Information Sciences 644 (2023) 119170

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

EventMix: An efficient data augmentation strategy for event-based learning

Guobin Shen [a,d], Dongcheng Zhao [a], Yi Zeng [a,b,c,d,*]

[a] Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China
[b] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, China
[c] Center for Excellence in Brain Science and Intelligence Technology, CAS, Shanghai, China
[d] School of Future Technology, University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Event based data augmentation
Neuromorphic data
Spiking neural networks
Reasonable label assignment
Gaussian mixture model

ABSTRACT

High-quality and challenging event stream datasets play an important role in the design of an efficient event-driven mechanism that mimics the brain. Although event cameras can provide high dynamic range and low-energy event stream data, the scale is smaller and more difficult to obtain than traditional frame-based data, which restricts the development of neuromorphic computing. Data augmentation can improve the quantity and quality of the original data by processing more representations from the original data. This paper proposes an efficient data augmentation strategy for event stream data: **EventMix**. We carefully design the mixing of different event streams by Gaussian Mixture Model (GMM) to generate random 3D masks and achieve arbitrary shape mixing of event streams in the spatio-temporal dimension. By computing the relative distances of event streams, we propose a more reasonable way to assign labels to the mixed samples. The experimental results on multiple neuromorphic datasets have shown that our strategy can improve performance on neuromorphic classification tasks as well as neuromorphic human action recognition tasks both for ANNs and SNNs, and we have achieved state-of-the-art performance on DVS-CIFAR10, N-Caltech101, and DVS-Gesture datasets.

1. Introduction

The event camera, such as the Dynamic Vision Sensor (DVS), is a bionic vision sensor that mimics how the human retina works. In contrast to the conventional cameras, the intensity change of each pixel is recorded asynchronously in an event-driven manner rather than capturing intensity images at a fixed rate. Because of the high temporal resolution, high dynamic range, low time latency, and energy consumption of event cameras, they are widely used in several domains, such as image reconstruction [1], flow estimation [2], motion segmentation [3], and recognition [4], which also promotes the construction of frame-based datasets.

In contrast to frame-based cameras, event cameras perceive intensity changes independently for each pixel and output asynchronous event streams. These event streams are highly compressed representations of the visual signal with low latency and high temporal resolution. Encouraged by the great success of deep learning in frame-based image processing, recent work has favored the use of trained Convolutional Neural Networks (CNNs) to process event data. The event stream is first sliced along the tempo-

* Corresponding author at: Brain-inspired Cognitive Intelligence Lab, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China.
E-mail address: yi.zeng@ia.ac.cn (Y. Zeng).

https://doi.org/10.1016/j.ins.2023.119170
Received 21 November 2022; Received in revised form 12 May 2023; Accepted 14 May 2023

Available online 18 May 2023

Table 1
The characteristic of the famous existing datasets.

Dataset	Categories	Samples	Access
DVS-CIFAR10 [17]	10	10,000	Shot on static images
DVS-Gesture [20]	11	1,342	Shot on real scenes
N-MNIST [18]	10	70,000	Shot on static images
N-Caltech101 [18]	101	8,709	Shot on static images
N-CARS [19]	2	24,029	Shot on real scenes
N-Omniglot [21]	1623	32,460	Shot on static images
ASL-DVS [11]	24	100,800	Shot on real scenes
UCF101-DVS [11]	101	13,320	Shot on static images
HMDB51-DVS [11]	51	6,766	Shot on static images

ral axis, and the different parts are integrated into a frame-based representation. Then CNNs are applied to the transformed event data for feature extraction and processing. Zhu et al. [5] discretize the event stream data along the time axis to form a voxel grid representation. Gehrig et al. [6] proposed a trainable kernel that is able to weigh each event during the integration process.

Unlike static image data, event data retains rich spatio-temporal information after conversion to frames. Hence, some studies improve the model's performance on event data by introducing a spatio-temporal attention mechanism. Yu et al. [7] proposed a spatio-temporal synaptic connection SNN (STSC-SNN) model to improve the classification accuracy of SNNs on event streams. Zhu et al. [8] proposed a Temporal-Channel Joint Attention (TCJA) architectural unit based on the attention mechanism and can effectively enhance the association of spike sequences in spatial and temporal dimensions, thus achieving good performance on multiple DVS datasets. Yao et al. [9] used the attention concept to extend the temporal-domain input to learn the frame-level representation in processing event streams and proposed a Temporal-wise Attention SNN (TA-SNN) model to improve the classification accuracy of event data. The Spiking CapsNet [10] amalgamates the strengths of SNNs and CapsNet, introducing a biologically plausible spike-timing-dependent plasticity routing mechanism. It demonstrates remarkable robustness against noise and affine transformations.

In addition to converting events into frames to process event data using well-established tools of CNNs, some graph-based approaches attempt to process sparse event data directly. Bi et al. [11] introduced a graph-based learning system for object classification of event data. Mondal et al. [12] proposed an unsupervised graph clustering approach to achieve moving object detection based on event data. Zhang et al. [13] designed an adaptive event vision detection framework to overcome the problem that conventional cameras cannot detect high-speed moving objects. Schaefer et al. [14] proposed an Asynchronous, Event-based Graph Neural Network (AEGNN) that updates only the nodes affected by new events, significantly reducing computational complexity. This graph-based event representation embeds the event information into the graph, which reasonably exploits the sparse event data and achieves higher computational efficiency but still has some performance gap compared to the traditional methods.

As seen in Table 1, event-based datasets are smaller in scale compared to traditional frame-based datasets, and the DVS-Gesture has only $1,342$ samples for 11 categories. Such small and sparse datasets can easily lead to overfitting and unstable convergence, whether for artificial neural networks (ANNs) or spiking neural networks (SNNs). Moreover, it has restricted the development of the event-based algorithm. The intuitive idea is to collect more event-based data. However, the cost is expensive compared to collecting traditional frame-based datasets due to the scarcity of event cameras. An alternative approach is to apply the data augmentation to the existing datasets. The data augmentation approach improves the quality and quantity of the training data by adding prior knowledge, such as rotation and flipping, to generate more different representations of the training data. Researchers have proposed many strategies [15,16] for the traditional image data.

However, these strategies do not take the sparse and spatio-temporal dynamic nature of the event stream data and cannot be directly applied. This paper proposes the **EventMix** strategy, as shown in Fig. 1, which fully takes the sample mixing and the characteristics of event stream data into consideration. The experimental results on several famous datasets have demonstrated that our EventMix is an efficient augmentation strategy for event-stream data. Our contributions are summarized as follows:

- By thinking about the spatio-temporal dynamic of event stream data, we propose a random 3D mask generation method, which can generate masks of event stream data in temporal and spatial dimensions and provide a diverse way of mixing data stream data.
- We propose two mixing methods for event stream data labels. By calculating the relative distance between the original and mixed samples or the number of events in the mixed region, we achieve a reasonable label weight assignment for the event stream data labels.
- We validate the proposed EventMix event augmentation strategy on the DVS-CIFAR10 [17], N-Caltech101 [18], N-Cars [19], and DVS-Gesture [20] datasets. The experimental results show that the proposed EventMix can effectively augment the event stream data and achieve state-of-the-art.

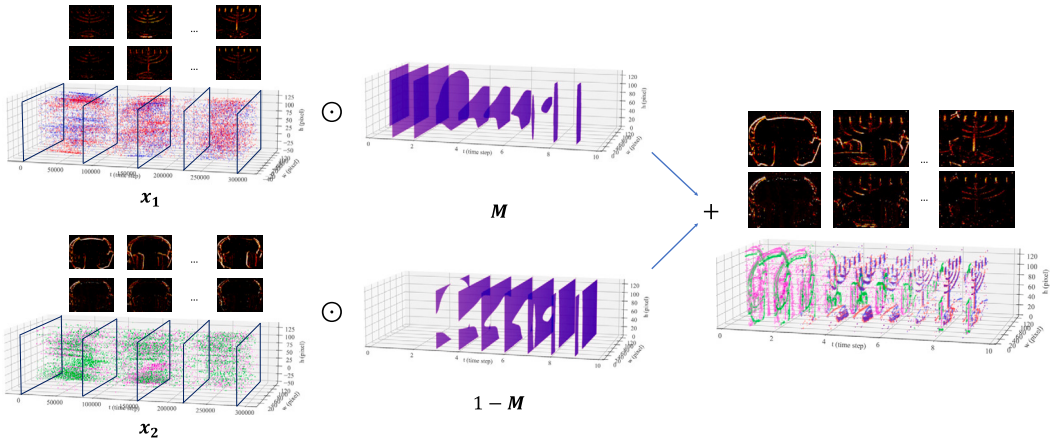


Fig. 1. The illustration of our EventMix data augmentation strategy on events data.

2. Related work

2.1. Event stream data

Unlike conventional image data that describe the environment by the brightness and color information, event stream data describe the visual information of the environment by the brightness change. The time stream information consists of four dimensions (t, x, y, p). Where t is the time of the event. (x, y) are the pixel coordinates that perceived the event. p is the polarity of the event. When the pixel perceives a brightness enhancement above a certain threshold, $+1$ is output; conversely, if the pixel perceives a brightness reduction above the threshold, -1 is output; otherwise, 0 is output. Event stream data is widely used in neuromorphic computing [10,22–24] because it is similar to the biological retina in that it describes the environment by events that are similar to spikes.

With the development of deep learning, event stream data with low energy consumption, low latency, and biological plausibility has also gained attention. Gehrig et al. [6] proposed a general framework that can process event stream data and output image-like, or video-like data, enabling neural networks designed for images and videos to be easily applied to event stream data. Schaefer et al. [14] designed an asynchronous event-based graph neural network that can process event stream data directly. However, its performance is not yet comparable to other frame-based methods. Therefore, in this study, we use the same method as Gehrig et al. for event stream data processing.

2.2. Data augmentation

It is widely believed that a larger number of training samples can reduce the overfitting of neural networks and enhance the generalization ability of the model and its adaptability to new samples. However, large data sets often require significant consumption. Data augmentation is a common technique in deep learning to generate virtual samples around the training samples through prior knowledge to diversify the datasets and make the trained model more generalizable. The augmentation techniques for image data have been extensively explored, including morphological transformations such as random flipping, random rotation, and random cropping. Additionally, pixel-level transformations such as noise addition, blurring, and erasing have also been widely studied.

Some data augmentation methods have been proposed to combine multiple samples to generate new samples in recent years. For example, MixUp [25] combines different samples convexly and performs the same operation on the corresponding labels. CutMix [26] combines different regions of image samples and mixes the corresponding labels according to the area of different samples in the mixed sample. Puzzle Mix [27] mixes the computed salient regions of different samples to generate more efficient mixing instances. These mixing-based methods are believed to lead the models to empirical risk minimization estimates, enabling many advanced deep neural network models to further improve their performance. However, since these methods are designed for image data, it is difficult to apply them directly to event stream data to achieve desired results.

Data augmentation strategies have improved deep learning performance in tasks such as image classification [18,28], face identification [29], person re-identification [30], and anomaly detection [31]. However, augmentation strategies for event stream data have not received sufficient attention, which also limits the further improvement of the performance of neural networks processing event streams. To the best of the authors' knowledge, only limited research has been conducted on augmenting event stream data. EventDrop [32] achieves augmentation of event stream data by randomly dropping events, but they do not take into account the relationship between different events in their approach. Li et al. [33] applied image morphological transformations (rolling, cutout, shear, and rotation) to the event stream data, but they ignored the temporal properties of the event stream data.

In this study, we introduce a novel approach to data augmentation called **EventMix**, which is specifically designed for event stream data. It leverages event streams' spatio-temporal dynamics and sparsity to generate augmented samples. To the best of our knowledge, this is the first time data augmentation based on sample mixing has been applied to event stream data. Experimental results demonstrate that our proposed method achieves state-of-the-art performance on several event stream datasets.

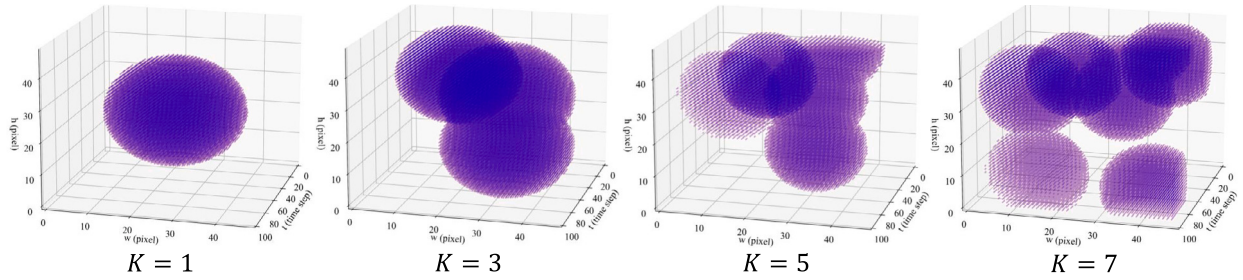


Fig. 2. Gaussian Mixture Models with different number of components.

3. EventMix

3.1. Event stream representation

The raw, direct event stream data from the DVS can be represented as:

$$\epsilon = \{e_i\}_{i=i}^{I} = \{t_i, x_i, y_i, p_i\}_{i=1}^{I} \tag{1}$$

Where t denotes the timestamp of the event, (x, y) denotes the position and p denotes the polarity, with $+1$ and -1 indicating the increase and decrease of light, respectively.

In order to enable neural networks to process the event stream data quickly and efficiently, we convert the event data into frame-based, which is commonly used for the event data. Similar to the method of [6], we first split the event stream data with duration T on average. The bin size of each frame is ΔT, and the events in ΔT are summed according to their polarities. Finally, we can get frame-based data with two channels and a length of $T/\Delta T$. The details are shown in Eq. (2):

$$E(c, x, y, p) = \sum_{e_i \in \epsilon} k(t_i - t_c, x_i - x, y_i - y, p_i - p) \tag{2}$$

where, t_c indicates the start time of the cth frame after splitting, $k(t, x, y, p)$ can be expressed as:

$$k(t, x, y, p) = \delta(x, y, p)(t < \Delta T) \tag{3}$$

In Eq. (3), $\delta(\cdot)$ denotes the discrete Dirac function. Our EventMix shares a similar motivation with MixUp and CutMix, in that it mixes two samples and labels to generate the new samples and their corresponding labels.

3.2. Improved 3D dynamic masking

Let x denote the sample in the training set after the above transformation, then the mixed samples $\widetilde{x}$ can be obtained from the two training samples (x_A, x_B):

$$\widetilde{x} = \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B \tag{4}$$

$\mathbf{M} \in \{0, 1\}$ denotes a binary mask, and the original data in the mask is discarded and filled with the corresponding part of the other data. For the shape of the mask, the original CutMix only considers the $\mathbf{M}$ in the spatial dimension and restricts the shape to be square. This unnecessary limitation on the data will make the models more biased toward the general features and is not conducive to the performance. Here, we remove the restriction that the mask must be square and fully consider the spatiotemporal characteristics of event data to extend the mask to 3 dimensions. To achieve mixing of samples of different shapes and scales, we generated a continuous 3D mask by random Gaussian Mixture Model (GMM) and then generated a binary 3D mask by binarization:

$\mathbf{M} \in \{0, 1\}$ denotes a binary mask, and the original data in the mask is discarded and filled with the corresponding part of the other data. For the mask's shape, the original CutMix only considers the $\mathbf{M}$ in the spatial dimension and restricts the shape to be square. This unnecessary limitation on the data will make the models more biased toward the general features and is not conducive to performance. Here, we remove the restriction that the mask must be square and fully consider the spatiotemporal characteristics of event data to extend the mask to 3 dimensions. To achieve the mixing of samples of different shapes and scales, we generated a continuous 3D mask by random Gaussian Mixture Model (GMM) and then generated a binary 3D mask by binarization:

$$\mathbf{M} = B(\sum_{k=1}^{K} \pi_k N(\mathbf{X} \mid \mu_k, \Sigma_k), \lambda) \tag{5}$$

K denotes the number of components in the GMM. As shown in Fig. 2, a smaller number of components will guide the model to generate continuous masks and achieve large-scale sample mixing. In comparison, a larger number of components will generate finer masks and achieve sample mixing in detail. π_k, μ_k, and Σ_k denote the mixing coefficient, mean, and variance of the kth component, respectively. The above variables are generated randomly. λ denotes the proportion of the randomly generated fraction to be mixed

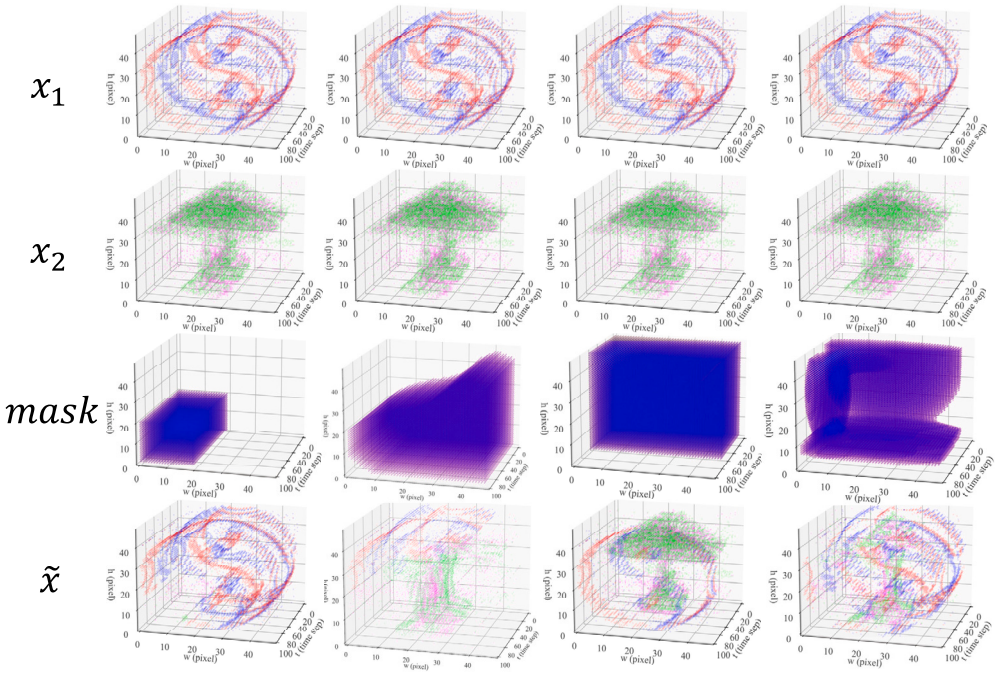


Fig. 3. Some examples of data augmentation using different masks. From left to right, we show the use of square masks, spatial masks, temporal masks, and spatio-temporal masks.

with the original sample and is obtained by sampling from the *beta* distribution. $B(\cdot, \lambda)$ represents the transformation of the actual mask obtained by the random GMM into a binarized form:

$$B(\mathbf{X}, \lambda)_{t,i,j} = \begin{cases} 1, & \mathbf{X}_{t,i,j} < top(\lambda \ size(\mathbf{X}), X) \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$top(k, \mathbf{X})$ denotes the value of the k_{th} largest number in $\mathbf{X}$. As shown in Fig. 3, By applying the above method, it is possible to generate diverse masks with temporal and spatial dimensions and achieve more mixing patterns of event stream data.

3.3. Mixing of event labels

Let y denote the labels, then the corresponding label $\tilde{y}$ of the mixed sample in Eq. (4) is shown as Eq. (7):

$$\tilde{y} = \alpha y_A + (1 - \alpha) y_B \tag{7}$$

As shown in Fig. 4, unlike the image data, the event stream data consists of sparse events, and meaningful regions in event stream data can be easily distinguished. It is not suitable to use the area for label weight assignment. Therefore, we design two reasonable label mixing methods: based on the number of events and the relative distance of events.

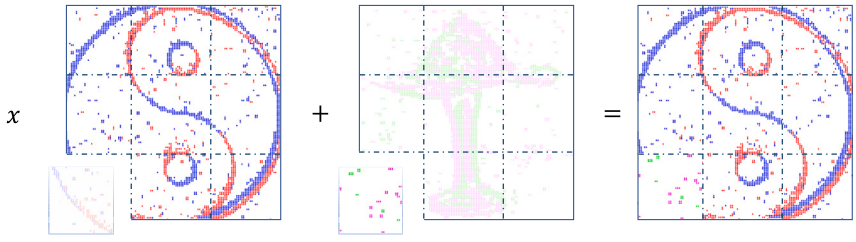
Based on the number of events. Sample mixing means that the events in the subregion of the original data x_A are deleted and filled with events from the same location of another event x_B. So a simple idea is to calculate the proportion of deleted events to x_A and the proportion of added events to x_B, and calculate the mixed label weight α:

$$\alpha = \frac{\frac{\sum \mathbf{M} \odot x_A}{\sum x_A}}{\frac{\sum \mathbf{M} \odot x_A}{\sum x_A} + \frac{\sum (\mathbf{1} - \mathbf{M}) \odot x_B}{\sum x_B}} \tag{8}$$

Where $\sum \cdot$ means summing over all elements of the tensor. Based on Eq. (8), the weights of the mixed labels can be calculated and combined with Eq. (4) to obtain a representation of the mixed labels based on the proportion of the number of events in the mixed region.

Based on the relative distance. Directly calculating the number of events in the mixed region solves the problem of biased label weight assignment when the mixed region is background. However, this approach only considers the number of events in the mixing region but ignores the distribution of events. Considering the sparse data of event streams and using only binary representation, we design a way to calculate the distance of event streams and use it as the basis for label weight assignment:

$$E(x_A, x_B) = e(f(x_A), f(x_B)) \tag{9}$$



y_{area}	$(1,0)$	$(0,1)$	$(\frac{8}{9},\frac{1}{9})$
y_{count}	$(1,0)$	$(0,1)$	$\approx(1,0)$
$y_{distance}$	$(1,0)$	$(0,1)$	$\approx(1,0)$

Fig. 4. An example of different label weight assignment methods of event stream data.

Where $e(\cdot,\cdot)$ indicates how to calculate the distance between two event streams. In this study, we use the mean square error (MSE) to represent the distance between event streams. The $f(\cdot)$ denotes the preprocessing of the event streams. Since calculating the difference of corresponding pixels directly is a strict metric, considering that the event streams have translation invariance, we first process the event stream using the average pooling operation and then calculate the distance.

$$\alpha = \frac{E(x_B,\widetilde{x})^2}{E(x_A,\widetilde{x})^2 + E(x_B,\widetilde{x})^2} \tag{10}$$

According to the characteristics of event stream data, we design the data augmentation strategy based on sample mixing. More diverse event stream mixing is achieved, as well as more reasonable label weight assignment rules.

4. Experiments

In this chapter, we will conduct experiments on several datasets to evaluate the effectiveness of our EventMix. The experiments are based on the PyTorch framework [34] on NVIDIA A100 GPU with several datasets to verify our method. We use the AdamW [35] as the optimizer, the learning rate lr is set with 1×10^{-3}. The training epochs are set at 500.

Random cropping, random flipping, and random rotation are used on the training samples before other mixing-based augmentation operations. If not mentioned, Eq. (9) is used as the default method. During data augmentation, EventMix is applied to the sample with probability $p=0.5$, the area of the mask is sampled from $beta(\lambda,\lambda)$ with $\lambda=1$, and the number of components of the GMM is set to $K=3$. We conduct experiments on ANNs and SNNs with PLIF [36] neurons on several network structures. Since in spiking neural networks, information is passed between layers in the form of spikes, which leads to low information density and tends to degrade performance. Therefore, we use the same method as [37] to preactivate each block of ResNet and pass the membrane potential between modules with identity connectivity, which ensures more information transfer without increasing the computation of the network. We compare the performance of our proposed EventMix with CutMix and MixUp for event stream data augmentation with the same neural network model and hyperparameter settings.

4.1. Implementation details

We validated EventMix on the following four DVS datasets:

DVS-CIFAR10 [17]. DVS-CIFAR10 is a neuromorphic version converted from the CIFAR10 dataset [45]. 10,000 frame-based images were transformed with DVS into 10,000 event streams. All event streams are sliced by Eq. (3) before training or validation, and in the same way as Li et al. [33] we slice the event stream data into 10 time steps. Before data augmentation, we resize all event streams to 48×48 in the spatial dimension. Since this dataset does not divide the training and valid sets, we divided the training and valid sets by $9:1$.

N-Caltech101 [18]. N-Caltech101 is the event stream version of Caltech101 [46]. The images in Caltech101 are displayed on the LCD monitor and captured by an automatically moving event camera. After this transformation, the N-Caltech101 dataset is obtained. Since there is a difference in the number of samples of different categories in N-Caltech101, we resample the training set according to the ratio of the number of different categories. As in DVS-CIFAR10, we resize the event stream to 48×48, divide it into 10 time steps, and divide the training and validation sets by $9:1$.

N-CARS [19] is a real-world event stream dataset acquired by an event camera placed behind the front windshield of a car in a real-world road environment. The samples in this data are divided into two categories: background and vehicle, with 15422 training samples and 8607 test samples.

Table 2
Comparison of classification results with other methods. † denotes the results reproduced in our training settings.

Method	Model	DVS-CIFAR10	N-Caltech101	N-CARS	DVS-Gesture
Rollout [38]	VGG16	66.5	-	94.07	95.68
SALT [39]	VGG11	67.1	55.0	-	-
tdBN [40]	ResNet19	67.8	73.10†	95.02†	89.01†
PLIF [36]	VGG11	74.8	74.25†	95.24†	94.07†
EST [6]	ResNet34-ANN	-	81.7	92.5	-
ECS [41]	ECSNet	72.7	69.3	94.6	-
Matrix-LSTM [42]	ResNet18-EST	-	84.31	94.37	-
AMAE [43]	AMAE	75.3	85.1	95.5	-
MVF-Net [44]	MVF-Net	76.2	87.1	**96.8**	-
w/o NDA [33]	ResNet19	67.9	62.8	82.4	-
w/. NDA [33]	ResNet19	78.0	78.6	87.2	-
w/o EventDrop [32]	ResNet34-ANN	77.2†	83.91	91.03	79.92†
w/. EventDrop [32]	ResNet34-ANN	82.9†	85.15	95.50	80.68†
Baseline	Resnet18	79.23	75.25	95.94	84.76
EventMix	Resnet18	**81.45**	**79.47**	**96.29**	**96.75**
Baseline	Resnet34-ANN	81.13	84.51	95.36	86.33
EventMix	Resnet34-ANN	**85.60**	**89.20**	**96.54**	**91.80**

DVS-Gesture [20] is a real-world gesture recognition dataset collected by the DVS camera. It contains 1342 different packs of 11 categories collected on 29 individuals. We used the same preprocessing approach as Gregor et al. [47], and the training and validation sets were divided by $8:2$.

UCF101-DVS is a neuromorphic version of UCF101 [48], containing a total of 13,320 event streams of 101 different human actions. The dataset was made under controlled illumination conditions, using a downward-fixed neuromorphic sensor to capture existing baseline video.

HMDB51-DVS is a neuromorphic version of HMDB51 [49], containing a total of 6,766 human action event streams of 51 different human actions. The dataset was made following the same recording procedure as UCF101-DVS.

4.2. Comparison with existing literature

To illustrate the superiority of our EventMix, we show experimental results against other state-of-the-art algorithms. As shown in Table 2, for those four datasets, both ANN and SNN, our EventMix brings a significant improvement. Especially for the DVS-Gesture dataset, EventMix improves by nearly 12% on SNNs and 5% on ANNs. Meanwhile, on the N-Caltech 101 dataset, our model outperforms Neuromorphic data augmentation by nearly 11%, and even compared to EventDrop, our model outperforms by nearly 4%. EventDrop just randomly drops the event, which destroys the distribution of the original event data but does not consider the change of the label. In addition to mixing samples, our EventMix fully considers the spatiotemporal characteristics of event data and aligns more accurate and reasonable labels to the mixed data, which greatly improves the performance of the algorithm.

4.3. Comparison with existing mixing-based methods

In 4.2, we show a comparison of our proposed EventMix strategy with some extant literature and confirm that our approach is able to achieve advanced performance on event stream data classification tasks. To further illustrate the efficiency and effectiveness of the EventMix strategy, we compare it with some advanced sample mixing-based data augmentation strategies applied directly to event stream data. We use MixUp and CutMix to compare with our proposed method. In order to apply the above two methods to 3D event data, we make a simple extension of them. For MixUp, we directly mix events from different event data with different weights. As for CutMix we generate a two-dimensional mask and then repeat it in the temporal dimension to generate a 3D mask. The results are shown in Table 3.

In Table 3, we compare our proposed EventMix strategy with some mixing-based data augmentation strategies designed for image data using the same ANN structure under the same hyperparameter settings. MixUp mixes all events in two event streams according to their weights, and since event data is very sparse and has simple semantics, mixing samples in this way often causes ambiguity. CutMix can avoid the above problem, but this approach restricts the data mask to rectangles and ignores the temporal dimensionality of the event data. In addition, the label weight assignment of these two methods does not consider the characteristics of the event data. Therefore, the data augmentation strategies MixUp and CutMix designed for image data can only provide limited performance improvement for event stream data. By carefully considering the characteristics of event stream data, our proposed EventMix strategy can improve the model performance by 2.45% on the DVS-CIFAR10 dataset, 2.36% on the N-Caltech101 dataset, and 4.69% on DVS-Gesture dataset when using the ResNet34-ANN model compared to the CutMix strategy.

Action recognition has a large number of applications in human behavior analysis and other motion-based tasks. Action recognition focuses more on the recognition of dynamic features, so the recognition of human action event data is a more challenging task than event classification. To further validate the effectiveness of the EventMix, we validate EventMix on the human action event

Table 3
Comparison of mixing-based data augmentation with ANN.

Model	Method	Accuracy (Improvement)		
		DVS-CIFAR10	N-Caltech101	DVS-Gesture
ResNet34	Baseline	$81.13_{+0.00}$	$84.51_{+0.00}$	$86.33_{+0.00}$
	MixUp	$82.93_{+1.80}$	$83.85_{+2.72}$	$89.06_{+2.73}$
	CutMix	$83.15_{+2.02}$	$86.84_{+2.33}$	$87.11_{+0.78}$
	EventMix	$\mathbf{85.60}_{+4.47}$	$\mathbf{89.20}_{+4.69}$	$\mathbf{91.80}_{+5.47}$
ResNet18	Baseline	$80.24_{+1.79}$	$82.03_{+1.79}$	$85.55_{+0.00}$
	MixUp	$83.15_{+2.91}$	$84.38_{+4.14}$	$86.72_{+1.17}$
	CutMix	$82.16_{+1.92}$	$81.58_{+1.34}$	$86.33_{+0.78}$
	EventMix	$\mathbf{84.38}_{+4.14}$	$\mathbf{84.71}_{+4.47}$	$\mathbf{89.45}_{+3.90}$
MobileV2	Baseline	$79.46_{+0.00}$	$76.30_{+0.00}$	$78.91_{+0.00}$
	MixUp	$82.37_{+2.91}$	$83.29_{+6.99}$	$81.25_{+2.34}$
	CutMix	$83.37_{+3.91}$	$81.77_{+5.47}$	$82.03_{+3.12}$
	EventMix	$\mathbf{83.70}_{+4.24}$	$\mathbf{83.85}_{+7.55}$	$\mathbf{82.93}_{+4.02}$

Table 4
Comparison of mixing-based data augmentation on human action recognition.

Model	Method	Accuracy (Improvement)	
		UCF101-DVS	HMDB51-DVS
ResNet18	Baseline	$61.62_{+0.00}$	$62.37_{+0.00}$
	MixUp	$58.87_{-2.75}$	$65.62_{+3.25}$
	CutMix	$62.95_{+1.33}$	$67.80_{+5.43}$
	EventMix	$63.90_{+2.28}$	$69.77_{+7.40}$
ResNet18	Baseline	$60.04_{+0.00}$	$61.47_{+0.00}$
	MixUp	$58.81_{-1.23}$	$65.86_{+4.39}$
	CutMix	$60.23_{+0.19}$	$68.73_{+7.26}$
	EventMix	$60.63_{+0.59}$	$70.25_{+8.78}$

Table 5
Component Analysis with ResNet34-ANN on DVS-CIFAR10.

Method	Area	Count	Distance
Square	$82.59_{+0.00}$	$82.14_{-0.45}$	$82.37_{-0.22}$
Spatial	$82.70_{+0.11}$	$82.59_{0.00}$	$83.48_{+0.89}$
Temporal	$84.15_{+1.56}$	$84.38_{+1.79}$	$83.37_{+0.78}$
S & T	$84.04_{+1.45}$	$85.16_{+2.57}$	$\mathbf{85.60}_{+3.01}$

dataset, as shown in Table 4. Compared with other data augmentation strategies that directly extend to event data, the proposed EventMix achieves better performance on tasks with richer dynamic features such as human action recognition.

4.4. Component analysis

Our proposed EventMix strategy redesigns the sample mixing-based data augmentation strategy from two aspects: mask generation and label weight assignment. On the one hand, we design a 3D mask generation method that can mix different samples in time and space dimensions according to the characteristics of event stream data and reduce the impact of monotonous mask shapes on the model's generalization ability. On the other hand, based on the sparse data characteristics of event streams, we first assume that each event in the event stream has the same contribution to the corresponding label and design two label mixing methods based on the number of events and the relative distance of the event streams. Therefore, we then compare different masking approaches and different label weight assignment approaches and further illustrate the effectiveness and efficiency of our proposed EventMix strategy.

Table 5 lists the performance contributions of the different components of our proposed EventMix strategy. The use of square masks and sample mixing in the spatial dimension is the result of directly applying CutMix to the event stream data. From Table 5, it can be seen that mixing event streams in the temporal dimension is essential and can improve the model's accuracy by almost 1%. Using the random GMM to generate 3D masks can enrich the pattern of sample mixing and further improve the generalization ability of the model. Our proposed event stream data label mixing methods can more accurately assign labels to the mixed samples and improve the model performance than the original area-based mixing method. Combining these two strategies can generate

Table 6
Ablation study of number of GMM components.

Components (K)	1	2	3	4	5
Top-1 Acc (%)	84.5	85.4	**85.6**	85.3	84.7

Table 7
Ablation study of sample mixing probability and mixing area distribution.

	$p = 0.25$	$p = 0.5$	$p = 0.75$	$p = 1.0$
$\lambda = 0.25$	84.7	85.1	84.9	85.1
$\lambda = 0.5$	85.1	85.2	85.3	85.2
$\lambda = 1.0$	85.1	**85.6**	85.5	85.3
$\lambda = 2.0$	85.0	85.4	85.3	85.2

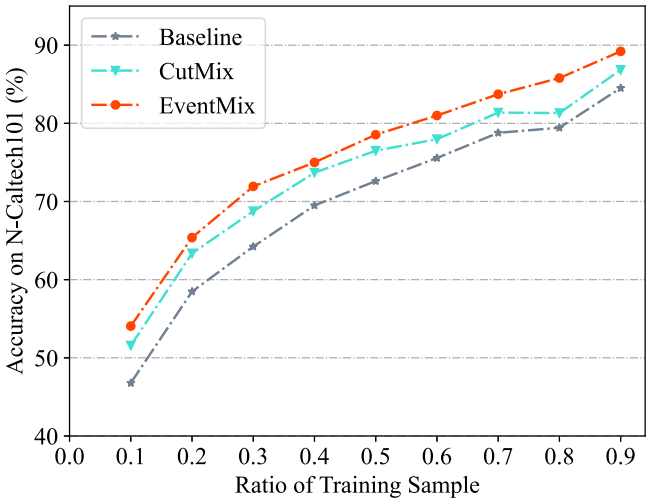


Fig. 5. Comparison of EventMix with CutMix and baseline when using different number of samples as training set.

more diverse event stream data mixing patterns, assign mixed sample labels more rationally, and achieve a 3.01% performance improvement over the original data augmentation strategy.

4.5. Ablation study of hyperparameters

We conducted an ablation study on the hyperparameters of EventMix on the DVS-CIFAR10 dataset using the same settings as in Section 4.1. We evaluated EventMix using different number of GMM components, the probability of EventMix being applied to the training data, and the shape of the *beta* distribution.

Table 6 shows the effect of the number of components of GMM on the performance of EventMix. A smaller number of components leads to a simple mask shape and degrades the performance of EventMix. A larger number of components causes the event data to be sliced into smaller chunks, confusing local details. Therefore, EventMix performs best when the number of components of GMM is $n = 3$.

Table 7 shows the effect of the probability of applying EventMix to the training samples, and the mask area distribution on EventMix. When EventMix is applied with low probability, the performance of EventMix is affected due to the small number of mixed samples, while the impact on the accuracy is not significant when EventMix is applied with probability $p \geq 0.5$. The λ controls the shape of the *beta* distribution, with larger λ tending to generate masks with uniform area (e.g., $0.5a + 0.5b$), while smaller λ tends to generate samples with larger differences in mixing area (e.g., $0.9a + 0.1b$). As shown in Table 7, the best performance is achieved at $\lambda = 1$, i.e., sampling the masked area from a uniform distribution.

4.6. Effect of the amount of training sample

To further demonstrate the effectiveness of the EventMix data augmentation strategy in extreme cases, we compared the performance of the EventMix strategy under different training sample sizes with some other data augmentation strategies.

Fig. 5 shows the performance of different data augmentation algorithms when using 10% - 90% of the total N-Caltech101 dataset as the training data. Our proposed data augmentation strategy guarantees 54.06% validation accuracy with only 10% of the data as the training set, a 7.28% improvement over the conventional morphological transformation strategy. It maintains a significant advantage over the baseline for different training set sizes.

5. Conclusion

Since the event data is small and difficult to obtain, this paper designs an efficient data augmentation strategy EventMix for the event data. EventMix has removed the limitation of the square mask in the CutMix and designs a three-dimensional mask according to the spatiotemporal characteristics of event data. Also, a more reasonable label assignment is designed for the mixed sample. We have tested on multiple event datasets, and the experimental results show that our EventMix can significantly improve the performance of ANNs and SNNs on the event-based dataset. For SNNs, our EventMix has reached state-of-the-art performance on DVS-CIFAR10, N-Caltech101, N-CARS, and DVS-Gesture datasets.

CRediT authorship contribution statement

Guobin Shen: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Dongcheng Zhao:** Data curation, Investigation, Methodology, Writing – review & editing. **Yi Zeng:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work was supported by the National Key Research and Development Program (Grant No. 2020AAA0104305), and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100).

References

[1] Y. Zou, Y. Zheng, T. Takatani, Y. Fu, Learning to reconstruct high speed and high dynamic range videos from events, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 2024–2033.
[2] A.Z. Zhu, L. Yuan, K. Chaney, K. Daniilidis, Ev-flownet: self-supervised optical flow estimation for event-based cameras, arXiv preprint arXiv:1802.06898, 2018.
[3] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, D. Scaramuzza, Event-based motion segmentation by motion compensation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7244–7253.
[4] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. Di Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, et al., A low power, fully event-based gesture recognition system, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7243–7252.
[5] A. Zihao Zhu, L. Yuan, K. Chaney, K. Daniilidis, Unsupervised event-based optical flow using motion compensation, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.
[6] D. Gehrig, A. Loquercio, K. Derpanis, D. Scaramuzza, End-to-end learning of representations for asynchronous event-based data, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Korea (South), 2019, pp. 5632–5642.
[7] C. Yu, Z. Gu, D. Li, G. Wang, A. Wang, E. Li, Stsc-snn: spatio-temporal synaptic connection with temporal convolution and attention for spiking neural networks, Front. Neurosci. 16 (2022).
[8] R.-J. Zhu, Q. Zhao, T. Zhang, H. Deng, Y. Duan, M. Zhang, L.-J. Deng, Tcja-snn: temporal-channel joint attention for spiking neural networks, arXiv preprint arXiv:2206.10177, 2022.
[9] M. Yao, H. Gao, G. Zhao, D. Wang, Y. Lin, Z. Yang, G. Li, Temporal-wise attention spiking neural networks for event streams classification, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10221–10230.
[10] D. Zhao, Y. Li, Y. Zeng, J. Wang, Q. Zhang, Spiking capsnet: a spiking neural network with a biologically plausible routing rule between capsules, Inf. Sci. 610 (2022) 1–13.
[11] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, Y. Andreopoulos, Graph-based spatio-temporal feature learning for neuromorphic vision sensing, IEEE Trans. Image Process. 29 (2020) 9084–9098, https://doi.org/10.1109/TIP.2020.3023597.
[12] A. Mondal, S. R, J.H. Giraldo, T. Bouwmans, A.S. Chowdhury, Moving object detection for event-based vision using graph spectral clustering, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 876–884.
[13] S. Zhang, W. Wang, H. Li, S. Zhang, Eventmd: high-speed moving object detection based on event-based video frames, https://doi.org/10.2139/ssrn.4006876, Jan. 2022.
[14] S. Schaefer, D. Gehrig, D. Scaramuzza, Aegnn: asynchronous event-based graph neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 12371–12381.
[15] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 60, https://doi.org/10.1186/s40537-019-0197-0.
[16] S. Lim, I. Kim, T. Kim, C. Kim, S. Kim, Fast autoaugment, Adv. Neural Inf. Process. Syst. 32 (2019).
[17] H. Li, H. Liu, X. Ji, G. Li, L. Shi, CIFAR10-DVS: an event-stream dataset for object classification, Front. Neurosci. 11 (2017) 309, https://doi.org/10.3389/fnins.2017.00309.

[18] G. Orchard, A. Jayawant, G.K. Cohen, N. Thakor, Converting static image datasets to spiking neuromorphic datasets using saccades, Front. Neurosci. 9 (Nov. 2015), https://doi.org/10.3389/fnins.2015.00437.
[19] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, R. Benosman, HATS: histograms of averaged time surfaces for robust event-based object classification, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, USA, 2018, pp. 1731–1740.
[20] 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, D. Modha, A low power, fully event-based gesture recognition system, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7388–7397.
[21] Y. Li, Y. Dong, D. Zhao, Y. Zeng, N-Omniglot: a large-scale neuromorphic dataset for spatio-temporal sparse few-shot learning, arXiv:2112.13230, Dec. 2021.
[22] D. Zhao, Y. Zeng, T. Zhang, M. Shi, F. Zhao, GLSNN: a multi-layer spiking neural network based on global feedback alignment and local STDP plasticity, Front. Comput. Neurosci. 14 (2020) 576841, https://doi.org/10.3389/fncom.2020.576841.
[23] G. Shen, D. Zhao, Y. Zeng, Backpropagation with biologically plausible spatio-temporal adjustment for training deep spiking neural networks, arXiv:2110.08858 [cs], arXiv:2110.08858, Oct. 2021.
[24] S. Nazari, K. Faez, Establishing the flow of information between two bio-inspired spiking neural networks, Inf. Sci. 477 (2019) 80–99.
[25] H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, Mixup: beyond empirical risk minimization, arXiv preprint arXiv:1710.09412, 2017.
[26] S. Yun, D. Han, S.J. Oh, S. Chun, J. Choe, Y. Yoo, Cutmix: regularization strategy to train strong classifiers with localizable features, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6023–6032.
[27] J.-H. Kim, W. Choo, H.O. Song, Puzzle mix: exploiting saliency and local statistics for optimal mixup, in: Proceedings of the 37th International Conference on Machine Learning, PMLR, 2020, pp. 5275–5285.
[28] Y. Ding, C. Liu, H. Zhu, Q. Chen, A supervised data augmentation strategy based on random combinations of key features, Inf. Sci. 632 (2023) 678–697.
[29] S. Ammar, T. Bouwmans, M. Neji, Face identification using data augmentation based on the combination of dcgans and basic manipulations, Information 13 (8) (2022) 370.
[30] F. Chen, N. Wang, J. Tang, D. Liang, A negative transfer approach to person re-identification via domain augmentation, Inf. Sci. 549 (2021) 1–12.
[31] S. Cohen, N. Goldshlager, L. Rokach, B. Shapira, Boosting anomaly detection using unsupervised diverse test-time augmentation, Inf. Sci. (2023).
[32] F. Gu, W. Sng, X. Hu, F. Yu, Eventdrop: data augmentation for event-based learning, arXiv preprint arXiv:2106.05836, 2021.
[33] Y. Li, Y. Kim, H. Park, T. Geller, P. Panda, Neuromorphic data augmentation for training spiking neural networks, in: Computer Vision–ECCV 2022: 17th European Conference, Proceedings, Part VII, Tel Aviv, Israel, October 23–27, 2022, Springer, 2022, pp. 631–649.
[34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.
[35] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv:1711.05101 [cs, math], Jan. 2019, arXiv:1711.05101.
[36] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, Y. Tian, Incorporating learnable membrane time constant to enhance learning of spiking neural networks, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021, pp. 2641–2651.
[37] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2016, pp. 630–645.
[38] A. Kugele, T. Pfeil, M. Pfeiffer, E. Chicca, Efficient processing of spatio-temporal data streams with spiking neural networks, Front. Neurosci. 14 (2020) 439, https://doi.org/10.3389/fnins.2020.00439.
[39] Y. Kim, P. Panda, Optimizing deeper spiking neural networks for dynamic vision sensing, Neural Netw. 144 (2021) 686–698, https://doi.org/10.1016/j.neunet.2021.09.022.
[40] H. Zheng, Y. Wu, L. Deng, Y. Hu, G. Li, Going deeper with directly-trained larger spiking neural networks, Proc. AAAI Conf. Artif. Intell. 35 (12) (2021) 11062–11070.
[41] Z. Chen, J. Wu, J. Hou, L. Li, W. Dong, G. Shi, Ecsnet: spatio-temporal feature learning for event camera, IEEE Trans. Circuits Syst. Video Technol. (2022) 1, https://doi.org/10.1109/TCSVT.2022.3202659.
[42] M. Cannici, M. Ciccone, A. Romanoni, M. Matteucci, A differentiable recurrent surface for asynchronous event-based data, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2020, pp. 136–152.
[43] Y. Deng, Y. Li, H. Chen, Amae: adaptive motion-agnostic encoder for event-based object classification, IEEE Robot. Autom. Lett. 5 (3) (2020) 4596–4603, https://doi.org/10.1109/LRA.2020.3002480.
[44] Y. Deng, H. Chen, Y. Li, Mvf-net: a multi-view fusion network for event-based object classification, IEEE Trans. Circuits Syst. Video Technol. 32 (12) (2022) 8275–8284, https://doi.org/10.1109/TCSVT.2021.3073673.
[45] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, 2009.
[46] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2004, p. 9.
[47] G. Lenz, K. Chaney, S.B. Shrestha, O. Oubari, S. Picaud, G. Zarrella, Tonic: event-based datasets and transformations, in: Zenodo, 2021.
[48] K. Soomro, A.R. Zamir, M. Shah, Ucf101: a dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402, 2012.
[49] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, Hmdb: a large video database for human motion recognition, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2556–2563.