Received 9 July 2023, accepted 6 September 2023, date of publication 18 September 2023, date of current version 26 September 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3316143

RESEARCH ARTICLE

CSTR: A Compact Spatio-Temporal Representation for Event-Based Vision

ZAID A. EL SHAIR, ALI HASSANI, (Member, IEEE), AND SAMIR A. RAWASHDEH, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, University of Michigan–Dearborn, Dearborn, MI 48128, USA
Corresponding author: Zaid A. El Shair (zelshair@umich.edu)

ABSTRACT Event-based vision is a novel perception modality that offers several advantages, such as high dynamic range and robustness to motion blur. In order to process events in batches and utilize modern computer vision deep-learning architectures, an intermediate representation is required. Nevertheless, constructing an effective batch representation is non-trivial. In this paper, we propose a novel representation for event-based vision, called the compact spatio-temporal representation (CSTR). The CSTR encodes an event batch's spatial, temporal, and polarity information in a 3-channel image-like format. It achieves this by calculating the mean of the events' timestamps in combination with the event count at each spatial position in the frame. This representation shows robustness to motion-overlapping, high event density, and varying event-batch durations. Due to its compact 3-channel form, the CSTR is directly compatible with modern computer vision architectures, serving as an excellent choice for deploying event-based solutions. In addition, we complement the CSTR with an augmentation framework that introduces randomized training variations to the spatial, temporal, and polarity characteristics of event data. Experimentation over different object and action recognition datasets shows that the CSTR outperforms other representations of similar complexity under a consistent baseline. Further, the CSTR is made more robust and significantly benefits from the proposed augmentation framework, considerably addressing the sparseness in event-based datasets.

INDEX TERMS Event-based vision, event representation, object recognition, data augmentation.

I. INTRODUCTION

Perception plays a crucial role in real-time robotic applications, enabling their operation in dynamic and unpredictable environments [1], [2]. These applications often operate under challenging lighting conditions, including high dynamic range (HDR) or high-speed motion scenes. Ensuring accurate perception and prompt responses under such conditions is vital for their success, especially in safety- or time-critical applications like autonomous vehicles [1] and industrial automation [2]. For instance, in an HDR scene such as when emerging from a tunnel in broad daylight, the failure to detect objects like vehicles or traffic signs can have severe consequences [3]. To address the challenges of robust operation in challenging lighting conditions (*e.g.*, HDR or high-speed motion scenes) and in potentially dynamic and unpredictable environments, many researchers have increasingly turned to event-based vision [4], [5] as a promising alternative visual sensing modality.

Event-based sensors, such as the Dynamic Vision Sensor (DVS) [6] or the Asynchronous Time-Based Image Sensor (ATIS) [7], operate by capturing per-pixel brightness changes asynchronously and at very high temporal resolutions [6], [7]. This results in a spatially sparse yet temporally dense output that effectively represents all visual changes in a scene over a specified time interval. In contrast, traditional cameras capture intensity images at a fixed rate, such as 24 frames per second [8]. This fixed rate can possibly lead to oversampling of static scenes, resulting in redundant data; or undersampling of scenes with high-speed motion, resulting in motion blur [5]. Overall, event-based vision offers several distinct properties that address dynamic range, response

The associate editor coordinating the review of this manuscript and approving it for publication was Claudio Loconsole.

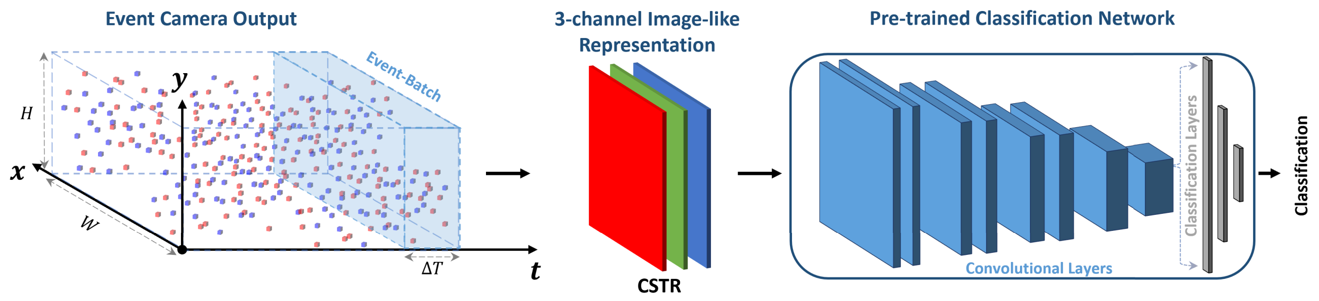


FIGURE 1. **Overview of the general framework of this paper. Sparse and asynchronous events, representing brightness changes at each pixel, are captured using an event-based sensor. To utilize this spatio-temporal event data, an intermediate representation is required to leverage modern deep-learning solutions when processing events in batches. In this work, we propose the Compact Spatio-Temporal Representation (CSTR) that encodes spatial, temporal, and polarity information of event data in a 3-channel image-like format. Accordingly, the CSTR is directly compatible with off-the-shelf pre-trained computer vision architectures.**

time, and motion blur issues. These properties include an HDR of >120 dB, microsecond-level temporal resolution, low output latency in the order of microseconds, and low power consumption averaging a few milliwatts [5], [6]. Consequently, these characteristics make event-based vision particularly well-suited for real-time robotic applications [9], [10]. Such applications require accurate perception and prompt response to visual changes, especially in challenging scenarios such as HDR scenes [11], low-light conditions [12], or high-speed motion environments [13]. In comparison, traditional cameras often struggle to perform effectively in such scenarios [3], [10].

While the properties of event-based vision are very compelling, effectively utilizing event data in various applications presents a challenge. The generated event stream is asynchronous and sparse, necessitating its transformation into a compatible format for established algorithmic methodologies. For instance, most traditional object detectors and classifiers employ a three-channel input designed for RGB imagery [14], [15]. However, the independence and sparsity of events make it non-trivial to establish batch relationships, often leading to the creation of hand-crafted representations tailored to specific applications [16], [17]. This inherent problem hampers generalization, as traditional frame-based cameras benefit from standardized formats that facilitate the canonical transfer learning of dataset weights across tasks. In contrast, event-based algorithms, are highly sensitive to the specific type of open-source data and its representation. This further exacerbates the data sparsity issue. As a result, the data needs to be closely associated with the particular task at hand, adversely impacting generalization and posing challenges for training convergence.

Accordingly, most works resort to using image-like representations in order to leverage pre-trained computer vision models. One common representation is the Event Frame [18], [19], chosen for its simplicity. This representation keeps track of whether any event has occurred at each pixel within a given time period (where the time period is a variable that can be adjusted per task). By doing so, the batch of events is effectively transformed into a single-channel image (or can be replicated to form a 3-channel image) that can be utilized with existing algorithms. While convenient, this approach has some limitations. Notably, it binarizes the behavior for the specified sampling period, losing temporal and polarity information (brightness changes), and is generally outperformed by more sophisticated approaches [20], [21], [22], [23]. Alternatively, more advanced representations have been explored to capture temporal and polarity contexts [20], [21], [22], [23]. These representations demonstrate better performance, but they come with either the trade-off of notable pre-processing overhead [20], [21] or are not directly compatible with pre-trained computer vision architectures that require a 3-channel input [22], [23].

To address these challenges, we propose a novel representation for event data called the Compact Spatio-Temporal Representation (CSTR). The CSTR efficiently encodes the spatial, temporal, polarity, and event count information of a given event batch while requiring minimal processing overhead. This is achieved by calculating the mean timestamps of the events per polarity type (positive or negative) and the normalized event counts at every spatial position in the resulting representation frame. This results in a 3-channel image-like format that is directly compatible with existing state-of-the-art networks [14], [15], allowing for seamless integration without the need for additional modifications. We visualize the general framework of this paper in Fig. 1.

We demonstrate the effectiveness of the CSTR through a comprehensive series of well-established event-based recognition benchmarks. This benchmarking includes six well-known representations that are similarly compatible with off-the-shelf networks over the following datasets: N-MNIST [24], N-CARS [25], N-Caltech101 [24], CIFAR10-DVS [26], ASL-DVS [27], and DVS-Gesture [28]. The CSTR is consistently an excellent performer, achieving the highest overall classification accuracy. Furthermore, the CSTR is stable when applying random augmentations; these are demonstrated to notably enhance classification accuracy, validating that the CSTR is a robust approach for encoding event data.

We summarize the contributions of this work as follows:

- We introduce the compact spatio-temporal representation (CSTR) for event-based vision, which efficiently encodes the spatio-temporal information of events in a 3-channel image-like format, directly compatible with modern computer vision architectures.
- We provide a comprehensive evaluation of the CSTR against foundational event representations of similar complexity using six event-based recognition datasets.
- We propose an augmentation framework for event data, significantly improving the performance of the CSTR and other spatio-temporal representations.
- We demonstrate the effectiveness of the CSTR and the data augmentation framework when combined with off-the-shelf pre-trained classifiers.

Our source code is available at: https://github.com/Zelshair/cstr-event-vision.

II. RELATED WORK

Event-based vision has recently seen significant advancements that leverage its unique characteristics for various applications [5], [10], [12], [13], [23]. There are two general approaches to effectively utilize the asynchronous and sparse event data. These include *event-by-event* and *batch* processing. In this section, we provide an overview of the relevant methods of each approach, highlighting their strengths and identifying their limitations. Next, we provide an overview of augmentation methods explored in the literature for enhancing event data. Finally, we introduce the proposed CSTR along with a new augmentation in the context of these limitations, noting how they address some of the remaining challenges.

A. EVENT-BY-EVENT PROCESSING

Event-by-event processing methods directly utilize events as they are received [29], [30], [31], [32]. This approach is intuitive and minimizes processing delays. The most prominent methods are spiking-neural-networks (SNNs) [32], [33], [34], [35], [36]. An SNN is a bio-inspired version of artificial neural networks comprising interconnected neurons. SNNs operate by integrating incoming spikes (events at the input layer) over time. An output spike is generated when the membrane potential of a neuron surpasses a certain threshold causing it to reset. The generated output spikes propagate information to other neurons in deeper layers, connected hierarchically. This neuron-activation threshold enables SNNs to be computationally efficient [35], [36], [37].

Despite the computational efficiency and minimal latency of event-by-event algorithms, they suffer from some limitations. Processing events individually inherently lacks temporal context, necessitating tailored solutions to compensate for the lack of event history [29], [30], [31]. Ironically, this approach can become computationally expensive during periods of high event density. Scenes with significant motion and texture can generate a substantial amount of events per second, requiring a proportional number of operations. As event-based sensors continue to improve their frame resolutions [8], [38], this computational challenge will only intensify. While SNNs somewhat address the latter with their energy-efficient design, they are non-trivial to set up and implement [32], [33], [34]. Moreover, SNNs require specialized hardware, which limits their widespread adoption, posing additional barriers to deployment.

B. BATCH PROCESSING

Batch processing methods accumulate, encode, and classify the events generated in a given time period. These approaches add temporal context with the capability to provide synchronous responses (*i.e.*, a classification per each batch period). By applying an intermediate encoding method, they have the key benefit of being able to employ modern computer-vision networks. This is directly germane to the problem statement of being able to leverage existing state-of-the-art networks (and corresponding training weights). Hence, we focus this survey on event-batch representations that are compatible with frame-based networks.

1) IMAGE-LIKE REPRESENTATIONS

Many opt to represent event batches in a simple image-like format. These representations encode spatial, temporal, and/or polarity information into traditional one, two, or three-channel images. Such approaches are popular because they enable rapid prototyping and demonstrate strong performance across various perception tasks [18], [19], [39]. For example, the Event Frame encodes the event's spatial information (*i.e.* the existence of any events per spatial position) [18], while the Event Count (also known as Event Histograms) [39], [40], [41] indicates the number of events recorded, instead. More advanced versions of these representations incorporate polarity information as well [19], [39], [41]. These representations, however, are inherently limited as they do not capture the temporal information of the event data. To address this limitation, more comprehensive representations have been developed to incorporate spatio-temporal information in an image-like format. One popular representation is Timestamp Images [42], also referred to as Time Surfaces [17]. Timestamp Images encode the timestamp information of the latest event at each spatial index [42], often represented using a separate channel per polarity type resulting in a 2-channel representation [42]. Recent advancements related to Timestamp Images have explored sophisticated techniques to enhance robustness against noise [25], [43]. For instance, DiST [43] incorporates temporal discounting by considering the ρ spatio-temporally neighboring events at each spatial position. Thus, discounting the timestamps of the latest events using a normalized time range of the neighboring pixels.

One challenge encountered in temporal representations is motion overwriting. While timestamp images excel in retaining contour information, the recent timestamps can be overwritten. This can happen when using long batch periods or in highly textured scenes. Accordingly, various representations have emerged that incorporate both the temporal and count information of events in different forms [44], [45], [46], [47]. For instance, a 4-channel representation, known as Event Image [45], [46], incorporates recent timestamps and event count per polarity. Another work by Bai *et al*. [47] proposes a more compact 3D representation that includes the temporal information of both polarities as well as the event count in separate channels. This forms a spatio-temporal image-like representation that encompasses vital information about the event data. The authors also investigate the advantages of this approach in the context of event-based object recognition.

Overall, the limitation of most spatio-temporal image-like representations can be distilled to overlapping events. A high number of overlapping events often results when using long batch periods or when operating in highly textured scenes. This can result in the overwriting of recent events causing a loss of information. Shortening the batch period can potentially limit this issue [45], however, this reduces temporal context and increases processing frequency.

As an alternative, image reconstruction from events is an effective approach that results in intensity images that enable the direct use of modern frame-based computer vision architectures [48]. However, generating images from events is a very processing-heavy task, making it not very suitable for real-time systems.

2) ADVANCED 4D GRID-LIKE REPRESENTATIONS

Advanced grid-like representations have been proposed to overcome the issue of event overlapping, thus, retaining more information [22], [23]. For example, TORE volumes [23] utilizes a first-in-first-out buffer at each spatial position to retain the temporal information of the last K events, for both polarity types, where $K > 1$. This results in a 4D representation with a resolution of $2 \times K \times H \times W$, where H and W are the frame's height and width, respectively. By doing so, TORE volumes [23] limit the problem of event-overwriting which is often encountered in image-like representations.

Another notable representation is Event Spike Tensors (EST) [22]. EST employs an end-to-end learning approach to derive event representations from input data. This is achieved by applying convolutional operations on a batch of events with a learned kernel comprising a multi-layer perceptron with two hidden layers. Then, the resulting convolutions are discretized, yielding a 4D grid-like representation with dimensions of $2 \times B \times H \times W$, where B is the pre-selected number of temporal bins.

Although these representations demonstrate remarkable performance in a multitude of tasks [22], [23], it is important to note that the choice of compatible deep learning architectures is somewhat limited. Consequently, an additional quantization step is often required to convert the 4D representation into a 3D format [22]. An alternative approach involves splitting the 4D grid along the polarity dimension (first dimension) and employing multiple deep learning models in parallel to process the resulting outputs, or modifying the input layers of a deep learning model to accommodate the higher-dimensional input. However, both approaches may lead to higher memory and computational requirements due to the increased dimensionality of the inputs.

3) VOXEL GRIDS

Voxel grids offer a precise means of capturing the spatial and temporal characteristics of events. A voxel represents a 3D point, traditionally denoting the height, width, and depth coordinates in a 3D model. Combining these voxels creates a 3D structure known as a voxel grid. Voxel grids are widely used in 3D computer vision, especially for representing a LiDAR-generated point cloud [49]. Similarly, it can be also used to handle sparse event data. Voxel grids are applied to event batches by converting the depth axis to a temporal axis using B temporal bins per event batch. This conversion is typically achieved through spatio-temporal quantization employing a designed sampling kernel. The resulting voxel grid has dimensions of $B \times H \times W$, allowing it to retain the essential spatio-temporal relationships within the event batches [16], [21], [50]. Accordingly, researchers have explored the application of voxel grids in various computer vision tasks, including optical flow estimation [16], [21], HDR video reconstruction [50], and object recognition [51].

Despite their advantages, the use of voxel grids poses two primary challenges. Firstly, generating voxel grids can be computationally demanding, especially when utilizing sophisticated sampling kernels. Secondly, the adoption of voxel grids may lead to high memory requirements due to the resulting increased input dimensionality, similar to the challenges with 4D representations discussed earlier. This issue becomes particularly prominent with high-resolution grids (*i.e.*, a large number of bins B) and long batch periods.

4) GRAPH-BASED REPRESENTATIONS

Alternative to voxel-grids, events can be represented as graphs [20], [27], [52]. Here, each sampled event in an event batch is treated as a vertex v_i. These vertices v (also referred to as nodes) are then connected to each other using edges ε, based on a pre-defined spatio-temporal distance metric, forming the graph G. This approach similarly captures the temporal relationships within the event batch and offers compatibility with existing graph-convolutional networks (GCNs) [20], [27]. Graph-based solutions provide flexibility in the processing of the event data, allowing for a natural way to incorporate their spatial and temporal information [20], [27], [52]. Compared to traditional CNNs, GCNs exhibit significantly lower inference computational complexity [52].

Nevertheless, generating the graphs can be computationally demanding. This is particularly true when dealing with high-density event streams, resulting in a large number of vertices and edges [53]. Consequently, it is often necessary to sample a subset of events from the batch to reduce storage and computational costs [20], [52]. Moreover, unlike CNNs in traditional computer vision, there is limited availability of GCN models pre-trained on large-scale datasets. This hampers the ability to leverage transfer learning. As a result, researchers often develop their own GCN architectures to accommodate the generated graphs [20], [27], [52].

C. AUGMENTATION METHODS FOR EVENT-BASED VISION

Data augmentation techniques play a crucial role in enhancing the performance and generalization of deep learning models. Given the limited availability of labeled event-based datasets, augmentation methods offer an effective approach to expand the training data and improve model robustness. In this subsection, we provide an overview of the different augmentation methods proposed for event data.

Li *et al*. [54] propose several randomized geometric augmentations for training SNNs. These include common techniques such as horizontal flip, translation, and rotation; as well as other unique techniques such as cutout, shear, and CutMix. These transformations introduce variations and enhance model performance. Gu *et al*. [55] introduce EventDrop, an augmentation framework for randomly dropping events within an event batch. It explores various event-dropping techniques, including dropping events within a random time period, pixel area, or a random portion of the sampled events. EventDrop improves robustness and has been evaluated for event-based object recognition. The authors also explore the use of EventDrop on different combinations of event representations and pre-trained classification models. EventMix [56] presents an advanced augmentation framework that uses a random 3D mask to mix different event-batch samples and their labels. This mixing technique enhances the diversity of the training data and has been evaluated on a set of event-based recognition benchmarks as well. Naeini *et al*. [57] propose spatial, noise, and time-series augmentations to improve contact-force estimation. Spatial augmentations include rotations and resizing. Noise augmentations add sequences of noise to the dataset, which are generated by recording similar sequences without any movement. Time-series augmentations include frame-shifting, which shifts all generated batch-representation frames within a given sequence; and temporal event shifting, where a fraction of events are randomly selected and removed from one frame and appended to an adjacent frame. For both types of time-series augmentations, the authors explore a fixed index-shift range of +3 to −3. These augmentation methods, along with others proposed in the literature, contribute to addressing the dataset scarcity issue in event-based vision. By applying these techniques, models can better handle variations in event data and improve their generalization capabilities. However, despite their importance, event data augmentation techniques are still not thoroughly explored in the literature.

D. LITERATURE CONTRIBUTION

In this paper, we present the CSTR, an alternative image-like representation for event-based vision. The CSTR offers a comprehensive representation of sparse event data when processed in batches while requiring minimal memory resources. It provides a choice that eliminates the need for manual parameter tuning and can be generated in an online manner. It is important to note that the CSTR is not meant to replace advanced or more sophisticated representations. Rather, it serves as an excellent representation choice for initial proof-of-concept and facilitates the rapid deployment of event-based solutions. This is due to the compact 3-channel image-like format of the CSTR, which enables the direct utilization of state-of-the-art computer vision architectures.

To validate the effectiveness of the CSTR, we conduct several experiments on various event-based recognition benchmarks comparing it to other image-like representations of similar complexity using various pre-trained classification networks. Additionally, we supplement our representation with several randomized augmentation methods that impact different components of events, including spatial, temporal, and polarity. These augmentation techniques further contribute to improving the performance and the generalization capabilities of event-based vision models.

III. METHODOLOGY

In this section, we present our proposed event-based representation. First, we provide a detailed overview of how events are generated. Then, we define the common and foundational image-like representations that form the basis of our work. These representations fundamentally encode the spatial and/or temporal components of events within the event batch. By analyzing the characteristics of these representations, we derive a more advanced spatio-temporal representation that enhances performance. We visualize these representations on the evaluation datasets in Fig. 2 (see: next page). Given that our approach aims to improve temporal context, we also introduce a novel temporal augmentation technique to address the sparseness of training data.

A. EVENT GENERATION MODEL

In contrast to traditional cameras, event-based sensors capture per-pixel brightness changes, asynchronously [6]. At a given pixel (x, y), an event e is generated whenever the logarithmic change in brightness intensity exceeds a predefined contrast threshold C. This can be expressed as follows:

$$|log(I(x, y, t)) - log(I(x, y, t - \Delta t))| \geq C, \tag{1}$$

where $I(x, y, t)$ represents the intensity measurement at spatial position (x, y) at time t, and Δt represents the time duration since the last generated event at the same spatial position. The polarity p of an event is determined by the sign of the brightness change. A brightness increase (on event) is assigned $p = +1$, while a brightness decrease (off event) is assigned $p = -1$. Thus, $p \in \{+1, -1\}$.

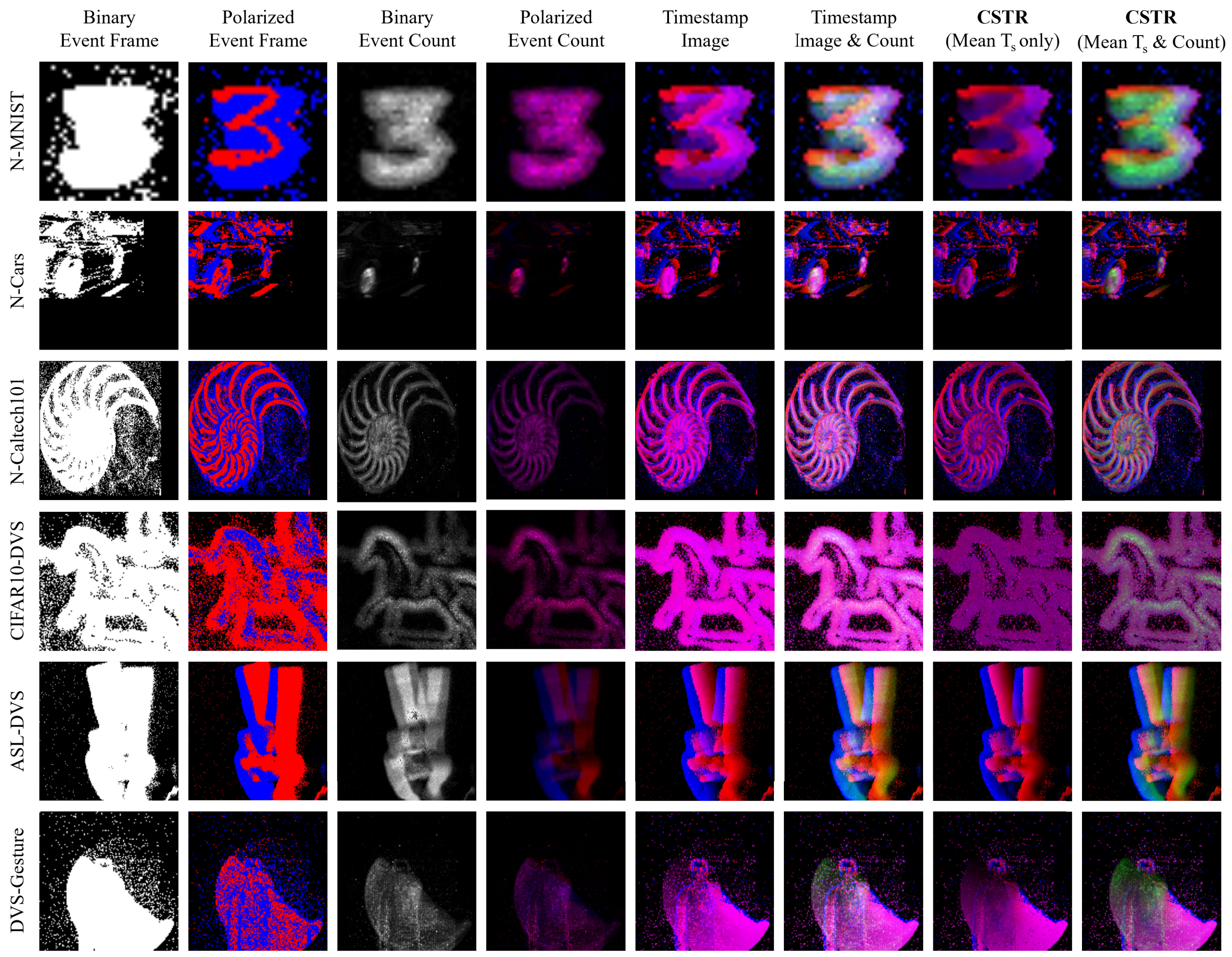


FIGURE 2. **Visualizations of the CSTR as well as the foundational event representations investigated in this work using various object and action recognition datasets. To enable visualization, we normalize the Binary and Polarized Event Count representations. Further, due to the significant event noise present in the N-Caltech101 [24] samples, we amplify the event count channels by a factor of 20 to improve visualization. This is shown in the 3rd row, columns 3, 4, 6, and 8.**

Event-based sensors report each captured event e_i as a combination of a microsecond timestamp t_i, a polarity p_i, and a two-dimensional spatial coordinate (x_i, y_i). In general, an event stream ε composed of n sequential events can be denoted as:

$$\varepsilon \rightarrow \{(t_1, x_1, y_1, p_1), (t_2, x_2, y_2, p_2), \ldots, (t_n, x_n, y_n, p_n)\}. \tag{2}$$

Events can be grouped into batches either based on a specified batch-sampling period ΔT or a fixed number of events. In this work, we focus on event batches accumulated using predefined batch periods to enable a synchronous response.

The event generation process outlined above captures the spatio-temporal dynamics of the scene. This is done by detecting changes in brightness intensity and encoding them as events with corresponding timestamps, spatial coordinates, and polarities.

B. FOUNDATIONAL EVENT REPRESENTATIONS

To represent a batch of events ε captured during a sampling period ΔT, several image-like representations can be formed. We identify five foundational approaches identified in the literature: Binary Event Frame, Polarized Event Frame, Binary Event Count, Polarized Event Count, and Timestamp Image. While these representations are not typically referred to as *Binary* or *Polarized*, we use these terms to distinguish between them clearly. We detail these approaches next.

1) BINARY EVENT FRAME

The Binary Event Frame binarizes whether any events are detected at a given spatial location. Each pixel position in the resulting two-dimensional $H \times W$ representation can be encoded as follows:

$$F_{\text{bin}}(x, y) = \begin{cases} 1, & \text{if } x = x_i \ \& \ y = y_i \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where x_i and y_i are the spatial coordinates of each event e_i in the batch ε. We encode the presence of an event as 1 and the absence of any as 0. This representation is visualized in Fig. 2, column one. Note how this approach is very simplistic and has low contrast; this is because it is highly sensitive to motion-overlapping, where multiple events occur at the same spatial location, as well as noise captured by the event camera. Accordingly, this representation suffers from frame saturation which results under almost any batch-sampling duration, as shown in Fig. 2.

2) POLARIZED EVENT FRAME

The Binary Event Frame can be extended to include polarity information. The Polarized Event Frame incorporates this in a $2 \times H \times W$ 3D matrix. The event batches are defined by:

$$F(x, y, p) = \begin{cases} 1, & \text{if } x = x_i \ \& \ y = y_i \ \& \ p = p_i \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

where x_i and y_i are the spatial coordinates and p_i is the polarity of each event e_i. We similarly encode detected events by 1 and the absence of events as 0, but for each polarity. This representation is visualized in Fig. 2 (second column), showing a notable contrast improvement. Similar to the Binary Event Frame, this representation also suffers from frame saturation. Accordingly, both Event Frame representations are more effective when generating batches based on a constant number of events (ideally a low number) instead of a fixed sampling duration [19].

3) BINARY EVENT COUNT

Alternative to the Binary Event Frame, the Binary Event Count representation captures the number of events at each spatial position. We encode this with the following equation:

$$C_{\text{bin}}(x, y) = \sum_{i=1}^{n} [x = x_i \ \& \ y = y_i], \tag{5}$$

where n is the number of events. The Iverson bracket here would be equal to 1 if the expression is true, which is whenever an event has the same spatial location as the pixel (x, y). This representation retains more information about the scene at each spatial location. Moreover, as visualized in Fig. 2 (third column), this representation shows high temporal precision, albeit at the cost of less sharp contour details.

4) POLARIZED EVENT COUNT

Analogous to the Polarized Event Frame, the Binary Event Count can be extended to include event-polarity context. We similarly represent this with a $2 \times H \times W$ matrix as follows:

$$C(x, y, p) = \sum_{i=1}^{n} [x = x_i \ \& \ y = y_i \ \& \ p = p_i], \tag{6}$$

where n is the number of events, x_i and y_i are the spatial coordinates and p_i is the polarity of each event e_i. This is visualized in Fig. 2 (fourth column), improving the contour details (though still not as sharp as the Polarized Event Frame). In contrast to the Event Frame representations, the Binary and Polarized Event Count representations do not suffer from frame saturation. Instead, they are robust to long batch-sampling durations, as shown in Fig. 2. Nevertheless, both Event Count representations require significant motion overlap and high event-density streams to yield a meaningful signal.

5) TIMESTAMP IMAGE

An alternative approach to tracking the number of events is to identify the most recent timestamp instead. This is achieved using the Timestamp Image representation [42], which is a 3D matrix of size $2 \times H \times W$. Assuming that the batch's events are sorted in chronological order (*i.e.*, from oldest to newest) we obtain this representation as follows:

$$T_s(x, y, p) = \begin{cases} \dfrac{t_i - t_s}{\Delta T}, & \text{if } x = x_i \ \& \ y = y_i \ \& \ p = p_i \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where t_s is the raw time offset representing the start of the event batch with temporal duration ΔT, and t_i is the timestamp of the event e_i. In (7), $T_s(x, y, p)$ represents the normalized timestamp (in the range of $[0, 1]$) of the latest event occurring at the pixel location (x, y) and polarity p. The subtraction of t_s removes the time offset from each event's timestamp. This representation is visualized in Fig. 2 (fifth column), where the normalized recent timestamp further improves contour details over the naive Event Frame representations. Note, however, that this improved contrast diminishes under high-density event streams with long batch periods. Additionally, the Timestamp Image is also susceptible to noise in more recent events.

6) COMBINING TIMESTAMP IMAGE AND EVENT COUNT

Given the inherent limitations of the Timestamp Image and the Event Count representations, combining them can enhance their robustness [47]. To achieve this, we concatenate the two-channel Timestamp Image T_s, defined in (7), with the normalized one-channel Binary Event Count. The normalized Binary Event Count $\hat{C}_{\text{bin}}$ is defined as follows:

$$\hat{C}_{\text{bin}}(x, y) = \frac{C_{\text{bin}}(x, y)}{\max(C_{\text{bin}})}, \tag{8}$$

where $\max(C_{\text{bin}})$ is the maximum event count in the frame. This combination results in a $3 \times H \times W$ 3D matrix, as visualized in Fig. 2 (sixth column). While the addition of the event-count information improves the contour details, the contrast of the recent timestamp channels is still affected by long batch periods with high event density.

C. COMPACT SPATIO-TEMPORAL REPRESENTATION

The combined Timestamp Image and Event Count representation is generally robust but can lose temporal context with motion-overlapping. A recent timestamp is most useful when the event data is temporally sparse; however, can lose general temporal context when there are many overlapping events. This bias can happen frequently when subjected to highly textured scenes or long batch periods. To address this, we introduce the compact spatio-temporal representation (CSTR).

The CSTR improves the timestamp information by utilizing the mean timestamp instead to better capture temporal context. Thus, we initially accumulate the normalized timestamp values of all events at each spatial position as follows:

$$S(x, y, p) = \sum_{i=1}^{n} \begin{cases} \dfrac{t_i - t_s}{\Delta T} & \text{if } x = x_i \ \& \ y = y_i \ \& \ p = p_i \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where $S(x, y, p)$ represents the sum of the normalized event timestamps at position (x, y, p). Then, we calculate the mean of events' timestamps by dividing (9) over (6) as follows:

$$\bar{T}_s(x, y, p) = \begin{cases} \dfrac{S(x, y, p)}{C(x, y, p)}, & \text{if } C(x, y, p) \neq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

where $\bar{T}_s(x, y, p)$ represents the mean timestamp at position (x, y, p). This is visualized in Fig. 2 (seventh column). Nevertheless, mean timestamps on their own can be insufficient to represent the event data. Incorporating the event count can provide vital event-overlap context. Therefore, we concatenate the 2-channel mean timestamp $\bar{T}_s$, defined in (10), with the normalized Binary Event Count $\hat{C}_{\text{bin}}$, defined in (8). This yields a 3-channel representation. We visualize the CSTR in Fig 2 (last column), showing that it retains strong temporal context and contour sharpness. Hence, the CSTR approach adds robustness to motion-overlapping while retaining direct compatibility with existing computer-vision networks.

D. EVENT-BASED DATA AUGMENTATION FRAMEWORK

Randomized data augmentations can improve the generalization of deep learning models. Further, they can complement the spatio-temporal representations in event-based solutions. Accordingly, we propose a simple framework for randomized event-data augmentations that affect the spatial, temporal, and polarity information of event data. These augmentations can be combined and applied when training an event-based deep learning model with a spatio-temporal representation.

1) SPATIAL AUGMENTATIONS

Spatial augmentations are a common solution for introducing variations across the spatial dimension. In our framework, we explore a combination of rotations, rescalings, crops, and horizontal flips, each with its own parameters to set. For optimal computational efficiency, we apply spatial augmentations to the generated image-like event-batch representations.

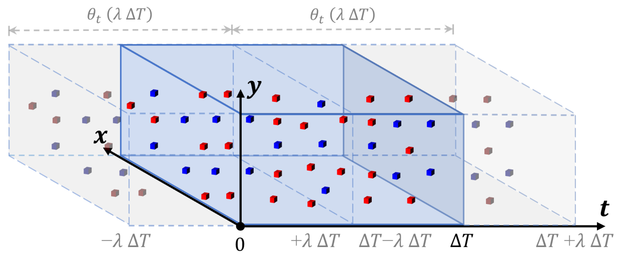

$\theta_t\ (\lambda\,\Delta T)$
$\theta_t\ (\lambda\,\Delta T)$
y
x
t
$-\lambda\,\Delta T$
0
$+\lambda\,\Delta T$
$\Delta T - \lambda\,\Delta T$
ΔT
$\Delta T + \lambda\,\Delta T$

FIGURE 3. Illustration of the proposed temporal augmentation method. Spatio-temporal events within a given batch are uniformly time-shifted by a randomized value λ multiplied by ΔT. Events that fall outside the original temporal range $[0, \Delta T]$ are subsequently removed. The maximum temporal shift θ_t that is demonstrated here is ±50% of the batch duration ΔT.

2) TEMPORAL AUGMENTATIONS

Rich temporal information is a major component of event data. Temporal augmentations can help enhance a model's ability to handle temporal dynamics. This is vital for representations that incorporate temporal information (*e.g.*, Timestamp Image [42]). As illustrated in Fig. 3, events are shifted based on a randomized value λ within the range of $[-1, +1]$, which is generated per event batch sample ε. This dynamic but consistent temporal shifting allows the model to learn from different temporal perspectives and improves its robustness to varying temporal dynamics. The temporal shift for each event e_i in the event batch ε can be expressed as:

$$t_i' = t_i + \theta_t(\lambda \Delta T), \tag{11}$$

where t_i' is the shifted timestamp of event e_i, θ_t is the max temporal shift threshold ($\theta_t \in (0, 1)$), and ΔT is the batch-sampling period. A balanced value for the max temporal shift threshold θ_t is 0.5, which indicates that the batch's events can be only shifted by a max of $\frac{\Delta T}{2}$ in either direction (shown in Fig. 3). Then, we filter out any events that fall outside the original batch's temporal range of $[0, \Delta T]$. Note that the proposed temporal augmentations are applied to a given event batch ε before generating an image-like representation.

3) POLARITY AUGMENTATIONS

Polarity augmentations introduce variations across the polarity domain, enabling the model to learn from varying polarity correlations of events. In our framework, we adopt a simple approach of inverting all the polarities in an event batch prior to frame transformation. This polarity inversion typically implies the reversal of the direction of motion and can introduce robustness to variations in lighting and motion. Hence, for each event e_i in an event batch ε, the polarity p_i is inverted to $\bar{p}_i$ if the threshold θ_p is met. The threshold θ_p is ideally set to 0.5, indicating a 50% chance of inverting the polarities of a given event batch ε. Similar to the proposed temporal augmentation method, the polarity augmentations are applied before generating the image-like representation.

TABLE 1. Statistics of the event-based object and action recognition datasets used in our experiments. The symbol † indicates that the referenced dataset does not have an official test split, while ‡ denotes that the dataset's original sequences were divided into samples of 500 ms with a 250 ms step size (following [58]).

Parameter	**Dataset**					
	Object Recognition				Action Recognition	
	N-MNIST [24]	N-Cars [25]	N-Caltech101† [24]	CIFAR10-DVS† [26]	ASL-DVS† [27]	DVS-Gesture‡ [28]
Number of classes	10	2	101	10	24	11
Dataset Type	Static	True	Static	Static	True	True
Event Camera / Event Sensor	ATIS [7]	ATIS [7]	ATIS [7]	DVS-128 [6]	DAVIS-240c [8]	DVS-128 [6]
Frame Dimension ($W \times H$)	35×35	128×128	240×180	128×128	240×180	128×128
# Total Samples	70000	24029	8709	10000	100800	38962
# Train Samples	60000 (86%)	15422 (64%)	6967 (80%)	8000 (80%)	80640 (80%)	30,978 (80%)
# Test Samples	10000 (14%)	8607 (36%)	1742 (20%)	2000 (20%)	20160 (20%)	7,984 (20%)
Avg ± Std of # samples/class	7000 ± 399.3	12015 ± 321.5	86 ± 119.3	1000 ± 0.0	4200 ± 0.0	3542 ± 1122
Min-Max range of # samples/class	6313-7877	11693-12336	31-800	1000-1000	4200-4200	2503-6676
Average # events/sample	4176	3966	115298	205072	28149	27339
Average event-batch duration	310 ms	100 ms	300 ms	1298 ms	110 ms	481 ms

IV. EXPERIMENT SETUP

In this section, we evaluate the proposed event-based representation for object and action recognition. Our primary comparison is evaluating our proposed event representation, the CSTR, against the foundational representations defined in the methodology (Section III-B). We do this over a series of well-known datasets to demonstrate our improvements in recognition tasks. Next, we take the best-performing spatio-temporal representations and do a second comparison while employing our proposed augmentation framework. Our experimental setup, including the network structures, datasets, augmentations, and training parameters are introduced next.

A. EXP I: BASELINE REPRESENTATION EVALUATION

In the baseline experiment, we compare the CSTR against the six foundational event representations presented in Section III-B. Recall that the Event Frame representations are traditionally encoded as either 0 or 1, while the foundational Event Count representations are encoded as the number of events (without scaling). However, the Event Count channel associated with the combined Timestamp Image & Event Count and the CSTR is normalized. This is done by dividing each event-count value by the maximum number of events in the frame as defined in (8). We apply this because the temporal representations are already scaled to be in the [0, 1] range.

We add rigor by exploring three-channel configurations for the one- and two-channel representations. We do this to enable direct compatibility with the classification networks' input structures and better leverage their pre-trained weights. In the case of the one-channel Binary Event Frame and Binary Event Count, we replicate the resulting channel three times. In the case of the Polarized Event Count, Timestamp Image, and the CSTR with mean timestamps only, we append an empty channel of zeros of the same spatial dimensions. Lastly, for the two-channel Polarized Event Frame, we first convert to an intermediary one-channel representation, where positive and negative events are denoted by values of $+1$ and -1 (following the approach proposed in [18]). We then replicate this three times instead of padding with a channel of zeros. These configurations are determined through experimentation to yield optimal results for each representation.

1) EVENT-BASED RECOGNITION DATASETS

Several event-based object and action recognition datasets are available in the literature. In this work, we utilize four commonly used event-based datasets to evaluate our proposed methods for object recognition: N-MNIST [24], N-Cars [25], N-Caltech101 [24], and CIFAR10-DVS [26]. Additionally, we evaluate our methods on two action recognition datasets, namely ASL-DVS [27] and DVS-Gesture [28]. In Table 1, we provide an overview of the main details and statistics of the selected recognition datasets.

For object recognition, all datasets except N-Cars [25] are effectively event-based versions of their frame-based counterparts commonly used in conventional computer vision. These datasets are generated using an event-based sensor, such as the DVS-128 [6] or the ATIS [7], mounted on a platform that moves in parallel to a screen displaying image samples of each dataset. The platform is programmed to move at various velocities and motions to simulate events similar to real-world sensor data. N-Cars [25], on the other hand, was generated using an event camera mounted on a moving vehicle driving on real-world roads. The dataset consists of events captured by the event camera as the vehicle encounters different objects, including cars and pedestrians, in various driving scenarios.

For action recognition, ASL-DVS [27] consists of 24 hand shapes resembling different letters from the American Sign Language. These shapes were recorded in an office environment with constant illumination using DAVIS240c [8]. For each letter, 4200 samples were collected at a sampling duration of 100 ms. Meanwhile, DVS-Gesture [28] consists of 1342 event-data sequence recordings of 11 different gestures. These sequences were captured under three lighting conditions and performed by 29 individuals. Due to the considerable length of the dataset's sequences (∼100 seconds on average), we divide each into shorter samples of a fixed batch-sampling period. Initially, each sequence is split into a subsequence per gesture. Then, the resulting subsequences are further divided into 500 ms samples with a 250 ms step size, following a similar approach used in previous works [20], [51], [58]. The resulting number of samples is presented in Table 1.

Except for DVS-Gesture [28], we use the provided samples with pre-defined batch periods ΔT from each dataset, as outlined in Table 1. The sampling periods range from 100 ms (N-Cars [25] and ASL-DVS [27]) to roughly 1300 ms (CIFAR10-DVS [26]). This enables us to analyze the robustness of different event representations to various batch-sampling periods.

Furthermore, Table 1 demonstrates an uneven distribution in the average number of samples per class across the datasets. N-MNIST [24], N-Cars [25], ASL-DVS [27], and DVS-Gesture [28] exhibit a substantial number of samples per class facilitating effective training and fine-tuning of classifiers. In contrast, CIFAR10-DVS [26] and N-Caltech101 [24] have significantly fewer average numbers of samples per class of 1000 and 81, respectively. While the samples of CIFAR10-DVS [26] are uniformly distributed among classes, the samples N-Caltech101 [24] are highly unbalanced, ranging from 31 to 800 samples per class, posing a challenge for object recognition tasks.

For datasets without an official test split (N-Caltech101 [24], CIFAR10-DVS [26], and ASL-DVS [27]), we adopt the 80%-20% training-testing dataset-split strategy employed in similar works [20], [25], [51]. These splits are generated once and utilized consistently throughout the experiments of this work to ensure consistent benchmarking and fair comparisons. In addition, to address the imbalance in the sample distribution within N-Caltech101 [24], we apply the same split ratios to each class's samples. This approach avoids imbalanced splits and maintains a fair and consistent benchmarking process across the different methods evaluated in this work.

2) CLASSIFICATION MODELS

We evaluate each event representation using six popular pre-trained CNN image classifiers. We do this both for completeness and to represent real-world use. These classifiers include: ResNet18 [15], ResNet50 [15], MobileNetV2 [59], both Small and Large variants of MobileNetV3 [60], and InceptionV3 [61] (limited to 3-channel representations only). We initialize all networks with weights pre-trained on ImageNet [62]. Then, we replace the final fully connected layer with a corresponding layer that matches the number of output classes in the utilized dataset. For representations with 1 or 2 channels, we replace the initial input convolutional layers of each CNN classifier with randomized weights to accommodate the desired number of input channels. Subsequently, we fine-tune these networks on the evaluation datasets. Throughout our experiments, we observed that utilizing the frame-based architectures as-is (*i.e.*, for 3-channel representations) yields better results due to more effective fine-tuning. Consequently, whenever possible, we present either a replicated or an extended 3-channel version of all tested representations.

3) TRAINING PARAMETERS

For all models trained in this work, we use the cross-entropy loss with the ADAM [63] optimizer (without weight decay), for up to 50 epochs. We utilize an initial learning rate of 1×10^{-3} for N-MNIST [24], N-Cars [25], and ASL-DVS [27]; and 3×10^{-4} for the more challenging N-Caltech101 [24], CIFAR10-DVS [26], and DVS-Gesture [28]. While more advanced learning rate schedulers can be employed, we avoid them to limit the number of hyper-parameters and simplify the comparison.

During training, each batch-representation sample is initially generated with a resolution matching the spatial dimensions of the utilized dataset (as shown in Table 1). The resulting 3D representations are then scaled to 224×224 for all classifiers, except for InceptionV3 [61] which requires a 3-channel input with the spatial dimensions of 299×299. After rescaling, we apply standardization to the resulting 3D matrices using normalization parameters derived from ImageNet [62] (*i.e.*, mean and standard deviation). Our experiments (using the CSTR with the object recognition datasets) consistently show an average classification accuracy improvement of approximately 5% when utilizing ImageNet normalization parameters. This improvement is observed compared to using each dataset's distribution parameters or when not applying normalization. It can be attributed to the suitability of ImageNet parameters for generalizing image-like representations. This is particularly important given the relatively low number of samples of the event-based datasets used in our experiments, compared to ImageNet [62], making them less optimal for removing input bias through standardization.

Furthermore, we randomly split the training set by 75% for training and 25% for validation. In addition, to ensure proper convergence and robust generalization, the samples of the validation split are randomly selected per each class's number of samples. This ensures a more balanced and well-representing validation set. For all models trained in the baseline experiment, we use early stopping to prevent overfitting. Specifically, we monitor the validation loss during training, and if it does not improve for 10 consecutive epochs, we stop the training early to avoid further overfitting. Afterward, we choose the model with the lowest validation loss that results during training. We follow the same procedure when not utilizing early stopping as well. Finally, we use a batch size of 64 for all the models we train throughout this work.

TABLE 2. Average test classification accuracy results for the foundational event representations and the CSTR across different recognition datasets. Each result is the average of up to 6 classification models as specified in Section IV-A2. Note that the 1 and 2-channel representations are additionally transformed into 3-channel representations as specified in Section IV-A, and indicated by the *. The best and second-best results are highlighted in bold and underlined, respectively.

Event Representation	Representation Components				Dataset						AVG.
	Timestamp	Polarity	Count	# Channels	N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	ASL-DVS	DVS-Gesture	
Binary Event Frame	×	×	×	1	95.1%	91.7%	68.5%	50.6%	99.6%	83.2%	81.4%
				3*	95.2%	92.6%	73.5%	52.8%	**99.7%**	84.2%	83.0%
Polarized Event Frame	×	✓	×	2	96.1%	88.4%	69.8%	62.0%	**99.7%**	90.6%	84.4%
				3*	98.9%	93.2%	81.5%	60.7%	**99.7%**	90.6%	87.4%
Binary Event Count	×	×	✓	1	98.6%	91.6%	75.2%	69.2%	45.8%	87.6%	78.0%
				3*	98.5%	91.1%	81.1%	**73.7%**	78.3%	89.1%	85.3%
Polarized Event Count	×	✓	✓	2	98.9%	91.8%	73.0%	69.9%	41.0%	91.8%	77.7%
				3*	98.5%	92.8%	81.6%	71.8%	52.0%	90.9%	81.3%
Timestamp Image	✓	✓	×	2	99.0%	85.5%	74.1%	67.7%	99.5%	91.4%	86.2%
				3*	99.0%	92.3%	81.3%	68.8%	**99.7%**	93.2%	89.0%
Timestamp Image & Count	✓	✓	✓	3	98.9%	92.2%	82.5%	72.6%	**99.7%**	92.9%	89.8%
CSTR (mean $\bar{T}_s$ only)	✓	✓	×	2	98.9%	92.4%	76.9%	63.5%	99.6%	92.8%	87.4%
				3*	99.0%	92.4%	**83.9%**	67.4%	99.6%	**93.6%**	89.3%
CSTR (mean $\bar{T}_s$ & Count)	✓	✓	✓	3	**99.1%**	**93.6%**	82.9%	71.6%	**99.7%**	**93.6%**	**90.1%**

B. EXP II: RANDOMIZED EVENT AUGMENTATIONS

With a baseline established, our next experiment aims to leverage the randomized augmentation framework introduced in Section III-D. Augmentations are a popular method for addressing data sparsity as they introduce variance in the spatial, temporal, and/or polarity characteristics. We believe these effects can also be used to further investigate batch-representation stability and explore how well the performance of spatio-temporal representations scales with the proposed randomized event-based augmentation framework.

In this experiment, we explore different settings for each type of randomized augmentation (spatial, temporal, and polarity). For spatial augmentations, we apply crops, rotations, and translations to the generated image-like representations. Initially, we randomly take crops of 90-100% of the spatial frame size with aspect ratios ranging from 3:4 to 4:3. We also apply translations of up to 10% in the x and y axis (up to 5% for N-Cars [25]) and rotations of up to $\pm 10°$ (up to $\pm 30°$ for N-MNIST [24]). Additionally, random horizontal flips are used with CIFAR10-DVS [26] (applied prior to the other spatial transformations) with a threshold of 0.5. For both temporal and polarity augmentations, we utilize a balanced value of 0.5 for both the maximum temporal shift θ_t and the polarity inversion thresholds θ_p. We note that all of the proposed randomized augmentations are only applied to the training splits (*i.e.*, excluding validation splits).

Furthermore, we explore different combinations of the proposed augmentation methods. Spatial augmentations can be highly beneficial as spatial dependencies are typically the most informative, especially when identifying the edges or contours of an object. However, when utilizing event data, they require careful manual tuning. On the other hand, the proposed temporal and polarity augmentations have minimal parameters to tune and can naturally complement the training of any event-based solution. Therefore, we focus on the temporal-polarity augmentation combination as an alternative that requires no tuning when using their default threshold values. Finally, for a more comprehensive approach, we explore a combination that incorporates all three event-based augmentation methods.

We perform this experiment only on the spatio-temporal representations presented in this work. This includes the proposed 3-channel variants of the CSTR and the Timestamp Image. These representations are selected because the proposed framework primarily affects the temporal and polarity information of event data, making them optimal for spatio-temporal representations. Additionally, we only utilize the three best classifiers found during the baseline experiment: ResNet18 [15], ResNet50 [15], and InceptionV3 [61]. The ASL-DVS [27] dataset is excluded from this experiment as its performance is already effectively saturated without the use of augmentations. Finally, we provide sufficient training time to ensure reaching an optimal global minimum, by training each model for 50 epochs without early stopping. We use an initial learning rate of 1×10^{-4} instead while keeping all the other evaluation parameters identical to the initial experiment.

V. EVALUATION RESULTS

In this section, we present our experimental results. We first do a baseline evaluation of the CSTR and six foundational representations across popular event-based recognition datasets. We then identify the best performers and re-evaluate them when using the proposed augmentation framework. These experiments help show that the proposed CSTR is a robust means of representing event batches, including ones with long temporal durations and high event density. Finally, we present a comparison with other works in the literature.

A. EXP I: BASELINE EVALUATION RESULTS

We present the baseline evaluation results in Table 2. This table shows the average performance of the representations with all six classification networks detailed in the Experimentation Setup (see: Section IV-A2). For space reasons,

TABLE 3. The effects of the proposed event-based augmentation framework on the average test classification performance of the different spatio-temporal representations explored in this work. Each result represents the average classification accuracy of the top three classifiers only (ResNet18, ResNet50, and InceptionV3) due to the complexity of training with augmentations. The first row represents the baseline results obtained without any augmentation, serving as a reference point for each representation. The subsequent rows demonstrate the performance improvements achieved when using the respective augmentation configurations. Notably, only the augmented three-channel representations are considered, as outlined in Section IV-A and indicated by the *. The best-performing baseline representation is indicated by the †, while the representations yielding the best and second-best performance with augmentations are highlighted in bold and underlined, respectively.

Representation	Augmentation Type			Dataset					AVG.
	Spatial	Temporal	Polarity	N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	DVS-Gesture	
Timestamp Image*		Baseline		99.1%	93.4%†	82.0%	72.1%	93.7%†	88.1%
	✓			$\underline{99.3\%}$ (+0.2%)	94.5% (+1.1%)	84.4% (+2.4%)	77.5% (+5.4%)	94.1% (+0.4%)	90.0% (+1.9%)
		✓		99.2% (+0.1%)	95.7% (+2.3%)	87.1% (+5.1%)	76.1% (+4.0%)	94.3% (+0.6%)	90.5% (+2.4%)
			✓	99.1% (+0.0%)	95.6% (+2.2%)	86.2% (+4.2%)	71.8% (–0.3%)	93.8% (+0.1%)	89.3% (+1.2%)
		✓	✓	99.2% (+0.1%)	95.8% (+2.4%)	86.9% (+4.9%)	76.3% (+4.2%)	93.9% (+0.2%)	90.4% (+2.3%)
	✓	✓	✓	99.2% (+0.1%)	96.3% (+2.9%)	85.2% (+3.2%)	78.0% (+5.9%)	94.9% (+1.2%)	90.7% (+2.6%)
Timestamp Image & Count		Baseline		99.1%	93.3%	84.5%	75.5%	93.0%	89.1%
	✓			**99.4%** (+0.3%)	95.7% (+2.4%)	84.4% (–0.2%)	**80.4%** (+4.9%)	94.6% (+1.6%)	90.9% (+1.8%)
		✓		99.2% (+0.1%)	95.4% (+2.1%)	87.1% (+2.6%)	77.2% (+1.7%)	94.6% (+1.6%)	90.7% (+1.6%)
			✓	99.2% (+0.1%)	96.3% (+3.0%)	86.4% (+1.9%)	73.5% (–2.0%)	93.3% (+0.3%)	89.8% (+0.7%)
		✓	✓	$\underline{99.3\%}$ (+0.2%)	95.7% (+2.4%)	87.3% (+2.8%)	78.1% (+2.6%)	94.4% (+1.4%)	91.0% (+1.9%)
	✓	✓	✓	$\underline{99.3\%}$ (+0.2%)	96.3% (+3.0%)	87.1% (+2.6%)	$\underline{80.2\%}$ (+4.7%)	94.4% (+1.4%)	$\underline{91.5\%}$ (+2.4%)
CSTR (mean $\bar{T}_s$ only)*		Baseline		99.2%†	92.7%	84.6%	71.5%	93.5%	88.3%
	✓			**99.4%** (+0.2%)	96.1% (+3.4%)	85.7% (+1.1%)	75.6% (+4.1%)	$\underline{95.5\%}$ (+2.0%)	90.4% (+2.1%)
		✓		$\underline{99.3\%}$ (+0.1%)	93.3% (+0.6%)	87.8% (+3.2%)	75.5% (+4.0%)	93.4% (–0.1%)	89.8% (+1.5%)
			✓	99.4% (+0.2%)	96.2% (+3.5%)	87.5% (+2.9%)	70.8% (–0.7%)	94.8% (+1.3%)	89.7% (+1.4%)
		✓	✓	99.2% (+0.0%)	$\underline{96.9\%}$ (+4.2%)	$\underline{88.3\%}$ (+3.7%)	74.8% (+3.3%)	93.7% (+0.2%)	90.6% (+2.3%)
	✓	✓	✓	$\underline{99.3\%}$ (+0.1%)	96.6% (+3.9%)	86.4% (+1.8%)	78.2% (+6.7%)	95.0% (+1.5%)	91.1% (+2.8%)
CSTR (mean $\bar{T}_s$ & Count)		Baseline		99.2%†	93.0%	84.9%†	75.8%†	93.4%	89.2%†
	✓			**99.4%** (+0.2%)	96.3% (+3.3%)	85.0% (+0.1%)	79.3% (+3.5%)	**95.7%**(+2.3%)	91.1% (+1.9%)
		✓		**99.4%** (+0.2%)	95.4% (+2.4%)	87.9% (+3.0%)	78.4% (+2.6%)	94.9% (+1.5%)	91.2% (+2.0%)
			✓	$\underline{99.3\%}$ (+0.1%)	96.1% (+3.1%)	87.0% (+2.1%)	72.2% (–3.6%)	95.1% (+1.7%)	89.9% (+0.7%)
		✓	✓	**99.4%** (+0.2%)	96.6% (+3.6%)	**88.4%** (+3.5%)	77.9% (+2.1%)	94.4% (+1.0%)	91.3% (+2.1%)
	✓	✓	✓	$\underline{99.3\%}$ (+0.1%)	**97.0%** (+4.0%)	86.1% (+1.2%)	79.8% (+4.0%)	**95.7%** (+2.3%)	**91.6%** (+2.4%)

we provide a full breakdown of each network's performance in Table 5 of the Appendix A. We note a few basic observations. First, including polarity improves generalization. We see this mainly in the Event Frame representations, as well as the Event Count representations but to a lesser extent. This aligns with the methodology expectations. Second, there is a benefit to maintaining the classification networks' native input structure. In all cases, transforming a one or two-channel representation into three channels (by either padding or replicating data) consistently improves classification accuracy. This reinforces the value of transfer-learning frame-based networks for event-based applications. Lastly, our representation, the CSTR, has the highest average classification accuracy and is the best overall in four of the six datasets.

The strength of the CSTR is in addressing motion-overlapping. We can see that of the foundational event representations, the simple Binary Event Count is rather robust. This implies that the number of events per batch is strongly correlated with the classification task, where adding polarity helps better describe the type of motion. Intuitively, this implies that better describing the event's temporal distribution should improve performance. While the Timestamp Image does this via recent timestamps, this approach can be biased for longer temporal periods. The CSTR addresses this by representing the aggregate behavior with the mean timestamp and generalizes very well across datasets, including those with long temporal durations and high event density.

We note the results get particularly interesting with the CIFAR10-DVS [26] dataset. In general, all classification networks for all representations notably overfit. This overfitting concern is verified by the simple Binary Event Count having the highest dataset classification accuracy, remaining in line with its accuracy on other datasets. We believe this overfitting is partially due to the dataset being generated by repeated back-and-forth motions (frequent direction change), causing very significant motion overlap [26]. Furthermore, the CIFAR10-DVS [26] data collection methodology uses up-scaled 32×32 RGB images that appear rather blurry [26]. This blurriness reduces the edge features the events depend on and inherently increases sensitivity to sensor noise. With this said the CSTR still does relatively well, but incrementally worse than the Timestamp Image representations. We hypothesize here that the timestamp recency better correlates with back-and-forth motions versus the timestamp mean.

Lastly, we observe that the optimal classification network can vary across representations and datasets. Intuitively, classification network accuracy should correlate with ImageNet accuracy; however, the expanded results given in Appendix A (Table 5) show that this is not always the case. We conjecture that this can be a function of dataset density and intra-class variance. When the variance is particularly high, such as in the CIFAR10-DVS [26] dataset, the smaller networks tend to generalize better. This is likely a result of overfitting, where the smaller parameter spaces inherently regularize themselves. However, we also note the large InceptionV3 [61] network is still the top performer for some representations.

TABLE 4. Comparison with the self-reported state-of-the-art works. Our proposed representation, the CSTR, yields very competitive results when compared with state-of-the-art event-based object and action recognition on the utilized datasets. For datasets without an official split, the † symbol denotes that the referenced result was based on a 90%-10% split, compared to the typical 80%-20% split. The best and second-best results are highlighted in bold and underlined, respectively.

Event Representation	Classifier Architecture	Data Augmentation	Dataset					
			N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	ASL-DVS	DVS-Gesture
HATS [25]	SVM	×	99.1%	90.2%	64.2%	52.4%	-	-
Event-by-event [33]	SNN	×	**99.6%**	-	-	69.0%	-	96.5%
Graphs [20]	Residual-GCN	✓(spatial)	99.0%	91.4%	65.7%	54.0%	90.10%	97.2%
Graphs [52]	GCN	×	-	94.5%	66.8%	-	-	-
Voxel-grid [51]	GCN	×	99.5%	93.2%	77.8%	69.0%	98.90%	**97.5%**
Event Clouds [58]	PointNet++	×	-	-	-	-	-	95.3%
EST [22]	CNN (ResNet34)	×	-	92.5%	81.7%	-	-	-
Timestamp Image & Count [47]	CNN (ResNet34)	×	**99.6%**	97.3%	89.2%	76.3%	-	-
TORE Volumes [23]	CNN (2×GoogLeNet)	×	99.4%	**97.7%**	83.4%	-	99.95%	96.2%
EST [22]	CNN (ResNet34)	EventDrop [55]	-	95.5%	85.2%	-	-	-
Polarized Event Count [56]	CNN (ResNet18)	EventMix [56]	-	-	84.7%†	84.4%†	-	89.5%
Polarized Event Count [56]	CNN (ResNet34)	EventMix [56]	-	96.6%	89.2%†	**85.6%**†	-	91.8%
CSTR (ours)	CNN (ResNet18)	×	99.1%	93.0%	81.6%	77.8% | 80.6%†	99.88%	95.5%
		TP	99.3%	96.6%	86.7%	77.9% | 81.8%†	**99.98%**	95.5%
		STP	99.3%	96.9%	84.0%	78.8% | 80.9%†	99.44%	96.9%
	CNN (ResNet50)	×	99.2%	92.5%	85.4%	70.6% | 70.4%†	99.94%	97.0%
		TP	99.4%	96.2%	88.6%	75.4% | 77.4%†	99.89%	**97.5%**
		STP	99.5%	96.9%	86.2%	78.7% | 80.9%†	99.84%	96.9%
	CNN (InceptionV3)	×	99.2%	93.5%	87.7%	79.0% | 77.2%†	99.89%	95.9%
		TP	99.4%	96.9%	**89.8%**	80.4% | 83.1%†	99.93%	96.3%
		STP	99.3%	97.2%	88.2%	**81.8%** | 83.7%†	99.74%	**97.5%**

This implies picking the optimal network may ultimately require experimentation. We recommend that the developer assess various networks and select the one that best fits their accuracy and run-time requirements.

B. EXP II: RANDOMIZED AUGMENTATIONS RESULTS

We present the results of the augmentation evaluation in Table 3. Starting with the baseline results, we observe that the CSTR consistently outperforms other representations when considering the top-3 classifiers (ResNet18, ResNet50, InceptionV3) on most datasets. This emphasizes the robustness of the CSTR in capturing spatio-temporal information across varying batch periods. The slight underperformance of the CSTR on the N-Cars dataset compared to the Timestamp Image representation can be attributed to the dataset's low event density and short batch periods. This causes larger classification networks to underfit with more complex representations. We observe this with DVS-Gesture as well. Nevertheless, the introduction of the proposed augmentations highlights the limitations of the Timestamp Image. Specifically, the CSTR demonstrates superior results on N-Cars when utilizing either the temporal-polarity augmentation combination or combining all three augmentation methods. This highlights the CSTR's ability to encode spatio-temporal information optimally when provided with sufficient training variations.

Overall, the augmentation framework shows significant performance improvements across all benchmarks. When using a single augmentation method, the proposed temporal augmentation method can match and even exceed the performance of hand-crafted spatial augmentations. This is evident in the highest average performance achieved by a single augmentation method (*i.e.*, 91.2% when using the CSTR). We find that the CSTR benefits the most from the temporal augmentations due to its effectiveness at encoding temporal information. On the other hand, spatial augmentations, while generally reliable, have limitations on datasets with challenging spatial characteristics like N-Caltech101 [24]. Furthermore, spatial augmentations require manual tuning for optimal results. In contrast, the proposed temporal and polarity augmentations serve as a promising alternative, requiring minimal tuning and consistently outperforming spatial augmentations on average across all evaluated representations. This makes them particularly advantageous for optimizing deep learning models in event-based applications.

Interestingly, we find that combining all augmentation methods (spatial, temporal, and polarity) does not consistently yield the best performance. The significant variations introduced by this combination can lead to underfitting, considering the utilized regularization approach. Therefore, we suggest exploring an alternative approach of randomly selecting one of the augmentation methods per event-batch sample during training. Additionally, we observe that spatial augmentations underperform polarity and temporal augmentations on the N-Caltech101 [24] dataset. This can be attributed to the dataset's imbalance, where typical spatial augmentations are insufficient to improve generalization.

In conclusion, our findings demonstrate the strength of the CSTR and its ability to leverage the proposed augmentation framework. The temporal augmentations prove to be the most advantageous on average for the CSTR, showcasing the CSTR's effectiveness in capturing temporal information. Moreover, combining multiple augmentation methods can enhance generalization performance. However, further exploration and optimization of the augmentation methods are necessary to maximize performance and address limitations.

C. COMPARISON WITH THE STATE-OF-THE-ART

In this section, we compare the performance of the CSTR with other approaches that utilize the same recognition datasets. Although each approach utilizes different methods and training configurations, our aim here is to highlight the efficacy of the CSTR when combined with off-the-shelf pre-trained classification networks. Furthermore, we emphasize how the performance can be further improved by leveraging the proposed augmentation framework for event data.

We present the performance comparison in Table 4. While most works report results for an 80-20% split, we provide the results of our framework on a 90-10% split for CIFAR10-DVS [26] as well to establish a fair comparison with those that utilize such a split. For our results on DVS-Gesture [28], we adopt a simple moving-majority filter to handle the long-term temporal dependencies, as applied in [23], [58]. This filter outputs the most frequent gesture classification out of the last 5 (*i.e.*, 1250 ms moving window). If there is more than one gesture with the same number of classifications (or none), the filter simply returns the classification result for the current event batch. It is worth noting that all the referenced works also utilize a 500 ms sampling period for splitting the event sequences of the DVS-Gesture [28] dataset.

Overall, the results show that the CSTR performs excellently across the employed benchmark datasets. In terms of the baseline performance (excluding augmentations), the CSTR notably achieves state-of-the-art results on CIFAR10-DVS [26] and consistently ranks as the second-best on ASL-DVS [27]. This demonstrates the robustness and versatility of the CSTR which requires minimal configuration and enables a direct and effective deployment for event-based solutions.

To demonstrate the impact of the proposed augmentation framework, we compare the results with other works that incorporate different augmentation techniques for event data. One such work utilizes EventMix [56] augmentations in combination with the Polarized Event Count representation. This work splits the provided batch samples of the N-Caltech101 and CIFAR10-DVS datasets into 10 slices of equal temporal duration. This effectively yields 10 times the original number of samples of each dataset. In contrast, we utilize the provided batch samples of each dataset as-is. Despite this, the CSTR with the randomized Temporal-Polarity augmentations proves to be highly competitive, even without splitting the datasets' samples. Accordingly, the CSTR demonstrates significant robustness to varying batch periods. Furthermore, we show that the CSTR, in combination with the proposed temporal and polarity augmentations, can achieve stronger results on N-Caltech101 [24] even with less training data. Lastly, the addition of the augmentation framework significantly improves the performance of the CSTR, surpassing more advanced representations such as EST [22] with the EventDrop [55] augmentation framework.

Our findings highlight the strength of the CSTR representation when combined with off-the-shelf pre-trained classifiers. They showcase the effectiveness of the CSTR in capturing temporal information and leveraging the robustness of pre-trained networks without any modification to the input layers. Thus, the CSTR retains a compact input dimensionality and effectively leverages transfer learning. Furthermore, the proposed augmentation framework offers a promising alternative for enhancing generalization performance without the need for significant manual tuning. Finally, we note that the results presented utilize a simple training framework. Therefore, various training optimization and batch-sampling techniques can be explored to further improve robustness.

VI. CONCLUSION

In this work, we introduce the compact spatio-temporal representation (CSTR) for event-based vision. When dealing with asynchronous event data, it is common to accumulate events in batches to generate a synchronous response. In order to do so, an intermediate representation is needed, especially when utilizing modern computer vision architectures. Thus, encoding the data into a representation compatible with existing classification networks is crucial for leveraging transfer learning and avoiding the complexity of designing custom deep-learning architectures. Foundational event representations typically encode either the number of events or the most recent event's timestamp per spatial location (based on polarity). These approaches are convenient and relatively robust but can be sensitive to motion-overlapping (common in long sampling duration) and possibly deficient for high event-density streams.

The CSTR improves upon the foundational event representations by better describing the temporal behavior of the asynchronous event data while retaining similar computational complexity. This is done by calculating the average of the normalized timestamps per each event polarity, combined with the polarity-agnostic number of events at each spatial index of the frame. Besides, the CSTR imposes minimal processing overhead given that each event is only processed once and that each spatial position is updated independently (*i.e.*, without the need to maintain any spatial dependencies), as indicated in the methodology. Accordingly, the CSTR generates a compact image-like representation that is more robust to high-motion scenes and long temporal durations. We validate this hypothesis through rigorous benchmarking against similar representations.

Combining the CSTR with off-the-shelf pre-trained classifiers demonstrates its ability to effectively leverage the power of transfer learning without modifying the input layers, thereby retaining its compact input dimensionality. We also propose a simple yet effective augmentation framework for event data, significantly improving the performance and generalization capabilities of the CSTR. This framework highlights the potential of augmentations in event-based recognition without the need for extensive manual tuning.

Experimental validation confirms that the CSTR outperforms foundational event representations in popular event-based applications. Benchmarking the CSTR against six foundational representations and six common recognition datasets (using six popular classification networks) consistently shows its superior performance. Additionally, incorporating random augmentations during training, including our proposed temporal augmentation, further enhances results on all representations, with the CSTR generally benefiting the most from the proposed augmentation framework. This overall improvement validates the CSTR's ability to robustly encode temporal information.

The CSTR achieves our goal of providing a robust event-batch representation that is directly compatible with existing computer vision architectures, maintaining similar inference complexity. As a result, the CSTR is an excellent choice for developing event-based solutions. The combination of the CSTR with the proposed augmentation framework further enhances its performance and generalization capabilities, requiring minimal tuning and enabling direct deployment.

While the CSTR excels as a versatile representation, it does not directly address certain prominent challenges in event-based vision, such as sensor noise [43]. To mitigate these issues effectively, additional techniques may be necessary.

Future work involves exploring the use of the CSTR in other perception tasks, such as object detection, and investigating additional optimization techniques to enhance robustness. Additionally, evaluating the suitability of the CSTR for real-time applications, where latency is a primary concern, would be an interesting avenue to explore.

APPENDIX A SUPPLEMENTARY TABLES

In Table 5, we provide a full breakdown of the first experiment's results (presented in Table 2). This experiment rigorously compares the CSTR against different foundational representations using the event-based object and action recognition datasets utilized in this work. Accordingly, the result of each of the six pre-trained classification networks, specified in Section IV-A2, are shown. Additionally, in Table 6, we provide a full breakdown of the second experiment's results (presented in Table 3). This experiment evaluates the effects of the proposed event-based augmentation framework on the CSTR and the other spatio-temporal representations explored in this work. The results are shown for each of the three-best classification networks utilized in this work (ResNet18 [15], ResNet50 [15], InceptionV3 [61]).

TABLE 5. A full breakdown of the test classification accuracy results that are presented in Table 2 for the event-based recognition datasets. The results of each evaluated representation configuration are demonstrated for 6 different pre-trained classification networks which are fine-tuned on each dataset. The best values are highlighted in bold per dataset and number of input channels.

Representation	**Pre-trained Classifier**	**Dataset** (# of channels)																	
		N-MNIST			N-Cars			N-Caltech101			CIFAR10-DVS			ASL-DVS			DVS-Gesture		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
Binary Event Frame	ResNet18	95.5%	95.0%	95.0%	92.6%	91.7%	93.0%	71.9%	64.8%	74.8%	61.4%	60.2%	61.3%	99.7%	99.3%	99.7%	87.3%	84.1%	85.4%
	ResNet50	95.1%	95.0%	95.3%	90.3%	90.5%	90.0%	69.1%	67.9%	74.7%	53.4%	49.6%	52.7%	99.5%	99.7%	**99.9%**	81.6%	82.9%	81.5%
	MobileNetV2	95.0%	94.6%	95.6%	92.1%	**94.3%**	94.4%	67.3%	68.3%	72.6%	48.2%	29.5%	48.6%	99.4%	99.6%	99.5%	82.5%	86.3%	84.3%
	MobileNetV3-L	95.0%	95.2%	95.3%	93.1%	88.5%	90.4%	69.7%	66.7%	71.1%	43.6%	51.1%	49.2%	99.6%	98.6%	99.7%	86.0%	84.7%	84.6%
	MobileNetV3-S	94.9%	95.2%	95.1%	90.2%	90.6%	93.1%	64.5%	64.0%	71.2%	46.7%	28.9%	45.9%	99.5%	95.6%	99.7%	78.7%	81.9%	83.5%
	InceptionV3	-	-	94.7%	-	-	94.5%	-	-	76.8%	-	-	59.1%	-	-	99.8%	-	-	85.8%
Polarized Event Frame	ResNet18	98.8%	96.7%	99.2%	93.2%	89.6%	94.0%	**80.8%**	72.6%	78.4%	67.8%	72.4%	68.3%	99.7%	**99.9%**	**99.9%**	90.4%	91.6%	91.8%
	ResNet50	98.8%	96.3%	99.0%	88.8%	90.9%	89.2%	77.1%	72.6%	84.5%	60.4%	64.4%	58.8%	**99.8%**	**99.9%**	**99.9%**	90.0%	90.9%	90.1%
	MobileNetV2	**99.1%**	95.9%	99.0%	92.6%	93.0%	94.0%	76.0%	69.0%	80.0%	45.7%	57.0%	55.4%	99.7%	99.8%	99.5%	88.5%	91.0%	88.5%
	MobileNetV3-L	98.6%	96.3%	98.2%	**95.0%**	76.1%	92.7%	74.2%	67.6%	80.7%	58.7%	59.1%	57.3%	99.2%	99.6%	99.6%	89.3%	88.7%	91.4%
	MobileNetV3-S	98.8%	95.2%	99.0%	90.3%	92.6%	94.5%	71.8%	67.4%	79.8%	53.2%	57.0%	57.8%	99.7%	99.4%	99.4%	88.3%	90.8%	90.5%
	InceptionV3	-	-	99.1%	-	-	95.1%	-	-	85.4%	-	-	67.0%	-	-	99.8%	-	-	91.3%
Binary Event Count	ResNet18	98.4%	98.5%	98.5%	89.3%	93.0%	91.3%	79.4%	76.9%	81.1%	**79.4%**	**78.3%**	78.7%	90.4%	53.0%	59.2%	84.7%	89.0%	86.3%
	ResNet50	98.8%	98.8%	98.7%	92.6%	91.3%	92.2%	78.0%	72.6%	80.8%	72.8%	70.0%	75.1%	85.3%	52.6%	61.8%	88.8%	85.0%	91.2%
	MobileNetV2	98.8%	98.7%	98.5%	93.0%	93.0%	93.3%	74.2%	74.8%	81.4%	65.4%	66.2%	67.1%	5.5%	35.4%	90.8%	**91.1%**	89.7%	90.2%
	MobileNetV3-L	98.3%	98.5%	98.3%	90.9%	90.9%	90.4%	76.4%	69.7%	80.1%	64.0%	70.6%	70.3%	39.5%	9.5%	76.3%	88.6%	88.3%	90.9%
	MobileNetV3-S	98.6%	98.7%	98.1%	92.0%	93.8%	87.8%	67.9%	71.9%	78.4%	64.6%	67.2%	71.6%	8.3%	9.0%	86.2%	85.0%	83.7%	84.1%
	InceptionV3	-	-	99.1%	-	-	91.6%	-	-	84.9%	-	-	**79.5%**	-	-	95.7%	-	-	91.9%
Polarized Event Count	ResNet18	-	99.1%	98.9%	-	92.5%	92.9%	-	75.7%	82.0%	-	77.5%	79.3%	-	17.1%	8.6%	-	92.1%	91.4%
	ResNet50	-	98.9%	97.6%	-	91.2%	92.8%	-	73.0%	82.2%	-	73.4%	72.7%	-	28.9%	55.4%	-	92.8%	90.5%
	MobileNetV2	-	98.9%	98.9%	-	92.4%	94.1%	-	73.8%	80.7%	-	64.1%	66.6%	-	41.8%	72.5%	-	90.6%	92.2%
	MobileNetV3-L	-	98.9%	98.7%	-	90.4%	92.6%	-	72.4%	79.5%	-	66.4%	65.2%	-	67.7%	54.8%	-	92.1%	93.0%
	MobileNetV3-S	-	98.7%	98.0%	-	92.4%	92.2%	-	70.4%	79.6%	-	68.2%	73.0%	-	49.7%	63.6%	-	91.2%	91.6%
	InceptionV3	-	-	99.1%	-	-	92.4%	-	-	85.9%	-	-	74.0%	-	-	56.9%	-	-	86.9%
Timestamp Image	ResNet18	-	99.0%	99.0%	-	86.9%	91.8%	-	73.1%	82.3%	-	74.7%	76.4%	-	99.8%	99.5%	-	91.6%	92.9%
	ResNet50	-	99.1%	99.1%	-	92.1%	94.8%	-	76.8%	80.4%	-	67.8%	67.6%	-	**99.9%**	**99.9%**	-	89.9%	93.5%
	MobileNetV2	-	98.9%	99.1%	-	83.2%	95.3%	-	75.1%	82.0%	-	63.5%	65.1%	-	99.7%	**99.9%**	-	92.3%	92.5%
	MobileNetV3-L	-	99.0%	98.6%	-	90.9%	85.5%	-	73.0%	80.5%	-	67.4%	62.3%	-	99.4%	99.8%	-	92.8%	92.8%
	MobileNetV3-S	-	98.9%	98.7%	-	74.5%	92.5%	-	72.7%	79.7%	-	65.4%	69.1%	-	98.5%	99.3%	-	90.4%	92.9%
	InceptionV3	-	-	**99.3%**	-	-	93.6%	-	-	83.2%	-	-	72.3%	-	-	99.8%	-	-	**94.7%**
Timestamp Image & Count	ResNet18	-	-	99.0%	-	-	92.2%	-	-	84.4%	-	-	78.7%	-	-	99.8%	-	-	93.6%
	ResNet50	-	-	99.1%	-	-	92.6%	-	-	82.7%	-	-	71.6%	-	-	**99.9%**	-	-	91.2%
	MobileNetV2	-	-	98.7%	-	-	93.1%	-	-	82.0%	-	-	66.8%	-	-	99.6%	-	-	91.7%
	MobileNetV3-L	-	-	98.7%	-	-	89.8%	-	-	82.1%	-	-	70.1%	-	-	99.7%	-	-	92.4%
	MobileNetV3-S	-	-	98.7%	-	-	90.3%	-	-	77.4%	-	-	72.3%	-	-	99.7%	-	-	94.0%
	InceptionV3	-	-	99.1%	-	-	95.1%	-	-	86.5%	-	-	76.2%	-	-	99.4%	-	-	94.3%
CSTR (mean $\bar{T}_s$ only)	ResNet18	-	**99.3%**	99.2%	-	93.1%	92.7%	-	**80.8%**	84.9%	-	74.2%	74.2%	-	**99.9%**	99.8%	-	93.9%	92.6%
	ResNet50	-	99.2%	99.1%	-	92.4%	93.7%	-	79.6%	84.0%	-	63.8%	67.3%	-	99.8%	99.8%	-	92.5%	93.4%
	MobileNetV2	-	99.1%	99.3%	-	93.6%	92.5%	-	77.9%	85.0%	-	59.5%	61.7%	-	99.8%	**99.9%**	-	**94.2%**	94.5%
	MobileNetV3-L	-	99.0%	98.5%	-	90.8%	92.9%	-	73.8%	83.0%	-	63.6%	63.2%	-	99.0%	98.8%	-	91.8%	93.0%
	MobileNetV3-S	-	98.0%	98.9%	-	92.0%	90.7%	-	72.5%	81.6%	-	56.7%	64.6%	-	99.6%	99.4%	-	91.8%	93.5%
	InceptionV3	-	-	99.2%	-	-	91.7%	-	-	85.0%	-	-	73.1%	-	-	99.8%	-	-	94.5%
CSTR (mean $\bar{T}_s$ & Count)	ResNet18	-	-	99.1%	-	-	93.0%	-	-	81.6%	-	-	77.8%	-	-	**99.9%**	-	-	92.8%
	ResNet50	-	-	99.2%	-	-	92.5%	-	-	85.4%	-	-	70.6%	-	-	**99.9%**	-	-	94.2%
	MobileNetV2	-	-	99.2%	-	-	**95.6%**	-	-	83.0%	-	-	65.2%	-	-	99.8%	-	-	94.6%
	MobileNetV3-L	-	-	98.9%	-	-	93.6%	-	-	82.2%	-	-	65.9%	-	-	99.1%	-	-	93.5%
	MobileNetV3-S	-	-	98.8%	-	-	93.6%	-	-	77.9%	-	-	71.0%	-	-	99.7%	-	-	93.1%
	InceptionV3	-	-	99.2%	-	-	93.5%	-	-	**87.7%**	-	-	79.0%	-	-	**99.9%**	-	-	93.3%

TABLE 6. A full breakdown of the test classification accuracy results that are presented in Table 3 for the event-based recognition datasets. The results of each evaluated spatio-temporal representation are demonstrated for the top-3 pre-trained classification networks which are fine-tuned on each dataset. The best values are highlighted in bold per dataset.

Representation	Augmentation Type			Classifier	Dataset				
	Spatial	Temporal	Polarity		N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	DVS-Gesture
Timestamp Image				ResNet18	99.0%	91.8%	82.3%	76.4%	92.9%
		Baseline		ResNet50	99.1%	94.8%	80.4%	67.6%	93.5%
				InceptionV3	99.3%	93.6%	83.2%	72.3%	94.7%
				ResNet18	99.4%	94.6%	81.6%	76.4%	94.2%
	✓			ResNet50	99.3%	93.9%	85.8%	76.6%	94.2%
				InceptionV3	99.3%	95.1%	85.8%	79.5%	93.9%
				ResNet18	99.1%	96.1%	85.3%	74.9%	93.3%
		✓		ResNet50	99.3%	93.7%	88.2%	73.0%	95.3%
				InceptionV3	99.2%	97.2%	87.7%	80.4%	94.4%
				ResNet18	99.2%	96.0%	83.7%	71.9%	92.9%
			✓	ResNet50	99.0%	95.5%	85.9%	68.5%	94.5%
				InceptionV3	99.2%	95.1%	89.0%	75.2%	94.1%
				ResNet18	99.2%	93.9%	83.6%	75.3%	93.0%
		✓	✓	ResNet50	99.2%	96.2%	88.9%	74.8%	94.0%
				InceptionV3	99.1%	**97.4%**	88.1%	78.8%	94.6%
				ResNet18	99.3%	95.9%	82.6%	78.0%	94.5%
	✓	✓	✓	ResNet50	99.1%	96.3%	86.5%	76.5%	95.0%
				InceptionV3	99.2%	96.8%	86.5%	79.7%	95.2%
Timestamp Image & Count				ResNet18	99.0%	92.2%	84.4%	78.7%	93.6%
		Baseline		ResNet50	99.1%	92.6%	82.7%	71.6%	91.2%
				InceptionV3	99.1%	95.1%	86.5%	76.2%	94.3%
				ResNet18	99.4%	96.3%	82.2%	80.0%	94.4%
	✓			ResNet50	99.4%	94.7%	84.9%	78.5%	94.3%
				InceptionV3	99.4%	96.2%	86.1%	82.6%	95.1%
				ResNet18	99.0%	95.5%	84.8%	76.4%	94.6%
		✓		ResNet50	99.2%	95.5%	87.4%	74.6%	94.6%
				InceptionV3	99.4%	95.1%	89.3%	80.7%	94.6%
				ResNet18	99.4%	96.1%	84.2%	74.0%	91.4%
			✓	ResNet50	99.0%	95.6%	86.4%	69.7%	93.6%
				InceptionV3	99.3%	97.1%	88.8%	76.9%	94.9%
				ResNet18	99.2%	95.9%	86.2%	76.9%	94.2%
		✓	✓	ResNet50	99.2%	94.2%	86.3%	76.6%	94.0%
				InceptionV3	99.4%	96.9%	89.2%	81.0%	95.0%
				ResNet18	99.3%	96.9%	84.3%	79.4%	94.0%
	✓	✓	✓	ResNet50	99.2%	95.1%	87.5%	78.6%	95.2%
				InceptionV3	99.4%	97.1%	87.7%	**82.8%**	94.1%
CSTR (mean $\bar{T}_s$ only)				ResNet18	99.2%	92.7%	84.9%	74.2%	92.6%
		Baseline		ResNet50	99.1%	93.7%	84.0%	67.3%	93.4%
				InceptionV3	99.2%	91.7%	85.0%	73.1%	94.5%
				ResNet18	99.4%	95.7%	83.8%	75.2%	94.5%
	✓			ResNet50	99.3%	95.5%	85.3%	73.6%	96.1%
				InceptionV3	**99.5%**	97.1%	88.0%	78.2%	95.5%
				ResNet18	99.3%	95.1%	85.8%	75.9%	92.3%
		✓		ResNet50	99.4%	96.1%	88.8%	73.8%	94.0%
				InceptionV3	99.3%	88.6%	88.1%	76.7%	93.9%
				ResNet18	99.4%	96.3%	86.5%	70.4%	94.5%
			✓	ResNet50	**99.5%**	96.3%	87.9%	66.7%	95.8%
				InceptionV3	99.2%	96.1%	89.0%	75.5%	94.1%
				ResNet18	99.3%	97.3%	85.8%	75.4%	93.8%
		✓	✓	ResNet50	99.2%	96.2%	89.0%	73.5%	92.6%
				InceptionV3	99.1%	97.1%	**90.1%**	75.5%	94.8%
				ResNet18	99.3%	96.4%	84.9%	77.4%	93.5%
	✓	✓	✓	ResNet50	99.4%	96.4%	86.3%	77.5%	96.0%
				InceptionV3	99.2%	97.0%	88.1%	79.9%	95.6%
CSTR (mean $\bar{T}_s$ & Count)				ResNet18	99.1%	93.0%	81.6%	77.8%	92.8%
		Baseline		ResNet50	99.2%	92.5%	85.4%	70.6%	94.2%
				InceptionV3	99.2%	93.5%	87.7%	79.0%	93.3%
				ResNet18	99.4%	95.9%	81.5%	78.6%	94.5%
	✓			ResNet50	99.4%	96.0%	86.6%	78.0%	96.1%
				InceptionV3	**99.5%**	96.9%	86.8%	81.3%	**96.5%**
				ResNet18	99.3%	94.4%	85.6%	78.2%	94.5%
		✓		ResNet50	99.4%	96.2%	88.4%	76.6%	94.8%
				InceptionV3	99.4%	95.5%	89.8%	80.4%	95.3%
				ResNet18	99.1%	96.4%	86.1%	73.1%	95.3%
			✓	ResNet50	99.4%	94.8%	86.7%	66.8%	95.5%
				InceptionV3	99.3%	97.1%	88.4%	76.9%	94.4%
				ResNet18	99.3%	96.6%	86.7%	77.9%	93.4%
		✓	✓	ResNet50	99.4%	96.2%	88.6%	75.4%	95.4%
				InceptionV3	99.4%	96.9%	89.8%	80.4%	94.5%
				ResNet18	99.3%	96.9%	84.0%	78.8%	95.1%
	✓	✓	✓	ResNet50	**99.5%**	96.9%	86.2%	78.7%	95.7%
				InceptionV3	99.3%	97.2%	88.2%	81.8%	96.3%

REFERENCES

[1] R. Benenson, S. Petti, T. Fraichard, and M. Parent, ''Toward urban driverless vehicles,'' *Int. J. vehicle Auto. Syst.*, vol. 1, no. 6, pp. 4–23, 2008.

[2] A. Bonci, P. D. Cen Cheng, M. Indri, G. Nabissi, and F. Sibona, ''Human-robot perception in industrial environments: A survey,'' *Sensors*, vol. 21, no. 5, p. 1571, Feb. 2021.

[3] E. Onzon, F. Mannan, and F. Heide, ''Neural auto-exposure for high-dynamic range object detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 7710–7720.

[4] G. Chen, H. Cao, J. Conradt, H. Tang, F. Rohrbein, and A. Knoll, ''Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception,'' *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 34–49, Jul. 2020.

[5] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, ''Event-based vision: A survey,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022.

[6] P. Lichtsteiner, C. Posch, and T. Delbruck, ''A 128$\times$128 120 db 15 μs latency asynchronous temporal contrast vision sensor,'' *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Mar. 2008.

[7] C. Posch, D. Matolin, and R. Wohlgenannt, ''A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS,'' *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[8] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, ''A 240 $\times$ 180 130 dB 3 μs latency global shutter spatiotemporal vision sensor,'' *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct. 2014.

[9] D. Falanga, K. Kleber, and D. Scaramuzza, ''Dynamic obstacle avoidance for quadrotors with event cameras,'' *Sci. Robot.*, vol. 5, no. 40, Mar. 2020, p. eaaz9712.

[10] D. Falanga, S. Kim, and D. Scaramuzza, ''How fast is too fast? The role of perception latency in high-speed sense and avoid,'' *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1884–1891, Apr. 2019.

[11] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, ''DSEC: A stereo event camera dataset for driving scenarios,'' *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4947–4954, Jul. 2021.

[12] C. Zhou, M. Teng, J. Han, J. Liang, C. Xu, G. Cao, and B. Shi, ''Deblurring low-light images with events,'' *Int. J. Comput. Vis.*, vol. 131, no. 5, pp. 1284–1298, May 2023.

[13] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, ''Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios,'' *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[14] J. Redmon and A. Farhadi, ''YOLOv3: An incremental improvement,'' 2018, *arXiv:1804.02767*.

[15] K. He, X. Zhang, S. Ren, and J. Sun, ''Deep residual learning for image recognition,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[16] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, ''Unsupervised event-based learning of optical flow, depth, and egomotion,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 989–997.

[17] X. Lagorce, G. Orchard, F. Galluppi, B. E. Shi, and R. B. Benosman, ''HOTS: A hierarchy of event-based time-surfaces for pattern recognition,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[18] M. Iacono, S. Weber, A. Glover, and C. Bartolozzi, ''Towards event-driven object detection with Off-the-Shelf deep learning,'' in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1–9.

[19] R. Ghosh, A. Mishra, G. Orchard, and N. V. Thakor, ''Real-time object recognition and orientation estimation using an event-based camera and CNN,'' in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 544–547.

[20] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, ''Graph-based spatio-temporal feature learning for neuromorphic vision sensing,'' *IEEE Trans. Image Process.*, vol. 29, pp. 9084–9098, 2020.

[21] A. Zihao Zhu, L. Yuan, K. Chaney, and K. Daniilidis, ''Unsupervised event-based optical flow using motion compensation,'' in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, pp. 1–4.

[22] D. Gehrig, A. Loquercio, K. Derpanis, and D. Scaramuzza, ''End-to-end learning of representations for asynchronous event-based data,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5633–5643.

[23] R. W. Baldwin, R. Liu, M. Almatrafi, V. Asari, and K. Hirakawa, ''Time-ordered recent event (TORE) volumes for event cameras,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, pp. 2519–2532, Feb. 2023.

[24] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, ''Converting static image datasets to spiking neuromorphic datasets using saccades,'' *Frontiers Neurosci.*, vol. 9, p. 437, Nov. 2015.

[25] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, ''HATS: Histograms of averaged time surfaces for robust event-based object classification,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1731–1740.

[26] H. Li, H. Liu, X. Ji, G. Li, and L. Shi, ''CIFAR10-DVS: An event-stream dataset for object classification,'' *Frontiers Neurosci.*, vol. 11, p. 309, May 2017.

[27] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, ''Graph-based object classification for neuromorphic vision sensing,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 491–501.

[28] 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, and D. Modha, ''A low power, fully event-based gesture recognition system,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7243–7252.

[29] C. Scheerlinck, N. Barnes, and R. Mahony, ''Continuous-time intensity estimation using event cameras,'' in *Computer Vision—ACCV 2018*. Perth, SC, Australia: Springer, 2019, pp. 308–324.

[30] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, ''Event-based, 6-DOF camera tracking from photometric depth maps,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2402–2412, Oct. 2018.

[31] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. Davison, ''Simultaneous mosaicing and tracking with an event camera,'' in *Proc. Brit. Mach. Vis. Conf.* Durham, U.K.: BMVA Press, 2014, pp. 1–9.

[32] J. Dong, R. Jiang, R. Xiao, R. Yan, and H. Tang, ''Event stream learning using spatio-temporal event surface,'' *Neural Netw.*, vol. 154, pp. 543–559, Oct. 2022.

[33] W. Fang, Z. Yu, Y. Chen, T. Masquelier, T. Huang, and Y. Tian, ''Incorporating learnable membrane time constant to enhance learning of spiking neural networks,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2661–2671.

[34] F. Paredes-Vallés, K. Y. W. Scheper, and G. C. H. E. de Croon, ''Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2051–2064, Aug. 2020.

[35] M. Yao, H. Gao, G. Zhao, D. Wang, Y. Lin, Z. Yang, and G. Li, ''Temporal-wise attention spiking neural networks for event streams classification,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10221–10230.

[36] Q. Meng, M. Xiao, S. Yan, Y. Wang, Z. Lin, and Z.-Q. Luo, ''Training high-performance low-latency spiking neural networks by differentiation on spike representation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12444–12453.

[37] M. Pfeiffer and T. Pfeil, ''Deep learning with spiking neurons: Opportunities and challenges,'' *Frontiers Neurosci.*, vol. 12, p. 774, Oct. 2018.

[38] Y. Suh, S. Choi, M. Ito, J. Kim, Y. Lee, J. Seo, H. Jung, D.-H. Yeo, S. Namgung, and J. Bong, ''A 1280$\times$ 960 dynamic vision sensor with a 4.95-μm pixel pitch and motion artifact minimization,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[39] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, ''Event-based vision meets deep learning on steering prediction for self-driving cars,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 5419–5427.

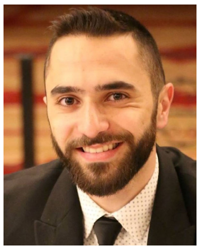
[40] M. Liu and T. Delbruck, ''Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors,'' in *Proc. BMVC*, 2018, pp. 1–12.

[41] F. Baghaei Naeini, D. Makris, D. Gan, and Y. Zweiri, ''Dynamic-vision-based force measurements using convolutional recurrent neural networks,'' *Sensors*, vol. 20, no. 16, p. 4469, Aug. 2020.

[42] P. K. J. Park, B. Hwan Cho, J. Man Park, K. Lee, H. Young Kim, H. Ah Kang, H. Goo Lee, J. Woo, Y. Roh, W. Jo Lee, C.-W. Shin, Q. Wang, and H. Ryu, ''Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique,'' in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1624–1628.

[43] J. Kim, J. Bae, G. Park, D. Zhang, and Y. M. Kim, ''N-ImageNet: Towards robust, fine-grained object recognition with event cameras,'' in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2126–2136.

[44] I. Alonso and A. C. Murillo, "EV-SegNet: Semantic segmentation for event-based cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1624–1633.
[45] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Proc. Robot., Sci. Syst.*, Pittsburgh, PA, USA, Jun. 2018, pp. 1–9.
[46] Y. Wang, X. Zhang, Y. Shen, B. Du, G. Zhao, L. Cui, and H. Wen, "Event-stream representation for human gaits identification using deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3436–3449, Jul. 2022.
[47] W. Bai, Y. Chen, R. Feng, and Y. Zheng, "Accurate and efficient frame-based event representation for AER object recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–6.
[48] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3857–3866.
[49] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.
[50] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "High speed and high dynamic range video with an event camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 1964–1980, Jun. 2021.
[51] B. Xie, Y. Deng, Z. Shao, H. Liu, and Y. Li, "VMV-GCN: Volumetric multi-view based graph CNN for event stream classification," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1976–1983, Apr. 2022.
[52] S. Schaefer, D. Gehrig, and D. Scaramuzza, "AEGNN: Asynchronous event-based graph neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12371–12381.
[53] H. Ren, W. Lu, Y. Xiao, X. Chang, X. Wang, Z. Dong, and D. Fang, "Graph convolutional networks in language and vision: A survey," *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109250.
[54] Y. Li, Y. Kim, H. Park, T. Geller, and P. Panda, "Neuromorphic data augmentation for training spiking neural networks," in *Computer Vision–ECCV*. Tel Aviv, Israel: Springer, Oct. 2022, pp. 631–649.
[55] F. Gu, W. Sng, X. Hu, and F. Yu, "EventDrop: Data augmentation for event-based learning," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, Aug. 2021, pp. 1–13.
[56] G. Shen, D. Zhao, and Y. Zeng, "EventMix: An efficient data augmentation strategy for event-based learning," *Inf. Sci.*, vol. 644, Oct. 2023, Art. no. 119170.
[57] F. B. Naeini, S. Kachole, R. Muthusamy, D. Makris, and Y. Zweiri, "Event augmentation for contact force measurements," *IEEE Access*, vol. 10, pp. 123651–123660, 2022.
[58] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, "Space-time event clouds for gesture recognition: From RGB cameras to event cameras," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1826–1835.
[59] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[60] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015.



ZAID A. EL SHAIR received the B.Sc. degree in computer engineering from the Princess Sumaya University for Technology, Amman, Jordan, in 2018, and the M.S.E. degree in computer engineering from the University of Michigan–Dearborn, Dearborn, MI, USA, in 2019, where he is currently pursuing the Ph.D. degree in computer engineering.

He has been a Graduate Student Instructor with the University of Michigan–Dearborn, since 2018, and has interned twice with Ford Motor Company, in 2020 and 2021. During the graduate studies, he has published several peer-reviewed research papers in different domains, such as computer vision, machine learning, and embedded systems. His research interests include the development of innovative deep learning-based solutions for computer vision tasks and advancing event-based vision techniques to enhance object detection and tracking, among other applications.

ALI HASSANI (Member, IEEE) was born in Ann Arbor, MI, USA, in 1992. He received the B.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2014, and the master's and Ph.D. degrees in computer engineering from the University of Michigan–Dearborn, Dearborn, MI, USA, in 2019 and 2022, respectively.

He has been a Research Engineer with Ford Motor Company since starting his professional career, in 2014. He has produced over 150 patent applications for various automotive technologies, as well as a small collection of peer-reviewed academic works in the facial biometrics field. His research interests include facial analysis using AI/ML, including recognition, liveliness, emotion, and other relevant biometrics.

SAMIR A. RAWASHDEH (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from The University of Jordan, Amman, Jordan, in 2007, and the M.S.E.E. and Ph.D. degrees in electrical engineering from the University of Kentucky, Lexington, KY, USA, in 2009 and 2013, respectively. In 2014, he joined the Electrical and Computer Engineering Department, University of Michigan–Dearborn, where he is currently an Associate Professor. His research interests include robot perception, embedded systems, and smart health.