IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 17, 2021, accepted May 23, 2021, date of publication June 3, 2021, date of current version June 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3085708

Making Sense of Neuromorphic Event Data for Human Action Recognition

SALAH AL-OBAIDI, HIBA AL-KHAFAJI, AND CHARITH ABHAYARATNE, (Member, IEEE)
Department of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S1 3JD, U.K.

Corresponding author: Charith Abhayaratne (c.abhayaratne@sheffield.ac.uk)

ABSTRACT Neuromorphic vision sensors provide low power sensing and capture salient spatial-temporal events. The majority of the existing neuromorphic sensing work focus on object detection. However, since they only record the events, they provide an efficient signal domain for privacy aware surveillance tasks. This paper explores how the neuromorphic vision sensor data streams can be analysed for human action recognition, which is a challenging application. The proposed method is based on handcrafted features. It consists of a pre-processing step for removing the noisy events followed by the extraction of handcrafted local and global feature vectors corresponding to the underlying human action. The local features are extracted considering a set of high-order descriptive statistics from the spatio-temporal events in a time window slice, while the global features are extracted by considering the frequencies of occurrences of the temporal event sequences. Then, low complexity classifiers, such as, support vector machines (SVM) and K-Nearest Neighbours (KNNs), are trained using these feature vectors. The proposed method evaluation uses three groups of datasets: Emulator-based, re-recording-based and native NVS-based. The proposed method has outperformed the existing methods in terms of human action recognition accuracy rates by 0.54%, 19.3%, and 25.61% for E-KTH, E-UCF11 and E-HMDB51 datasets, respectively. This paper also reports results for three further datasets: E-UCF50, R-UCF50, and N-Actions, which are reported for the first time for human action recognition on neuromorphic vision sensor domain.

INDEX TERMS Neuromorphic vision sensing (NVS), event cameras, dynamic vision sensing (DVS), human action recognition (HAR), local features, global features.

I. INTRODUCTION

Neuromorphic vision sensing (NVS), also known as dynamic vision sensing and event camera sensing, which has emerged recently, is capable of capturing fast spatio-temporal spikes (changes) in a scene with low power consumption [1]–[8]. Such data is of the form of a continuous stream of spatio-temporal *events* or *spikes*, as opposed to regularly uniformly spatio-temporal sampled values traditions imaging systems, as in active pixel sensing (APS). This allows NVS to measure changes in intensity at each pixel asynchronously, instead of acquiring the intensity of that pixel, *i.e.*, non-uniformly sampling temporally leading to a rendering frame rate up to 2000 fps with consuming low power. It encodes the intensity change at each pixel in the form of an *event* or a *spike*. FIGURE 1 shows an example of a stream of events for a person running. This stream is represented as 3D points, referring to the spatial location (in terms of (x,y) coordinates and the time of the event. An event is recorded either as an initiation (ON) and a termination (OFF), shown in green and red points, respectively in FIGURE 1. Each event in this figure has potentially valid information that can be explored to understand the scene, in terms of object and action recognition. Although success has been reported in visual content understanding using traditional imaging in the

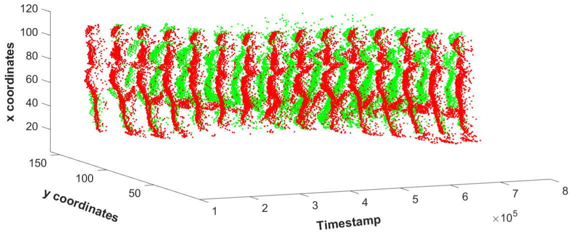


FIGURE 1. Representation of the events of a running action using an emulator to generate the events. Green/Red points are for visualisation of ON and OFF events.

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

literature, in order to optimally use event or spike data from NVS, novel algorithms for processing and learning such data are needed. In this work, we explore how NVS data can be analysed for human action recognition (HAR).

HAR from video sequences captured using conventional APS imaging systems primarily detect and model motion patterns to learn important features of an action to train a classifier [9]–[22]. The features can be either handcrafted or learned by deep learning approaches. Although these have shown very high accuracy rates for benchmark datasets, such conventional vision systems often suffer from many limitations, such as, limited frame rate, high redundancy within the successive frames and motion blurring, affecting the performance of action recognition [22], [23]. Also, pre-processing steps, such as estimating motion from video pixels (block matching, optic flow or phase correlation) are computationally expensive. Furthermore, conventional video-based HAR has also caused privacy issues in the context of assisted living [24]–[26]. Exploration of NVS data for HAR also enables to overcome some of these limitations intrinsic to conventional imaging-based HAR. As NVS encodes the intensity change at each pixel and samples at non-uniform sampling rates, events with high frequency of occurrences correspond to high motion present in the scene, which is a solution for motion blurring due to high speed motion as often seen in conventional APS cameras. Such a high motion response means that NVS based camera is regarded as a data-driven sensor since the output NVS depends on the magnitude of the apparent motion in the scene [23]. These advantages combined with low power consumption and low throughput for streaming have emerged NVS as a suitable vision sensor for robotics and mobile-based applications [27]–[29].

Although NVS-based vision applications have seen emerged fast recently [23], it has not resulted in many works in human motion analysis. Most recent works exploring NVS data for human motion analysis consists of low semantic tasks, such as, hand or finger movement analysis [30]–[37] and human fall detection [38]. However, exploring NVS data for higher-level semantic tasks, such as, multi-class HAR, is still in early stages [39]–[43]. One reason for this slow progress of NVS domain HAR is the high cost of NVS devices compared to the conventional APS cameras [8] leading to insufficient annotated NVS domain HAR training datasets [41], [44]. Recently emerged software-based emulators for converting APS data into NVS data [45], [46] were also found useful for generating test data.

However, rather than extending conventional HAR approaches used in classical computer vision, new paradigms are need to be explored for efficiently understanding NVS data for HAR and other applications. Some of the challenges in NVS data include presence of noisy events, understanding true motion, lack of clarity of contexts in object boundaries due to lack of intensity data, high sparsity of data and missing spatio-temporal connectivity in NVS data. Therefore, effective NVS-domain feature extraction algorithms are needed for the advancement of usage of NVS devices in real applications. In this work, we present a novel methodology for efficient understanding of NVS data for HAR applications. The main contributions of our work include:

1) A methodology for pre-processing NVS data including a new algorithm for de-noising NVS data, *i.e.*, to remove the noisy events that may have been resulted in due to certain acquisition parameters used in NVS devices;
2) A methodology for extracting a new set of global temporal features to model the global (long term) motion patterns considering a long duration NVS event data stream;
3) A methodology for extracting a new set of local spatio-temporal features to model local (short term) motion patterns considering a shorter durations of connected events in short durations of an NVS event data stream; and
4) Fusion of features for training a classifier and evaluation of the proposed method for various types of NVS data (real data, emulated data and recorded NVS from an RGB playback) covering various types of actions.

The rest of this paper is organized as follows: Section II reviews the related work on exploring the NVS domain for HAR. In Section III, we present the proposed methodology for understanding NVS data for HAR. Section IV shows the experimental evaluation of the performance of the proposed method and discussion followed by the concluding remarks in Section V.

II. RELATED WORK

The existing work on neuromorphic vision sensing in computer vision can be grouped into three themes: object detection [47]–[49], pedestrian detection [50], [51] and hand gesture recognition [33]–[35]. There is only a little work on exploring the neuromorphic data beyond object detection addressing highly semantic applications, such as, multi class action recognition, which still poses an important challenge. As mentioned in Section I, work on using NVS data for HAR is still in early stages [39]–[43]. Most of these methods start with temporally aggregating the polarities into a collection of NVS data frames by considering a non-overlapping time window corresponding to the frame rate of conventional APS cameras. This is followed by using these NVS frames for either extracting handcrafted motion features or learned features for representing the actions in the test sequences.

In [41], 8-bit gray-scale frames are constructed from the events. Pixels of these frames are initialised with 128 and then either are increased or decreased considering the polarities of the events recorded at each spatial location (pixel) by considering the time interval corresponding to an actual frame. This is followed by extracting motion event features (magnitude and direction of motion) considering stacked event frames with variable stack sizes depending on the

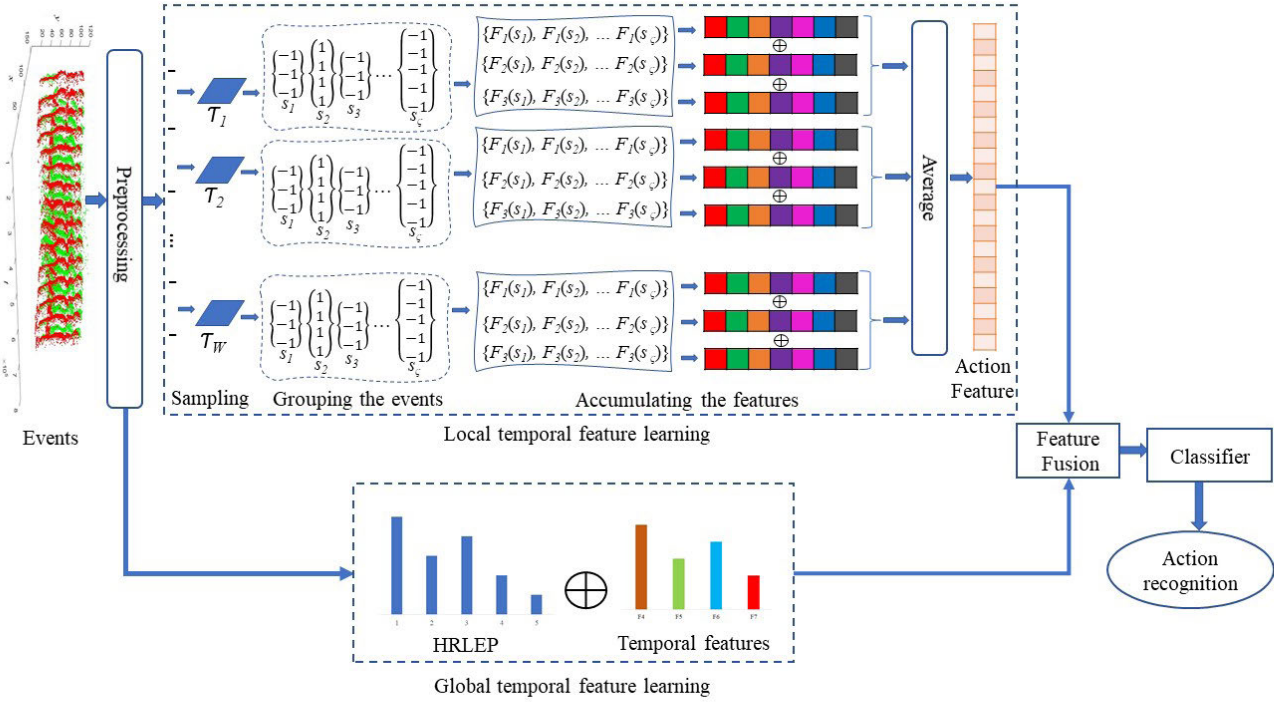

Events
Preprocessing
T_1
T_2
T_W
s_1 s_2 s_3 ... s_ς
$\{F_1(s_1), F_1(s_2), \dots F_1(s_\varsigma)\}$
$\{F_2(s_1), F_2(s_2), \dots F_2(s_\varsigma)\}$
$\{F_3(s_1), F_3(s_2), \dots F_3(s_\varsigma)\}$
⊕
Average
Action Feature
Sampling
Grouping the events
Accumulating the features
Local temporal feature learning
Feature Fusion
Classifier
Action recognition
HRLEP
Temporal features
Global temporal feature learning

FIGURE 2. The pipeline of the proposed method for NVS domain HAR.

motion level in the activity. Finally, these motion information is fed into a convolution neural network (CNN) for feature learning for HAR. They demonstrate the usability of event data in HAR compared to conventional camera-based vision systems, where complex optic flow estimation is required. A similar approach was followed in [40] by using two CNNs to learn features from event frames and corresponding optic flow from the original RGB. Converting the events into frames is also applied in [43] to classify the actions of the neuromorphic version of UCF11 dataset. A time stamp aggregation algorithm is used to create the frames from the events, where these frames are fed into CNN for classification with a 92.90% of accuracy. In [39], three 2D motion maps (on x-y, x-z and y-z planes) and Motion Boundary Histogram (MBH) are constructed from the events. Speeded Up Robust Features (SURF) are extracted through grid search on the 2D motion maps followed by k-means clustering to create a Bag of visual vocabulary (BoVV) of k words from motion maps and MBH. Finally, the feature vectors constructed from BoVV are used to train the linear SVM. Also Graph CNN based methods were reported for NVS-domain object recognition, with case studies on HAR in [42].

The existing work that uses hand-crafted features has achieved an accuracy rate of 75.13% [39], while the works that have used deep learning have achieved accuracy rates ranging from 51.5% to 92.9% [40]–[43] as detailed in Section IV-B. It can be also observed that the accuracy rates of these methods depend on the quality of the constructed event frames. The choice of time intervals plays a significant role in this. All these methods create motion maps from NVS events followed by either handcrafted feature learning or deep learning. In either way, they do not take the full advantage of NVS event data, which can be considered as motion information. Following the approach of frame creation or motion parameter estimation has added complexity similar to conventional cameras based vision algorithms. Therefore, in our present work, we focus on extracting features on the NVS event domain, *i.e.*, exploring the events directly, for HAR. Accordingly, we propose a new method that explores the NVS domain alone by considering the temporal patterns of ON and OFF events locally and globally to extract robust description for HAR. The proposed method analyses the patterns of the polarities using only NVS domain events and avoids converting the events into other domains without losing the essence of neuromorphic computing.

III. THE PROPOSED NVS DOMAIN FEATURE LEARNING

This section presents the proposed method for NVS domain HAR including the novel contributions on noise removal and constructing local and global spatio temporal event descriptors. FIGURE 2 depicts the block diagram of the proposed method with the pipeline of operations. The proposed method is divided into five main steps: pre-processing for noisy event removal, NVS domain local feature extraction, NVS domain global feature extraction, feature fusing, and classification. We start this section by introducing the NVS operation and the notation followed by the description of the main steps of the proposed method in subsequent subsections.

A. NVS OPERATION AND OUR NOTATIONS

In contrast to the standard pixel-domain based camera, where the sensors record the information of pixels at a constant frame rate, the NVS acquires the change of luminance with a variable sampling rate at each pixel. Accordingly, an event is triggered if the luminance at a pixel changes, *i.e.*, the log intensity exceeding a predefined threshold is sufficient to be considered as an event. This mechanism is performed independently and continuously for each pixel in the chip's array in NVS cameras, and the pixel is set to idle in case there is no luminance change has been detected, leading to temporally and spatially adapted independent and non-uniform temporal sampling for each pixel.

We denote an event, e_k, acquired at the coordinates, x_k and y_k, corresponding to a pixel, P_k, in the sensor array and at the timestamp, t_k, with the polarity p_k, *i.e.*, the orientation of the shifted log intensity, $\mathcal{L}P_k = \log(I_k)$, where k is the event index and I_k is the intensity at P_k. Thus, an event is represented as $e_k = (x_k, y_k, t_k, p_k)$ as soon as the magnitude of $\mathcal{L}(P_k)$ is shifted since the last event recorded at P_k, *i.e.*,

$$\Delta\mathcal{L}\left(x_k, y_k, t_k\right) = \mathcal{L}\left(x_k, y_k, t_k\right) - \mathcal{L}\left(x_k, y_k, t_k - \Delta t\right), \quad (1)$$

exceeds a temporal contrast threshold [7]. Δt is the time when the pixel P_k is idle since the last event at P_k. When the log intensity at P_k exceeds the, e_k is triggered with the polarity, $p_k \in \{-1, 1\}$, *i.e.*, the orientation of log intensity change, $\Delta\mathcal{L}$. It can be noticed that Eq. (1) is similar to finding the pixel difference between successive frames in conventional cameras based computer vision. This pixel difference, *i.e.*, log intensity, is evidence of the presence of motion in the scene. Therefore, this allows us to infer the implied motion in the scene by exploiting the events statistics rather than going through computationally expensive motion estimation algorithms often used in computer vision applications.

B. PRE-PROCESSING THE NOISY EVENTS

Depending on the threshold magnitude, some events are recorded in isolation without leading to any semantic meaning. We denote such events as noisy events and a pre-processing step for removing such events (de-noising) is applied on the events stream.

Let $\mathbb{E} = \{e_n | e_n = (x_n, y_n, t_n, p_n), \text{ and } 1 \leq n \leq N\}$, is a stream of events, where N is the length of the event stream. $\mathbb{E}$ is partitioned into time slices, $\mathbb{T} = \{\mathcal{T}_w | 1 \leq w \leq W\}$, where $\mathcal{T}_w$ is the time slice w. This partitioning is based on the principle of the frame rate that one would expect for a conventional camera video sequence. For example, if we have an NVS stream for 5 seconds, we generate 150 event slices assuming a 30 frames per second frame rate.

After partitioning the stream into event slices, for each slice let $\mathbf{E}_w = \{e_\ell | e_\ell = (x_\ell, y_\ell, t_\ell, p_\ell), \text{ and } 1 \leq \ell \leq L\}$, be the event stream in slice w, where L is the length of the total event stream in a slice, the following operations are applied. For each event e_ℓ at spatio-temporal location (x_ℓ, y_ℓ, t_ℓ), a 3 × 3 window on xy plane centered on the event location (x_ℓ, y_ℓ, t_ℓ) is considered and the number of events $C_{\ell_{(x,y)}}$ recorded on each of nine spatial coordinates (x, y) of the window over the total time of the slice is counted. This is followed by computing the total number of events in the 3D window-slice, S_ℓ, and the maximum events over the slice length, m_ℓ, as follows:

$$S_\ell = \sum_{i=x_\ell-1}^{x_\ell+1} \sum_{j=y_\ell-1}^{y_\ell+1} C_{\ell_{(i,j)}}. \quad (2)$$

$$m_\ell = \max_{i=x_\ell-1, j=y_\ell-1}^{i=x_\ell+1, j=y_\ell+1} C_{\ell_{(i,j)}}. \quad (3)$$

Finally, e_ℓ is processed to obtain new polarity, p'_ℓ, of the event as follows:

$$p'_\ell = \begin{cases} p_\ell & \text{if } S_\ell < (k \times 3 \times 3 \times m_\ell), \\ 0 & \text{otherwise}, \end{cases} \quad (4)$$

where $\{k \in \mathbb{R}^+ | k < 1\}$ is a user defined parameter for controlling the number of events to be removed. We present a discussion on the choice of the parameter k in Section IV-A.

C. LOCAL SPATIO-TEMPORAL FEATURE EXTRACTION

An action event stream can be represented considering the overall spatio-temporal patterns appear in the overall action sequence, as well as considering the local variations corresponding to the actions. In this section we address how to extract local features from the events stream, considering the events in partitioned time slices, $\mathcal{T}_w$. Since each action results in different spatio-temporal patterns of events at each time window, the local descriptors aim to recognise these patterns leading to representing discriminating features for specific action streams.

The process is started with $\mathbf{E}_w$ at $\mathcal{T}_w$, by sorting all e_ℓ in the ascending order of the x coordinate followed by grouping these events in $\mathcal{T}_w$ into $\left\{s_g | 1 \leq g \leq G\right\}$, where s_g defines ρ events that are successive and have the same polarity, such that,

$$s_g = \{e_i | e_i = (x_i, y_i, t_i, p_i), \text{ and } 1 \leq i \leq \rho\}, \quad (5)$$

where $x_{i+1} \geq x_i$ and $p_{i+1} = p_i \; \forall i$. According to Eq. (5), all events in s_g represent a pattern of log intensity change. Processing such patterns of polarities contributes to tracking the dynamic changes for each action and capturing the local structure of the events. This is achieved by modelling these changes in terms the relationship of horizontal and vertical locations, *i.e.*, (x, y) coordinates of the events in each set, s_g in terms of the following quantities:

$$m_g = \mu_x(s_g) - \mu_y(s_g), \quad (6)$$

$$v_g = \sigma_x^2(s_g) - \sigma_y^2(s_g), \quad (7)$$

$$d_g = \sigma_x(s_g) - \sigma_y(s_g), \quad (8)$$

where μ, σ^2 and σ are the mean, variance and the standard deviation of the spatial coordinates x and y of the events in s_g, respectively. This gives us three data vectors, $\mathbf{M}_w = \{m_g | 1 \leq g \leq G\}$, $\mathbf{V}_w = \{v_g | 1 \leq g \leq G\}$ and $\mathbf{D}_w = \{m_d | 1 \leq g \leq G\}$, for each $\mathcal{T}_w$. Then these data vectors are transformed

Algorithm 1 RLE of Polarities in a Stream of Events With N Events

```
Initialize $Count \longleftarrow 0$.
Initialize $RunLengths \longleftarrow [\,]$.
for $i \longleftarrow 1$ to $N$ do
    if $p_i = p_{i+1}$ then
        $Count \longleftarrow Count + 1$,
    else
        $RunLengths \longleftarrow [RunLengths\ \ Count]$.
        $Count \longleftarrow 0$.
    end if
end for
Return $RunLengths$.
```

into 3 vectors containing higher order statistics of the data vectors as follows:

$$F_{1_w} = [\mu(\mathbf{M}_w), \max(\mathbf{M}_w), \min(\mathbf{M}_w), \sigma(\mathbf{M}_w), \ldots \ldots \sigma^2(\mathbf{M}_w), \gamma(\mathbf{M}_w), \kappa(\mathbf{M}_w)], \tag{9}$$

$$F_{2_w} = [\mu(\mathbf{V}_w), \max(\mathbf{V}_w), \min(\mathbf{V}_w), \sigma(\mathbf{V}_w), \ldots \ldots \sigma^2(\mathbf{V}_w), \gamma(\mathbf{V}_w), \kappa(\mathbf{V}_w)], \tag{10}$$

$$F_{3_w} = [\mu(\mathbf{D}_w), \max(\mathbf{D}_w), \min(\mathbf{D}_w), \sigma(\mathbf{D}_w), \ldots \ldots \sigma^2(\mathbf{D}_w), \gamma(\mathbf{D}_w), \kappa(\mathbf{D}_w)], \tag{11}$$

where γ and κ denote the skewness and the kurtosis, respectively. Then for each element in feature vectors, F_{1_w}, F_{2_w} and F_{3_w} the average over all W slices are computed to get the average feature vectors, F_1, F_2 and F_3, respectively. These three vectors are concatenated to get the local feature vector, $\mathcal{F}_L = \{F_1, F_2, F_3\}$, with 21 feature elements for the event stream $\mathbb{E}$. As an example, mean values of these feature vector elements for six sequences of one of the datasets (E-KTH) in FIGURE 3.

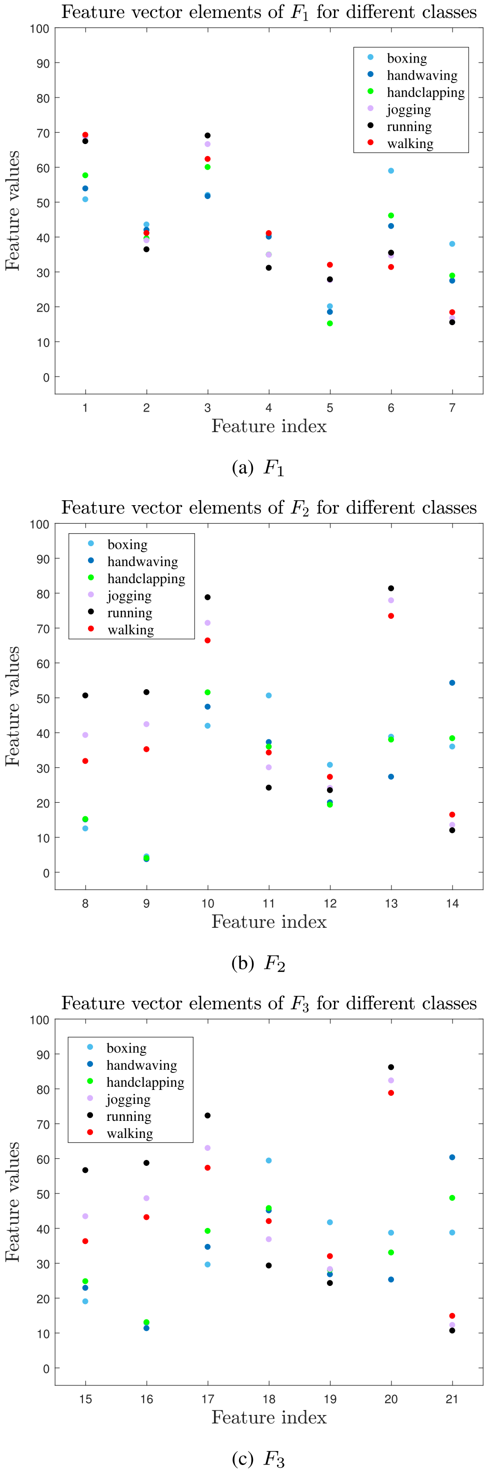
D. GLOBAL FEATURE EXTRACTION

Global features are extracted by considering the event stream for an action as a whole without resorting it into time-based slices. On the spatio-temporal event space, for each spatial coordinate (x, y), all temporal events are stacked into temporal groups, $\mathcal{H}_{\mathbb{E}} = \{\delta_h | 1 \leq h \leq H\}$, where H is the total number of temporal groups for the given (x, y). A group is defined as the continuous occurrence of events (either $p_l = +1$ or $p_l = -1$) at user-specified temporal sampling periods. The minimum events for a group is considered as 2, while just the isolated single events are disregarded as noise. For all events in δ_h, the consecutive similar polarity counts recorded as run-length encoding (RLE) as detailed in Algorithm 1. RLE keeps only the counts of consecutive occurrences without keeping the magnitudes of the polarities. Run lengths of all $\mathcal{H}_{\mathbb{E}}$ for all spatial locations are collected as a set, $\mathbb{R}$.

The first part of the global feature vector represents the spatial locations, $\mathbb{R}$, by computing the histogram of run-length encoded polarities (HRLEP), $\mathbb{H}$. Our experiments have found that partitioning HRLEP into 5 bins is sufficient to capture the discriminative features from $\mathbb{R}$. Then the global temporal feature vector, $\mathcal{F}_G(\mathbb{E})$ consisting of the 5-bin $\mathbb{H}$ and four other global features considering both $\mathbb{R}$ and $\mathbb{E}$ for the whole event stream as follows: $\mathcal{F}_G(\mathbb{E}) = \{\mathbb{H}, F_4, F_5, F_6, F_7\}$, where

$$F_4 = \max(\mathbb{R}), \tag{12}$$

$$F_5 = \max(W), \tag{13}$$

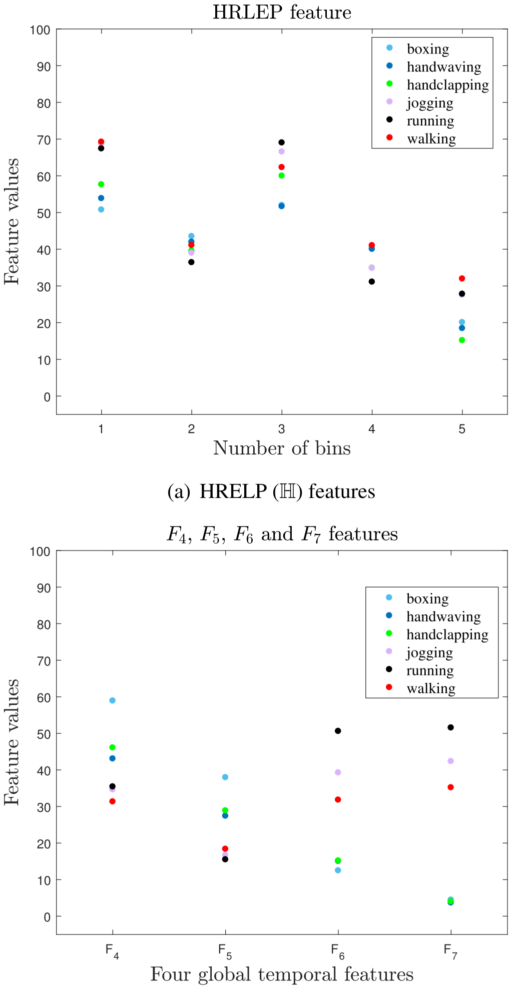


(a) F_1

(b) F_2

(c) F_3

FIGURE 3. Local features ($\mathcal{F}_L$) for six human actions in E-KTH dataset. (Values are normalized in the 0-100 region for visualization).



(a) HRELP ($\mathbb{H}$) features

(b) F_4, F_5, F_6, F_7 features

FIGURE 4. Global features ($\mathcal{F}_G$) for six human actions in E-KTH dataset. (Values are normalized in the 0-100 region for visualization).

F_6 and F_7 are the number of ON and OFF events in $\mathbb{E}$, respectively. The global features extracted from six sequences of the E-KTH dataset are shown in FIGURE 4 as an example.

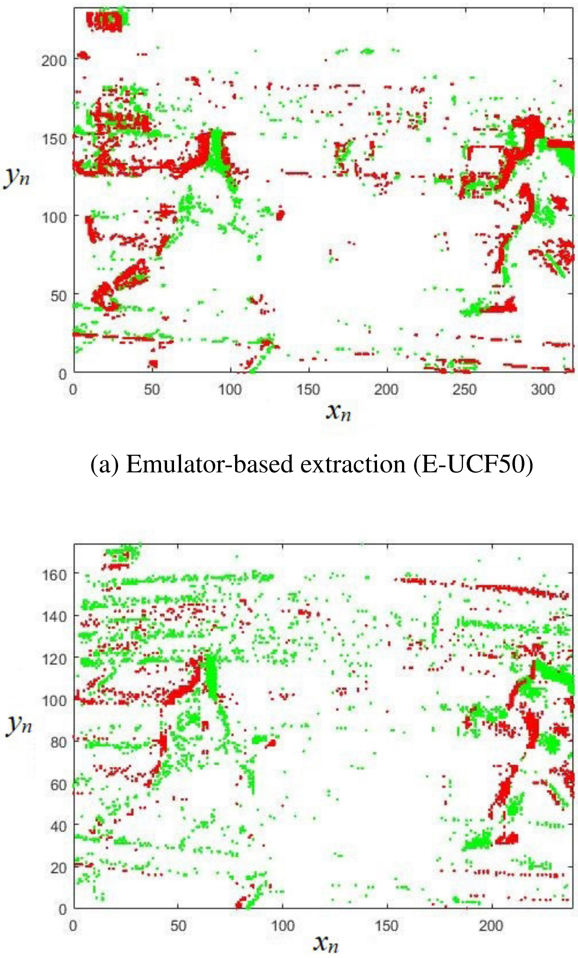
E. FEATURE FUSION AND CLASSIFICATION

Finally, both $\mathcal{F}_L(\mathbb{E})$ and $\mathcal{F}_G(\mathbb{E})$ are fused to construct an overall feature vector, $\mathbb{F}(\mathbb{E})$, as

$$\mathbb{F}(\mathbb{E}) = \{\mathcal{F}_L(\mathbb{E}), \mathcal{F}_G(\mathbb{E})\}. \tag{14}$$

This $\mathbb{F}(\mathbb{E})$ is a 30 dimensions feature vector to represent the action in $\mathbb{E}$, and it is used to train the classifier for recognising the actions.

We conducted our experiments with several classifiers and found that the best results are obtained with KNN and QSVM. On one hand, from the complexity perspective, these classifiers have less complexity, especially KNN, compared to other classifiers. On the other hand, these classifiers are commonly used in the applications of computer vision for their efficiency, therefore, it is easy to compare with the existing work.



(a) Emulator-based extraction (E-UCF50)

(b) NVS device re-recording-based (R-UCF50)

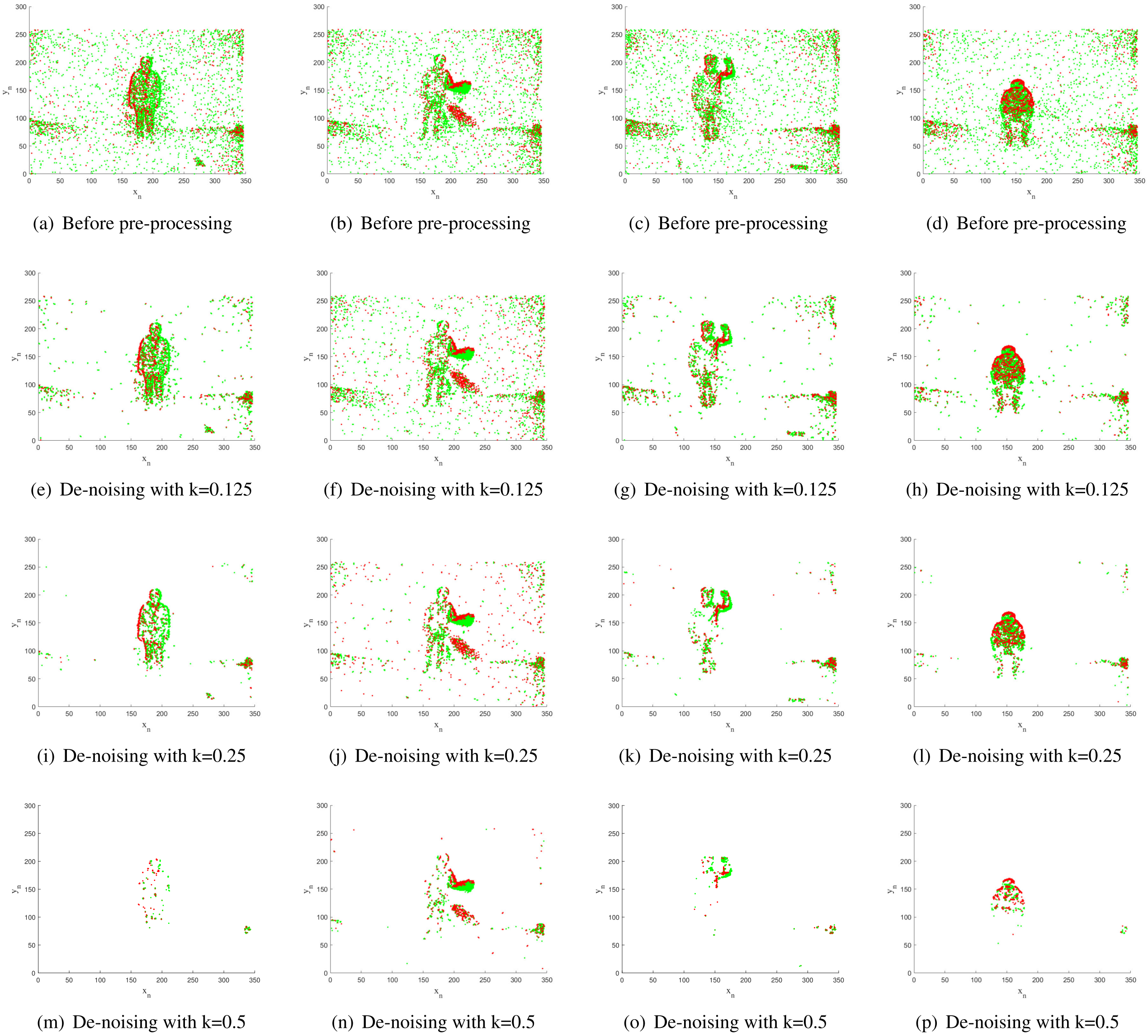
FIGURE 5. Two examples for the same frame from a fencing sequence in UCF50 dataset explaining the amount and the distribution of the events in each frame: (a) PIX2NVS emulator has been used to generate the stream of the events and (b) The DVS240C camera has been used to acquire the events. For visualisation, the ON and OFF events are plotted with green and red colours, respectively.

IV. PERFORMANCE EVALUATION

This section reports the extensive experiments conducted using challenging datasets to evaluate the performance of the proposed methodology for using NVS data for human action recognition.

A. DATASETS AND EXPERIMENTS SET UP

The publicly available and widely used NVS datasets can be categorised into three main groups: Emulator based datasets generated from the commonly used RGB datasets; datasets of NVS devices based re-recording of RGB video displayed on a monitor and datasets of actions acquired by native NVS devices. In our naming of datasets we identify these three groups with the prefixes E-, R- and N-, respectively in



(a) Before pre-processing
(b) Before pre-processing
(c) Before pre-processing
(d) Before pre-processing
(e) De-noising with k=0.125
(f) De-noising with k=0.125
(g) De-noising with k=0.125
(h) De-noising with k=0.125
(i) De-noising with k=0.25
(j) De-noising with k=0.25
(k) De-noising with k=0.25
(l) De-noising with k=0.25
(m) De-noising with k=0.5
(n) De-noising with k=0.5
(o) De-noising with k=0.5
(p) De-noising with k=0.5

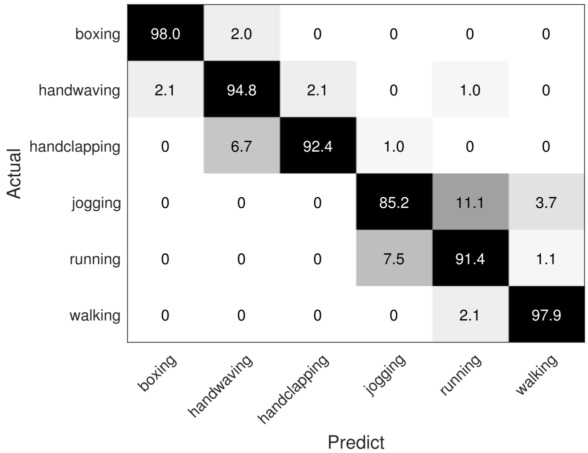
FIGURE 6. An example of pre-processing considering 0.03s time slice and three different user defined values; (b) $k = 0.125$, (c) $k = 0.25$, and (d) $k = 0.5$ applied on four actions: *Column 1* walking; *Column 2* waving; *Column 3* throwing; and *Column 4* get-up.

the dataset names. More details about these datasets are as follows:

1) *Emulator-Based:* In this group of datasets, the neuromorphic data for corresponding RGB sequences is generated by using an emulator. There are several emulators, such as, PIX2NVS [46], pyDVS [45] and ESIM [52], that are designed to simulate the native DVS cameras. In our experiments, PIX2NVS emulator was used to generate the events from the video sequences since the work in the literature is based on PIX2VNS. We used four datasets, KTH [53], UCF11 [54], UCF50 [55] and HMDB51 [56] and converted them into the neuromorphic datasets, E-KTH, E-UCF11, E-UCF50 and E-HMDB51, respectively. E-KTH dataset contains 597 sequences showing 6 action classes performed by 25 different subjects and 4 different camera views. E-UCF11 dataset contains 11 action classes, while E-UCF50 contains 50 different classes in 6681 sequences. E-HMDB51 dataset, which is one of the largest datasets used in HAR, contains 6766 clips distributed in 51 action classes. The action categories of this dataset can be grouped into five types based on the body movements. The RGB version of this dataset is considered challenging due to containing clips collected from the Internet and YouTube.
2) *Re-Recording-Based:* In this group, we used R-UCF11 and R-UCF50 datasets, which have been acquired by playing the original RGB versions of UCF11 and UCF50, respectively on the monitor and positioning the DAVIS240C vision sensor camera in the opposite

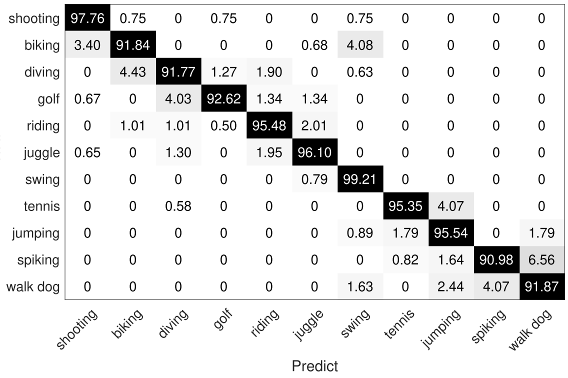
TABLE 1. Detailed statistical information of the datasets used in the experiments.

Dataset	No. of sequences	No of actions	No. of clips per action	Dataset source
E-KTH	597	6	10	PIX2NVS emulator [46]
E-UCF11	1576	11	100	
E-UCF50	6681	50	100	
E-HMDB51	6766	51	101	
R-UCF11	1576	11	100	DAVIS240C [44]
R-UCF50	6681	50	100	DAVIS240C [44]
N-Actions	450	10	30	DAVIS346red -Color [57]



Actual \ Predict	boxing	handwaving	handclapping	jogging	running	walking
boxing	98.0	2.0	0	0	0	0
handwaving	2.1	94.8	2.1	0	1.0	0
handclapping	0	6.7	92.4	1.0	0	0
jogging	0	0	0	85.2	11.1	3.7
running	0	0	0	7.5	91.4	1.1
walking	0	0	0	0	2.1	97.9

FIGURE 7. Confusion matrix of the proposed HAR on E-KTH dataset using QSVM (Overall accuracy: 93.14%).



Actual \ Predict	shooting	biking	diving	golf	riding	juggle	swing	tennis	jumping	spiking	walk dog
shooting	97.76	0.75	0	0.75	0	0	0.75	0	0	0	0
biking	3.40	91.84	0	0	0	0.68	4.08	0	0	0	0
diving	0	4.43	91.77	1.27	1.90	0	0.63	0	0	0	0
golf	0.67	0	4.03	92.62	1.34	1.34	0	0	0	0	0
riding	0	1.01	1.01	0.50	95.48	2.01	0	0	0	0	0
juggle	0.65	0	1.30	0	1.95	96.10	0	0	0	0	0
swing	0	0	0	0	0	0.79	99.21	0	0	0	0
tennis	0	0	0.58	0	0	0	0	95.35	4.07	0	0
jumping	0	0	0	0	0	0	0.89	1.79	95.54	0	1.79
spiking	0	0	0	0	0	0	0	0.82	1.64	90.98	6.56
walk dog	0	0	0	0	0	0	1.63	0	2.44	4.07	91.87

FIGURE 8. Confusion matrix of the proposed HAR on E-UCF11 dataset using QSVM (Overall accuracy: 94.43%).

of the monitor to record the events [44]. R-UCF11 and R-UCF50 datasets contain the same number of video clips and the classes as the original RGB versions converted to the NVS domain with 240 × 180 spatial resolution. More details about these datasets can be found in [54], [55].

3) *Native NVS-Based:* In these datasets, NVS devices are used to acquire real NVS data. We used the dataset in [57] which was acquired by recording 10 real human actions in an office environment using DAVIS346redColor camera. Herein, we refer to this dataset as N-Actions dataset. N-Actions dataset contains 10 action classes captured in 450 NVS sequences with 346 × 260 spatial resolution.

The statistical details of all these datasets used in the experiments reported in are summarized in TABLE 1.

The native NVS-based datasets show a high presence of noisy events compared to the other groups of datasets. As an example, FIGURE 5 compares a time slice of an action sequence from E-UCF50 dataset and the corresponding slice from R-USCF50 dataset. They show presence of various amounts of noisy events. Sometimes, the number of noisy events is much higher than the number of events related to the action. Although NVS data can be intrinsically noisy depending on the threshold used for an event determination, we have noticed that the noise can be as high as 70% of the overall captured events for data streams in N-Actions dataset. In such cases, application of our proposed pre-processing presented in Section III-B is helpful in removing such noise.

FIGURE 6 shows an example for de-noising a sampled slice from walking action acquired by a native neuromorphic camera from the dataset N-Actions. Using different values for the parameter k in the pre-processing noisy event removal algorithm. On one hand, it can be observed that the highest value for k, *e.g.*, $k = 0.5$, as shown in the fourth row in FIGURE 6, removes the majority of the noisy events as well as a large portion of events corresponding to the action. Removing this amount of events can lead to the loss of important features of the action representation. On the other hand, choosing a small value for k, *e.g.*, $k = 0.125$, as shown in second row in FIGURE 6, can result in retaining many noisy events leading to generating inaccurate features. Thus, the value of k aims to remove the majority of the noisy events while retaining the events corresponding to the action. This can be observed in the third row in FIGURE 6, when the best filtering result is presented with $k = 0.25$. With $k = 0.25$, we demonstrate that the filtering algorithm keeps the most important events that represent the action. In this case, the pre-processing step has resulted in removing approximately 70% of events in each slice and retaining the relevant events of the action. This representation is important in our method because it aims to model the dynamics of the action instead of measuring the speed of the action. In the experiments, we have used the noisy event removal pre-processing with $k = 0.25$.

Since the last two groups of datasets were captured using real NVS devices, there is no notion of temporal frame rate for these datasets. Therefore, a time window for extracting local features needs to be determined. In order to correspond with the first group of sequences, which were emulated using the conventional RGB video with 30 frames per second frame rate, for these experiments, we have defined the size of the window, w, to be $\mathcal{T}_w = \frac{1}{30} = 0.033$ seconds.

In this section, we report the human action recognition performance of the proposed algorithm using the two

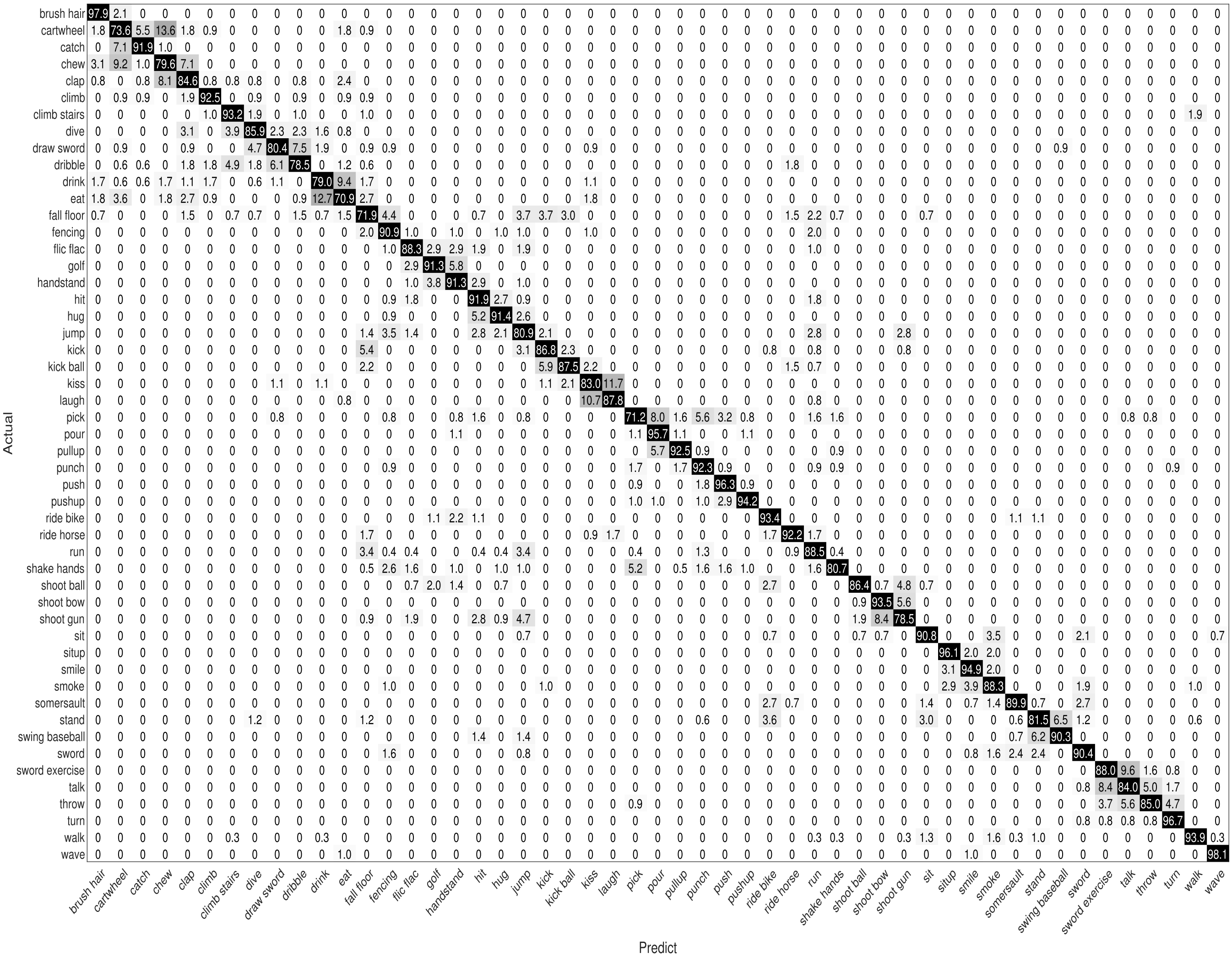


FIGURE 9. Confusion matrix of the proposed HAR on E-HMDB51 dataset using QSVM (Overall accuracy: 87.61%).

TABLE 2. HAR accuracy rates(%) of the proposed method.

Method	E-KTH	E-UCF11	E-HMDB51	E-UCF50	R-UCF11	R-UCF50	N-Actions
local features ($\mathcal{F}_L$) only + KNN	51.17	82.15	73.60	63.66	80.83	52.07	51.21
local features ($\mathcal{F}_L$) only + QSVM	61.04	76.21	65.41	49.6	79.13	42.49	58.48
global($\mathcal{F}_G$) features only + KNN	91.47	89.36	73.32	36.62	43.34	44.64	44.29
global features ($\mathcal{F}_G$) only + QSVM	92.47	92.99	82.82	40.46	46.14	43.6	43.25
local and global features ($\mathbb{F}$)+ KNN	80.27	93.36	86.38	**69.45**	81.68	**68.96**	53.29
local and global features ($\mathbb{F}$)+ QSVM	**93.14**	**94.43**	**87.61**	65.07	**82.61**	65.32	**61.94**

classifiers, KNN and QSVM. KNN classifier is set up with K = 1 neighbour and Mahalanobis distance measuring. The reported results were generated using the 5-fold cross validation.

B. PERFORMANCE OF THE PROPOSED HUMAN ACTION RECOGNITION METHODOLOGY

TABLE 2 shows the overall recognition accuracy percentages for the seven datasets, E-KTH, E-UCF11, E-HMDB51, E-UCF50, R-UCF50, R-UCF11 and N-Actions for the proposed algorithm with various feature variants. It shows the performance of the handcrafted local and global features separately and together using two different classifiers: KNN and QSVM. TABLE 3 compares the performance of our proposed method with that of the existing algorithms in the literature. The corresponding confusion matrices are shown in FIGURE 7, FIGURE 8, FIGURE 9, FIGURE 10, FIGURE 11, FIGURE 12 and FIGURE 13, respectively.

The overall recognition accuracy percentages using the proposed features were evaluated individually (as local only

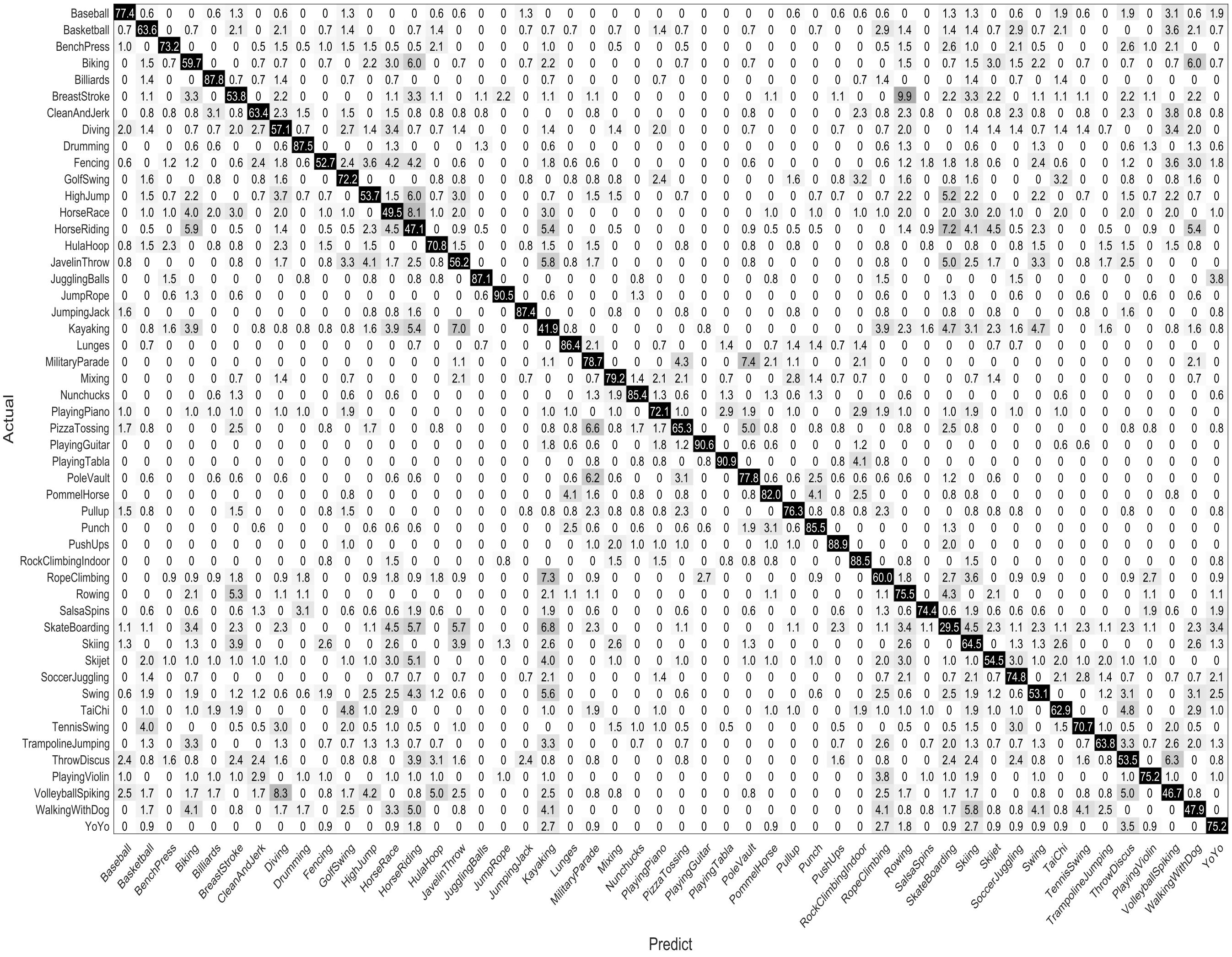


FIGURE 10. Confusion matrix of the proposed HAR on E-UCF50 dataset using KNN (Overall accuracy: 69.81%).

TABLE 3. HAR accuracy rates(%) of the proposed method comparison with the existing work.

Method		E-KTH	E-UCF11	E-HMDB51	E-UCF50	R-UCF11	R-UCF50	N-Actions
Existing work	handcrafted features	–	75.13 [39]	–	–	–	–	–
	CNN	92.6 [41]	–	51.5 [42]	–	**92.90** [43]	–	–
	RGB + CNN	–	–	62.0 [40]	–	–	–	–
Proposed method	*local and global features* ($\mathbb{F}$)+ *KNN*	80.27	93.36	86.38	**69.45**	81.68	**68.96**	53.29
	local and global features ($\mathbb{F}$)+ *QSVM*	**93.14**	**94.43**	**87.61**	65.07	**82.61**	65.32	**61.94**

and global only) and as a combined feature vector. It is evident from the results in TABLE 2 that using the concatenated local and global feature vector has resulted in the best averages for each of the datasets. It has achieved the best average recognition accuracy rates of 93.14%, 94.43%, 87.61%, 69.45%, 68.96%, 82.61% and 61.94% for E-KTH, E-UCF11, E-HMDB51, E-UCF50, R-UCF50, R-UCF11 and N-Actions respectively. For all datasets apart from E-UCF50 and R-UCF50 datasets, QSVM classifier has performed better than the KNN classifier. The proposed method have outperformed the accuracy rates achieved by the state of the art on exploring the neuromorphic data streams for HAR by 0.54%, 19.3% and 25.61% for E-KTH, E-UCF11 and E-HMDB51, respectively, as observed in TABLE 3. There is no reported previous work for HAR using the E-UCF50, R-UCF50, and N-Actions datasets to our best knowledge. Furthermore, we summarize the comparison between our proposed method and deep learning based work on neuromorphic sensing data in the domain of HAR in TABLE 3. Note that the reference [43] is not peer reviewed at the time of preparation of the present article. For the datasets of E-HMDB51 and E-KTH, our proposed handcrafted feature based method has outperformed the deep learning based methods.

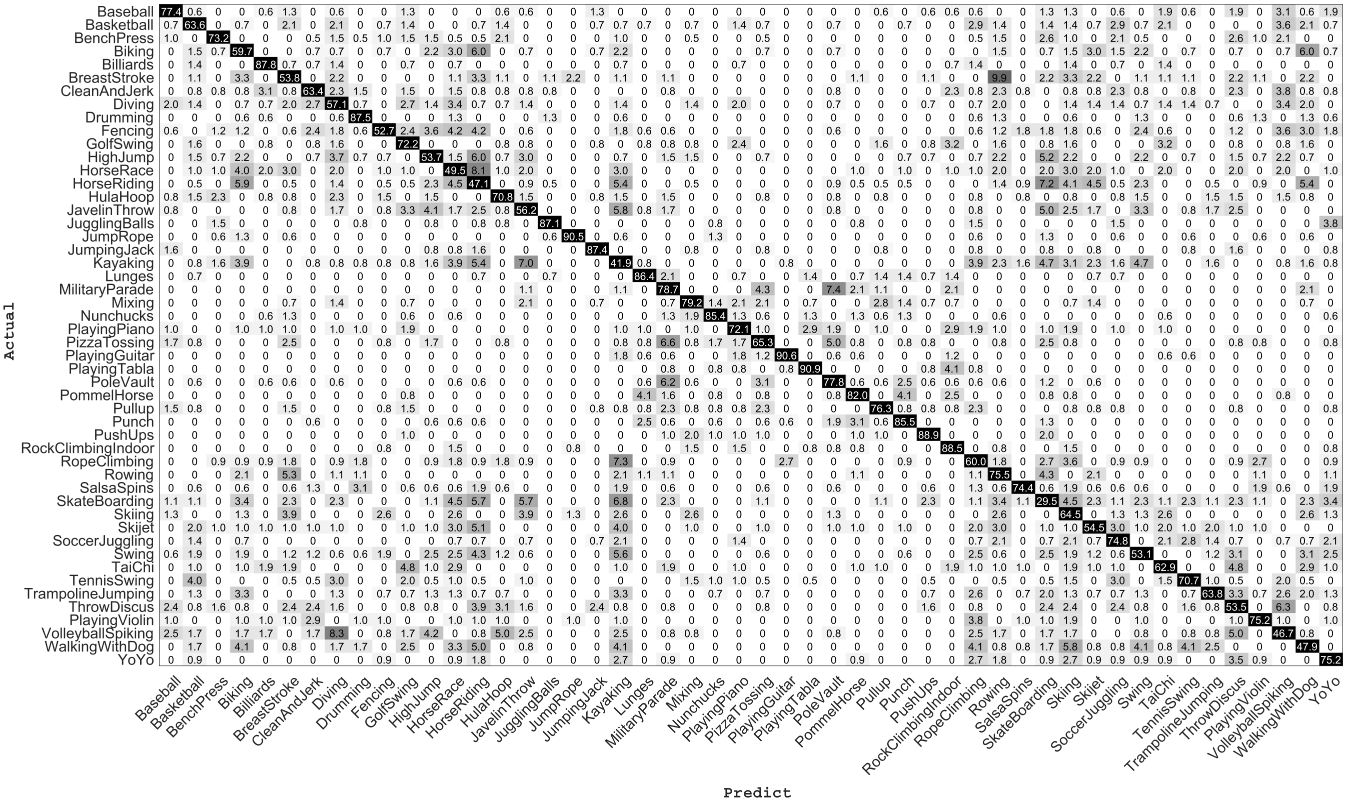


FIGURE 11. Confusion matrix of the proposed HAR on R-UCF50 dataset using KNN (Overall accuracy: 68.96%).

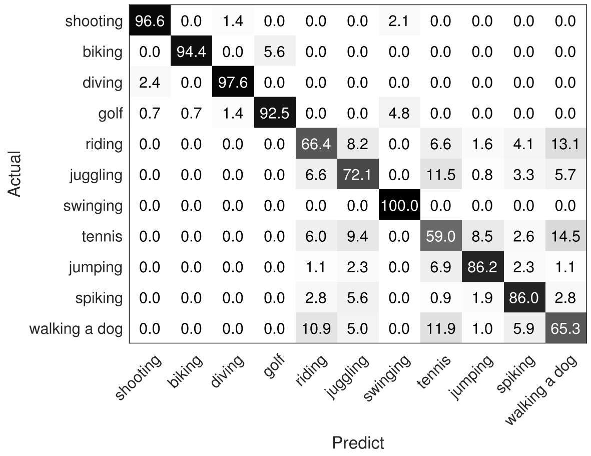


FIGURE 12. Confusion matrix of the proposed HAR on R-UCF11 dataset using QSVM (Overall accuracy: 82.61%).

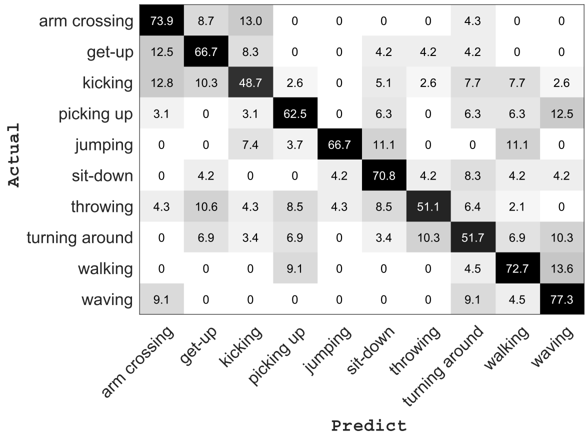
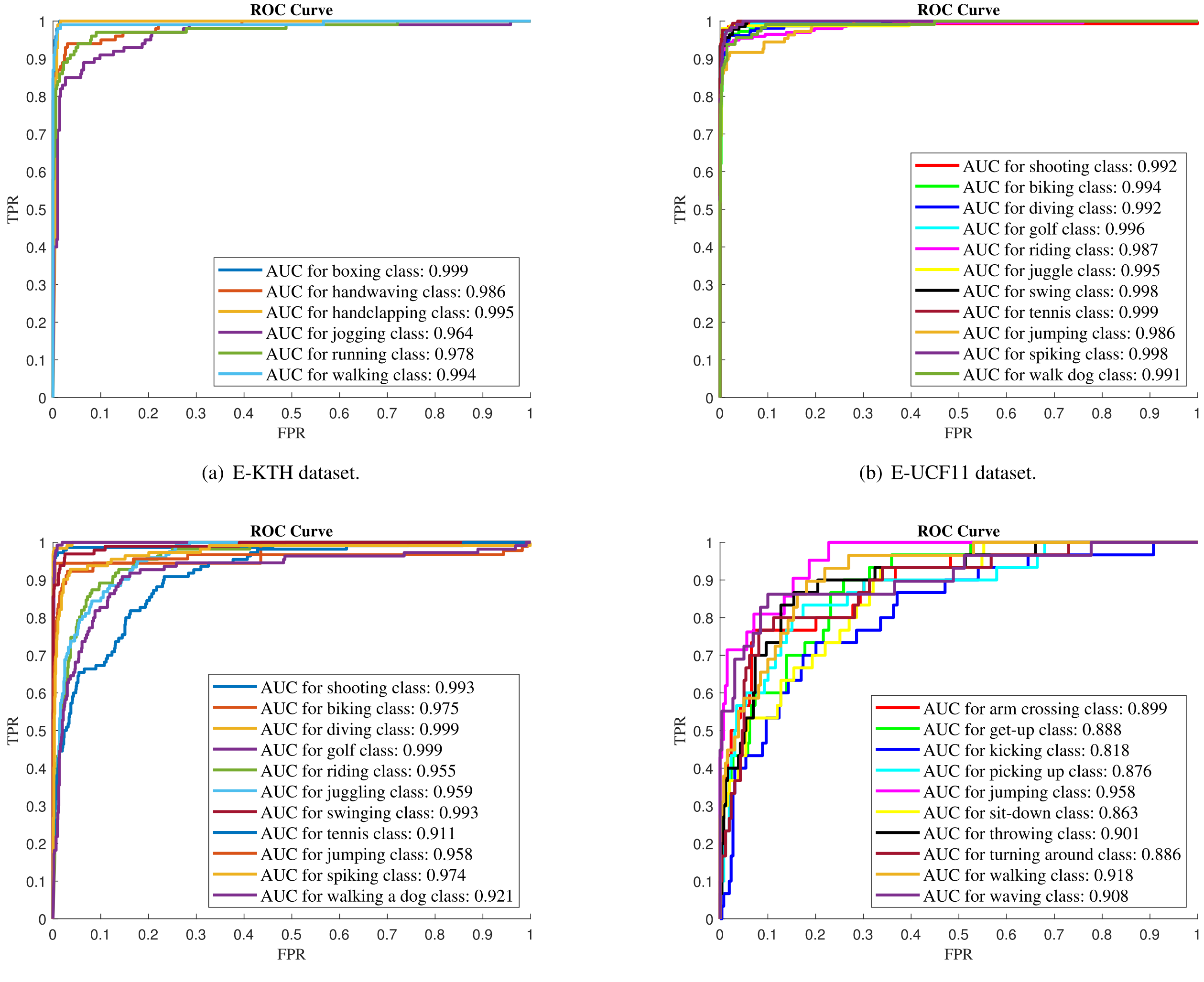


FIGURE 13. Confusion matrix of the proposed HAR on N-Actions dataset using QSVM (Overall accuracy: 61.94%).

In some cases, the confusion matrix shows that it is hard for the classifier to discriminate between the actions accurately using the NVS data. For example, some very high similarity actions like, jogging, running and walking in EKTH have resulted in some confusion among them in the classification, as can be seen in FIGURE 7. This is because our proposed method focuses on modelling the dynamic patterns of the actions instead of their speed. Since these actions have similar patterns during the time, similar description features due to the similar temporal changes in the action are obtained. These actions have similar global features due to the similar temporal changes in the action. This is evident from the global features depicted in FIGURE 4, as those features are very close to each other for these three classes.

This issue of high rate of similarities can also be observed in the confusion matrix of E-UCF11 dataset in FIGURE 7. For example, the spiking action archived a high rate of similarity with both jumping and walking actions, since there are similarities in the dynamics of these actions in some



(a) E-KTH dataset.

(b) E-UCF11 dataset.

(c) R-UCF11 dataset.

(d) N-Actions dataset.

FIGURE 14. **ROC curves and their corresponding AUC values for E-KTH, E-UCF11, R-UCF11, N-Actions datasets.**

situations. The biking also provides another example of a high rate of similarity with swing activity since they have similar instants of motion. Other cases can be found clearly in FIGURE 9, where the eat action shows a rate of 12% of similarity with drink action since these two actions include a similarity in performing these actions.

N-Action dataset shows more examples of similarities among the actions since this dataset contains several actions that have the same dynamic in achieving these actions, such as kicking, which has a similarity with most of the actions in this dataset. The lower accuracy rates in these cases are partly due to the non-uniform temporal sampling in neuromorphic sensing failing to capture the speed differences in some similar actions. The presence of a large number of noisy events also plays a part in this issue.

We notice in TABLE 2, the HAR accuracy rates of N-Actions, R-UCF50 and R-UCF11 datasets are much lower compared to those of most of the emulator-based datasets. This is likely to be due to the generation of a high proportion of unnecessary events (so called noisy events) in the data streams captured by the actual NVS devices, compared with the emulator based datasets, where most events are concentrated around the pixels correspond to the human actions in the sequences, as shown in FIGURE 5.

For the further justification of the classification results, the Receiver Operating Curve (ROC) plots for each class in the smallest datasets (E-KTH, E-UCF11, R-UCF11, and N-Actions) are shown in FIGURE 14. The corresponding Area Under Curve (AUC) values are also shown in FIGURE 14. The minimum and the maximum AUC values for the classes in all datasets are shown in TABLE 4. The majority of the classes show large AUC values. However, the classes that are highly similar with the other classes show lower AUC values. As an example, the jogging action in E-KTH dataset, which has the lowest accuracy rate, shows the lowest AUC with 0.964 compared to the other actions in the dataset. This lowest AUC value correlates with the lowest accuracy rate of the same action appeared in the confusion matrix in FIGURE 14. The tennis action in R-UCF11 dataset also shows the lowest AUC value for that dataset. This

TABLE 4. AUC for the classification of the seven datasets used in the proposed method.

[Dataset	min AUC	max AUC
E-KTH	0.964	0.999
E-UCF11	0.986	0.999
E-HMDB51	0.958	0.999
E-UCF50	0.612	0.993
R-UCF11	0.911	0.999
R-UCF50	0.597	0.964
N-Actions	0.818	0.958

corresponds to showing a high rate of similarity with other actions in this dataset as evident from the accuracy rate of this action shown in FIGURE 12.

C. COMPUTATIONAL COMPLEXITY OF THE PROPOSED METHOD

All experiments in this paper were implemented using Matlab R2018a on a PC with Intel processor, CPU@3.6GHz and RAM 16GB. The proposed feature extraction algorithm consists of three main steps: De-noising, local feature extraction and global feature extraction. The computational complexity for each of these steps is $\mathcal{O}(N)$, where N is the number of the events in an action. Thus, leading to the total computational complexity for including all three steps in the order of $\mathcal{O}(3N)$.

V. CONCLUSIONS

In this work, we have presented a new methodology for learning the data streams from emerging neurmorphic vision sensing devices. Our proposed method consists of a pre-processing step followed by the generation of a feature vector to capture local and global features correspond to the underlying human action. The local features were extracted considering a set of high-order descriptive statistics from the spatio-temporal events in a time window slice, while the global features were extracted by considering the frequencies of occurrences of the temporal event sequences. Then a classifier was trained using these feature vectors. The proposed method was evaluated using three groups of datasets: Emulator-based, re-recording-based and native NVS-based. The proposed method has outperformed the HAR accuracy rates of the existing methods by 0.54%, 19.42% and 25.61% for E-KTH, E-UCF11 and E-HMDB50 datasets, respectively. This paper also reported the results for three further datasets, which were used for the first time in the literature for human action recognition on neuromorphic vision sensor domain. It was also noted that the re-recording-based and native NVS-based datasets were providing lower rates of HAR accuracy compared to those for emulator-based datasets, due to the presence of a high number of noisy events in the sequences directly captured by the NVS devices.

ACKNOWLEDGMENT

Salah Al-Obaidi and Hiba Al-Khafaji like to thank the University of Babylon and the Ministry of Higher Education and Scientific Research (MOHESR) in Iraq for their Ph.D. studentships.

REFERENCES

[1] T. Delbruckl, "Neuromorophic vision sensing and processing," in *Proc. Conf. 42nd Eur. Solid-State Circuits Conf. (ESSCIRC)*, Sep. 2016, pp. 7–14.

[2] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opinion Neurobiol.*, vol. 20, no. 3, pp. 288–295, Jun. 2010.

[3] T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2426–2429.

[4] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. V. Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, May 2011.

[5] K. A. Boahen, "A burst-mode word-serial address-event link—I: Transmitter design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.

[6] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. Hoboken, NJ, USA: Wiley, 2014.

[7] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Jan. 2008.

[8] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 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] T. V. Nguyen, Z. Song, and S. Yan, "STAP: Spatial-temporal attention-aware pooling for action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 1, pp. 77–86, Jan. 2015.

[10] Z. Zhang, S. Liu, S. Liu, L. Han, Y. Shao, and W. Zhou, "Human action recognition using salient region detection in complex scenes," in *Proc. 3rd Int. Conf. Commun., Signal Process., Syst.*, 2015, pp. 565–572.

[11] H.-B. Zhang, Q. Lei, B.-N. Zhong, J.-X. Du, J. Peng, T.-C. Hsiao, and D.-S. Chen, "Multi-surface analysis for human action recognition in video," *SpringerPlus*, vol. 5, no. 1, p. 1226, Dec. 2016.

[12] A.-A. Liu, Y.-T. Su, W.-Z. Nie, and M. Kankanhalli, "Hierarchical clustering multi-task learning for joint human action grouping and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 102–114, Jan. 2017.

[13] Z. Gao, Y. Zhang, H. Zhang, Y. B. Xue, and G. P. Xu, "Multi-dimensional human action recognition model based on image set and group sparisty," *Neurocomputing*, vol. 215, pp. 138–149, Nov. 2016.

[14] F. Angelini, Z. Fu, S. A. Velastin, J. A. Chambers, and S. M. Naqvi, "3D-HOG embedding frameworks for single and multi-viewpoints action recognition based on human silhouettes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4219–4223.

[15] S. Al-Obaidi and C. Abhayaratne, "Temporal salience based human action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2017–2021.

[16] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[18] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 816–833.

[19] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4305–4314.

[20] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 7444–7452.

[21] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector CNNs," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2326–2339, May 2018.

[22] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[23] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based vision: A survey," 2019, *arXiv:1904.08405*. [Online]. Available: http://arxiv.org/abs/1904.08405

[24] F. Cardinaux, D. Bhowmik, C. Abhayaratne, and M. S. Hawley, "Video based technology for ambient assisted living: A review of the literature," *J. Ambient Intell. Smart Environ.*, vol. 3, no. 3, pp. 253–269, 2011.

[25] S. Al-Obaidi and C. Abhayaratne, "Privacy protected recognition of activities of daily living in video," in *Proc. 3rd IET Int. Conf. Technol. Act. Assist. Living (TechAAL)*, 2019, pp. 1–6.

[26] S. Al-Obaidi, H. Al-Khafaji, and C. Abhayaratne, "Modeling temporal visual salience for human action recognition enabled visual anonymity preservation," *IEEE Access*, vol. 8, pp. 213806–213824, 2020.

[27] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 142–149, Feb. 2017.

[28] K. D. Fischl, G. Tognetti, D. R. Mendat, G. Orchard, J. Rattray, C. Sapsanis, L. F. Campbell, L. Elphage, T. E. Niebur, A. Pasciaroni, V. E. Rennoll, H. Romney, S. Walker, P. O. Pouliquen, and A. G. Andreou, "Neuromorphic self-driving robot with retinomorphic vision and spike-based processing/closed-loop control," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2017, pp. 1–6.

[29] D. P. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbruck, "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *Proc. 2nd Int. Conf. Event Control, Commun., Signal Process. (EBCCSP)*, Jun. 2016, pp. 1–8.

[30] J. H. Lee, P. K. J. Park, C.-W. Shin, H. Ryu, B. C. Kang, and T. Delbruck, "Touchless hand gesture UI with instantaneous responses," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 1957–1960.

[31] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. J. Park, C.-W. Shin, H. Ryu, and B. C. Kang, "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2250–2263, Dec. 2014.

[32] 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.*, Jul. 2017, pp. 7243–7252.

[33] 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.

[34] A. M. George, D. Banerjee, S. Dey, A. Mukherjee, and P. Balamurali, "A reservoir-based convolutional spiking neural network for gesture recognition from DVS input," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–9.

[35] G. Chen, Z. Xu, Z. Li, H. Tang, S. Qu, K. Ren, and A. Knoll, "A novel illumination-robust hand gesture recognition system with event-based neuromorphic vision sensor," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 2, pp. 508–520, Apr. 2021.

[36] G. Chen, J. Chen, M. Lienen, J. Conradt, F. Röhrbein, and A. C. Knoll, "FLGR: Fixed length gists representation learning for RNN-HMM hybrid-based neuromorphic continuous gesture recognition," *Frontiers Neurosci.*, vol. 13, Feb. 2019, Art. no. 73.

[37] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1412–1421.

[38] A. N. Belbachir, S. Schraml, and A. Nowakowska, "Event-driven stereo vision for fall detection," in *Proc. CVPR WORKSHOPS*, Jun. 2011, pp. 78–83.

[39] S. A. Baby, B. Vinod, C. Chinni, and K. Mitra, "Dynamic vision sensors for human activity recognition," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 316–321.

[40] A. Chadha, Y. Bi, A. Abbas, and Y. Andreopoulos, "Neuromorphic vision sensing for CNN-based action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 7968–7972.

[41] K. Sullivan and W. Lawson, "Representing motion information from event-based cameras," in *Proc. 26th IEEE Int. Symp. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2017, pp. 1465–1470.

[42] 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.

[43] C. Huang, "Event-based action recognition using timestamp image encoding network," 2020, *arXiv:2009.13049*. [Online]. Available: http://arxiv.org/abs/2009.13049

[44] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers Neurosci.*, vol. 10, p. 405, Aug. 2016.

[45] G. P. Garcia, P. Camilleri, Q. Liu, and S. Furber, "PyDVS: An extensible, real-time dynamic vision sensor emulator using off-the-shelf hardware," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2016, pp. 1–7.

[46] Y. Bi and Y. Andreopoulos, "PIX2NVS: Parameterized conversion of pixel-domain video frames to neuromorphic vision streams," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1990–1994.

[47] 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.

[48] W. Shi and R. Rajkumar, "Point-GNN: Graph neural network for 3D object detection in a point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1711–1719.

[49] Q. He, Z. Wang, H. Zeng, Y. Zeng, S. Liu, and B. Zeng, "SVGA-net: Sparse voxel-graph attention network for 3D object detection from point clouds," 2020, *arXiv:2006.04043*. [Online]. Available: http://arxiv.org/abs/2006.04043

[50] G. Chen, H. Cao, C. Ye, Z. Zhang, X. Liu, X. Mo, Z. Qu, J. Conradt, F. Röhrbein, and A. Knoll, "Multi-cue event information fusion for pedestrian detection with neuromorphic vision sensors," *Frontiers Neurorobotics*, vol. 13, p. 10, Apr. 2019.

[51] Y. Wang, X. Zhang, Y. Shen, B. Du, G. Zhao, L. C. Cui Lizhen, and H. Wen, "Event-stream representation for human gaits identification using deep neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 27, 2021.

[52] H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: An open event camera simulator," in *Proc. Conf. Robot Learn.*, 2018, pp. 969–982.

[53] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3, 2004, pp. 32–36.

[54] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2009, pp. 1996–2003.

[55] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of Web videos," *Mach. Vis. Appl.*, vol. 24, no. 5, pp. 971–981, Jul. 2013.

[56] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2556–2563.

[57] S. Miao, G. Chen, X. Ning, Y. Zi, K. Ren, Z. Bing, and A. Knoll, "Neuromorphic vision datasets for pedestrian detection, action recognition, and fall detection," *Frontiers Neurorobotics*, vol. 13, p. 38, Jun. 2019.

SALAH AL-OBAIDI received the B.E. and M.Sc. degrees in computer science from the University of Babylon, Iraq, in 2000 and 2004, respectively. He is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. Prior to that, he was a Lecturer with the Department of Computer Science, University of Babylon. His research interests include image and video processing, signal processing, video saliency, surveillance, human action recognition, assisted living, and computer vision. He was a recipient of a Ph.D. Scholarship from the University of Babylon and the Ministry of Higher Education and Scientific Research (MOHESR), Iraq.

HIBA AL-KHAFAJI received the B.E. and M.Sc. degrees in computer science from the College of Science, University of Babylon, Iraq, in 1999 and 2003, respectively. She is currently pursuing the Ph.D. degree with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. Her research interests include signal and image processing, security, and computer vision. She was a recipient of the Ph.D. Scholarship from the University of Babylon and the Ministry of Higher Education and Scientific Research (MOHESR), Iraq.

CHARITH ABHAYARATNE (Member, IEEE) received the B.E. degree in electrical and electronic engineering from The University of Adelaide, Australia, in 1998, and the Ph.D. degree in electronic and electrical engineering from the University of Bath, U.K., in 2002. He is currently a Lecturer with the Department of Electronic and Electrical Engineering, The University of Sheffield, U.K. His research interests include multidimensional signal processing, image and video compression, visual content understanding, multimedia content security, and forensics. He is currently serving as a member of the U.K. Engineering and Physical Science Research Council (EPSRC) peer review college, the British Standards Institution (BSI) group IST/037 committee for coding of picture, audio, multimedia and hypermedia information, the Multimedia Security and Forensics Technical Committee of Asia-Pacific Signal and Information Processing Association (APSIPA), the Audio/Video Systems and Signal Processing Technical Committee of the IEEE Consumer Technologies Society, and the IEEE Multimedia Communications Technical Committee. He was a recipient of the European Research Consortium for Informatics and Mathematics (ERCIM) Postdoctoral Fellowship (2002–2004) to carry out research at the Centre of Mathematics and Computer Science (CWI), The Netherlands, and the National Research Institute for Computer Science and Control (INRIA), Sophia Antipolis, France. He currently serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE ACCESS, and *Journal of Information Security and Applications (JISA)* (Elsevier).