



Learning Bottleneck Transformer for Event Image-Voxel Feature Fusion Based Classification

Chengguo Yuan, Yu Jin , Zongzhen Wu, Fanting Wei, Yangzirui Wang,
Lan Chen^(✉), and Xiao Wang

Anhui University, Hefei 230601, Anhui, China
{e21301283,E02114335,E02114336}@stu.ahu.edu.cn,
{chenlan,xiaowang}@ahu.edu.cn

Abstract. Recognizing target objects using an event-based camera draws more and more attention in recent years. Existing works usually represent the event streams into point-cloud, voxel, image, etc., and learn the feature representations using various deep neural networks. Their final results may be limited by the following factors: monotonous modal expressions and the design of the network structure. To address the aforementioned challenges, this paper proposes a novel dual-stream framework for event representation, extraction, and fusion. This framework simultaneously models two common representations: event images and event voxels. By utilizing Transformer and Structured Graph Neural Network (GNN) architectures, spatial information and three-dimensional stereo information can be learned separately. Additionally, a bottleneck Transformer is introduced to facilitate the fusion of the dual-stream information. Extensive experiments demonstrate that our proposed framework achieves state-of-the-art performance on two widely used event-based classification datasets. The source code of this work is available at: https://github.com/Event-AHU/EFV_event_classification.

Keywords: Event Camera · Graph Neural Networks · Transformer Network · Bottleneck Fusion

1 Introduction

Recognizing the category of a given object is a fundamental problem in computer vision. Most of the previous classification models are developed for frame-based cameras, in other words, these recognition models focus on encoding and learning the representation of RGB frames. With the rapid development of deep learning, frame-based classification achieves significant improvement in recent years. Representative deep models (e.g., the AlexNet [1], ResNet [2], and Transformer [3]) and datasets (e.g., ImageNet [4]) are proposed one after another. However, the recognition performance in challenging scenarios is still far from unsatisfactory, including heavy occlusion, fast motion, and low illumination.

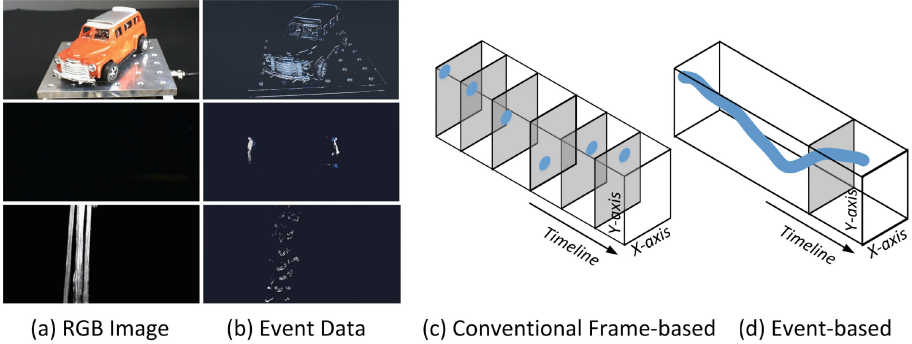


Fig. 1. Comparison of the frame- and event-based cameras <https://youtu.be/6xOmo7Ikzwk>. (a, b) shows representative samples in regular scenarios, low-illumination, and fast motion. (c, d) illustrates the different types of raw data representation of frame- and event-based cameras.

To improve object recognition in challenging scenarios, some researchers have started leveraging other sensors to obtain more effective signal inputs, thus enhancing recognition performance [5]. Among them, one of the most representative sensors is the event camera, also known as DVS (Dynamic Vision Sensor), which has been widely exploited in computer vision [6–8]. This paper focuses on using event cameras for object recognition. As shown in Fig. 1, different from the frame-based camera which records the light intensity for each pixel simultaneously, the event camera captures pulse signals asynchronously based on changes in light intensity, recording binary digital values of either zero or one. Typically, an increase in brightness is denoted as an ON event, while a decrease corresponds to an OFF event. An event pulse signal can be represented as a quadruple (x, y, t, p) , where x, y represents the spatial position information, t represents the timestamp, and p represents the polarity, i.e., ON/OFF event. Many works demonstrate that the event camera performs better in High Dynamic Range (HDR), high temporal resolution, low latency response, and strong robustness. Therefore, utilizing event cameras for object recognition is a research direction that holds great research value and practical potential.

Recently, researchers have already conducted studies on object recognition using event cameras and have proposed various approaches to address this task, including CNN (Convolutional Neural Network) [9], GNN (Graph Neural Network) [10], Transformer [3], etc. Although these methods have achieved good accuracy by representing and learning events from different perspectives, they are still limited by the following aspects: **Firstly**, they rely on a single event representation form, such as images, point clouds, or voxels, which may limit the expressiveness and versatility of the learned features. Different event representation forms may capture different aspects of the data, and using only one representation may lead to the loss of valuable information. **Secondly**, the current methods are constrained to using only one of the deep learning architectures,

such as CNNs, GNNs, or Transformers, for feature learning. Each architecture has its strengths and limitations in capturing different types of patterns and dependencies in data. By restricting the choice to a single architecture, the methods may not fully exploit the potential benefits and complementary strengths of different architectures. To address these limitations, future research should explore approaches that can integrate multiple event representation forms and leverage the combined power of different deep learning architectures. This could involve developing novel fusion techniques or hybrid architectures that can effectively capture and leverage diverse features and dependencies present in event data. By doing so, we can potentially enhance the performance and flexibility of event-based object recognition methods.

To address the aforementioned issues, in this work, we propose an effective dual-stream event information processing framework, referred to as EFV, as shown in Fig. 2. Specifically, we first transform the dense event point cloud signals into event images and event voxel representations. For the input of image frames, we utilize advanced spatiotemporal Transformer networks to learn spatiotemporal features. For voxel input, considering the sparsity of events, we employ a top-k selection method to sample meaningful signals for constructing a structured graph, and then use GNN (Graph Neural Network) to learn these volumetric structured features. Importantly, we introduce the Bottleneck Transformer to integrate these two types of feature representations, which are ultimately input to the dense layer for classification. It is easy to find that our proposed EFV possesses the characteristics of efficient event information processing, integration of multiple feature representations, spatiotemporal modeling capability, consideration of event sparsity, and accurate classification capability.

To sum up, the main contributions of this work can be concluded as the following two aspects:

- We propose an effective framework for recognition in event-based cameras, utilizing Event Image-Voxel feature representation and fusion.
- The introduction of the Bottleneck Transformer enables the interaction and fusion of dual-stream information, leading to improved recognition results.

2 Related Work

In this section, we give an introduction to Event-based Recognition¹, Graph Neural Networks, and Bottleneck Transformer.

Event-Based Recognition. Current research on event-based recognition can be divided into three distinct streams: CNN-based [9], SNN (Spiking Neural Networks)-based [11, 12], and GNN-based models [13–15]. For the CNN-based models, Wang et al. [9] proposed an event-based gait recognition (EV-gait) method, which effectively removes noise via motion consistency. SNN is also utilized for encoding the event stream in order to achieve energy-efficient recognition. A kind of highly efficient conversion of ANN to SNN method is put

¹ https://github.com/Event-AHU/Event_Camera_in_Top_Conference.

forward by Peter and others [16], the method involves the balance of the weights and thresholds, while achieving lower latency and requiring fewer operations. In [17], a sparse backpropagation method for SNN was introduced by redefining the surrogate gradient function form. Fang et al. [18] propose spike element-wise (SEW) ResNet to implement residual learning for deep SNNS, while proving that SEW ResNet can easily implement identity mapping and overcome the vanishing/exploding gradient problem of Spiking ResNet. Wang et al. propose a hybrid SNN-ANN framework for RGB-Event based recognition by fusing the memory support Transformer and spiking neural networks, termed SSTFormer [19]. Jiang et al. propose to aggregate the event point and voxel using absorbing graph neural networks for event-based recognition [20].

For point cloud based representation, Wang et al. [21] treat the event stream as a set of 3D points in space-time, i.e., space-time event clouds, and adopt the PointNet [22] architecture, which directly takes the point cloud as input and outputs the class label for the entire input or each point segment/part label for each input point. Xie et al. [23] propose VMV-GCN, a voxel-wise graph learning model designed to integrate multi-view volumetric data. Li et al. [24] introduce a Transformer network to directly process event sequences in its native vector tensor format to effectively represent the temporal and spatial correlations of input raw events, thereby generating effective spatio-temporal features for the task. Different from previous works, this paper designs an event recognition method based on Transformer and graph convolutional neural network, which transmits bimodal information through a specific method and learns a unified feature representation, so as to represent event data more effectively.

Graph Neural Networks. One notable application of GNNs in event data recognition is gait recognition. Wang et al. [15] propose a 3D graph neural network specifically designed for gait recognition. The model leverages the graph structure to capture the spatial and temporal dependencies in gait patterns. Bi et al. introduce the concepts of residual Graph Convolutional Neural Networks (RG-CNN) and Graph2Grid blocks [13, 14], which exploit graph structure to extract and exploit spatial and temporal information from event data. The Asynchronous, Event-based Graph Neural Networks (AEGNN) proposed by [25] addresses the processing of events as “evolving” spatio-temporal graphs. In the field of object recognition, Li et al. [26] introduce SlideGCN, a GNN-based model that focuses on fast graph construction using a radius search algorithm. Different from previous works, we adopt a graph Convolutional Neural Network (GCN) to process the graph data and connect the outputs of GCN and ST-Transformer module for accurate event-based pattern recognition.

Bottleneck Transformer. The traditional Transformer model has the problem of large computation and memory overhead when processing large-size images. Srinivas et al. [27] propose a novel network architecture called Bottleneck Transformer, which achieves dimensionality reduction of spatial attention by introducing a “bottleneck layer” between high-resolution and low-resolution representations of input features, thereby reducing computational consumption and increasing model scalability. Li et al. [28] introduce a local multi-head self-

attention mechanism and a novel position encoding method to solve the scalability bottleneck of Transformer under GPU memory constraints. Nagrani et al. [29] propose a multimodal bottleneck converter (MBT) and guided the bottlenecks in it to connect across modes. Song et al. [30] propose a new model BS2T that captures long-range dependencies between pixels in HS images by leveraging the self-attention mechanism in Transformers. In addition, we innovatively introduce bottleneck Transformer to promote the fusion of dual-stream information and improve the performance of module fusion.

3 Our Proposed Approach

3.1 Overview

Given an input event stream consisting of hundreds of thousands of events, our approach involves several steps to enhance the representation. Initially, we employ event frame stacking and voxel construction techniques to generate event frame and voxel representations, respectively. Subsequently, we utilize two intermediate representations, namely event frame and voxel graph, to capture the spatio-temporal relationships within the event stream. To further improve the feature descriptors for event frame and graph-based event representation, we propose a novel dual branch learning network. Finally, we combine these representations to create a comprehensive representation for event data, enabling effective recognition. The overall framework is depicted in Fig. 2. In the following sections, we provide a detailed explanation of each module.

3.2 Network Architecture

Input Representation. Considering the large amount of data and computational complexity, it is necessary to employ some down-sampling techniques to reduce the number of events. In this paper, we adopt two kinds of sampling techniques to obtain the compressed event representations. We first transform the asynchronous event flow into the synchronous event images by stacking the events in a time interval based on the exposure time. We also employ voxelization to obtain voxel representations. Specifically, given the original event stream \mathcal{E} with range H, W, T , we divide the spatio-temporal 3D space into voxels with the size of each voxel being h', w', t' . Hence, each voxel generally contains several events and the resulting event voxels in spatio-temporal space are of size $H/h', W/w', T/t'$. In practice, the above voxelization usually still produces tens of thousands of voxels. In order to further reduce the number of voxels and alleviate the effect of noisy voxels, we also adopt a voxel selection process to select top K voxels based on the number of events contained in each voxel. Let $\mathcal{O} = \{o_1, o_2 \dots o_K\}$ denote the collection of the final selected voxels. Each event voxel o_i is associated with a feature descriptor $\mathbf{a}_i \in \mathbb{R}^C$ which integrates the attributes (polarity) of its involved events. Hence, each $o_i \in \mathcal{O}$ is represented as: $o_i = (x_i, y_i, t_i, \mathbf{a}_i)$, where x_i, y_i, t_i denotes the 3D coordinate of each voxel.

Graph Neural Networks for Event Voxel Encoding. We similarly construct a geometric neighboring graph $G^o(V^o, E^o)$ for voxel event data \mathcal{O} . To be specific, each node $v_i \in V^o$ represents a voxel $o_i = (x_i, y_i, t_i, \mathbf{a}_i) \in \mathcal{O}$ which is described as a feature vector $\mathbf{a}_i \in \mathbb{R}^C$. The edge $e_{ij} \in E^o$ exists between node v_i and v_j , if the Euclidean distance between their 3D coordinates is less than a threshold R . We adapt Gaussian Mixture Model (GMM), convolution to learn the effective representations for voxel graph. To be specific, in each GCN layer, each event node v_i aggregates the features from its adjacency nodes as

$$f'_d(v_i) \leftarrow \sigma \left(\sum_{v \in V} \omega_d(v_i, v) f(v) \right), d = 1, 2 \dots D \quad (1)$$

where $\sigma(\cdot)$ denotes the activation function, such as ReLU. V denotes adjacency nodes of v_i . $\omega_d(v_i, v)$ denote the learnable convolution kernel weights. Finally, we adapt average graph pooling to get the global representation of voxel graph.

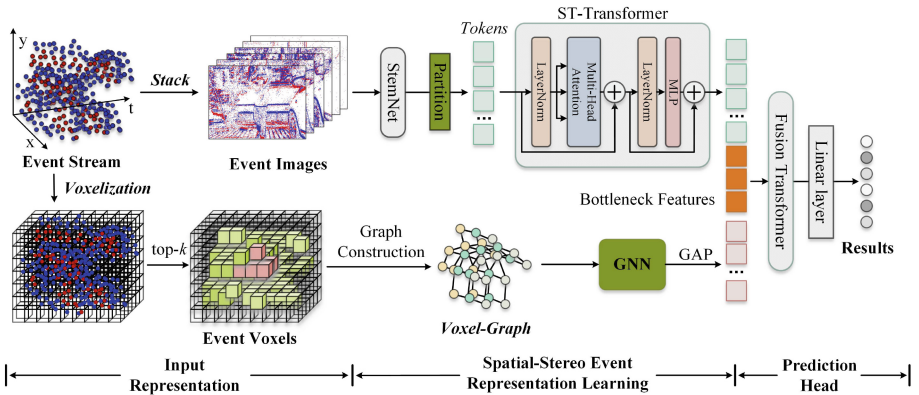


Fig. 2. An overview of our proposed Image-Voxel Feature Learning framework for event-based recognition.

Spatial-Temporal Transformer for Event Frame Encoding. After a series of data augmentation, each video sample obtained T event frames with a size of $H \times W$. We extract initial CNN features and embed event frames through StemNet (ResNet18 [2] is used in our experiments). After obtaining the initial features, we designed an ST-Transformer module to further achieve a better representation of spatio-temporal information. The proposed module consists of multi-head self-attention (MSA), MLP, and LayerNorm (LN). As shown in Fig. 2, T event frames are divided into N patches in spatial dimension, therefore, the $T \times N$ tokens can be obtained. We add learnable location encoding to these tokens and feed them into the ST-Transformer module to fully extract the enhanced spatio-temporal features, as shown in Eq. 2 and Eq. 3:

$$Y = X^{in} + MSA(LN(X^{in})) \quad (2)$$

$$X^{out} = Y + MLP(LN(Y)) \quad (3)$$

Bottleneck Transformer. In order to achieve the interaction between Event Images and Event Voxels information representations and learn a unified spatio-temporal context data representation. We also designed the Fusion Transformer module and introduced the Bottleneck mechanism. Specifically, let $X^{image} \in \mathbb{R}^{T \times N \times d}$ and $X^{voxel} \in \mathbb{R}^{1 \times d}$ represent the outputs of the previous ST Transformer and GNN modules, respectively. We first collect the $T \times N$ image and $T \times N$ randomly initialized Bottleneck tokens together and feed them to Fusion Transformer which includes multi-head self-attention (MSA) and MLP submodule, i.e.,

$$F^1 = [X^{image}, X^{bottleneck}] \in \mathbb{R}^{2 \times T \times N \times d} \quad (4)$$

$$\tilde{F}^1 = FusionTransformer(F^1) \quad (5)$$

We then split \tilde{F}^1 into two parts, i.e., the images feature representation \tilde{F}^{image} and the bottleneck feature representation $\tilde{F}^{bottleneck}$. The latter one will be concatenated with X^{voxel} and fed into the Fusion-Transformer module for interactive learning of the two representations. Similarly,

$$F^2 = [X^{bottleneck}, X^{voxel}] \in \mathbb{R}^{(T \times N + 1) \times d} \quad (6)$$

$$\tilde{F}^2 = FusionTransformer(F^2) \quad (7)$$

Finally, we concatenate both \tilde{F}^2 and \tilde{F}^{image} together and flattened them into a feature representation. After that, we utilize a two-layer MLP to output the final class label prediction, as shown in Fig. 2. We adopt the Negative Log Likelihood Loss function [31] to train the whole network.

4 Experiment

4.1 Dataset and Evaluation Metric

In this work, we utilized two datasets, namely DVS128-Gait-Day [10], N-MNIST [32], and ASL-DVS [14], to evaluate our proposed model. Here is a brief introduction to these datasets:

- **ASL-DVS** [14]: This dataset consists of 100,800 samples, with 4,200 samples available for each letter. The focus was on the 24 letters representing the handshapes of American Sign Language. Each video in this dataset has a duration of approximately 100 ms. The author captured these samples using an iniLabs DAVIS240c camera under realistic conditions.
- **DVS128-Gait-Day** [10] dataset is proposed for event-based gait recognition. It contains 4,000 videos corresponding to 20 classes. 20 volunteers are recruited for data collection using a DVS128 Dynamic Vision Sensor (the pixel resolution is 128×128).
- **N-MNIST** [32] dataset is obtained by recording the display equipment when visualizing the original MNIST (28×28 pixels). The ATIS event camera is used for the data collection and each event sample lasts about 10 ms. There are 70,000 event files for this dataset, the training and testing subset contains 60,000 and 10,000 videos, respectively. The resolution of this dataset is 28×28 .

Note that the top-1 and top-5 accuracy are employed as the evaluation metrics throughout our study.

4.2 Implementation Details

Our proposed dual-stream event-based recognition framework can be trained in an end-to-end manner. The initial learning rate is set as 0.001 and multiplied by 0.1 for every 60 epochs. We select eight frames for each video sample and divide each frame into eight tokens. For the constructed voxel graph, the threshold R is set to 2. The scale of the voxel grid is (4, 4, 4) for the ASL-DVS dataset. We select 512 voxels as the graph node for the structured graph representation learning. Our code is implemented using Python 3.8 and trained on a server with RTX3090 GPUs.

Table 1. Results on the ASL-DVS [14] dataset.

EST [33]	AMAE [34]	M-LSTM [35]	MVF-Net [36]	EventNet [37]
0.979	0.984	0.980	0.971	0.833
RG-CNNs [14]	EV-VGCNN [38]	VMV-GCN [23]	EV-Gait-3DGraph [9]	Ours
0.901	0.983	0.989	0.738	0.996

4.3 Comparison with Other SOTA Algorithms

As shown in Table 1, previous works already achieve high performance on the ASL-DVS [14] dataset. For example, the EST [33] (0.979), AMAE [34] (0.984), M-LSTM [35] (0.980), and MVF-Net [36] (0.971). Note that the GCN-based model, VMV-GCN [23], achieves better results, i.e., 0.989 on the top-1 accuracy. Thanks to the spatial-temporal feature learning and fusion network proposed in this work, we set new state-of-the-art performance on this dataset, i.e., 0.996. On the N-MNIST [32] dataset, as shown in Table 2, we also achieve SOTA performance compared with recent strong models. These comparisons fully validated the effectiveness of our proposed framework for event-based recognition. We provide two figures to better illustrate our results, as shown in the left subfigure of Fig. 3.

Table 2. Results on the N-MNIST [32] dataset.

EST [33]	M-LSTM [35]	MVF-Net [36]	Gabor-SNN [39]	EvS-S [26]
99.0	98.6	98.1	83.7	99.1
HATS [39]	EventNet [37]	RG-CNNs [14]	EV-VGCNN [38]	Ours
99.1	75.2	99.0	99.4	98.9

4.4 Ablation Study

To help researchers better understand the method we proposed, in this section, we conduct comprehensive experiments of component analysis on the DVS128-Gait-Day dataset and ASL-DVS dataset to check their influence on the overall model.

Component Analysis. Table 3 shows the effect of using different components on experimental results. In this part, we didn't use the Bottleneck, and the dataset we use is DVS128-Gait-Day. **Event image only** indicates that we only transform the event flow into the synchronous event images, which gets the result of 95.2. **Event voxel only** indicates that we only employ voxelization to obtain the compressed event representation and it achieves 98.0. We also use Event image and Event voxel together when obtaining event representation, denoted by **Event Image+Voxel**. It gets the result of 98.7. We can easily draw the conclusion by comparing the above three cases that using Event Image and Event voxel together can achieve higher performance, which reflects the effectiveness of our method.

Effect of Bottleneck. In this paper, we use Bottleneck Transformer to enhance the performance when fusing the modules. As shown in Table 4, **w/o Bottleneck Feature** means we do not feed the learning token into Bottleneck. It gets the result of 98.5. **w/o FusionFormer** means we do not use Fusion Transformer before the linear layer, in other words, we use the all components proposed in this paper except for the Bottleneck, and the result is 98.3. We found that compared with these two experiments, the result has increased after introducing the Bottleneck which indicates that the Bottleneck is a better choice for our framework. At the same time, a comparison with the experimental results in Table 1 shows that the experimental result drops when there is no Bottleneck, which also demonstrates Bottleneck has a positive effect on our proposed model (Table 3).

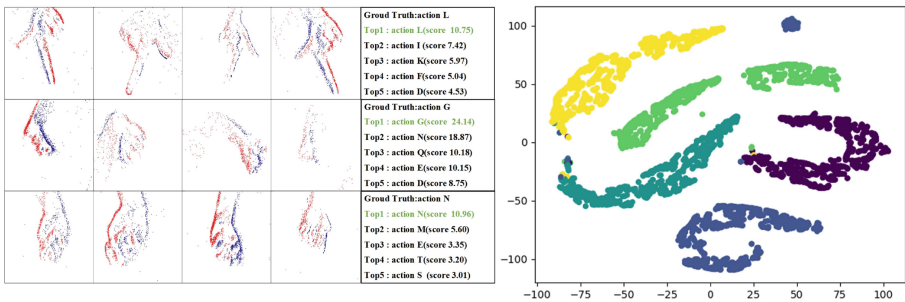


Fig. 3. Visualization of top-5 recognition results and feature distribution on the ASL-DVS dataset.

Table 3. Ablation study on DVS128-Gait-Day dataset [10].

Index	Component	Results
1	Event image only	95.2
2	Event voxel only	98.0
3	Event Image + Voxel	98.7

Table 4. Ablation study on ASL-DVS [14].

Index	Component	Results
1	w/o Bottleneck Feature	98.5
2	w/o FusionFormer	98.3

4.5 Parameter Analysis

The storage space of our proposed method is 220.3 MB. Our model spends 16.7 ms for each video in ASL-DVS dataset.

5 Conclusion

Previous event-based recognition approaches typically represented event streams as point clouds, voxels, or images, and employed various deep neural networks to learn feature representations. However, these approaches are usually challenged by monotonous modal expressions and the design of the network structure. To overcome these challenges, this paper introduces a novel dual-stream framework for event representation, extraction, and fusion. The proposed framework simultaneously models two common representations: event images and event voxels. By leveraging Transformer and Structured Graph Neural Network (GNN) architectures, spatial information and three dimensional stereo information can be learned separately. Moreover, the introduction of a bottleneck Transformer facilitates the fusion of the dual-stream information. Extensive experiments were conducted to evaluate the performance of our framework, using two widely used event-based classification datasets. The results demonstrate that our proposed framework achieves state-of-the-art performance. These findings highlight the effectiveness of the dual-stream framework in addressing the limitations of existing approaches and improving the recognition accuracy in event-based object recognition tasks.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 62102205).

References

1. ul Hassan, M.: AlexNet ImageNet classification with deep convolutional neural networks (2018)
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
3. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)

4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
5. Sun, Z., Ke, Q., Rahmani, H., Bennamoun, M., Wang, G., Liu, J.: Human action recognition from various data modalities: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(3), 3200–3225 (2023)
6. Wang, X., et al.: VisEvent: reliable object tracking via collaboration of frame and event flows arXiv preprint [arXiv:2108.05015](https://arxiv.org/abs/2108.05015) (2021)
7. Tang, C., et al.: Revisiting color-event based tracking: a unified network, dataset, and metric, arXiv preprint [arXiv:2211.11010](https://arxiv.org/abs/2211.11010) (2022)
8. Zhu, L., Wang, X., Chang, Y., Li, J., Huang, T., Tian, Y.: Event-based video reconstruction via potential-assisted spiking neural network. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3594–3604 (2022)
9. Wang, Y., et al.: EV-Gait: event-based robust gait recognition using dynamic vision sensors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6358–6367 (2019)
10. Wang, Y., et al.: Event-stream representation for human gaits identification using deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 3436–3449 (2021)
11. Fang, H., Shrestha, A., Zhao, Z., Qiu, Q.: Exploiting neuron and synapse filter dynamics in spatial temporal learning of deep spiking neural network, arXiv preprint [arXiv:2003.02944](https://arxiv.org/abs/2003.02944) (2020)
12. Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., Tian, Y.: Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2661–2671 (2021)
13. Bi, Y., Chadha, A., Abbas, A., Bourtsoulatze, E., Andreopoulos, Y.: Graph-based object classification for neuromorphic vision sensing. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 491–501 (2019)
14. Bi, Y., Chadha, A., Abbas, A., Bourtsoulatze, E., Andreopoulos, Y.: Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Trans. Image Process.* **29**, 9084–9098 (2020)
15. Wang, Y., et al.: Event-stream representation for human gaits identification using deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(7), 3436–3449 (2021)
16. Diehl, P.U., Neil, D., Binas, J., Cook, M., Liu, S.-C., Pfeiffer, M.: Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. *IEEE* (2015)
17. Perez-Nieves, N., Goodman, D.: Sparse spiking gradient descent. *Adv. Neural Inf. Process. Syst.* **34**, 11 795–11 808 (2021)
18. Fang, W., Yu, Z., Chen, Y., Huang, T., Masquelier, T., Tian, Y.: Deep residual learning in spiking neural networks. *Adv. Neural Inf. Process. Syst.* **34**, 21 056–21 069 (2021)
19. Wang, X., et al.: SSTFormer: bridging spiking neural network and memory support transformer for frame-event based recognition arXiv preprint [arXiv:2308.04369](https://arxiv.org/abs/2308.04369) (2023)
20. Jiang, B., Yuan, C., Wang, X., Bao, Z., Zhu, L., Luo, B.: Point-voxel absorbing graph representation learning for event stream based recognition arXiv preprint [arXiv:2306.05239](https://arxiv.org/abs/2306.05239) (2023)

21. Wang, Q., Zhang, Y., Yuan, J., Lu, Y.: Space-time event clouds for gesture recognition: from RGB cameras to event cameras. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1826–1835. IEEE (2019)
22. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 652–660 (2017)
23. Xie, B., Deng, Y., Shao, Z., Liu, H., Li, Y.: VMV-GCN: volumetric multi-view based graph CNN for event stream classification. *IEEE Robot. Autom. Lett.* **7**(2), 1976–1983 (2022)
24. Li, Z., Asif, M.S., Ma, Z.: Event transformer, arXiv preprint [arXiv:2204.05172](https://arxiv.org/abs/2204.05172) (2022)
25. Schaefer, S., Gehrig, D., Scaramuzza, D.: AEGNN: asynchronous event-based graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12 371–12 381 (2022)
26. Li, Y., et al.: Graph-based asynchronous event processing for rapid object recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 934–943 (2021)
27. Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., Vaswani, A.: Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision And Pattern Recognition, pp. 16 519–16 529 (2021)
28. Li, S., et al.: Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Adv. Neural Inf. Process. Syst.*, **32** (2019)
29. Nagrani, A., Yang, S., Arnab, A., Jansen, A., Schmid, C., Sun, C.: Attention bottlenecks for multimodal fusion. *Adv. Neural Inf. Process. Syst.* **34**, 14 200–14 213 (2021)
30. Song, R., Feng, Y., Cheng, W., Mu, Z., Wang, X.: BS2T: bottleneck spatial-spectral transformer for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–17 (2022)
31. Miranda, L.J.: Understanding softmax and the negative log-likelihood. [ljvmiranda921.github.io](https://github.com/ljvmiranda921) (2017)
32. Orchard, G., Jayawant, A., Cohen, G.K., Thakor, N.: Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* **9**, 437 (2015)
33. Gehrig, D., Loquercio, A., Derpanis, K.G., Scaramuzza, D.: End-to-end learning of representations for asynchronous event-based data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5633–5643 (2019)
34. Deng, Y., Li, Y., Chen, H.: AMAE: adaptive motion-agnostic encoder for event-based object classification. *IEEE Robot. Autom. Lett.* **5**(3), 4596–4603 (2020)
35. Cannici, M., Ciccone, M., Romanoni, A., Matteucci, M.: A differentiable recurrent surface for asynchronous event-based data. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12365, pp. 136–152. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58565-5_9
36. Deng, Y., Chen, H., Li, Y.: MVF-Net: a multi-view fusion network for event-based object classification. *IEEE Trans. Circuits Syst. Video Technol.* **32**(12), 8275–8284 (2021)
37. Sekikawa, Y., Hara, K., Saito, H.: EventNet: asynchronous recursive event processing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3887–3896 (2019)

38. Deng, Y., Chen, H., Chen, H., Li, Y.: EVVGCNN: a voxel graph CNN for event-based object classification, arXiv preprint [arXiv:2106.00216](https://arxiv.org/abs/2106.00216), vol. 1, no. 2, p. 6 (2021)
39. Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., Benosman, R.: HATS: histograms of averaged time surfaces for robust event-based object classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1731–1740 (2018)