TRANSFORMER-BASED DOMAIN ADAPTATION FOR EVENT DATA CLASSIFICATION

Junwei Zhao, Shiliang Zhang, Tiejun Huang

School of Computer Science, Peking University, Beijing, China

ABSTRACT

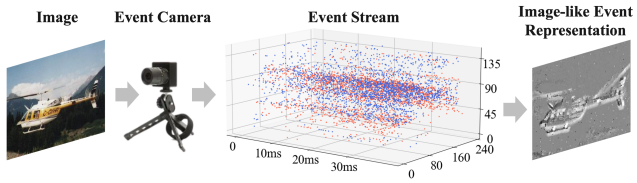
Event cameras encode the change of brightness into events, differing from conventional frame cameras. The novel working principle makes them to have stronger potential in high-speed applications. However, the lack of labeled event annotations limits the applications of such cameras in deep learning frameworks, making it appealing to study more efficient deep learning algorithms and architectures. This paper devises the Convolutional Transformer Network (CTN) for processing event data. The CTN enjoys the advantages of convolution networks and transformers, presenting stronger capability in event-based classification tasks compared with existing models. To address the insufficiency issue of annotated event data, we propose to train the CTN via the source-free Unsupervised Domain Adaptation (UDA) algorithm leveraging large-scale labeled image data. Extensive experiments verify the effectiveness of the UDA algorithm. And our CTN outperforms recent state-of-the-art methods on event-based classification tasks, suggesting that it is an effective model for this task. To our best acknowledge, it is an early attempt of employing vision transformers with the source-free UDA algorithm to process event data.

***Index Terms*—** Neuromorphic Signal Processing, Event Camera, Deep Learning Applications, Transfer Learning

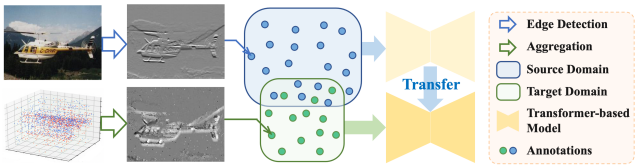
1. INTRODUCTION

Event cameras are commonly regarded as bio-inspired vision sensors [1, 2], where each pixel asynchronously measures brightness changes and emits an event once the changes exceed a threshold [3, 4]. The distinct working principle leads to advantages of high dynamic range, high temporal resolution, etc., thus allow it to tackle limitations confronted by standard cameras, e.g., the difficulty to record high-speed moving objects. Compared with frame-based cameras, event cameras present better potentials in high-speed applications [5].

Recent years have witnessed the success of frame-based cameras and deep neural networks in computer vision tasks. Deeper networks like ResNet and VGGNet have achieved significant performance on classification leveraging large-scale annotated datasets such as ImageNet [6]. Compared with frame cameras, event cameras are novel and have not been fully explored in computer vision community. Moreover, the novelty and high cost of such cameras result in the scarcity of labeled event data, which further limits the performance of deep learning frameworks on event-based vision tasks.



(a) The difference and correlation among image, event stream, and image-like event representation. Red/Blue points represent ON/OFF events, respectively.



(b) Leveraging labeled images to learn transferable features for event classification.

Fig. 1. We devise a transformer-based network to tackle event classification tasks via source-free UDA (i.e., transferring features from edge maps to event representations).

This paper tries to unlock the potentials of deep learning models in processing event data. To tackle the scarcity of labeled event annotations, we leverage large-scale image datasets to pre-train the model with UDA algorithms. UDA have been commonly used to alleviate the scarcity of labeled annotations in image classification. A typical approach in UDA is to use a feature extractor, e.g., deep neural networks, to learn transferable features in source domains and then transfer to target domain. Such method has achieved success in conventional image classification, but can not be directly applied to our task. This is mainly because the output of event cameras is asynchronous event streams, as illustrated in Fig. 1 (a). Most recent works convert events into image-like event representation such as Voxel Grid [7], which can be processed by off-the-shelf Convolution Neural Networks (CNNs). Considering that events and images can be regarded as different representations of the same visual cues, they share feature-level correlations. The difference is that events mainly contain edge cues due to the sensing principle of event cameras [8]. To reduce the domain gap between image and event, our insight is to transfer the features learned from the edge maps (extracted from images) to event domain, as

This work was supported by the National Key Research and Development Program of China (2018YFE0118400) and the National Natural Science Foundation of China (U20B2052, 61936011).

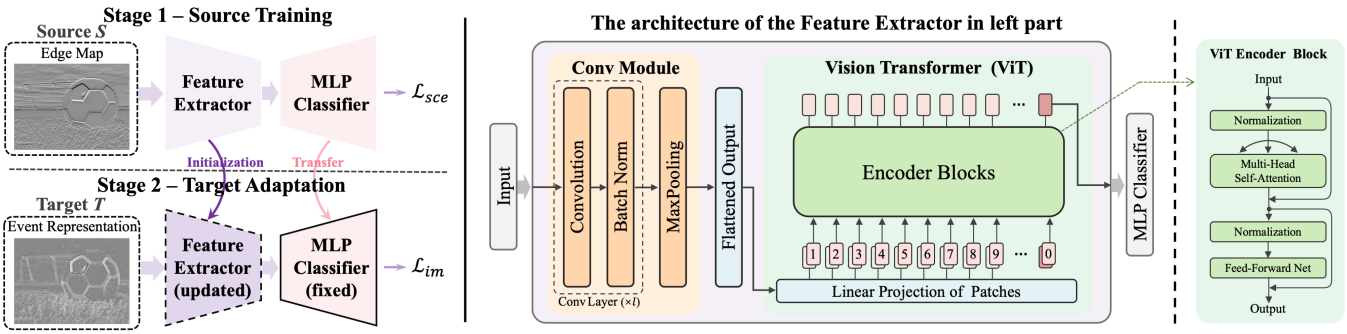


Fig. 2. Overview of the proposed method. The CTN is composed of a multi-layer convolution module and a vision transformer. The complementary combination of the two modules improves the classification performance on event data. The training involves two stages. First, we train the model on labeled edge maps (source) with the loss $\mathcal{L}_{sce}$. Afterwards, we freeze the classifier and update the feature extractor on unlabeled event representations (target) with the loss $\mathcal{L}_{im}$.

illustrated in Fig. 1 (b).

The property of event representations, e.g., lack details and textures, makes CNNs not efficient in processing event data. As each convolution filter of CNNs focuses on a small local region, they are more efficient in modeling local details. Different from the architecture of CNNs, Vision Transformers (ViT) [9] have exhibited stronger capability in modeling cues at larger scale, and have attained competitive performance on a wide range of vision tasks. Transformers utilizes self-attention mechanism to capture global interactions, thus is more effective in modeling long-range structures.

In this paper, we propose the Convolutional Transformer Network for event-based classification. The CTN comprises of a multi-layer convolution module and a ViT, as illustrated in Fig. 2. Convolution module is good at early visual processing, while ViT is efficient in capturing long-range features. The integration of CNN and ViT exploits the advantages of both models. However, insufficiency of event annotations has limited the performance of deep neural networks. Considering that event cameras extract edge cues from visual scenes, we propose to detect edges from RGB images to simulate event representations. Afterwards, we adopt the source-free UDA algorithm that leverages annotated edge maps to train deep networks and transfers the learned features to event domain. Extensive experiments demonstrate the effectiveness of domain adaptation algorithm. Moreover, our CTN achieves state-of-the-art (SOTA) performances on multiple event-based datasets.

2. METHOD

Source-free UDA only uses the pre-trained source model and without the requirement of source data in target adaptation stage. Given that N_s labeled samples $\{(x_s^i, y_s^i)\}_{i=1}^{N_s}$ from the source domain $\mathcal{S}$ where $x_s^i \in \mathcal{X}_s, y_s^i \in \mathcal{Y}_s$, and also N_t unlabeled samples $\{(x_t^i)\}_{i=1}^{N_t}$ from the target domain $\mathcal{T}$ where $x_t^i \in \mathcal{X}_t$, the goal of UDA is to predict the labels $\{y_t^i\}_{i=1}^{N_t}$ in the target domain, where $y_t^i \in \mathcal{Y}_t$. Here, our task aims to learn a target function $f_t : \mathcal{X}_t \to \mathcal{Y}_t$ and infer $\{y_t^i\}_{i=1}^{N_t}$, with only $\{(x_t^i)\}_{i=1}^{N_t}$ and the source function $f_s : \mathcal{X}_s \to \mathcal{Y}_s$ available.

In our task, the source data are the edge maps generated from RGB images, and the target data are events, as shown in Fig. 3. To make the target data can be directly processed by deep learning models, we convert asynchronous event streams into image-like event representations (illustrated in Sec. 2.1). To classify the event representations, we devise a hybrid model (illustrated in Sec. 2.2). The training of the model involves two stages, i.e., source training and target adaptation, which is introduced in Sec. 2.3.

2.1. Event Representation

Given an event stream $\varepsilon = \{(y_i, x_i, t_i, p_i)\}_{i=1}^{M}$, we convert it into a tensor-like representation $E \in \mathbb{R}^{H\times W\times C}$ for the processing of deep learning networks, where M is the number of events, $H\&W$ are the spatial resolution of the input ($(y, x) \in \mathbb{R}^{H\times W}$), t is the timestamp, and p is the polarity of events. A feasible choice is to encode the events into voxel grid [7] due to this method outperforms existing representations on multiple vision tasks. To be specific, the time domain of event timestamps $\Delta T = t_M - t_1$ need to be discretized into C temporal bins (similar to image-like channels). The resulting representation can be expressed as,

$$E_{b\in[0,C-1]} = \sum_{m=1}^{M} p_i \max(0, 1 - |t_m^* - b|), \qquad (1)$$

where $t_m^* = \frac{B-1}{\Delta T}(t_m - t_1)$ is the timestamp normalized to the range $[0, C-1]$.

2.2. CTN Architecture

To utilize the advantages of convolution in extracting low-level features and self-attention in building long-range dependencies, we incorporate the convolution module into the basic ViT. We first elaborate on the proposed convolution module. Then, we illustrate the basic components of ViT.

Convolution Module As illustrated in Fig. 2, the convolution module consists of multi-layer (l layers) convolution and BatchNorm (BN), followed by a MaxPooling (MP) layer. Given the event representations E, we need to extract low-level features through convolution module and thus

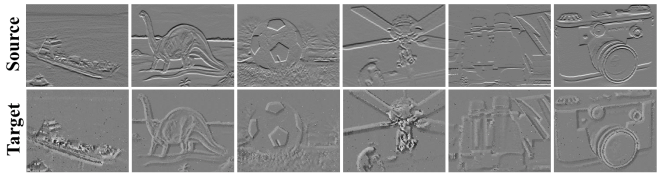


Fig. 3. Visualization of samples in source and target domain.

generate feature maps E_c. Suppose that the number of convolution channel is d, kernel size is k, convolution stride is s_c, padding is p and pooling stride is s_p, the output size (h, w) of the one-layer convolution module can be calculated as: $h = \lfloor (\frac{H+2p-s}{s_c} - 1)/s_p \rfloor$ (as same as w), and $E_c \in \mathbb{R}^{h \times w \times d}$. The above convolution process can be formulated as,

$$E_c = MP(BN(Conv(E))) \,. \tag{2}$$

Then, we reshape E_c into a 2D sequential patch $E_t \in \mathbb{R}^{N \times P^2}$ for the input of ViT, where (P, P) is the resolution of each patch, and $N = hw/P^2$ is the number of patch. These token embeddings are flattened and mapped to latent embeddings with a size of D. Besides, an extra trainable classification token is added into the embedding sequences for classification, resulting in the input with a size of $\mathbb{R}^{(N+1) \times D}$. Afterwards, the output of ViT is fed into a multi-layer perceptron for distinguishing classes.

We briefly revisit the basic components of ViT model as follows. As shown in Fig. 2, each ViT encoder block consists of Multi-head Self-Attention (MSA) and Feed-Forward Network (FFN). Residual connection and normalization are applied in each sub-modules to enhance the performance.

MSA Multi-head attention boosts the performance of single-head attention via extracting features from different representation subspace at different positions. For a single-head self-attention module, the sequence of input tokens E_t are first transformed into three vectors: query Q, key K and value V with dimension $Q, K, V \in \mathbb{R}^{(N+1) \times D}$. Subsequently, the calculation process of attention function among different vectors can be unified as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D}})V \,. \tag{3}$$

FFN A feed-forward network is followed after the self-attention module in each encoder blocks. It performs point-wise operations on each tokens separately, consisting of two linear transformation layers and a non-linear activation between layers. We formulate it as the following function:

$$FFN(X) = \sigma(XW_1 + b_1)W_2 + b_2 \,, \tag{4}$$

where X is the output of attention layer, W_1 and W_2 are the weights of the first and second layer, b_1 and b_2 are bias, and $\sigma(\cdot)$ is the non-linear activation of GELU.

2.3. Model Training

In the source training stage, we target to learn a source model with labeled image data. Given a source model $f_s : \mathcal{X}_s \to \mathcal{Y}_s$,

Conv Module Config				N-MNIST	CIF10-DVS	N-CAL101
l	s	n_k	s_p			
–	–	–	–	98.2%	75.3%	88.3%
1	3×3	16	2×2	98.5%	76.2%	89.1%
1	5×5	16	2×2	98.4%	76.9%	89.6%
1	3×3	32	2×2	98.7%	77.1%	89.9%
1	5×5	32	2×2	98.4%	77.3%	90.2%
1	5×5	32	4×4	98.3%	75.9%	88.7%
2	3×3	16	2×2	99.3%	77.4%	90.3%
2	5×5	16	2×2	99.1%	78.5%	91.7%
2	3×3	32	2×2	**99.7%**	78.2%	91.5%
2	5×5	32	2×2	99.3%	**79.3%**	**92.4%**
2	5×5	32	4×4	98.6%	77.5%	89.8%

Table 1. Ablation study results of CTN model on N-MNIST, CIFAR10-DVS (CIF10-DVS), and N-CALTECH101 (N-CAL101) datasets. Baseline model is ViT-B/32.

we train it with the label smoothing cross-entropy loss [10]:

$$\mathcal{L}_{sce} = -\mathbb{E}_{x_s \in \mathcal{X}_s} \sum_{k=1}^{K} \hat{y}_k \log \delta(f_s(x_s)) \,, \tag{5}$$

where $\hat{y}_k = (1 - \alpha)y_k + \alpha/K$, y_k is the desired one-hot label, K refers to the number of classes in source domain, $\delta(\cdot)$ is the softmax operation function, and α is the smoothing parameter which is empirically set to 0.1. Compared with standard cross-entropy loss, the label smoothing technique can increase the discriminability of source model.

In the target adaptation, we are given a pre-trained source model and unlabeled event data. We fix the classifier to maintain the data distribution information of the source domain, and update the feature extractor using the information maximization loss [10] as expressed in Eq. 6, which consists of a conditional entropy term and a diversity term:

$$\mathcal{L}_{im} = -\mathbb{E}_{x_t \in \mathcal{X}_t} \sum_{k=1}^{K} \delta(f_t(x_t)) \log \delta(f_t(x_t)) + \sum_{k=1}^{K} \hat{r}_k \log \hat{r}_k, \tag{6}$$

where $\hat{r} = \mathbb{E}_{x_t \in \mathcal{X}_t}[\delta(f_t(x_t))]$ is the mean of the softmax outputs for the current batch. Loss $\mathcal{L}_{im}$ can make the target outputs individually certain and globally diverse, and reduce the feture-level gap between target and source domains.

3. EXPERIMENT

3.1. Datasets

We evaluate the proposed method on three public neuromorphic datasets, i.e. N-MNIST [11], CIFAR10-DVS [12], and N-CALTECH101 [11] datasets. N-MNIST is the event conversion of MNIST dataset, which contains 10 classes of digits and 60000/10000 training/testing samples. CIFAR10-DVS cconsists of 10000 event streams in 10 classes. N-CALTECH101 is the event version of CALTECH101 dataset, which has the highest number of categories (101 classes) among existing neuromorphic datasets.

3.2. Implementation Details

Event streams are converted into event representations (C is set to 3) via the method described in Sec. 2.1. Then, we re-

Model	N-MNIST		CIFAR10-DVS		N-CALTECH101	
	S-only	S→T	S-only	S→T	S-only	S→T
VGG-11	80.7%	97.5%	50.2%	70.8%	68.4%	81.2%
VGG-13	81.4%	98.2%	51.7%	72.5%	70.2%	83.6%
VGG-16	83.1%	98.7%	53.1%	75.0%	73.3%	88.2%
VGG-19	83.5%	99.1%	53.8%	75.6%	73.5%	88.7%
ResNet-18	82.6%	98.6%	54.1%	76.5%	72.8%	88.5%
ResNet-34	84.3%	99.0%	55.7%	76.7%	73.2%	89.3%
ResNet-50	84.7%	99.3%	56.5%	77.3%	74.7%	90.1%
ViT-B/16	74.2%	97.9%	51.7%	75.2%	72.3%	88.1%
ViT-B/32	74.8%	98.2%	52.2%	75.3%	72.7%	88.3%
ViT-L/32	75.9%	98.3%	51.9%	74.6%	72.1%	87.7%
Ours	85.9%	**99.7%**	58.5%	**79.3%**	75.3%	**92.4%**

* S-only: Source-only, →: Domain Adaptation.

Table 2. Validation experiment results on the effectiveness of source-free UDA algorithm.

shape the spatial size of event representations to 256×256 and randomly cropping to 224×224 (a unified input size of ViT). RGB images are converted into edge maps and pre-processed by the method same as event representations. As for the models, we adopt the public backbones pre-trained on ImageNet [6], e.g., VGG-16, ResNet-34, and ViT-B/16. Training batch size N_b is fixed to 64 in all experiments. Training optimizer is SGD with the learning rate initialized as 0.01 and declined by a cosine decay schedule. Image augmentation strategies such as random flipping and rotation are performed.

3.3. Validation Experiments

Ablation Study: We test the influencing factors in the convolution module including the number of convolution layers (l), the kernel size (s), the number of kernels per layer (n_s), and the pooling stride (s_p). Other factors are set with empirical values, e.g., the convolution stride is 1×1, the padding size is $\lfloor s/2 \rfloor$. ViT-B/32 is adopted as the baseline. Experimental results are given in Tab. 1. The results in the first row show that, the performance suffers from a large drop without the convolution module. This suggests the effectiveness of the convolution structure. Besides, different configurations will influence the performance as well. Thus, we adopt the structure corresponding to the best performance (with underlines in the table) for each dataset in subsequent experiments.

Effectiveness of UDA: We conduct extensive experiments on a variety of models to verify the source-free UDA algorithm, as recorded in the Tab. 2. S-only denotes that training the model on source data only and directly testing on target data. S→T denotes that training the model via source-free UDA algorithm. The significant improvement of results suggests that UDA is effective for boosting the event-based classification accuracy. Note that the source-free UDA algorithm does not need the source data in the target adaptation stage, which improves the learning efficiency compared with other UDA methods. Besides, compared with conventional convolution models (i.e., VGGNet, ResNet) and pure ViT models (i.e., ViT-base), our CTN model achieves higher performance. Those results indicate that CTN is an effective model for event-based classification tasks.

Method	Year	Type	N-MNIST	CIFAR10 -DVS	N-CAL 101
SNN-SPA [13]	2020	SNN	96.3%	32.2%	-
SNN-IF [14]	2021	SNN	99.3%	63.7%	-
ASF-BP [15]	2021	SNN	-	62.5%	-
BSNN [16]	2021	SNN	99.5%	62.9%	-
H2L-SNN[17]	2021	SNN	99.0%	63.0%	-
E2V-CNN [18]	2019	ANN	98.3%	-	86.6%
Event BLS[19]	2020	ANN	98.4%	58.8%	72.2%
RG-CNN [20]	2020	ANN	99.0%	54.0%	65.7%
MatrixLSTM[21]	2020	ANN	98.9%	-	85.7%
VGCNN[22]	2021	ANN	99.4%	65.1%	74.6%
MVF-Net [23]	2021	ANN	99.3%	76.2%	87.1%
Deep SNN [24]	2020	A2S	99.0%	63.8%	-
ConvertSNN[25]	2020	A2S	99.5%	65.6%	-
Ours	2021	ANN	**99.7%**	**79.3%**	**92.4%**

Table 3. Comparison with Recent Works

3.4. Comparison with Recent Works

We compare the proposed method with SOTA works on N-MNIST, CIFAR10-DVS, and N-CALTECH101 datasets. As recorded in Tab. 3, our model achieves superior performances among existing methods. The compared methods can be regrouped into Spiking Neural Network (SNN), Artificial Neural Network (ANN), and ANN-to-SNN Conversion (A2S). Specifically, SNN includes: SNN-SPA [13], SNN-IF [14], ASF-BP [15], BSNN [16], H2L-SNN [17]; ANN includes: E2V-CNN [18], Event BLS [19], RG-CNN [20], MatrixLSTM [21], VGCNN [22], MVF-Net [23]; A2S includes: Deep SNN [24], ConvertSNN [25]. Among those works, ANN-based methods achieve superior performances. We observe that there are few SNN-based methods evaluated on the N-CALTECH101 dataset. The reason could be that this dataset is challenging for SNNs, thus requiring deeper models. However, the learning algorithms of SNNs are struggling in training very deep networks. Besides, most ANN-to-SNN conversion methods evaluate only on image datasets (e.g., MNIST and CIFAR10 dataset) while do not perform experiments on event datasets. This could be attributed to the mismatch of data type between images and events.

4. CONCLUSION

Event cameras, commonly regarded as bio-inspired sensors, have a large potential for computer vision in the challenging scenarios for conventional cameras. However, the difficulty of collecting and labelling event annotations largely limit the applications of such cameras in deep models. To address this issue, this paper adopts the source-free UDA algorithm to train deep learning networks, removing the dependency of labeled event annotations. Besides, we devise a transformer-based network to process event-based data. To the best of our acknowledge, it is an early work to reveal the potential of transformers in event-based classification tasks. The superior performance demonstrates the effectiveness of the proposed method. We believe that this work will ignite more efforts in neuromorphic vision community.

5. REFERENCES

[1] Tobi Delbruck, Bernabe Linares-Barranco, Eugenio Culurciello, and Christoph Posch, "Activity-driven, event-based vision sensors," in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2010, pp. 2426–2429.

[2] Shoushun Chen and Menghan Guo, "Live Demonstration: CeleX-V: a 1M pixel multi-mode event-based sensor," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 1682–1683.

[3] Jinjian Wu, Chuanwei Ma, Xiaojie Yu, and Guangming Shi, "Denoising of event-based sensors with spatial-temporal correlation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 4437–4441.

[4] Shane Harrigan, Sonya Coleman, Dermot Kerr, Pratheepan Yogarajah, Zheng Fang, and Chengdong Wu, "Neural coding strategies for event-based vision data," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 2468–2472.

[5] Aaron Chadha, Yin Bi, Alhabib Abbas, and Yiannis Andreopoulos, "Neuromorphic vision sensing for cnn-based action recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 7968–7972.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 989–997.

[8] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, et al., "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020.

[10] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng, "Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, doi:10.1109/TPAMI.2021.3103390.

[11] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in neuroscience*, vol. 9, pp. 437, 2015.

[12] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi, "Cifar10-dvs: an event-stream dataset for object classification," *Frontiers in neuroscience*, vol. 11, pp. 309, 2017.

[13] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 1308–1315.

[14] Jibin Wu, Yansong Chua, Malu Zhang, Guoqi Li, Haizhou Li, and Kay Chen Tan, "A tandem learning rule for effective training and rapid inference of deep spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2021, doi:10.1109/TNNLS.2021.3095724.

[15] Hao Wu, Yueyi Zhang, Wenming Weng, Yongting Zhang, Zhiwei Xiong, Zheng-Jun Zha, Xiaoyan Sun, and Feng Wu, "Training spiking neural networks with accumulated spiking flow," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 10320–10328.

[16] GC Qiao, N Ning, Y Zuo, SG Hu, Q Yu, and Y Liu, "Direct training of hardware-friendly weight binarized spiking neural network with surrogate gradient learning towards spatio-temporal event-based dynamic data recognition," *Neurocomputing*, vol. 457, pp. 203–213, 2021.

[17] Ling Liang, Zheng Qu, Zhaodong Chen, Fengbin Tu, Yujie Wu, Lei Deng, Guoqi Li, Peng Li, and Yuan Xie, "H2learn: High-efficiency learning accelerator for high-accuracy spiking neural networks," *arXiv preprint arXiv:2107.11746*, 2021.

[18] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3857–3866.

[19] Shan Gao, Guangqian Guo, Hanqiao Huang, Xuemei Cheng, and CL Philip Chen, "An end-to-end broad learning system for event-based object classification," *IEEE Access*, vol. 8, pp. 45974–45984, 2020.

[20] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 9084–9098, 2020.

[21] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci, "A differentiable recurrent surface for asynchronous event-based data," in *European Conference on Computer Vision*. Springer, 2020, pp. 136–152.

[22] Yongjian Deng, Hao Chen, Huiying Chen, and Youfu Li, "Evvgcnn: A voxel graph cnn for event-based object classification," *arXiv preprint arXiv:2106.00216*, 2021.

[23] Yongjian Deng, Hao Chen, and Youfu Li, "MVF-Net: A multi-view fusion network for event-based object classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, doi:10.1109/TCSVT.2021.3073673.

[24] Ruizhi Chen and Ling Li, "Analyzing and accelerating the bottlenecks of training deep snns with backpropagation," *Neural Computation*, vol. 32, no. 12, pp. 2557–2600, 2020.

[25] Alexander Kugele, Thomas Pfeil, Michael Pfeiffer, and Elisabetta Chicca, "Efficient processing of spatio-temporal data streams with spiking neural networks," *Frontiers in Neuroscience*, vol. 14, pp. 439, 2020.