensors. Datasets for event cameras are quite rare, even more so for multi-modal data. Unfortunately, no datasets were found featuring both modalities simultaneously. However, several datasets that have at least partial correspondence with the experimental setup and are open source were found. One with event-based data and three with depth data, as shown in Table II. These datasets diverge quite a bit in size, scope, as well as resolution. The datasets, listed above, do not support any multi-functionality. Also, they lack consistency between labels and samples are partially unstable and incomplete. Consequently, a new dataset



(a) (b) (c) (d)

Fig. 3: Examples of gestures from the newly recorded dataset, featuring the encoded depth data from the RealSense in (a) and (c). Respectively, events from the ATIS are visualized in (b) and (d). Featured gestures are throwing an object ((a), (b)) and crossing the arms in front of the chest ((c), (d)).

is recorded for evaluation purposes. Nevertheless, the above datasets are used for pretraining. Gestures that are too similar to be effectively differentiated are omitted. Furthermore, gestures for which samples exist for both modalities are preferred so that both feature extraction networks are trained on similar data. Strict avoidance of gestures that do not occur in both modalities would limit the set for pretraining to only seven different gestures.

The newly developed dataset for this paper combines

Fig. 4: Sensor setup of the ATIS and Intel Realsense, enabling the development of a synchronized bimodal dataset.

synchronized recordings of an event camera and a depth sensor. The specific sensor model, chosen to provide event data, is the the Prophesee Evaluation Kit Gen3 HVGA-EM, realizing the design of the ATIS [26]. Respectively, for the depth data, the Intel Realsense is used. A large pool of gestures was aimed at when creating the dataset, which can be increased by modifications. Thus, in total 31 gestures were recorded which were partially mirrored and thus increased to 35. A core dataset of 14 gestures, listed in Table III, was defined, as these gestures are also contained in the dataset DVS-Gesture [23]. The extended set contains the core dataset and an additional 21 gestures. However, even the extended set is not overly large, as only two participants completed five repetitions each, resulting in a total of 310 samples.

B. Simulation Parameters

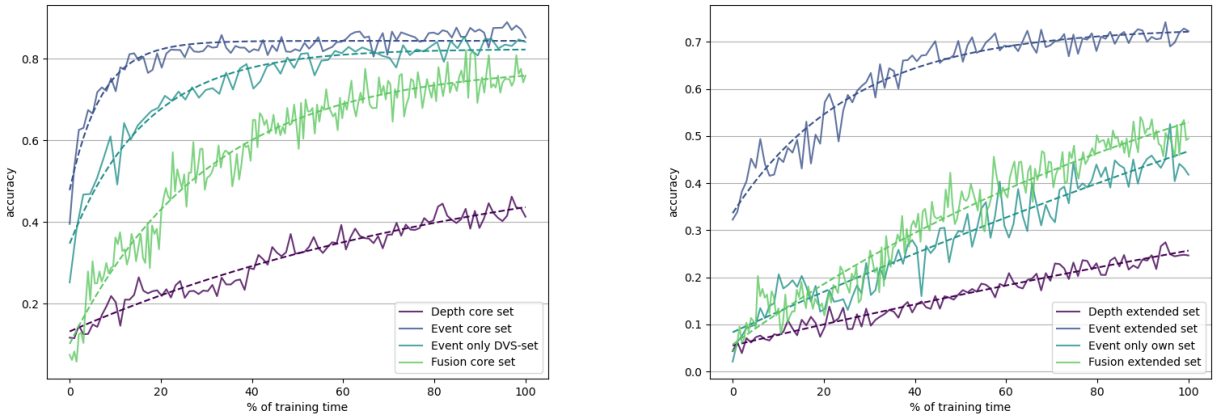
To ease comparisons of networks and keep the network optimization space manageable, neuron parameters and network and simulation parameters are kept constant between runs. Regarding the neuron and synapse-related parameters, an overview is given in Table IV. Only those values are listed that deviate from the default setting. The $weight\ scale$ is layer-dependent, thus, lower layers have a bigger $weight\ scale$. Additionally, in Table V all parameters regarding the simulation are stated. The epochs are set to 100 for the single modalities and to 200 for the fusion network, as less data and more weights need to be handled here. As the $learning\ rate$ defines a 0.05 annealing, it is halved every 25 epochs.

VI. RESULTS

Previous work [10], [23], [17] suggests that training SNN on event data for gesture recognition leads to a solid performance. Due to the limited amount of classes that were usually aimed for, a sharp drop in classification accuracy is likely when increasing the number of gestures to be learned. However, regarding depth data used with a temporal encoding scheme, any concrete performance estimates are difficult to obtain. By combining both modalities, the fusion network is expected to tie the best performance of its sub-networks. Additionally, better generalization capabilities across a larger number of classes are expected. The approach is also expected to perform well on small datasets since it can simultaneously make use of information in both modalities. For deployment, the performance in terms of classification strength is expected to be close to the performance in an offline setting. Due to the encoding

dataset	DVS-Gesture [23]	UTD-MHAD [24]	UTD-Kinect2 [25]	Own
modality	events	depth	depth	event and depth
device	DVS128	Kinect	Kinect 2	ATIS and RealSense
datatype	aedat3.1	MAT file	MAT file	ROSbag
resolution	128×128	320×240	512×424	480×360 and 640×480
# actions	10 + 1	27	10	30
subjects	29	8	6	2
trials	5	4	5	5

TABLE II: A comparison of the datasets used, where '# Actions' refers to the number of different classes of gestures in the dataset, 'subjects' represents the number of unique persons featured across the dataset and 'trials' means the number of sequences per action and subject. Regarding 'resolution', 480×360 refers to the event and 640×480 to the depth data.



(a) (b)

Fig. 5: Development of the mean accuracy. The plots neglect pre-training and show only the final learning process. (a) shows the accuracy of networks using the different modalities on a core set of 10 gestures, see Table III. In (b) the same networks are trained on the extended data set.

and processing approach, the network is designed to allow for low-latency applications.

A. *Offline Training*

For the pre-training step, the single modality networks were trained on all datasets containing the respective modality. Regarding the fused network, the newly created dataset including both modalities is used for training. The performance during offline training is visualized in Figure 5. As seen in Figure 5(a), the classification of the core set between the different modalities is dominated by the network using event data exclusively, reaching accuracies of up to 88% on the training and 91.48% on the testing set. Similar results were achieved by the fusion approach achieving classification accuracies of 86.67% during training and 80% during testing. The lowest performance can be observed with the network working on depth data exclusively showing accuracies of 46.79% and 45.68% during training and testing respectively.
A similar distribution of performance can be observed when training on datasets with more classes. In Figure 5(b) it can be seen that the best performance is offered by the network utilizing only event data, although its classification accuracy drops to a maximum of 71.38% during testing and 79.4% during training. The classification accuracy using other modalities decreases as well, with the fusion approach reaching up to 54.73% during training and 58.9% during testing. Using depth data exclusively results in accuracies up to 26.78% and 28.09% during training and testing.
To analyze the impact of the choice of dataset on training, a comparison of training the event network on each dataset is carried out. The Dynamic Vision Sensor (DVS)-Gesture dataset, which only features core labels, was achieved when considered by itself 84.42% as training and 89.02% as testing accuracies. Omission of the DVS-Gesture set, however, shows a significant drop in classification accuracy, only ever reaching a maximum of 64.93% during training and 43.66% during testing with both trajectories diverging further.

1	hand clap
2	right hand wave
3	left hand wave
4	right arm clockwise
5	right arm counterclockwise
6	left arm clockwise
7	left arm counterclockwise
8	arm roll
9	air drums
10	air guitar
11	right arm draw X
12	right arm arm throw
13	right hand catch object
14	right hand forward punch

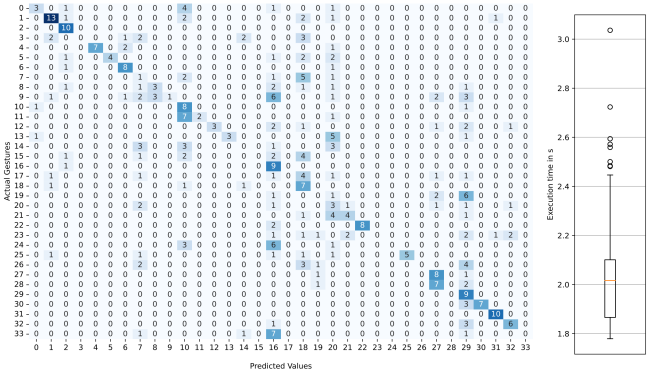
TABLE III: The core set embodies these 14 gestures which are also part of the DVS-Gesture [23]. In the extended dataset, 21 additional gestures are included.

$membrane\ threshold$	1.25
$current\ decay$	1
$voltage\ decay$	0.3
$dropout$	0.1
$weight\ scale$	$1-50$
$weight\ norm$	$true$

TABLE IV: Overview of the parameters regarding the neuron and synapse model.

$sequence\ length$	2000 ms
$learning\ rate$	0.05
$epochs$	100/200
$batch\ size$	8/3
$optimizer$	$ADAM$
$loss\ function$	$Spike\ Rate\ Loss$

TABLE V: Overview of the simulation-related parameters. Regarding the $batch\ size$ and the $epochs$, the first value refers to the network representing a single modality, and the second to the fusion network.



(a) (b)

Fig. 6: Experimental results of the network deployment onto an NVIDIA Jetson Xavier AGX. (a) shows the confusion matrix of the classification output, while (b) shows the distribution of execution times per sample of 2000ms length

B. Deployment

The results when deploying the network onto embedded hardware are summarized in Figure 6. The accuracy concerning the complete dataset is 39.55%, which is significantly lower than the expected mean accuracy of 56.81% (mean of training and testing performance). In the confusion matrix, in Figure 6(a), a distinguishable diagonal can be seen alongside some prediction outliers which tend to accumulate into rows of increased activity. The execution time of the processing of a 2-second sample of spikes is visualized in Figure 6(b), ranging from 1.78s to 3.04s with a mean execution time of 2.02s.

C. Discussion

The network performance using only event data shows performance analogous to related work and serves as a baseline performance test for the own dataset. While the depth and fusion approaches are underwhelming in terms of raw classification accuracy, it does seem that at least some learning can be achieved using only depth data. The approach using modality fusion also stands out for its ability to make use of very limited amounts of training data at the cost of needing multi-modal recordings. This is amplified by its solid performance on a set on which a purely event-based approach overfitted. This refers to the extended dataset that contains all 31 gestures. The reason for overfitting is the lack of balance between event-based and depth data.

One potential source of the depth networks' low performance is assumed in the lack of hyperparameter analysis. While a lot of different configurations and parameterizations of the depth data encoding, neuron models and network topologies have been tested, the testing was far from exhaustive. This could also explain the unexpected drop in performance of the fusion approach when compared to a single modality network. Another aspect to consider is the heterogeneity of the datasets. While previous event-based approaches mainly relied on a single dataset with a consistent setup, the addition and omission of different datasets under different conditions may impact performance negatively, as can be seen with the training runs using single sets.

The results of the deployment show lower classification accuracies than in the offline settings and execution times that are on average slightly higher than the chosen sample length. These circumstances prevent the system from being run in a real-time manner in its current state. However, optimization might decrease execution times further. Its current state already manages to show the viability of limited power simulation of SNN on embedded hardware. More specifically the confusion matrix seems to indicate an accumulation of activity and therefore output spikes even for unrelated classes. Most of the wrong predictions were not a case of

sensibly similar activities that have movements in common, but rather specific neurons which show a high frequency of activation (e.g. Neurons 10, 16, 18, 29 in Figure 6(a)). This could indicate variability in the model execution due to underlying architectural differences.
In summary, while this approach of depth encoding and multi-modal fusion does not yield a new benchmark for classification accuracy on gesture recognition or performance in terms of processing speed and resource management, it does show that temporal encoding of depth data is useful for gesture recognition tasks. Combined with multi-modal approaches, effective use of limited datasets for training can be achieved.

VII. CONCLUSIONS AND FUTURE WORKS

In this work, the temporal encoding of depth images as well as their fusion with event data was conceptualized and implemented. This approach was then used to train and compare networks for exclusive event- and depth-based gesture recognition, as well as a network using sensor fusion of both modalities. To this end, a multi-modal dataset was recorded using an event and depth camera stereo setup. The trained network was then deployed and evaluated on an embedded system. Thereby, the viability of using temporally encoded depth data in gesture recognition was shown. Additionally, fusing the modalities proved useful in remedying the typical signs of overfitting on the small available dataset. This indicates good generalization capabilities and effective use of the samples. Finally, the deployment of SNN models onto embedded systems featuring conventional hardware was shown to be possible. These achievements show the potential benefits of using encoded modalities in SNN, which allow for the integration of readily available data for the training of SNN. In particular, the approach for sensor fusion enables a wide range of potential new research fields without the strict need for excessively large available datasets. The possibility of SNN running on embedded systems further allows for contesting conventional deep learning models in many applications. To compete with the classic deep learning applications, further optimization of the deployment workflow is needed. An in-depth analysis of the encoding techniques regarding depth data and sensor fusion may also lead to a more competitive performance with SNN. This could also be extended by an approach to event-based spatial perception, as proposed in [27]. A very interesting extension of this approach is its deployment on the neuromorphic hardware Loihi. The implementation of the SNN and especially the learning process on a neuromorphic chip is more efficient than on GPU and therefore, promises an increase in performance. In a broader context, the reactive gesture recognition approach presented here can generally be used for robotic applications that include a gesture recognition component, as in [28]. Additionally, it could be adapted and extended for a human-robot collaborative assembly. A neural system as presented in [29], whereby motion planning is carried out in the configuration space by mimicking neural structures, would be particularly interesting for this. The system enables highly reactive path planning in a workspace shared between humans and robots.

ACKNOWLEDGMENT

This research has been supported by the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3).

REFERENCES

[1] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. abs/1904.0, apr 2019. [Online]. Available: http://arxiv.org/abs/1904.08405

[2] W. Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models, Tech. Rep. 9, 1997.

[3] H. Ene Paugam-Moisy and S. Bohte, "Computing with Spiking Neuron Networks," in *Handbook of Natural Computing*, 2012.

[4] R. B. Stein, "A Theoretical Analysis of Neuronal Variability," *Biophysical journal*, vol. 5, no. 2, pp. 173–194, 1965. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/14268952/

[5] W. Guo, M. E. Fouda, A. M. Eltawil, and K. N. Salama, "Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems," *Frontiers in Neuroscience*, vol. 15, p. 638474, mar 2021.

[6] M. Yao, H. Zhang, G. Zhao, X. Zhang, D. Wang, G. Cao, and G. Li, "Sparser spiking activity can be better: Feature Refine-and-Mask spiking neural network for event-based visual recognition," *Neural Networks*, vol. 166, pp. 410–423, sep 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0893608023003660

[7] R. Johansson and I. Birznieks, "First spikes in ensembles of human tactile afferents code complex spatial fingertip events," *Nature neuroscience*, 2004. [Online]. Available: https://www.nature.com/articles/nn1177

[8] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature 1986 323:6088*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: https://www.nature.com/articles/323533a0

[9] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards Biologically Plausible Deep Learning," *arXiv preprint*, vol. arXiv:1502, feb 2015. [Online]. Available: https://arxiv.org/abs/1502.04156v3

[10] S. B. Shrestha and G. Orchard, "SLAYER: Spike Layer Error Reassignment in Time," *Int. Conf. on Neural Information Processing Systems (NeurIPS)*, 2018.

[11] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-based optimization to spiking neural networks," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 51–63, nov 2019.

[12] S. Zhou, Y. Chen, X. Li, and A. Sanyal, "Deep SCNN-Based Real-Time Object Detection for Self-Driving Vehicles Using LiDAR Temporal Data," *IEEE Access*, vol. 8, pp. 76 903–76 912, 2019.

[13] J. Lee, T. Delbruck, P. K. Park, M. Pfeiffer, C. W. Shin, H. Ryu, and B. C. Kang, "Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors," *Int. Symposium on Circuits and Systems (ISCAS)*, pp. 742–745, 2012.

[14] A. Amir, B. Taba, D. Berg, T. Melano, J. Mckinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A low power, fully event-based gesture recognition system," *Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 30, pp. 7388–7397, nov 2017.

[15] A. Chadha, Y. Bi, A. Abbas, and Y. Andreopoulos, "Neuromorphic Vision Sensing for CNN-based Action Recognition," *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2019-May, pp. 7968–7972, may 2019.

[16] J. Chen, J. Meng, X. Wang, and J. Yuan, "Dynamic graph CNn for event-camera based gesture recognition," *Int. Symposium on Circuits and Systems*, vol. 2020-Octob, 2020.

[17] K. Stewart, G. Orchard, S. B. Shrestha, and E. Neftci, "Online Few-shot Gesture Learning on a Neuromorphic Processor," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 512–521, aug 2020. [Online]. Available: https://arxiv.org/abs/2008.01151v2

[18] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[19] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, "Space-time event clouds for gesture recognition: From RGB cameras to event cameras," *Winter Conference on Applications of Computer Vision (WACV)*, pp. 1826–1835, mar 2019.

[20] E. Ceolini, C. Frenkel, S. B. Shrestha, G. Taverni, L. Khacef, M. Payvand, and E. Donati, "Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing," *Frontiers in Neuroscience*, vol. 14, pp. 637–637, aug 2020.

[21] N. Messikommer, D. Gehrig, M. Gehrig, and D. Scaramuzza, "Bridging the Gap between Events and Frames through Unsupervised Domain Adaptation," *IEEE Robotics and Automation Letters (RAL)*, vol. 7, no. 2, pp. 3515–3522, sep 2021. [Online]. Available: http://arxiv.org/abs/2109.02618http://dx.doi.org/10.1109/LRA.2022.3145053

[22] "Lava Software Framework," 2022. [Online]. Available: https://lava-nc.org

[23] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. D. Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "DVS128 Gesture Dataset - IBM Research," 2017. [Online]. Available: https://research.ibm.com/interactive/dvsgesture/

[24] C. Chen, R. Jafari, and N. Kehtarnavaz, "UTD-MHAD: A Multimodal Dataset for Human Action Recognition Utilizing a Depth Camera and a Wearable Inertial Sensor," *Int.Conf. on Image Processing*, 2015.

[25] ——, "Fusion of depth, skeleton, and inertial data for human action recognition," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2016-May, pp. 2712–2716, may 2016.

[26] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, jan 2011.

[27] L. Steffen, S. Ulbrich, A. Roennau, and R. Dillmann, "Multi-view 3D reconstruction with self-organizing maps on event-based data," *Int. Conf. on Advanced Robotics, (ICAR)*, vol. 19, 2019.

[28] G. Bolano, A. Tanev, L. Steffen, A. Roennau, and R. Dillmann, "Towards a Vision-Based Concept for Gesture Control of a Robot Providing Visual Feedback," in *2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018*, 2018.

[29] L. Steffen, T. Weyer, S. Ulbrich, A. Roennau, and R. Dillmann, "Reactive Neural Path Planning with Dynamic Obstacle Avoidance in a Condensed Configuration Space," *Int. Conf. on Intelligent Robots and Systems (IROS)*, 2022.