A Study On the Effects of Pre-processing On Spatio-temporal Action Recognition Using Spiking Neural Networks Trained with STDP

Mireille El-Assal, Pierre Tirilly, Ioan Marius Bilasco

A Study On the Effects of Pre-processing On Spatio-temporal Action Recognition Using Spiking Neural Networks Trained with STDP

Mireille El-Assal*, Pierre Tirilly*, and Ioan Marius Bilasco*
* *Univ. Lille, CNRS, Centrale Lille, UMR 9189 – CRIStAL – Centre de Recherche en Informatique, Signal et Automatique de Lille*
F-59000, Lille, France
Email: mireille.elassal2@univ-lille.fr, pierre.tirilly@univ-lille.fr, marius.bilasco@univ-lille.fr

***Abstract*—There has been an increasing interest in spiking neural networks in recent years. SNNs are seen as hypothetical solutions for the bottlenecks of ANNs in pattern recognition, such as energy efficiency [1]. But current methods such as ANN-to-SNN conversion and back-propagation do not take full advantage of these networks, and unsupervised methods have not yet reached a success comparable to advanced artificial neural networks. It is important to study the behavior of SNNs trained with unsupervised learning methods such as spike-timing dependent plasticity (STDP) on video classification tasks, including mechanisms to model motion information using spikes, as this information is critical for video understanding. This paper presents multiple methods of transposing temporal information into a static format, and then transforming the visual information into spikes using latency coding. These methods are paired with two types of temporal fusion known as early and late fusion, and are used to help the spiking neural network in capturing the spatio-temporal features from videos. In this paper, we rely on the network architecture of a convolutional spiking neural network trained with STDP, and we test the performance of this network when challenged with action recognition tasks. Understanding how a spiking neural network responds to different methods of movement extraction and representation can help reduce the performance gap between SNNs and ANNs. In this paper we show the effect of the similarity in the shape and speed of certain actions on action recognition with spiking neural networks, we also highlight the effectiveness of some methods compared to others.**

***Index Terms*—spiking neural networks, STDP, pre-processing, action recognition, temporal fusion, optical flow, SVM, sequence preparation, spatio-temporal features.**

I. Introduction

Spiking neural networks are biologically-inspired third generation neural networks modelled after the human brain [2]. In these networks the communication between neurons is done by broadcasting spike trains. Some of their advantages over ANNs are biological plausibility, fast information processing when implemented on dedicated hardware, and energy efficiency [2], [3], [4], not to mention that spiking events are sparse in time, which means that these spikes can potentially hold a large amount of information [2]. Despite all of these advantages, and the many theories in which SNNs are capable of avoiding certain bottlenecks of ANNs, current methods such as spatio-temporal back-propagation [5] and ANN-to-SNN conversion [6] do not completely overcome the bottlenecks of ANNs nor certain SNN related limits, such as frequency loss [7]. On the other hand, models trained with spike-timing dependent plasticity (STDP) allow local computations, thus enabling the implementation of these networks on larger ranges of devices, but these models still do not compete with the results achieved by ANNs [8]. Human action recognition is a standard computer vision problem, that can be addressed with many neural network models. Because of its wide range of applications, it is valuable to challenge a spiking neural network with video analysis tasks, in order to inspect the ability of SNNs in processing visual information. But spiking models are still far behind traditional models. Therefore, in order for SNNs to stand out, there is a need for unsupervised methodologies that can make them effectively learn spatio-temporal features, because unsupervised learning has the ability to develop new biologically plausible self-learning methods without needing excessive amounts of labeled data.

In this work, we aim to learn spatio-temporal features in an unsupervised manner with STDP, by pairing different pre-processing methods with two temporal fusion methods, thus, generating static representations that encode local movement. After that, we evaluate the performance achieved by these representations on a convolutional SNN. Experiments are performed on the KTH and Weizmann datasets, which are natural datasets similar to real live applications in computer vision; although ideal recognition rates have already been achieved on these datasets using traditional computer vision approaches, their simplicity makes them good basic benchmarks to study the performance of new models like SNNs trained with STDP when challenged with spatio-temporal information. Thus, attempting unsupervised feature learning using STDP on these datasets is a first step towards bridging the performance gap between SNNs and other deep learning solutions.

II. Related Work

Common spiking neural network learning methods. The most common spiking neural network models featured in the literature are based on ANN-to-SNN transformation, and supervised learning techniques, such as adapting back-propagation on SNNs [9], [10], which target high recognition rates, setting aside many of the advantages of SNNs, such

as energy efficiency during training on dedicated hardware. Some of these models are described in this section. An ANN-to-SNN transformation is applied in [11] where a regular ANN is trained for a given sequence of input frames, and streaming rollouts are used to compute the activations of all ANN units over time. They applied back-propagation-through-time in order to train their network, and then they transformed their ANN into an SNN. However, in their work they do not address the ANN bottlenecks SNNs were made to avoid, because they use a regular ANN to conduct the training. A supervised approach is also suggested in [5], where the authors use a supervised spatio-temporal back-propagation (STBP) algorithm for training SNNs. They solved the "non-differentiable" problem caused by the nature of spikes by using surrogate gradients as approximate derivatives for spike activity. But, in their approach, each convolutional neuron receives pre-prepared convoluted results as input, thus using heavy and costly pre-processing instead of training the convolutional kernels from scratch with spike information. SPAN [12] is a spiking neural network used to classify spatio-temporal data by transforming spike trains during the learning phase into analog signals. In [12], the authors are able to successfully teach their system to recognise certain simple patterns of numbers that they created. But they did not conduct any experiments on a reference video dataset. Therefore, there is no evidence that their model would be realistically applicable on more complex datasets. Another SNN learning method is the BCM (Bienenstock, Cooper, and Munro) learning rule. In [13], the authors proposed a BCM-based spiking neural network model that classifies human action recognition videos. Another learning rule is STDP, which is a biologically plausible unsupervised learning rule [14]. In [15], the authors use STDP learning on a deep SNN. They use temporal coding, and train their network on natural images for the sake of object recognition. In [16] the authors use reward-modulated Spike-Timing-Dependent-Plasticity (R-STDP) and reinforcement learning to train their network to perform action classification. In [17], the authors use a supervised reward-modulated Spike-Timing-Dependent-Plasticity (R-STDP) learning rule to train two SNN-based sub-controllers on obstacle avoidance tasks. In this work we explore another way of training the network to perform action classification. We use the biological STDP learning rule [18] in an unsupervised manner to train our convolutional spiking neural network to learn spatio-temporal features. This unsupervised learning gives the advantage of not needing a large amount of labeled data.

Spatio-temporal information learning. In this section, we review traditional models, such as ConvNets, that can learn spatio-temporal information from videos. In [19], the authors evaluate multiple approaches of extending CNNs into video classification. Then they highlight an architecture that separates the spatial information of the input into a low-resolution and a high-resolution context stream. After that, they describe multiple fusion methods to fuse the information across the temporal domain. In [20], a two-stream model is introduced. In this model, two deep convolutional networks are used to separate the spatial and temporal recognition streams. The spatial stream relies on still frames and is responsible for the information regarding appearance, while the temporal stream relies on multi-frame dense optical flow and is responsible for the movement information found in the motion between frames. These two streams are combined by a type of feature fusion which is late fusion [21]. This fusion forms the complementary information needed to achieve action recognition in videos. A more complex approach is explained in [22], where a spatio-temporal pyramid architecture is introduced. In this paper, the authors used an architecture similar to the two-stream method in [20] in the first stage. The spatial stream is represented by still RGB frames that contain the appearance information, while optical flow is used to capture the motion between frames, in the temporal stream. Then these channels are fused together in the first step, as in two-stream methods, but a multi-level fusion pyramid of spatio-temporal features is added when the same streams are fused again in step two with the result of their previous fusion. Although not implemented in the context of SNNs, these models are interesting because they show the importance of both the spatial as well as the temporal information in action recognition. They also highlight the importance of feature fusion in creating a complete data representation. Temporal fusion is introduced in [21] where the authors create a Dual Temporal Scale Convolutional Neural Network (DTSCNN) architecture to recognize spontaneous micro-expression.

In brief, there is a need for understanding different methods that represent the spatio-temporal information found in videos, and their implementation with SNNs. This work can serve as a first step towards creating models that can learn spatio-temporal features and conduct their training locally, in a processing-cost friendly manner that can be used in real-world applications. This energy efficiency can be achieved with STDP learning. This paper contributes to the study of bringing closer the nature of the spatio-temporal information and the nature of STDP trained spiking neural network, in order to insure better performance.

III. Network Architecture

The general architecture. In this paper we use a state-of-the-art convolutional SNN model from [18] which consists of feed-forward layers that contain IF neurons [23], and trained using the biological STDP learning rule [24]. An on-center/off-center filter is used to pre-process the data before latency coding is applied to transform this data into spikes. The threshold adaptation method described in [18] is used in order to maintain a state of homeostasis. The SNN we chose uses only one (convolution/pooling) stage for simplicity, as shown in (Fig. 1). This is because training multi-layer SNNs with STDP is still an open problem [3]. The objective of this paper is to focus on how spatio-temporal data can be pre-processed in order to feed a convolutional SNN with temporal information aggregated from a video sequence. We also explore early and late fusion techniques [21] in order to evaluate the benefits of such techniques in encoding spatio-

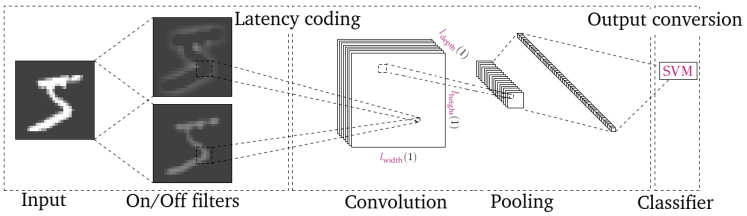

Fig. 1. Network topology (figure from [18]).

temporal information. The output of this network is a dense array that represents the processed sample, flattened as a linear array and introduced into a support vector machine (SVM) that is used to make the action classification. An SVM is used to classify the samples because we focus on the unsupervised learning of features. Any other supervised method can be used to do the classification, but we chose to use an SVM for its simplicity and its effectiveness.

Feature fusion. Temporal fusion is one way of aggregating the temporal information in a sequence of frames. In deep neural networks, there are multiple types of fusion [21], but in this paper, only early and late fusions are studied and implemented. They are a good fit to our study because we are using a single layer SNN model, which makes implementing other fusion methods that require multiple layers inapplicable, such as slow fusion [21]. Early and late fusion techniques operate differently and are implemented separately in this work. Early fusion is implemented by taking multiple samples and fusing them together row by row, into one big frame, as shown in the following equation: $I^o_{kj} = I^f_{ij}$ with $k = i * n + f$, $f \in [0, n-1], i \in [0, h-1], j \in [0, w-1]$, where I^f is the input frame of index f, and I^o is the output frame. On the other hand, late fusion is implemented by taking multiple flattened samples at the output of the network and concatenating them together. The main difference between early and late fusion is the stage at which the fusion takes place. In early fusion, shown in (Fig. 2), the sample frames are fused together before training the convolutional kernels. On the other hand, in late fusion, see (Fig. 3), the features that result from the processing of the video frames are fused together in the last stage. In the late fusion method implemented in this work, the samples are flattened and then joined together using a sequential queue.

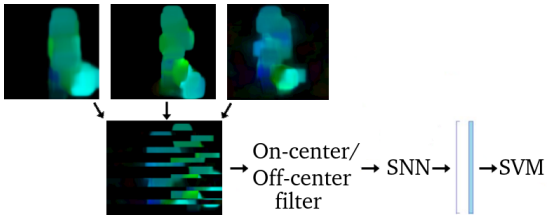

Fig. 2. Early fusion, multiple input frames are fused together.

The process. The general sequence preparation (SP) is a pre-processing procedure (see Fig. 4) that consists in applying background subtraction to every two consecutive frames, because it reduces the noise that can results from optical flow

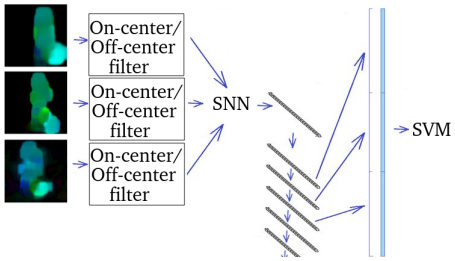

Fig. 3. Late fusion: the features obtained at the end of the last pooling layer are fused together before entering the SVM.

computation. After that, frames that do not contain significant motion are dropped. This is done by averaging the values of pixels and checking if this average is greater than a threshold estimated by trial and error, where each method has its own threshold. Two frames are also dropped between every two frames that are selected to be used in the sequence; this speeds up the action and helps recording it in a relatively smaller number of frames. Then, a pre-processing method based on Farneback's dense optical flow [25] is applied. Early fusion can be applied directly after implementing the optical flow representation, unless late fusion is going to be applied at the end of the process. Then, on-center/off-center filtering is applied and the data is introduced into the convolutional SNN.

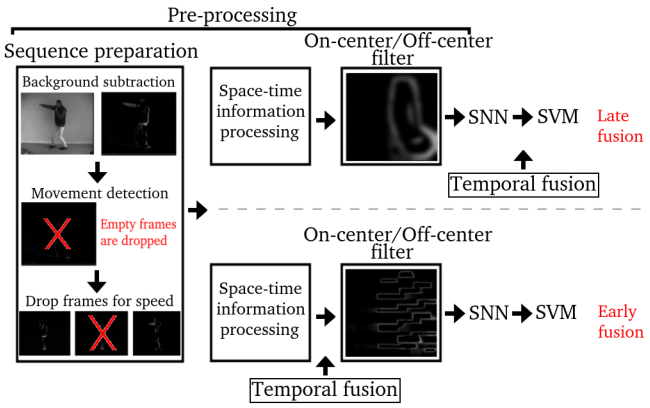

Fig. 4. The general pre-processing procedure.

IV. Space-Time Information Processing

In this section, we present different pre-processing methods for action recognition video datasets, and evaluate their suitability to spatio-temporal feature learning with STDP-based SNNs. We rely on deriving different data representations from sequences of static frames. We base our pre-processing on Farneback's dense optical flow because it is a reference method in motion representation. We define five representations that exploit various aspects of the optical flow: the horizontal and vertical displacement (DXDY), the Orientation and Amplitude (OA), the Composite Channel information (CC), the Edges Grid (EG) and the Motion Grid (MG). The initial processes described in Section III-C are used for all the pre-processing methods except the EG and MG methods.

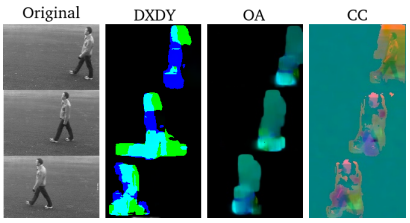

Fig. 5. A Walking action. (A) The original frame. (B) The DXDY representation (in RGB, G: D_x, B: D_y). (C) The OA representation(in HSV, H: orientation, V:amplitude). (D) The CC representation (in RGB, R: the moving part of the original grey-scale image G: D_x, B: D_y).

The DXDY representation. DXDY is made up of 2 channels, a horizontal displacement D_x in channel C_1 and a vertical displacement D_y in channel C_2. A sample produced by using this method is shown in (Fig. 5(B)).

The OA representation. OA separates optical flow vectors into orientation and magnitude values. The orientation data is periodic, therefore, it is difficult to apply latency coding to it. Thus, the information is displayed in the HSV color space, which is then converted into RGB color space (see Fig. 5(C)).

The CC representation. CC is created by combining the channels of the DXDY representation with the gray scale appearance information of the moving subject. A sample that results from this method contains three channels (see Fig. 5(D)). The first channel C_1 represents the horizontal displacement D_x, the second channel C_2 represents the vertical displacement D_y, and the third channel C_3 represents the original gray scale information of only the moving parts of the subject. The gray scale illumination of each pixel corresponds to the mean value of the channels in the original image, and pixels with significant motion are detected as follows: $|D_x| + |D_y| > \theta$, we use $\theta = 30$ in the experiments.

The EG representation. EG is based on extracting the edges of motion from optical flow frames using the Canny edge detection approach [26]. In spirit, the EG representation resembles motion boundary descriptors [27], except that we group these edges of motion into a grid as shown in Fig. 6(A). Each sample is constructed using 36 optical flow frames, and sample frame overlapping is used to increase the number of samples. Many different grid sizes were tested but 36 proved to be the most suitable value. This creates an early fusion of 36 frames, and therefore, the feature fusion methods in Section III-B are not applied with this method.

The MG representation. MG groups the movement information into a composite grid that is made up of 4×12 optical flow frames. Each frame is divided into 4 channels that are placed in separate frames one after the other as shown in (Fig. 6(B)). These channels are: the horizontal displacement to the left $|-D_x|$, the horizontal displacement to the right $+D_x$, the vertical displacement in the upwards direction $|-D_y|$, and the vertical displacement in the downwards direction $+D_y$, resulting in 16×12 channels per grid. This creates an early fusion of 48 frames, therefore, the feature fusion methods in Section III-B are also not applied with this method.

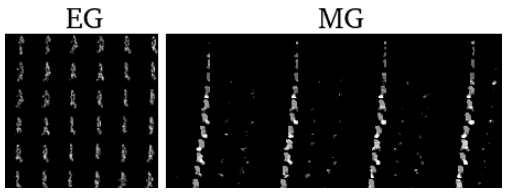

Fig. 6. A Walking action. (A) EG representation. (B) MG representation.

V. Evaluation

Data Sets. The KTH dataset [28] contains 600 videos of 25 subjects, performing 6 actions in 4 scenarios. The subjects 11, 12, 13, 14, 15, 16, 17 and 18 are used for training, while 02, 03, 05, 06, 07, 08, 09, 10 and 22 are used for testing, as indicated in the KTH protocol. The Weizmann dataset [29] contains 90 videos of 9 subjects performing 10 actions. The experiments on this dataset are done using the leave-one-out strategy. Each sample from both datasets has 10 frames prepared as described in Section III-C. This applies for all representations, except the MG and EG representations which need 48 and 36 frames per sample respectively.

Meta-parameters of the Model. The meta-parameters used in this work are presented in Table I. A difference-of-Gaussian (DoG) filter is used to simulate on-center/off-center cells. This filter creates a motion boundary effect [27] that increases the classification rates. Experiments with and without this filter were conducted, and experiments without this filter gave inferior results. Different numbers and sizes of filters were tested, but we only reported the most suitable values: 128 convolutional kernels of size 5 x 5, with a padding of 2 and a stride of 1. The convolutional SNN tested in this work is simulated using the falez-csnn-simulator [18].

Learning
$\alpha = 0.95$, $n_{epoch} = 100$
STDP
$W_{min} = 0.0$, $W_{max} = 1.0$, $\eta_{w(0)} = 0.1$, $\beta = 1.0$, $\tau_{STDP} = 0.1$, $w(0) \sim U(0,1)$
Neural Coding
$t_{exposition} = 1.0$
Threshold Adaptation
$t_{expected} = 0.95$, $\eta_\theta(0) = 1.0$, $th_{min} = 1.0$, $v_\theta(0) \sim G(5,1)$, $v_{inh} = 1.0$
Difference-of-Gaussian
$DoG_{center} = 1.0$, $DoG_{surround} = 4.0$, $DoG_{size} = 7.0$

TABLE I
THE META-PARAMETER VALUES USED IN THE EXPIREMENTS. SEE [18] FOR NOTATIONS.

Baseline. We cannot compare our evaluation with the state-of-the-arts because, to the best of our knowledge, none of the previous work in the literature use STDP with unsupervised learning for video classification as mentioned in Section II. Table II displays the classification rates obtained by training the convolutional SNN using raw video frames. Each sequence fused using early and late fusion is made up of 10 frames. The sequence preparation (SP) process mentioned in section (III-C) is applied. This table serves as a baseline in order to compare the effectiveness of the pre-processing techniques.

Dataset	KTH	Weizmann	KTH + SP	Weizmann + SP
No Fusion	19.54	18.88	34.84	**24.49**
Early Fusion	24.50	22.11	30.52	20.73
Late Fusion	26.38	21.03	**35.26**	23.58

TABLE II
CLASSIFICATION RATE IN % USING EARLY, LATE, AND NO FUSION WITH THE KTH AND WEIZMANN DATASETS AS RAW FRAMES, WITH AND WITHOUT SEQUENCE PREPARATION (SP).

Sequence preparation yields higher classification rates. This is because background subtraction removes some unnecessary spatial information, such as clothing and surrounding objects (see Fig. 4). Without sequence preparation, temporal fusion increases the classification rate with respect to no fusion. This is due to the SNN depending on the temporal information to classify the action. But early fusion decreases the classification rate after sequence preparation. This is because the SNN is no longer confused by the extra spatial information, and only considers the form of the action. Another reason is that fusion decreases the number of training samples by 10, since every 10 frames are processed as one sample.

Evaluation of the Five Representations. Preparing pre-processed samples from the action recognition videos using the DXDY, OA, CC and MG representations, yields the results displayed in Table III.

Dataset	KTH		Weizmann	
Fusion Method	Early	Late	Early	Late
DXDY	26.85	28.70	12.86	11.11
Orientation and Amplitude	41.05	45.83	**50.42**	44.44
Composite Channels	45.78	54.21	36.75	30.68
Edges Grid	63.01	-	43.83	-
Motion Grid	**77.69**	-	28.86	-

TABLE III
CLASSIFICATION RATE IN % USING EARLY FUSION AND LATE FUSION WITH THE KTH AND WEIZMANN DATASETS AS PRE-PROCESSED FRAMES.

DXDY depends only on the displacement information. The datasets contain some actions that are similar in form, and others that are similar in their amount of displacement. Thus, DXDY is not efficient enough to discriminate actions. The OA representation gives a slightly higher classification rate. With this method, the SNN is able to learn that the features of the first set of three actions Boxing, Clapping, and Waving are different from the other set of three actions Jogging, Walking, and Running (see Fig. 7), but actions that are relatively similar in form are not well discriminated. The biggest confusion was recorded between the Jogging and Running actions, which are similar. The same applies to the Weizmann dataset, where the actions Bend and Jack are well differentiated, while there is confusion between similar actions such as Walk and Run. The CC representation aims to add spatial information to the movement information, and shows a slightly better performance than the OA method in the case of KTH dataset.

The MG outperforms all the other pre-processing methods when implemented with the KTH dataset (see Table III). This is because it densely represents the temporal movement

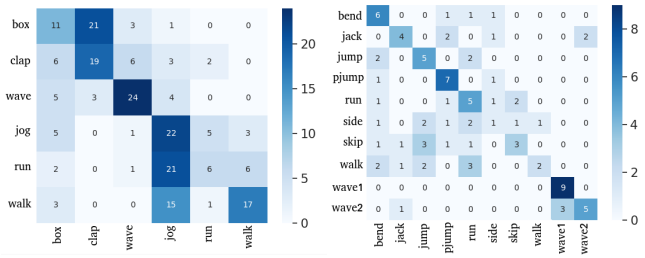

Fig. 7. Confusion Matrix of the (A) KTH dataset and (B) Weizmann dataset pre-processed with the OA method and introduced into the SNN.

information. On the other hand, the performance of this method on the Weizmann dataset is poor, which is due to the lack of training samples. The Weizmann dataset videos are scarcer and have shorter durations than the KTH ones, and each MG sample requires 48 frames, thus not enough training samples can be generated from the Weizmann dataset using this method. To highlight this, we increased the number of training samples with the MG representation from 110 to 220 by horizontally flipping the videos. Then we added Gaussian noise to the set of frames, thus doubling the number of samples again, to 440 training samples. The classification rate increased to 45%. We also experimented with decreasing numbers of samples on the KTH dataset, and the accuracy decreased accordingly. Finally, the EG representation gives an inferior classification rate to the MG representation when using the KTH dataset. On the other hand, using the Weizmann dataset with the EG representation gives a higher classification rate than the MG representation. This is because fusing 36 frames per sample along with sample overlapping generates more samples than fusing 48 frames per sample: here again, the number of training samples in critical in reaching good performances. It is important to note that we also tested the MG method applied on the KTH dataset with a regular CNN. This test gave a classification rate of (77%), which is similar to that obtained with our SNN. Although the MG does not give results that are state-of-the-art in comparison to the classification rates obtained by other methods in testing human action recognition datasets with ANNs, it does show potential in understanding how SNNs may handle spatio-temporal information.

VI. DISCUSSION

Using multiple pre-processing techniques in addition to early and late fusion methods helps in understanding how SNNs can achieve human action recognition. The velocity distribution of the moving components in the videos and the shapes of these components are two very important aspects in action classification. When using a pre-processing method that clearly highlights these two aspects, the spiking neural network was able to reach higher classification rates. The MG grid is able to represent at least one complete cycle of the action being performed with the KTH dataset, thus forming a more complete data representation. Spiking neural networks are still not able to achieve the recognition rates that

regular convolutional neural networks can achieve with action classification tasks, and therefore more research needs to be conducted in this field. As a result it may be a good idea to implement the two-stream method [20] which is similar in spirit to the CC representation, except that in this method, spatial information has an entire dedicate stream, while in the CC method, the spatial information is processed by the same neurons. Another idea would be to implement the same experiments with a multi-layer spiking neural network, which is a very challenging task [3].

VII. Conclusion

This study was carried out in order to give an assessment of the effect of different data representations on spatio-temporal feature learning. The result of testing these representations on an SNNs trained with STDP yields several conclusions. The first conclusion is that the spatial information improves the classification rate with respect to using the displacement information alone. The second conclusion is that the best action classification rate is recorded when there is at least a full cycle of motion, like in the case of the MG representation with KTH dataset videos. The same MG representation gave inferior results using the Weizmann dataset, because the videos are not long enough to fill this grid with multiple cycles of motion (some videos contain only one action, others contain two, etc.), or to create enough samples. Testing the MG representation with regular 2D CNNs gives a similar recognition rate, which proves that a suitable pre-processing method can help bridge the gap between SNNs and CNNs. The MG shows an improvement compared to the other methods of pre-processing, and serves as a good starting point in improving human action recognition with SNNs.

Acknowledgments

This work has been partially funded by IRCICA (USR 3380) under the bio-inspired project.

References

[1] S. Ghosh-Dastidar and H. Adeli, "Third generation neural networks: Spiking neural networks," in Advances in Computational Intelligence, W. Yu and E. N. Sanchez, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 167–178.

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

[3] P. Falez, P. Tirilly, I. Marius Bilasco, P. Devienne, and P. Boulet, "Multi-layered spiking neural network with target timestamp threshold adaptation and stdp," in 2019 International Joint Conference on Neural Networks (IJCNN), 2019, pp. 1–8.

[4] Z. Bing, I. Baumann, Z. Jiang, K. Huang, C. Cai, and A. Knoll, "Supervised Learning in SNN via Reward-Modulated Spike-Timing-Dependent Plasticity for a Target Reaching Vehicle," Frontiers in Neurorobotics, vol. 13, 2019.

[5] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," Frontiers in Neuroscience, vol. 12, 2018.

[6] B. Rueckauer, I. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," Frontiers in Neuroscience, vol. 11, 2017.

[7] P. Falez, P. Tirilly, I. M. Bilasco, P. Devienne, and P. Boulet, "Mastering the output frequency in spiking neural networks," in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8.

[8] P. Falez, P. Tirilly, I. M. Bilasco, P. Devienne, and P. Boulet, "Unsupervised visual feature learning with spike-timing-dependent plasticity: How far are we from traditional feature learning approaches?" Pattern Recognition, vol. 93, p. 418–429, 2019.

[9] A. V. Gavrilov and K. O. Panchenko, "Methods of learning for spiking neural networks. a survey," in 2016 13th International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), vol. 02, 2016, pp. 455–460.

[10] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," Frontiers in Neuroscience, vol. 14, 2020.

[11] A. Kugele, T. Pfeil, M. Pfeiffer, and E. Chicca, "Efficient Processing of Spatio-Temporal Data Streams With Spiking Neural Networks," Frontiers in Neuroscience, vol. 14, 2020.

[12] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "Span: Spike pattern association neuron for learning spatio-temporal spike patterns," International journal of neural systems, vol. 22, p. 1250012, 2012.

[13] Y. Meng, Y. Jin, J. Yin, and M. Conforth, "Human activity detection using spiking neural networks regulated by a gene regulatory network," The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–6, 2010.

[14] C. Lee, P. Panda, G. Srinivasan, and K. Roy, "Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning," Frontiers in Neuroscience, vol. 12, 2018.

[15] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," Neural Networks, vol. 99, p. 56–67, 2018.

[16] J. Berlin and M. John, "R-stdp based spiking neural network for human action recognition," Applied Artificial Intelligence, vol. 34, pp. 1–18, 2020.

[17] Z. Bing, I. Baumann, Z. Jiang, K. Huang, C. Cai, and A. Knoll, "Supervised Learning in SNN via Reward-Modulated Spike-Timing-Dependent Plasticity for a Target Reaching Vehicle," Frontiers in Neurorobotics, vol. 13, 2019.

[18] P. Falez, "Improving spiking neural networks trained with spike timing dependent plasticity for image recognition," Ph.D. Thesis, Université de Lille, 2019.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[20] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 568–576.

[21] C. Wang, "Dual temporal scale convolutional neural network for micro-expression recognition," Frontiers in Psychology, vol. 8, p. 1745, 2017.

[22] Y. Wang, M. Long, J. Wang, and P. S. Yu, "Spatiotemporal pyramid network for video action recognition," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2097–2106.

[23] A. Burkitt, "A review of the integrate-and-fire neuron model: I. homogeneous synaptic input," Biological cybernetics, vol. 95, pp. 1–19, 2006.

[24] G.-q. Bi and M.-m. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," Journal of Neuroscience, vol. 18, no. 24, pp. 10 464–10 472, 1998.

[25] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in Image Analysis, J. Bigun and T. Gustavsson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370.

[26] J. Canny, "A computational approach to edge detection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. PAMI-8, pp. 679 – 698, 12 1986.

[27] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," International Journal of Computer Vision, vol. 103, 2013.

[28] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03, ser. ICPR '04. USA: IEEE Computer Society, 2004, p. 32–36.

[29] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2247–2253, December 2007.