Event-based Action Recognition Using Motion Information and Spiking Neural Networks

Qianhui Liu[1,2] , **Dong Xing**[1,2] , **Huajin Tang**[1,2] , **De Ma**[1*] and **Gang Pan**[1,2*]
[1]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[2]Zhejiang Lab, Hangzhou, China
{qianhuiliu, dongxing, htang, made, gpan}@zju.edu.cn

Abstract

Event-based cameras have attracted increasing attention due to their advantages of biologically inspired paradigm and low power consumption. Since event-based cameras record the visual input as asynchronous discrete events, they are inherently suitable to cooperate with the spiking neural network (SNN). Existing works of SNNs for processing events mainly focus on the task of object recognition. However, events from the event-based camera are triggered by dynamic changes, which makes it an ideal choice to capture actions in the visual scene. Inspired by the dorsal stream in visual cortex, we propose a hierarchical SNN architecture for event-based action recognition using motion information. Motion features are extracted and utilized from events to local and finally to global perception for action recognition. To the best of the authors' knowledge, it is the first attempt of SNN to apply motion information to event-based action recognition. We evaluate our proposed SNN on three event-based action recognition datasets, including our newly published *DailyAction-DVS* dataset comprising 12 actions collected under diverse recording conditions. Extensive experimental results show the effectiveness of motion information and our proposed SNN architecture for event-based action recognition.

1 Introduction

Event-based cameras are a novel class of vision devices imitating the mechanism of human retina. Contrary to conventional cameras, which record the visual input from all pixels as images at a fixed rate, with event-based cameras, each pixel individually emits events when it observes sufficient changes of light intensity in its receptive field. Thus, event-based cameras naturally respond to moving objects and ignore static redundant information, resulting in significant reduction of memory usage and energy consumption. The final output of the camera is a stream of events collected from each pixel, forming an asynchronous and sparse representation of the scene.

This event-based representation is inherently suitable to cooperate with the spiking neural network (SNN) since SNN also has the event-based property [Hu *et al.*, 2018]. SNN uses discrete spikes to transmit information between units which mimics the behavior of biological neural systems. Benefiting from this event-driven processing paradigm, SNN is energy efficient on neuromorphic hardware and has a powerful ability in processing spatio-temporal information. Recent years, SNN has been increasingly applied to the task related to event-based cameras.

Existing works of SNNs cooperating with event-based cameras mainly focus on the object recognition tasks [Orchard *et al.*, 2015; Xiao *et al.*, 2019; Liu *et al.*, 2020b]. However, since the event-based camera naturally captures movements in the visual scene, it is a good fit for the action recognition task. Nevertheless, works of SNN on event-based action recognition are still limited.

Humans can recognize actions accurately, which motivates us to explore the biological visual cortex to gain experience for event-based recognition. The visual cortex is organized in two different pathways [Jhuang *et al.*, 2007]. One is a ventral stream dealing with shape information, which has been widely used in the existing spiking object recognition models [Orchard *et al.*, 2015; Liu *et al.*, 2020b]. The other is a dorsal stream involved with the analysis of motion information. Since the event streams representing the actions contain rich motion information, motion features of event stream may be an ideal choice for action recognition tasks. Further, the organization of dorsal stream is hierarchical. Neurons gradually increase their receptive field along the hierarchy, as well as their selectivity and invariance to features. Inspired by the current theory of the visual cortex, we will make steps towards the solution of event-based action recognition.

We propose a hierarchical SNN architecture for event-based action recognition using motion information. Motion features are extracted and utilized from events to local and finally to global perception for action recognition. Specifically, we first adopt motion-sensitive neurons to estimate optical flow for the purpose of local motion (direction and speed) perception. Then, we perform a motion pooling and a spatial pooling to mitigate the effect of the aperture problem [Orchard *et al.*, 2013] and increase the spatial invariance

[*] Corresponding author

respectively. As the final stage of the architecture, a SNN classifier in which the spiking neurons are fully connected to the previous pooling layer, is adopted as global perception for recognition results. To the best of the authors' knowledge, it is the first attempt of SNN to apply the motion information to action recognition tasks.

Besides, due to the lack of event-based action datasets and their importance for the algorithm development, we present a new event-based action recognition dataset called *DailyAction-DVS*. The dataset comprises 15 subjects performing 12 daily actions under 2 lighting conditions and 2 camera positions (with different distances and angles to the subjects). This setting increases the challenge of dataset while gaining more practical significance.

We evaluate the proposed SNN on the new event-based action recognition dataset and two other challenging ones. Experimental results show the effectiveness of motion information and our proposed SNN architecture for event-based action recognition.

2 Related Work

2.1 Event-based Action Recognition

Action recognition task has drawn a significant amount of attention from the academic community, owing to its applications in many areas like security and behavior analysis. With the popularity of event-based cameras, they have been found to be ideal choices to capture human actions since they only record the activity in the field of view and automatically partition the foreground and background. Recently, research on event-based action recognition has emerged progressively. One approach is to convert the output of an event camera into frames and use standard computer vision methods, such as [Innocenti *et al.*, 2020]. However, these works mainly focus on how to aggregate events and deal with frames. Another approach is to directly deal with events. [Maro *et al.*, 2020] introduced a framework for dynamic gesture recognition relying on the concept of time-surfaces introduced in [Lagorce *et al.*, 2017]. In addition, SNN is trying to solve the event-based gesture recognition. [George *et al.*, 2020] presented a SNN which uses the idea of convolution and reservoir computing in order to classify human hand gestures. Since SNN has the event-based property and the ability in processing spatio-temporal information, it has great potential to solve the event-based action recognition, but related works are still limited.

2.2 Event-based Features

[Lagorce *et al.*, 2017] proposed the spatio-temporal features based on recent temporal activity of events within a local spatial neighborhood called time-surfaces. [Sironi *et al.*, 2018] presented local memory time surfaces to leverage past temporal information and improve robustness. [Ramesh *et al.*, 2019] encoded the structural context using log-polar grids for event stream, which is robust to moderate scale and rotation variations. Inspired by the function of ventral stream in visual cortex, [Orchard *et al.*, 2015] proposed a spiking architecture to extract HMAX-based shape features for event-based object recognition. This work pioneers a series of related works [Xiao *et al.*, 2019; Liu *et al.*, 2020a; 2020b] and the proposed features become one of the most commonly used features when using SNN to process the event-based recognition.

We are inspired by the function of dorsal stream also in visual cortex and make steps towards using motion information for action recognition tasks. Since event-based cameras provide an efficient way for encoding light and its temporal variations [Benosman *et al.*, 2012], we introduce the optical flow estimation for motion (direction and speed) perception. Existing works on SNN-based optical flow estimation adopted motion-sensitive neurons with synaptic delays [Orchard *et al.*, 2013; Paredes-Vallés *et al.*, 2019]. We here adopt neurons in [Orchard *et al.*, 2013] due to their effectiveness and simplicity.

2.3 Event-based Datasets

Existing datasets on event-based action recognition can be divided into two categories: one is recorded with a static event-based camera facing a monitor on which video-based datasets were set to play automatically [Hu *et al.*, 2016]. However, this recording way will lose real dynamics of moving objects between two frames. There is no guarantee that a method tested on this kind of artificial data will behave similarly in real-world conditions [Sironi *et al.*, 2018]. The other is to record directly by event-based cameras in the real scene. Among them, several datasets are proposed for gestures. [Amir *et al.*, 2017] proposed an event-based hand gesture dataset captured by a fixed DVS camera [Lichtsteiner *et al.*, 2008]. [Maro *et al.*, 2020] also proposed a gesture dataset but was recorded by an ATIS camera [Posch *et al.*, 2011] connected to the smartphone. As for human action, [Miao *et al.*, 2019] proposed an event-based action recognition dataset using a DAVIS camera [Brandli *et al.*, 2014] with 3 different positions. However, this dataset is recorded under single light condition and is relatively small (291 recordings released).

Our proposed event-based action recognition *DailyAction-DVS* dataset has 1440 recordings of 12 daily actions. The dataset is captured by a DVS camera with 2 different lighting conditions and 2 different camera positions (with different distances and angles), which brings more challenges to the dataset and is also more in line with the realistic situation.

3 Method

In this section, we introduce the proposed SNN for event-based action recognition, which extracts and utilizes motion information from events to local and finally to global perception. The architecture of the proposed SNN is shown in Figure 1.

3.1 Events From Event-based Camera

Given an event-based camera with pixel grid size $N \times M$, the i-th event can be described as:

$$e_i = [e_{t_i}, e_{x_i}, e_{y_i}, e_{p_i}], \quad i \in \{1, 2, \dots, I\} \tag{1}$$

where $e_{t_i} \geq 0$ is the timestamp at which the event is generated, $(e_{x_i}, e_{y_i}) \in \{1, 2, \dots, N\} \times \{1, 2, \dots, M\}$ is the position of the pixel generating the i-th event, $e_{p_i} \in \{-1, 1\}$

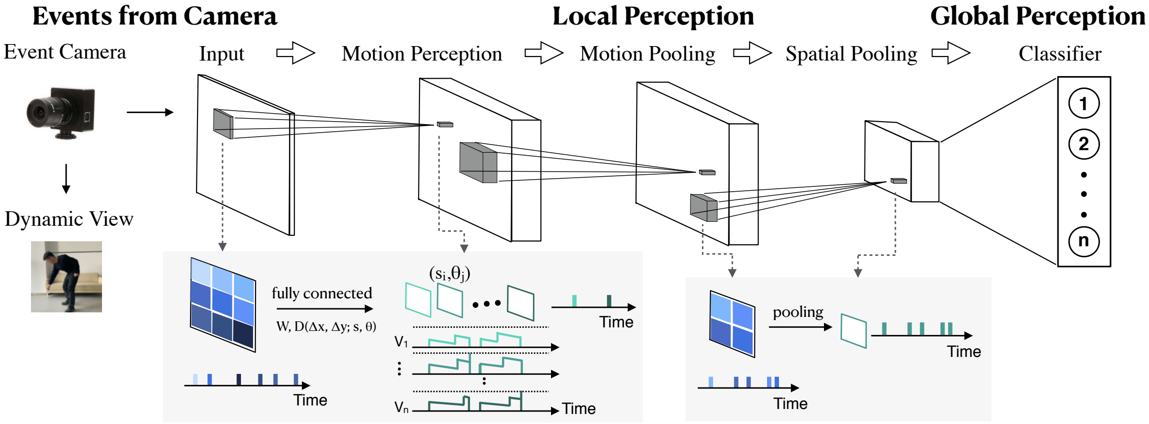

Figure 1: The architecture of the proposed SNN for event-based action recognition. The network consists of 5 layers: the first layer is the *Input* layer, the following 3 layers are used for local perception, and the last layer provides global perception for recognition results. Events from the event-based camera are first encoded in *Input* layer to a compatible format and then sent to *Motion Perception* layer *(MP1)*. *MP1* consists of motion-sensitive neurons and neurons of the same sensitive motion (direction and speed) are organized into one neural map. If the membrane voltage of neuron exceeds its threshold, the neuron will fire a spike to *Motion Pooling* layer (*MP2*) and all neurons in the same position of different neural maps will be reset. Neurons in *MP2* have the same motion sensitivity but a larger receptive field. Neurons in *Spatial Pooling* layer (*SP*) fuse spikes from their receptive field and transmit them to the *Global Perception* layer (*GP*). Finally, a SNN classifier, in which the spiking neurons are fully connected with *SP*, receives all the extracted motion feature spikes and outputs action recognition results.

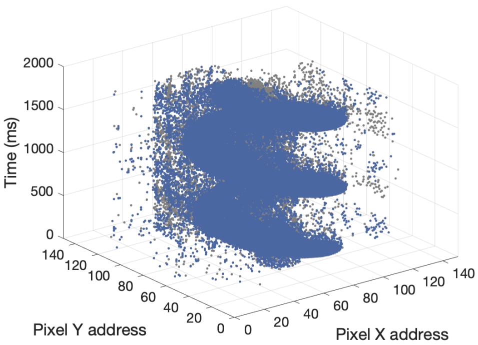

Figure 2: Visualization of one event stream representing the *left arm clockwise* in *DvsGesture* dataset. ON and OFF events are represented by blue and gray respectively.

is the polarity of the event, with $-1, 1$ meaning respectively OFF and ON events, and I is the number of events. Figure 2 shows a visualization of the event stream representing *left arm clockwise* gesture in *DvsGesture* dataset [Amir *et al.*, 2017].

The events from camera are first encoded in *Input* layer to a compatible format for the following processing. This layer can been seen as having two neural maps, one per polarity. Each map is comprised of $N \times M$ spiking neurons with no internal dynamics that emit spikes when receiving the corresponding events.

3.2 Local Perception

Motion Perception (*MP1*)

The events are sent to *Motion Perception* layer (*MP1*) for local motion estimation. The synaptic connections between Input and *MP1* have transmission delay, which is related to motion that the postsynaptic neurons are sensitive to. As in [Orchard *et al.*, 2013], we specify the delay as a function of speed s and direction θ to which the neuron is tuned as well as the position of the pixel relative to the neuron. The function of delay can be described with the following equation:

$$D(\Delta x, \Delta y; s, \theta) = -\frac{\Delta x cos\theta + \Delta y sin\theta}{s} \quad (2)$$

where Δx and Δy are the spatial offsets between the neuron position (x,y) and the event address (e_x, e_y). Considering both the coverage of various directions and speeds and the complexity of implementing the algorithm, we set the directions θ varing in increment of 45 degrees and speeds s varing by a factor of 2. Neurons of the same sensitive direction and speed are organized into one neural map. On each map, the membrane voltage of the neuron at position (x, y) and time t can be described as:

$$V(x, y, t) = \sum_i \mathbb{1}\{x \in \mathcal{X}(e_{x_i})\}\mathbb{1}\{y \in \mathcal{Y}(e_{y_i})\} \\ W(-\frac{t - e_{t_i}}{\tau_m}) \quad (3)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\mathcal{X}(e_{x_i}) = [e_{x_i} - r_{mp1}, e_{x_i} + r_{mp1}]$ and $\mathcal{Y}(e_{y_i}) = [e_{y_i} - r_{mp1}, e_{y_i} + r_{mp1}]$ denote the receptive field of the neuron in this layer, r_{mp1} denotes the receptive field size, W denotes the synaptic weight and τ_m denotes the decay time constant.

Each spiking neuron has equally weighted connections to $r_{mp1} \times r_{mp1}$ neurons in the previous layer, but the connections have specific delays such that when neuron's sensitive motion (direction and speed) pattern of events emits, all these events will arrive at the neuron in a small time interval and trigger the membrane voltage of neurons to respond. When the neuron voltage exceeds its threshold V^{thr}_{mp1}, the neuron will fire a spike. The threshold V^{thr}_{mp1} is set according to the

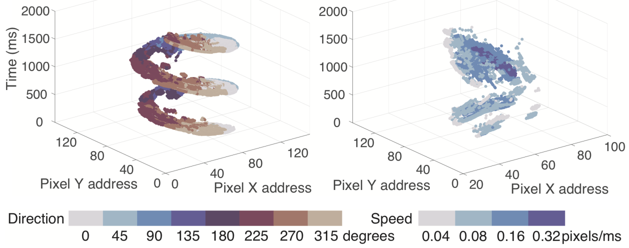

Figure 3: Visualization of the extracted motion feature spikes. *Left*: direction response of one *left arm clockwise* sample in *DvsGesture* dataset. *Right*: speed response of one *bend* sample in *DailyAction-DVS* dataset. The color bar shows the direction or speed represented by each color.

sum of weights of input events in the receptive field. In this way, when stimulus inputs, the neuron which most closely matches the stimulus will most likely fire first. Meanwhile, the spike will inhibit neurons in other neural maps sharing the same position and force them to reset. This Winner-Take-All (WTA) mechanism will eliminate erroneous feature spikes when the next stimulus passes.

Motion Pooling (*MP2*)

Spiking neurons in this layer have the same neural dynamics and connection delays as the *Motion Perception* layer, but a larger receptive field size r_{mp2} ($r_{mp2} > r_{mp1}$). Large receptive field size increases the probability that edges in different directions will be included in the receptive field [Orchard *et al.*, 2013]. Therefore, local motion pooling mitigates the aperture problem and provides accurate motion features of event streams.

Figure 3 shows the motion feature spikes of *left arm clockwise* in *DvsGesture* dataset [Amir *et al.*, 2017] and *bend* in *DailyAction-DVS* dataset. We can see in *left arm clockwise*, the direction of motion gradually changes over time. In *bend*, the middle part which represents head trajectory, is faster than the side part which represents body trajectory. These observations are consistent with our experience.

Spatial Pooling (*SP*)

Before the global perception, we here employ the spatial pooling in the SNN architecture, aiming to increase the feature spatial invariance and reduce the spatial dimensionality. Neural maps in the previous *Motion Pooling* layer are divided into adjacent non-overlapping $r_{sp} \times r_{sp}$ regions and feature spikes emitted from the same region will be transmitted to the same neurons in this layer. Neurons have no specific dynamics and emit spikes when receiving the spikes in their receptive field.

3.3 Global Perception (*GP*): SNN Classifier

In this section, we will describe how we perform global perception for the final action recognition result. The spiking neurons in this layer are fully connected to all neurons in the previous *Spatial Pooling* layer. The connection weights are trained by the Segmented Probability-Maximization (SPA) [Liu *et al.*, 2020b], an effective SNN learning algorithm that is specifically designed for processing event-based data.

We employ the Leaky Integrate-and-Fire (LIF) model. The neural dynamics can be described as:

$$V(t) = \sum_i w_i \sum_{t_i} K(t - t_i) + V_{rest} \tag{4}$$

where w_i and t_i are the synaptic weight and the firing time of the afferent i. V_{rest} denotes the resting potential of the neuron. K is the normalized postsynaptic potential (PSP) kernel which is defined as follows:

$$K(t - t_i) = V_0(\exp(\frac{-(t - t_i)}{\tau_m}) - \exp(\frac{-(t - t_i)}{\tau_s})) \tag{5}$$

where V_0 denotes the coefficient to normalize the maximum value of the kernel as 1, τ_m and τ_s denote decay time constants of membrane integration and synaptic currents respectively.

The SPA learning algorithm aims to train the connection weights so that the neurons can respond more active to the input patterns of the class they represent. According to the input pattern of class c_k and the class j neuron represent, the weights should be updated in the following way. First, we find the peak membrane voltage V^j_{peak} of the neuron representing class j and label the corresponding time stamp as t_{peak}. Second, we define the normalized output firing rate f^j_{out} of the neuron representing class j as:

$$f^j_{out} = \log(\exp(V^j_{peak}) + 1) \tag{6}$$

Third, we update the weights using the equation:

$$\Delta w_i = \begin{cases} \lambda(f'_{out})^j \frac{f_{sum} - f^j_{out}}{f_{sum} f^j_{out}} \sum\limits_{t_i < t^j_{peak}} K(t^j_{peak} - t_i) & j = c_k \\ -\lambda(f'_{out})^j \frac{1}{f_{sum}} \sum\limits_{t_i < t^j_{peak}} K(t^j_{peak} - t_i) & j \neq c_k \end{cases} \tag{7}$$

where λ is the learning rate. We use f_{sum} to denote $\sum_{j'=1}^{n} f^{j'}_{out}$ for convenience. When training is done, we keep the synaptic weights fixed, and set the threshold of neurons V^{gp}_{thr}. The predicted class for the input is determined by averaging the firing rates of neurons per class and then choosing the class with the highest average firing rate.

4 Experimental Results

In this section, we evaluate the performance of our proposed SNN on three event-based gesture/action recognition datasets and compare it with other SNN methods.

4.1 Datasets

We analyze the performance of our SNN on three event-based datasets, i.e., our proposed *DailyAction-DVS* dataset[1], publicly available *DvsGesture* dataset and *Action Recognition* dataset. Figure 4 shows some samples of these three datasets.

DailyAction-DVS dataset: It comprises 1440 recordings of 15 subjects acting 12 different actions, including *bend, climb, fall down, get up, jump, lie down, carry box, run, sit*

[1] https://github.com/qianhuiliu/SNN-action-recognition

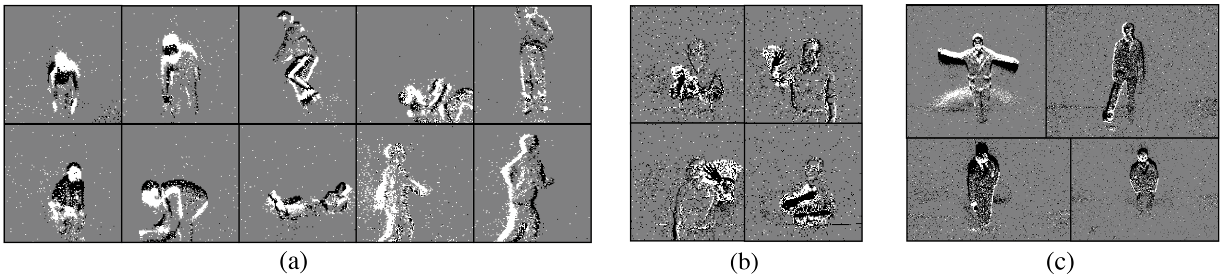

Figure 4: Sample snapshots from the used datasets. Black and white pixels represent OFF and ON events, respectively. (a) Our proposed *DailyAction-DVS* dataset; (b) *DvsGesture* dataset; (c) *Action Recognition* dataset.

	Method	DailyAction-DVS	DvsGesture	ActionRecognition
[Amir *et al.*, 2017]	Deep SNN (16 layers)	-	91.8%	-
[Shrestha and Orchard, 2018]	Deep SNN (8 layers)	-	93.6%	-
[Gu *et al.*, 2019]	Deep SNN (6 layers)	-	-	71.2%
[Xiao *et al.*, 2019]	HMAX-based SNN	68.3%	-	55.0%
[Xing *et al.*, 2020]	Conv-RNN SNN (5 layers)	-	92.0%	-
[George *et al.*, 2020]	Conv+Reservoir SNN	-	65.0%	-
[Liu *et al.*, 2020b]	HMAX-based SNN	76.9%	70.1%	-
This Work	Motion-based SNN	90.3%	92.7%	78.1%

Table 1: Comparison of recognition accuracy on three datasets.

down, stand up, walk and *pick up*. A DVS camera [Lichtsteiner *et al.*, 2008] was placed at 2 positions that had different distances and angles to the subject, as shown in Figure 5. The actions were captured under 2 lighting conditions including *natural light* and *LED light*. Each subject performed each action under the same camera position and lighting condition. The duration of each recording is within $6s$.

The actions we choose in this dataset are common in daily life, thus we call it *DailyAction-DVS*. With the diverse recording conditions, the dataset gains more practical significance. Notice that some actions are potentially dangerous (such as fall down, climb), this dataset may also contribute to detecting whether someone is in danger in the environment.

DvsGesture dataset [Amir *et al.*, 2017]: It comprises 1342 recordings of 29 subjects performing 11 different actions (including one rejected class with random gestures) under 3 different lighting conditions. The DVS camera [Lichtsteiner *et al.*, 2008] was mounted on a stand while the subjects stood still in front of it performing gestures.

Action Recognition dataset [Miao *et al.*, 2019]: It has released 291 effective recordings of 15 subjects acting 10 different actions. The DAVIS camera [Brandli *et al.*, 2014] was set to 3 positions to the subject for recording each action.

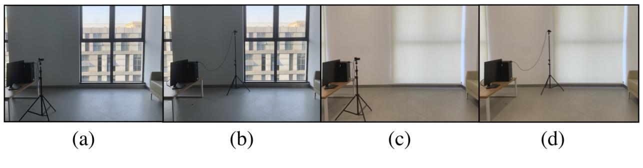

Figure 5: Environmental setup of *DailyAction-DVS* dataset.

4.2 Performance on Event-based Action Recognition Tasks

On DailyAction-DVS Dataset

Table 1 shows our model achieves the recognition accuracy of 90.3% on average. Although the diverse recording conditions of this dataset increase the difficulty of recognition, our model outperforms [Xiao *et al.*, 2019] and [Liu *et al.*, 2020b] by 22.0% and 13.4%, respectively. This suggests that our model is robust against these environmental variances and can be generalized to more realistic scenarios. Notice that [Xiao *et al.*, 2019], [Liu *et al.*, 2020b] and our model all employ a single-layer classifier, but the adopted features are different. The results indicate that motion features used in our model are more effective than the HMAX-based shape features used in [Xiao *et al.*, 2019] and [Liu *et al.*, 2020b]. This suggests that in the action recognition task, how the action moves is potentially more important than what the action looks like.

On DvsGesture Dataset

Our model achieves the recognition accuracy of 92.7% on average. Table 1 shows that our model outperforms [George *et al.*, 2020] and [Liu *et al.*, 2020b] by a large margin of 27.7% and 22.6% respectively. In addition, our model (with one trained layer) also achieves better performance than some deep SNNs. The recognition accuracy of our model is 0.9% and 0.7% higher than [Amir *et al.*, 2017] (with 16 trained layers) and [Xing *et al.*, 2020] (with 5 trained layers). This indicates that the motion features extracted in our model are representative enough to compare with those of deep SNNs. Moreover, they are more light-weighted, making it more efficient in computation.

On Action Recognition Dataset

The recognition task on this dataset is relatively more challenging because of its limited training samples (about 5 times fewer than the other two datasets), therefore the recognition results of all the compared methods on this dataset are relatively poor. Nevertheless, our model achieves the recognition accuracy of 78.1%, which is higher than [Gu *et al.*, 2019] and [Xiao *et al.*, 2019]. Notice that [Gu *et al.*, 2019] reaches a lower accuracy of 71.2% with a 6-layer fully connected SNN. The reason is that [Gu *et al.*, 2019] is not designed for event-based data that has high temporal resolution and sparse representation. Thus, it will lose the precise temporal information which is critical for event-based action recognition.

4.3 Ablation Study of Proposed SNN

In this section, we present the ablation study of our proposed SNN on *DailyAction-DVS* and *DvsGesture* datasets. Based on the full model, we bypass *Spatial Pooling* layer (*SP*), *Motion Pooling* layer (*MP2*) and *Motion Perception* and *Pooling* layers (*MP1&2*) to verify their effects on the original architecture. Table 2 reports the recognition accuracy and the number of activated synapses in each layer under different settings.

We can observe from Table 2 that for the setting of our proposed SNN bypassing *SP* layer, the recognition accuracy on two datasets drops by 0.7% and 0.8% respectively, meanwhile the required computation is increased to nearly three times. This indicates that *SP* layer contributes to a more compact feature format, which is beneficial to both the accuracy and computation. We can observe a similar phenomenon when bypassing *MP2* layer. For example, on *DailyAction-DVS* dataset, the average number of activated synapses increases to 1.3 and 1.6 times for *SP* and *GP* respectively. This indicates that *MP2* layer filters out redundant and erroneous motion feature spikes. For the setting of our proposed SNN bypassing *MP1* and *MP2* layers, a minimum amount of computation is required since there are no motion-sensitive neurons used to extract motion information. However, the recognition accuracy drops significantly, which in turn indicates the effectiveness of motion information for action recognition.

Task	Acc.	Input Synapse Activations			
		MP1	*MP2*	*SP*	*GP*
DailyAction-DVS					
Full model	90.3%	8.0k	4.0k	3.2k	229.5k
Bypass *SP*	89.6%	8.0k	4.0k	-	610.4k
Bypass *MP2*	90.0%	8.0k	-	4.0k	364.3k
Bypass *MP1&2*	58.1%	-	-	8.0k	216.6k
DvsGesture					
Full model	92.7%	8.8k	2.7k	2.8k	313.3k
Bypass *SP*	91.9%	8.8k	2.7k	-	913.6k
Bypass *MP2*	92.1%	8.8k	-	2.7k	415.5k
Bypass *MP1&2*	57.0%	-	-	8.8k	157.8k

Table 2: Recognition accuracy and required computation

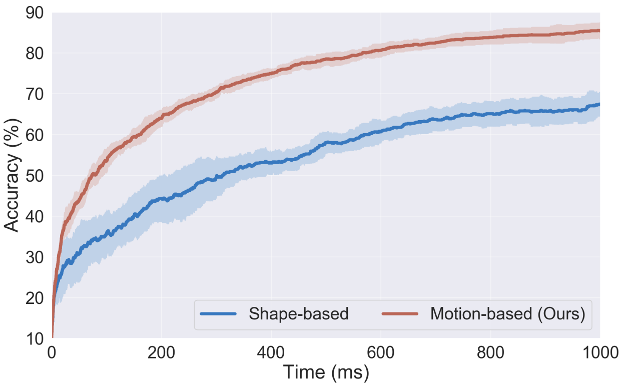

Figure 6: Performance comparison between two kinds of features on *DailyAction-DVS* dataset.

4.4 Effectiveness of Motion Information For Action Recognition

In this section, we validate the effectiveness of our extracted motion features for event-based action recognition. We compare our motion-based features with the shape-based features, which have been widely adopted in the previous works [Orchard *et al.*, 2015; Xiao *et al.*, 2019; Liu *et al.*, 2020b] and employ SPA classifier [Liu *et al.*, 2020b] for both features. The experiments are conducted on *DailyAction-DVS* dataset. We use the full recordings (within $6000ms$) for training and observe the performance of each method within the first $1000ms$ recordings.

As the event in the recording flows in, the recognition accuracy of two features keeps increasing. Nevertheless, the motion-based method has a higher performance than the shape-based one. We also notice that within the first $400ms$, when the input information is extremely incomplete to describe the actions, our motion-based method also has a higher growth rate and smaller variance. The result indicates that our extracted motion features are more representative to actions, compared to HMAX-based shape features. Therefore, we conclude that motion information is more effective for action recognition.

5 Conclusion

In this paper, we propose a SNN architecture for event-based action recognition, which utilizes the motion information hierarchically from events to local and finally to global perception. The ablation study validates our proposed SNN architecture. This work is the first attempt of SNN to apply motion information to event-based action recognition. Experimental results show that this attempt is not only feasible but also effective.

Acknowledgments

This work is supported by the Natural Science Foundation of China (No. 61925603), the Key Research and Development Program of Zhejiang Province in China (2020C03004) and Zhejiang Lab. Qianhui Liu is partly supported by the Zhejiang Lab's International Talent Fund for Young Professionals.

References

[Amir *et al.*, 2017] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, et al. A low power, fully event-based gesture recognition system. In *CVPR*, pages 7243–7252, 2017.

[Benosman *et al.*, 2012] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012.

[Brandli *et al.*, 2014] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.

[George *et al.*, 2020] Arun M George, Dighanchal Banerjee, Sounak Dey, Arijit Mukherjee, and P Balamurali. A reservoir-based convolutional spiking neural network for gesture recognition from dvs input. In *IJCNN*, pages 1–9. IEEE, 2020.

[Gu *et al.*, 2019] Pengjie Gu, Rong Xiao, Gang Pan, and Huajin Tang. STCA: Spatio-temporal credit assignment with delayed feedback in deep spiking neural networks. In *IJCAI*, pages 1366–1372, 2019.

[Hu *et al.*, 2016] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. DVS benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in Neuroscience*, 10:405, 2016.

[Hu *et al.*, 2018] Yangfan Hu, Huajin Tang, Yueming Wang, and Gang Pan. Spiking deep residual network. *arXiv preprint arXiv:1805.01352*, 2018.

[Innocenti *et al.*, 2020] Simone Undri Innocenti, Federico Becattini, Federico Pernici, and Alberto Del Bimbo. Temporal binary representation for event-based action recognition. *arXiv preprint arXiv:2010.08946*, 2020.

[Jhuang *et al.*, 2007] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio. A biologically inspired system for action recognition. In *ICCV*, pages 1–8. IEEE, 2007.

[Lagorce *et al.*, 2017] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. HOTS: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1346–1359, 2017.

[Lichtsteiner *et al.*, 2008] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, 2008.

[Liu *et al.*, 2020a] Qianhui Liu, Gang Pan, Haibo Ruan, Dong Xing, Qi Xu, and Huajin Tang. Unsupervised AER object recognition based on multiscale spatio-temporal features and spiking neurons. *IEEE Trans. Neural Networks Learn. Syst.*, 31(12):5300–5311, 2020.

[Liu *et al.*, 2020b] Qianhui Liu, Haibo Ruan, Dong Xing, Huajin Tang, and Gang Pan. Effective AER object classification using segmented probability-maximization learning in spiking neural networks. In *AAAI*, pages 1308–1315, 2020.

[Maro *et al.*, 2020] Jean-Matthieu Maro, Sio-Hoi Ieng, and Ryad Benosman. Event-based gesture recognition with dynamic background suppression using smartphone computational capabilities. *Frontiers in Neuroscience*, 14:275, 2020.

[Miao *et al.*, 2019] Shu Miao, Guang Chen, Xiangyu Ning, Yang Zi, Kejia Ren, Zhenshan Bing, and Alois Knoll. Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection. *Frontiers in Neurorobotics*, 13:38, 2019.

[Orchard *et al.*, 2013] Garrick Orchard, Ryad Benosman, Ralph Etienne-Cummings, and Nitish V Thakor. A spiking neural network architecture for visual motion estimation. In *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 298–301. IEEE, 2013.

[Orchard *et al.*, 2015] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. HFirst: A temporal approach to object recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):2028–2040, 2015.

[Paredes-Vallés *et al.*, 2019] Federico Paredes-Vallés, Kirk Yannick Willehm Scheper, and Guido Cornelis Henricus Eugene De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[Posch *et al.*, 2011] Christoph Posch, Daniel Matolin, and Rainer 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*, 46(1):259–275, 2011.

[Ramesh *et al.*, 2019] Bharath Ramesh, Hong Yang, Garrick Michael Orchard, Ngoc Anh Le Thi, Shihao Zhang, and Cheng Xiang. DART: distribution aware retinal transform for event-based cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[Shrestha and Orchard, 2018] Sumit Bam Shrestha and Garrick Orchard. SLAYER: Spike layer error reassignment in time. In *NeurIPS*, pages 1412–1421, 2018.

[Sironi *et al.*, 2018] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *CVPR*, pages 1731–1740, 2018.

[Xiao *et al.*, 2019] Rong Xiao, Huajin Tang, Yuhao Ma, Rui Yan, and Garrick Orchard. An event-driven categorization model for AER image sensors using multispike encoding and learning. *IEEE Trans. Neural Networks Learn. Syst.*, 2019.

[Xing *et al.*, 2020] Yannan Xing, Gaetano Di Caterina, and John Soraghan. A new spiking convolutional recurrent neural network (SCRNN) with applications to event-based hand gesture recognition. *Frontiers in Neuroscience*, 14, 2020.