Event-Stream Representation for Human Gaits Identification Using Deep Neural Networks

Yanxiang Wang, Xian Zhang, Yiran Shen*, *Senior Member, IEEE,* Bowen Du, Guangrong Zhao, Lizhen Cui, *Member, IEEE* and Hongkai Wen, *Member, IEEE*

Abstract—Dynamic vision sensors (event cameras) have recently been introduced to solve a number of different vision tasks such as object recognition, activities recognition, tracking, etc. Compared with the traditional RGB sensors, the event cameras have many unique advantages such as ultra low resources consumption, high temporal resolution and much larger dynamic range. However, these cameras only produce noisy and asynchronous events of intensity changes, i.e., event-streams rather than frames, where conventional computer vision algorithms can't be directly applied. In our opinion the key challenge for improving the performance of event cameras in vision tasks is finding the appropriate representations of the event-streams so that cutting-edge learning approaches can be applied to fully uncover the spatio-temporal information contained in the event-streams. In this paper, we focus on the event-based human gait identification task and investigate the possible representations of the event-streams when deep neural networks are applied as the classifier. We propose new event-based gait recognition approaches basing on two different representations of the event-stream, i.e., graph and image-like representations, and use Graph-based Convolutional Network (GCN) and Convolutional Neural Networks (CNN) respectively to recognize gait from the event-streams. The two approaches are termed as EV-Gait-3DGraph and EV-Gait-IMG. To evaluate the performance of the proposed approaches, we collect two event-based gait datasets, one from real-world experiments and the other by converting the publicly available RGB gait recognition benchmark CASIA-B. Extensive experiments show that EV-Gait-3DGraph achieves significantly higher recognition accuracy than other competing methods when sufficient training samples are available. However, EV-Gait-IMG converges more quickly than graph-based approaches while training and shows good accuracy with only few number of training samples (less than 10). So image-like presentation is preferable when the amount of training data is limited.

Index Terms—Gait Recognition, Dynamic Vision Sensors, Graph-based Convolutional Networks.

1 Introduction

Inspired by the principles of biological vision, Dynamic Vision Sensors (DVS) [1], [2], [3] are a new sensing modality for a number of tasks such as visual odometry/SLAM [4], [5], [6], robotic perception [7], [8], [9], [10] and object recognition [11], [12]. Unlike the RGB cameras which produce synchronized frames at fixed rates, the pixels of DVS sensors are able to capture microseconds level intensity change independently, and generate a stream of asynchronous "events". The design of DVS sensors provides many benefits over the conventional RGB cameras. Firstly, DVS sensors require fewer resources including energy, bandwidth and computation as the events are sparse and only triggered when intensity changes are detected. For example, the DVS128 sensor platform consumes 150 times less energy than a CMOS camera [1]. Secondly, the temporal resolution of DVS sensors is tens of microseconds which means the DVS sensors are able to capture detailed motion phases or high speed movements without blur or rolling shutter problems. Finally, DVS sensors have significantly larger dynamic range (up to 140dB [1]) than RGB cameras (∼60dB), which allows them to work under more challenging lighting conditions. These characteristics make DVS sensors more appealing than RGB cameras for vision tasks with special requirements on latency, resources consumption and operation environments.

In this paper, we investigate the feasibility of using DVS to tackle the classic gait recognition problem. Specifically, it aims to determine human identities based on their walking patterns captured by the sensors. This is a fundamental building block for many real-world applications such as activity tracking, digital healthcare and security surveillance. In those contexts, DVS sensors have unique advantages over the standard RGB cameras because i) their low energy and bandwidth footprint makes them ideal for always-on wireless monitoring; and ii) the high dynamic range allows them to work under challenging lighting conditions without dedicated illumination control.

However as shown in Fig. 1 (a), DVS operates in a completely different way than the RGB cameras, which generates asynchronous and noisy events, termed as event-stream, rather than frames when capturing human motion. The conventional RGB-based algorithms are designed on top of feature extracted or learned from discrete 2D frames, therefore, existing image processing or deep neural networks can't be applied directly on the asynchronous event-streams. In this paper, we propose new event-based gait recognition approaches which are able to work with the noisy event-streams and accurately infer the identities based

*Corresponding author

- *Yanxiang Wang, Yiran Shen and Lizhen Cui are with School of Software and C-Fair, Shandong University, China. E-mail: fancyswift@outlook.com; yiran.shen@sdu.edu.cn;clz@sdu.edu.cn*
- *Xian Zhang and Guangrong Zhao are with College of Computer Science and Technology, Harbin Engineering University, China. E-mail:zhangxian@hrbeu.edu.cn, 379745905@hrbeu.edu.cn*
- *Bowen Du and Hongkai Wen are with Department of Computer Science, University of Warwick, UK. E-mail:{b.du, hongkai.wen}@dcs.warwick.ac.uk*

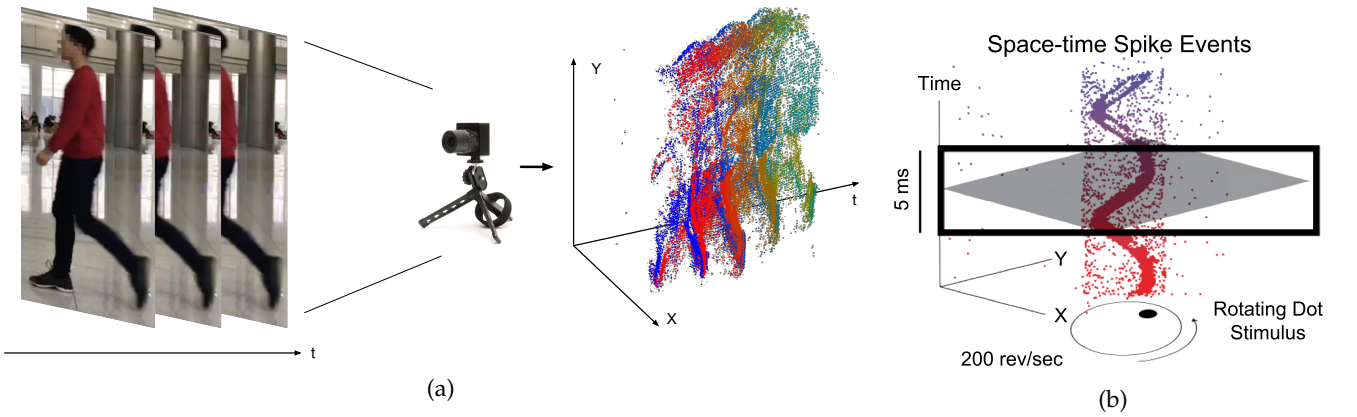


Fig. 1: (a) DVS sensor generates asynchronous event-stream when a subject is walking in front of it. The positive intensity changes (+1) are denoted in red and negative intensity changes (-1) are in blue. Red changes to yellow and blue changes to green gradually with time. (b) Noisy events stream caused by a rotating dot (adapted from [13]).

on gait. We represent the event-streams in either graphs or event images and design specific deep neural networks for recognition accordingly. Specifically, the technical contributions of this paper are as follows:

- We consider two types of representations for the asynchronous event-streams, i.e., 3D-Graph and event-image. To the best of our knowledge, this is the first piece of work using 3D-Graph to represent event-streams and we believe the 3D-Graph representation can better preserve spatio-temporal features of the event-streams inherently.
- Along with the new representations, two event-based gait recognition approaches, EV-Gait-3DGraph and EV-Gait-IMG, are designed by facilitating Graph-based Convolutional Networks (GCNs) for 3D-Graphs and Convolutional Neural Networks (CNNs) for image-like representation. They are able to recognize the identities from the asynchronous and sparse event data generated by human's gait effectively.
- We collect multiple event-based gait datasets. DVS128-Gait-Day and DVS128-Gait-Night were collected under practical settings and reasonable variance on the scale of subjects is allowed. EV-CASIA-B was transferred from RGB-based dataset. The evaluation on DVS128-Gait datasets shows that the proposed EV-Gait-3DGraph and EV-Gait-IMG can recognize identities up to 94.5% and 87.3% accuracy respectively when 100 training samples per subject are used and image-like representation shows good accuracy when only few number of training samples are available. The evaluation on EV-CASIA-B shows EV-Gait-IMG achieves comparable (even better in some viewing angles) performance with the state-of-the-art RGB-based approaches.

The rest of the paper is organized as follows. Section 2 reviews the related work of gait recognition using DVS sensors and applications of graph-based convolutional networks on vision tasks. Section 3 describes the workflow of the proposed approaches in detail. Section 4 evaluates the proposed approaches on both realworld and benchmark datasets. At last, Section 5 concludes the whole paper.

2 Related Work

Gait recognition has been intensively studied for decades in computer vision community [14], [15], [16], [17] and deep learning has been proven to provide state-of-the-art performance on gait recognition without tedious feature engineering [18], [19], [20], [21]. One classic approach for gait recognition proposed in [16] was based on extracting the silhouette using background subtraction and modeled the structural and transitional characteristics of gait. Han et al. [22] further improved the silhouette-based approach by extracting scale-invariant features from the gait template. Though template and feature based approaches were widely investigated [17], [23], [24], designing optimal features are still difficult tasks. Deep learning became popular in recent years to solve classification problems in an end-to-end way and requires no feature engineering. It has been introduced for solving gait recognition problem and produced state-of-the-art performance [18], [19], [20], [21]. CNNs are known to work well on extracting features from images. Wu et al. [18] proposed different CNN-based architectures for gait recognition and produced state-of-the-art recognition accuracy on CASIA-B dataset. One of our proposed event-based approaches also uses CNN, but our network is adapted to process the event data instead of the standard RGB frames.

We also review the related work of using DVS sensors for recognition or classification tasks. In [25], the authors applied CNN for identifying gestures, like hand-wave, circling and air-guitar actions. Lagorce at el. [12] proposed a new representation for event data called time-surface then a classification model was built to classify 36 characters(0-9, A-Z). Park et al. [26] employed a shallow neural network to extract the spatial pyramid kernel features for the hand motion recognition using DVS sensor. Graph-based representation of event-streams was first introduced by Bin et al. [27] to address the object recognition problem by constructing a 2D-Graph from a short-term event-stream and employing graph-based convolution for feature extraction. Then the work was extended to deal with action recognition task [28] where graph-based convolutions were applied on a sequence of 2D-Graphs to extract spatial features from discrete slices of event-streams and 3DCNN was used on

top of graph-based networks to extract temporal features. In addition, Gao at el. [29] used the DVS sensor to track the special markers equipped on the ankle joints of the subjects for gait analysis. However, unlike our approach it did not aim to recognize the identities and required attaching special markers to human bodies which was intrusive.

At last we have a brief discussion about GCNs and its applications on various vision tasks. CNNs have been proved to be powerful approach for addressing traditional vision-based applications in which the signals are represented in euclidean space. Similar convolution operations are then applied on signals represented as graphs or manifolds [30]. According to different convolution operators, GCNs can be vastly categorized as spectrum-based [31], [32] and spatial-based [33], [34] approaches. The spectral-based convolution on graphs was first proposed in [31]. The convolution operation exploits the normalized graph Laplacian matrix from spectral graph theory which had been proved to be a robust mathematical representation of undirected graphs [35]. Spectral-based approaches have been successfully applied to model the connectivity of the nodes and require a fixed number of nodes. However, in the case studied in this paper, the graphs constructed from different event-streams have various number of nodes. Not only the connectivity but also the locations of the nodes in spatio-temporal domain are important for a robust representation of the event-streams. Thus, the spatial-based approaches [36] which utilize the location information and relaxes constraints on the structure of the graphs are more appropriate on recognizing human gaits from event-streams. There are a number of different specific spatial-based convolution algorithms available [27], [34], [37]. In this paper, we choose Gaussian Mixture Model (GMM)-based convolution (proposed in Monet [34]) as the fundamental building block for our deep recognition network. There have been a number of successful applications of GCNs on vision tasks. For examples, graphs could be constructed from the RGB-D point clouds to preserve both the appearance and geometric relations [38]. Then appropriate GCNs were applied to extract features from graphs to improve the performance on semantic segmentation [39] and object detection [40]. Besides the point clouds, the GCNs have been also applied on 2D image processing tasks. Graphs could be constructed through consecutive frames to capture the spatio-temporal information in videos to infer the common foreground objects [41] or realize multi-object tracking [42], [43] with various sizes of objects. The skeleton data is another type of data suitable for GCNs since the graph is natural to represent the joints, bones and their connections. For instance, a directed graph neural network was proposed to predict the actions of human based on the constructed graph from human skeleton data [44] and achieved state-of-the-art performance.

3 Event-based Gait Recognition

In this paper, we propose new event-based gait recognition approaches, EV-Gait-3DGraph and EV-Gait-IMG, to identify gait from event-streams in the two types of representations. The deep neural networks for the two approaches are designed basing on GCNs and CNNs respectively.

3.1 EV-Gait-3DGraph

The workflow of EV-Gait-Graph3D and the key components of the proposed GCN are shown in Figure 2. It starts with collecting event-streams consisting of hundreds of thousands events. Considering the computational complexity, the OctreeGrid filtering algorithm is applied to significantly reduce the number of events while preserving most of the spatio-temporal structure of the event-streams. The connectivity between the remaining events after downsampling is calculated according to the predefined radius of neighborhood to construct 3D-Graph representation of the event-streams. Finally, the 3D-Graphs are taken as the inputs to train GCN for event-based gait recognition.

3.1.1 From Asynchronous Event-stream to 3D-Graph

Unlike the conventional CMOS/CCD cameras which produce synchronized frames at fixed rate, dynamic vision sensors (DVS) are a class of neuromorphic devices that can capture microsecond level pixel intensity changes as "events", asynchronously at the time they occur. Therefore they are often referred to as the "event cameras", whose output can be described as a stream of quadruplet, (t, x, y, p), where t is the timestamp of an event happens, (x, y) is the location of the event in the 2D pixel space, and p is the polarity. Without loss of generality, we often use $p = +1$ to denote the increase in pixel intensity and -1 as decrease. In practice, the DVS sensors only report such an event when the intensity change at a pixel exceeds certain threshold, i.e.,

$$|log\left(I^{x,y}_{now}\right) - log\left(I^{x,y}_{previous}\right)| > \theta \tag{1}$$

where $I^{x,y}_{now}$ and $I^{x,y}_{previous}$ are the current and previous intensity at the same pixel (x, y).

Fig. 1 shows an example of how the DVS sensors operate. When an object of interest is moving in the camera field of view, e.g. the rotating dot as in Fig. 1, rather than image frames, the DVS sensor generates an event-stream, i.e. the spiral-like shape in the spatio-temporal domain. The asynchronous and differential nature of the DVS sensors brings many unique benefits. For instance, they can have a very high dynamic range (140dB vs. 60dB of standard cameras), which allow them to work under more challenging lighting conditions. The event-streams produced by those sensors are at microseconds temporal resolution, which effectively captures details of the high speed motion. In addition, they are extremely power efficient, consuming approximately 150 times less energy than standard cameras, and have very low bandwidth requirement.

3.1.1.1 **Nonuniform OctreeGrid Filtering for Event-Stream Downsampling:** The upper-left part of Figure 2 shows an example of event-stream produced by human gait consisting of N asynchronously generated events. As N can be as large as tens of thousands to hundreds of thousands for just few seconds (3s-4s for our dataset), constructing graphs from raw event-stream directly is simply infeasible for computing and training consideration: the number of edges connecting neighboring events can even be order of magnitudes of the number of events. To reduce computational and training cost, we apply nonuniform OctreeGrid filtering algorithm [45], [46] to reduce the number

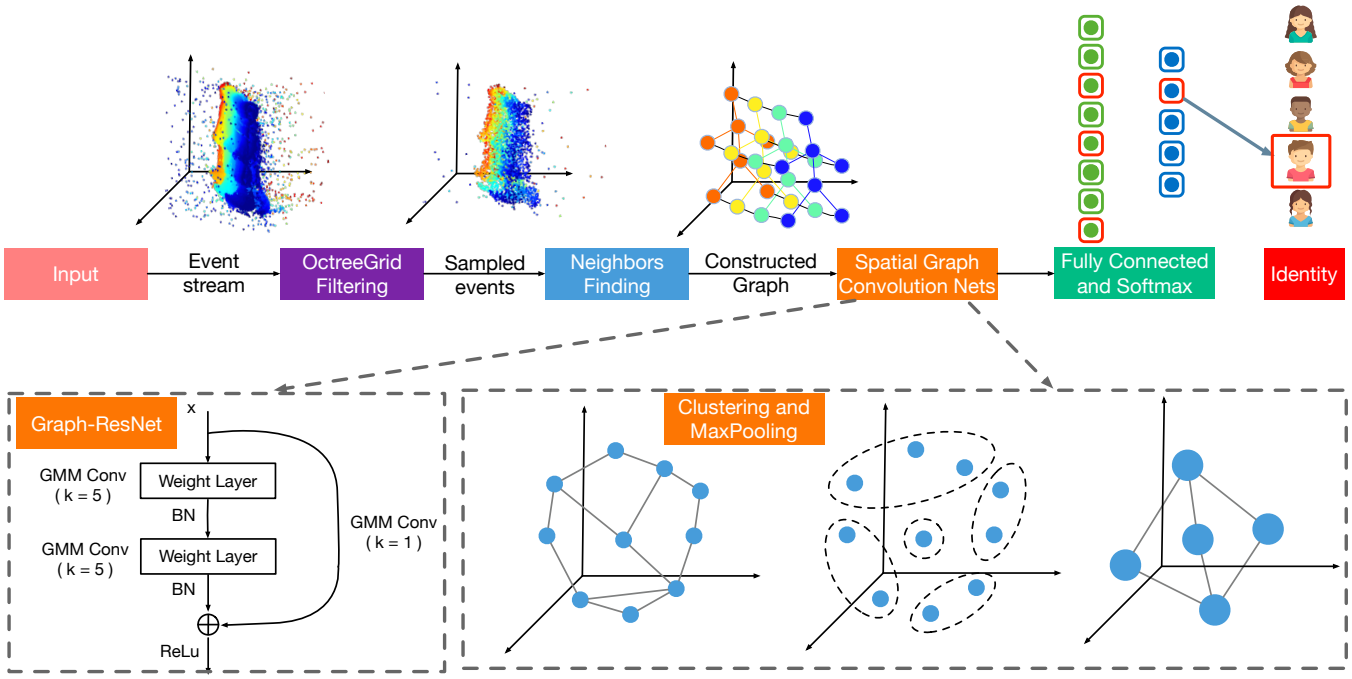


Fig. 2: Workflow of EV-Gait-Graph3D.

of events from N to M (where $M \ll N$) while the spatio-temporal structure of the original event-stream can still be well-preserved (see the example of downsampled event-stream on the upper-left of Figure 2). As its name suggests, the downsampling algorithm creates M nonuniform spatio-temporal grids according to the local density of the event-stream and randomly pick an event from the grid as a representative. *MaxNumEvents* is the parameter to determine the maximum number of points in each leaf node (or grid) when building the structure of octree, therefore, controls the downsampling rate.

3.1.1.2 **3D-Graph Construction:** After the event-stream is downsampled, the remaining events are regarded as vertices or nodes of a graph. A 3D-Graph is then constructed by connecting neighboring nodes with bi-directional edges. Two nodes $v_i = (x_i, y_i, t_i, p_i)$ and $v_j = (x_j, y_j, t_j, p_j)$ are neighbors if their predefined distance is less than the threshold of radius R:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha(t_i - t_j)^2} < R \qquad (2)$$

where α is a scaling factor to tune the difference between temporal and spatial resolution of the event-streams. A connected 3D-Graph is represented as G=(V, E, P) where V are the set of vertices and E are the set of the edges. The set of the polarity P are regarded as the input feature set for the graph-based convolution in the next step. After the connectivity of the 3D-Graph is determined, the adjacency matrix A of the graph can be generated whose element $A_{i,j}$ equals to 1 if nodes v_i and v_j are connected otherwise it equals to 0. In GCN, the elements on the diagonal of the adjacency matrix are also set to 1s to include the features of the center nodes when aggregating its neighbors.

3.1.2 GCN-based Deep Recognition Network

After the event-streams are downsampled and transformed to 3D-Graphs, we design a GCN-based deep recognition network for extracting features and recognizing human gaits. The key components of the network include Gaussian Mixture Model (GMM)-based graph convolution, Graph Residual Network, graph clustering and MaxPooling which are shown in lower part of Figure 2.

3.1.2.1 **GMM-based Graph Convolution:** Spatial-based convolution operation aggregates feature vectors among neighboring nodes by convolving with learned weights matrices to output a P-dimensional feature vector f'. The GMM-based convolution centered at node v_x can be expressed as weighted summation of J Gaussian kernels,

$$f'_d = \sum_{k=1}^{K} \sum_{y \in \mathcal{N}(x)} g_k w_k^p(u(x,y)) f(y) \quad p = 1, 2, 3, ..., P \qquad (3)$$

where f'_p is one entry of the P-dimensional output feature vector. g_k is the weight associated to the k_{th} Gaussian kernel and $f(y)$ is the feature vector of node v_y. $\mathcal{N}(x)$ are the collection of the neighbors of the node v_x. The learnable weighting function $w_k^p(u(x,y))$ is defined on the pseudo-coordinates $u(x,y)$ for aggregating feature vectors of the neighboring nodes. One of the key design factors of the graph-based convolutions is the choice of weighting functions or kernel functions such as B-spline kernels [33] and Gaussian Mixture Model (GMM)-based kernels [34]. In this paper, we choose GMM-based kernel for convolution operations. Specifically, GMM-based convolution adopts K Gaussian models as the kernel functions and the weighting function of the k_{th} Gaussian model can be written as:

$$w_k(u) = \exp\left(-\frac{1}{2}(u - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(u - \boldsymbol{\mu}_k)\right) \qquad (4)$$

where Σ_k^{-1} is the covariance matrix and the μ_k is the mean vector of the k_{th} Gaussian model. We denote the kernel size (number of Gaussian models) as K in the following manuscript.

The choice of the pseudo-coordinates is another important design factor for graph-based convolutions. In this paper, we use relative Cartesian coordinates in three dimensions (x, y, t) to estimate the relative position between neighbors so that both the spatial and temporal information can be extracted from the 3D-Graphs through GMM-based convolution.

3.1.2.2 **Graph-ResNet:** The Graph-ResNet layer of the GCN-based deep recognition network is designed according to the approach proposed in [27]. The major difference is the choice of the kernels and definition of the kernel size when operating graph convolution. Graph-ResNet is believed to be able to address the gradient degradation issue when the network depth goes deep. Lower-left of Figure 2 shows an example of the Graph-ResNet using GMM-based convolution. The kernel size K_1 in our Graph-ResNet is the number of Gaussian Models used for graph-based convolution (refer to Equation 4). Batch normalization (BN) is applied after each GMM-based convolution operation and a shortcut connection is added with kernel size $K_2 = 1$. As the results of our evaluation, the Graph-ResNet brings significant improvement on the recognition accuracy when incorporated in our GCN-based deep recognition network.

3.1.2.3 **Graph Nodes Clustering and MaxPooling:** Graph nodes clustering and MaxPooling strategy [47] is another important component in our approach. It is applied to reduce the complexity and alleviate the issue of overfitting of when the network goes deep. MaxPooling aggregates feature vectors of the nodes in the same cluster to obtain the abstract representation so that the dense graph is transformed to a coarsen graph. The clusters are formed by evenly dividing the spatio-temporal space into 3D grids with size d (number of pixels) in each dimension, which is also known as pooling size. The nodes falling into the same grid will be merged together via MaxPooling. MaxPooling picks up the maximum value from dimension of the feature vectors of the nodes clustered together as the representation of the corresponding node in the graph of the next layer. If the size of the spatio-temporal space in three dimensions is D_1, D_2, D_3 respectively, the maximum number of nodes after MaxPooling will be $\left\lceil \frac{D_1}{d} \right\rceil \times \left\lceil \frac{D_2}{d} \right\rceil \times \left\lceil \frac{D_3}{d} \right\rceil$.

3.1.2.4 **Detailed Network Architecture:** With the key components introduced above, We design a GCN-based deep recognition network for identifying gait from event-streams. The 3D-Graphs constructed from event-streams are taken as inputs to train the network. It starts with convolving the input graphs with a GMM-based Graph-ConvNet, GC_0(5,64), whose kernel size is 5 and output feature size is 64. A MaxPooling layer, MP_0(4), with grid size 4 is applied to merge the graph nodes from the first Graph-ConvNet layer. Then three Graph-ResNet layers, $GRes_1$(5,1,128), $GRes_1$(5,1,256) and $GRes_1$(5,1,512) with $K_1 = 5$ and $K_2 = 1$ are stacked sequentially whose output feature sizes are 128, 256 and 512 respectively. The resultant activations of ReLu [48] functions from each Graph-ResNet are passed to MaxPooling layers with pooling size $d = 6$, $d = 24$ and $d = 64$ respectively. At last, a fully-connected layer with 1024 nodes (FC(1024)) is connected to the last MaxPooling layer and softmax functions are used for obtaining the final recognition results. The detailed parameter settings of the network layers in sequence are GC_0(5, 64)- MP_0(4)- $GRes_1$(5, 1, 128)- MP_1(6)- $GRes_2$(5, 1, 256)- MP_2(24)- $GRes_3$(5, 1, 512)- MP_3(64)- FC(1024).

3.2 EV-Gait-IMG

Different from GCN-based gait recognition approach, EV-Gait-IMG utilizes image-like representation and CNN-based deep recognition networks (it was first introduced in our previous work published on CVPR in 2019 [49]). As shown in Figure 3(a), Ev-Gait-IMG starts from capturing asynchronous raw event-stream while the subject is walking through the view. Then the raw event-stream is preprocessed and represented according to the design of the input layer of the EV-Gait-IMG. Finally, we train our CNN-based deep network and apply it to recognize the identities of the subjects based on event-streams.

3.2.1 Image-like Representation

Image-like representation of asynchronous event-streams was proposed in [50] which can be directly fitted into state-of-the-art CNN-based structure. Event-streams are converted to image-like representation with four channels, known as event image, for our deep neural networks. The first two channels accommodate the counts of positive or negative events at each pixel respectively. These heatmap-like distributions can effectively describe the spatial characteristics of the event-stream. Then the other two channels are constructed from the timestamps of the positive and negative events respectively. They hold the ratios describing the temporal characteristics. The ratio $r_{i,j}$ at pixel (i, j) is defined as,

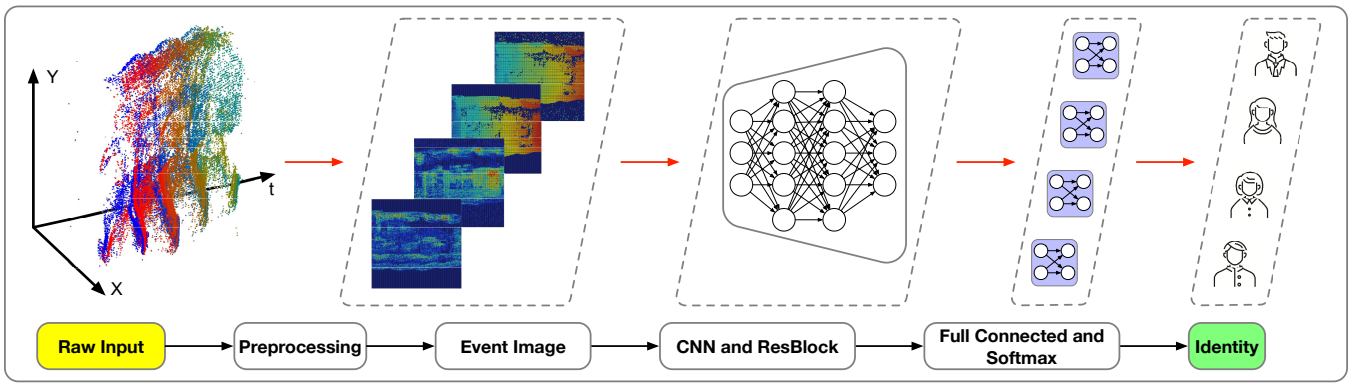
$$r_{i,j} = \frac{t_{i,j} - t_{begin}}{t_{end} - t_{begin}} \tag{5}$$

$t_{i,j}$ is the timestamp of the most recent positive (negative) event at pixel (i, j), t_{begin} is the timestamp of the first positive (negative) event and t_{end} is the last positive (negative) event of the whole stream. These ratios estimate the lifetime of object of interest at different locations.

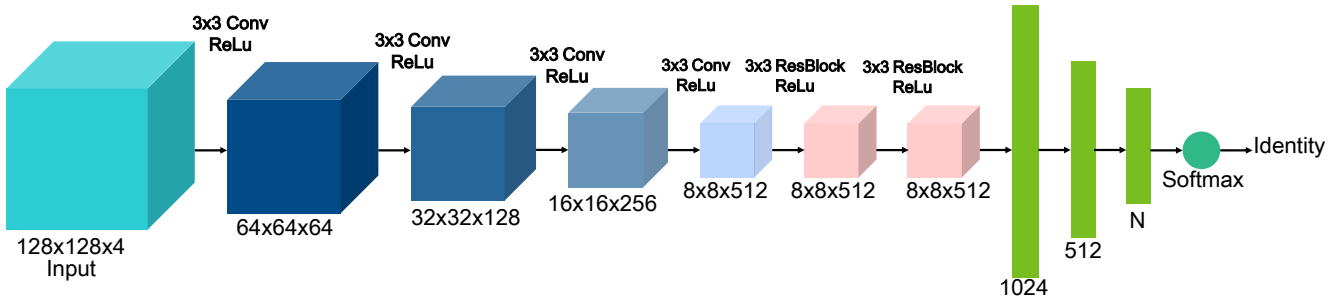
After the above processes, the event-streams are represented as event images ready for training the deep neural network.

3.2.2 CNN-based Deep Recognition Network

Our deep neural network for event-based gait recognition can be vastly divided into two major components: convolutional layers with Residual Block (ResBlock) layers are responsible for feature extraction and fully-connected layers with softmax associate the features to different identities. The convolutional layers have been proved an effective way to extract features and popularly applied in image classification tasks [51], [52], [53]. The ResBlock layers [54] are able to deal with the vanishing gradient problem when the network goes deeper so that features extracted by convolutional layers can be better integrated. The fully-connected layers



(a)



(b)

Fig. 3: Network architecture of the proposed EV-Gait.

decode the features and pass them to the softmax functions to execute classification tasks.

The detailed design of our network is shown in Figure 3(b). It starts from a special input layer to accommodate the event images presented in Section 3.2.1. The input image is passed through four convolutional layers whose filter size is 3×3 and stride is 2. The number of channels of the four convolutional layers are 64, 128, 256 and 512 respectively. After the convolutional layers, the resultant activations of the ReLu [48] functions are passed through two ResBlock layers to deal with the vanishing gradient problem and keep the features extracted from lower layers when our network goes deeper. The two ResBlock layers share the same parameters: the filter size is 3×3, the stride is 1 and the number of channels are 512. Then, two fully-connected layers with 1024 and 512 nodes respectively are connected to the ResBlock layers and softmax functions are stacked to finalize the whole network. Finally, the cross entropy loss function and Adam optimizer [55] are adopted to train the network.

4 Evaluation

In this section, we evaluate our proposed event-based human gait recognition approaches, EV-Gait-3DGraph and EV-Gait-IMG, on both data collected in real-world experiments and converted from publicly available RGB gait databases. In our experiments, we use a DVS128 Dynamic Vision Sensor from iniVation [56] operating at 128×128 pixel resolution. The event data is streamed to and processed on a desktop machine with Intel i9-9980Xe CPU and 128G DDR4 Ram running Ubuntu 16.04, and the deep networks (discussed in Section 3) are trained on two NVIDIA RTX Titan GPUs. In the following, we first evaluate performance of EV-Gait-3DGraph and EV-Gait-IMG with different parameter choices in Section 4.2 and Section 4.3, and then compare our proposed approaches to a number of existing event-based recognition methods in Section 4.4. Finally, the performance of the proposed approaches are benchmarked with RGB-based approaches in Section 4.5.

4.1 Realworld Dataset Collection and Implementation Details

We recruited a total number of 20 volunteers (14 males and 6 females) to contribute their data in two experiment sessions spanning over three weeks. In each session, the participants were asked to walk normally in front of a DVS128 sensor mounted on a tripod, and repeat walking for 100 times. The sensor viewing angle is set to approximately 90 degrees with respect to the walking directions. The second experiment sessions were conducted after at least one week after the previous sessions. We did not specify the distance between the walking subject and the device so that the position and scale of the human figures shown in the streams could be different. This setting introduces practical variance which is challenging for the methods sensitive to object alignment.In total we collected 4,000 samples of event streams capturing gait of the 20 volunteers. As the dataset is collected during daytime, it is named as DVS128-Gait-Day. Fig. 4 shows visualization of the data from 4 different identities (events accumulated within 20ms), where the color of pixels indicate polarity (red for +1, green for -1). We also collected another

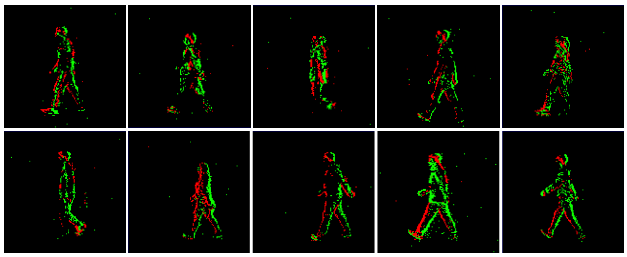


Fig. 4: Visualization of the event streams (accumulated over 20ms) of 10 different identities in the DVS128-Gait-Day dataset.

dataset at night without dedicated illumination to specifically evaluate the performance of EV-Gait under challenging lighting condition, which will be specified in Section 4.4.4.

We implement the proposed deep network in EV-Gait (discussed in Section 3) with PyTorch [57]. The dataset DVS128-Gait-Day is used to determine the parameters of the proposed deep networks. The data collected in the first session (100 samples per subject) is used for training, while for testing we use data from the second session (100 samples per subject). During training we set the batch size as 16 for both of the proposed methods and the learning rate for EV-Gait-3DGraph and EV-Gait-IMG is 1e-3 and 3e-7 respectively. Both training and testing were performed on a cluster of two NVIDIA RTX Titan GPU.

4.2 Parameter Choice of EV-Gait-3DGraph

To determine appropriate parameter settings of EV-Gait-3DGraph, we evaluate the recognition accuracy of the proposed method by varying the values of a number of different parameters including MaxNumEvents, neighboring range, last pooling size, convolution kernel size, the influence of Graph-ResNet and complexity of the network architecture. When evaluating one of the parameters, the other parameters remain unchanged at their default values which will be justified in each corresponding sections below.

4.2.1 Evaluation on MaxNumEvents

MaxNumEvent	80	60	40	20
Accuracy	87.2±1.1%	89.0±1.3%	93.8±0.8%	89.5±1.6%
Remaining Events	1001	1162	1994	3965

TABLE 1: Recognition accuracy and number of remaining events after downsampling with different values of MaxNumEvents.

As introduced in Section 3.1.1, the maximum number of events (*MaxNumEvents*) of each grid determines the rate of compression when downsampling the event-streams using nonuniform OctreeGrid filtering. We gradually reduce the *MaxNumEvents* from 80 to 20 and compute corresponding recognition accuracy of EV-Gait-3DGraph by repeating independent training and testing trials by 30 times. For each trial, the samples collected from the first session are used for training and those collected from the second session are for testing. The resultant average and standard deviation of the recognition accuracy and the number of remaining events of each event-stream after downsampling is shown in Table 1. From the results we can observe, the recognition accuracy is improved from 87.2% to 93.8% when *MaxNumEvents* decreases from 80 to 40 and the averaged number of remaining events 3D-Graph grows from 1001 to 1994. Then the accuracy significantly drops to 89.5% when the *MaxNumEvents* is further reduced to 20 and the corresponding number of remaining events are close to 4000. Therefore, the default value of the *MaxNumEvents* is chosen as 40. The improvement of accuracy at the beginning is derived from the information gain brought by the growth of the number of remaining events. However, the model complexity becomes the major issue when excessive events are included in graph construction which causes accuracy drop afterwards.

4.2.2 Evaluation on Neighboring Range

Neighboring range	3	4	5	6
Accuracy	87.3±0.7%	92.1±1.0%	93.8±0.8%	89.5±1.6%
Number of edges	1573	3660	6318	10016

TABLE 2: Recognition accuracy and number of edges of EV-Gait-3DGraph with different neighboring range.

We then evaluate the recognition accuracy of EV-Gait-3DGraph with various neighboring ranges (R). As the neighboring range determines the connectivity of the 3D-Graphs, the number of edges of each event-stream are also calculated. Table 2 presents the average and standard deviation of the recognition accuracy over 30 repeated experiments and the average number of edges of the 3D-Graphs. By analyzing the results we can see that the recognition accuracy improves significantly when neighboring range grows from 3 to 5 because local feature extraction is enhanced when more edges are generated to connect more nodes. However, in the meantime, the model complexity also grows significantly with the growth of the number of edges, which affects the performance of the deep neural networks, e.g., the recognition accuracy drops to 89.5% when the neighboring range is 6. Therefore, $R = 5$ is chosen as the default value for the neighboring range.

4.2.3 Evaluation on Kernel Size

Kernel Size	No ResNet	K=3	K=4	K=5	K=6
Accuracy	90.5±0.9%	92.9±0.9%	92.5±1.4%	93.8±0.8%	92.9±1.4%

TABLE 3: Recognition accuracy of EV-Gait-3DGraph with different sizes of convolution kernel and the impact of Graph-ResNet.

In this part of evaluation, we estimate the recognition accuracy of EV-Gait-3DGraph with different settings for GMM-based convolution. Again, the recognition accuracy presented in Table 3 is obtained from averaging the results of 30 repeated trials. By comparing the recognition accuracy of EV-Gait-3DGraph with or without Graph-ResNet, we find the Graph-ResNet in GCN is able to improve the recognition accuracy by up to 3.3% (93.8% v.s. 90.5%). Then, we investigate the recognition accuracy of EV-Gait-3DGraph with respect to different sizes of convolution K. The results show the recognition accuracy peaks at $K = 5$. Therefore,

$K = 5$ is chosen as the default kernel size for GMM-based convolution.

4.2.4 Evaluation on Pooling Size

Pooling size	16	32	64	128
Accuracy	86.1±1.5%	89.1±1.1%	93.8±0.8%	94.9±1.5%
Number of grids	512	64	8	1

TABLE 4: Recognition accuracy and number of grids of EV-Gait-3DGraph with different pooling size of the last MaxPooling layer.

When tuning the design factors of the graph-based deep recognition network, we find the change of the pooling size of the last MaxPooling layer has significant impact on the recognition accuracy of EV-Gait-3DGraph. We again retrain the model and report the averaged recognition accuracy over 30 independent trials in Table 4. The pooling size at the last MaxPooling varies from 16 to 128 and results in 512 to 1 nodes at the last layer. From the results we see that as higher pooling size at the last MaxPooling is applied, the recognition accuracy increases and the highest accuracy (94.5%) is achieved when the pooling size is 128. However, by carefully investigating the specific results of each trials, we find the recognition accuracy is not stable when pooling size is 128 because of the limited feature space (512). The graph is merged into only one node and it may not be able handle larger number of subjects. Therefore, we set the default pooling size of the last MaxPooling layer as 64 which aggregates the graph into 8 nodes at last.

4.2.5 Evaluation on Network Complexity of GCN

At last, we investigate influence of the network complexity on the recognition accuracy by removing one the of Graph-ResNet layers(along with the associated MaxPooling layer). We retrain the simplified models and report the average and standard deviation of the accuracy obtained from 30 independent training and inference trials in Table 5. Compared with the original accuracy (last column), we can observe removing one of the graph-based convolutional layers will lead to significant accuracy drop by at least 5%).

Operation	Remove ResGC0	Remove ResGC1	Remove ResGC2	Remove ResGC3	Original
Accuracy	83.9±1.8%	82.0±2.4%	88.9±1.3%	88.0±1.4%	93.8±0.8%

TABLE 5: The impact on recognition accuracy when removing one of the Graph-based convolution layers.

4.3 Parameter Choice of EV-Gait-IMG

We now evaluate the recognition accuracy of EV-Gait-IMG with different parameter settings, including the setup of the input representation, the use of ResBlocks, size of the convolution kernels and complexity of the network architecture.

4.3.1 Evaluation on the Setup of Representation

The image-like representation proposed in [50] converts the asynchronous event-streams to event-image with four channels: two channels accommodate the counts of positive or negative events at each pixel and the other two channels account for the temporal characteristics. In this section, we will evaluate four different setups of the representation to determine the importance of each type of channel:

- **All Channels.** All four channels are considered, which is the original setup of the image-like representation.
- **Counts Only.** Only the two channels accommodating the counts of positive or negative events are kept.
- **Time Only.** Only the two channels holding temporal characteristics are kept.
- **No Polarity.** The polarity of the events is removed.

Channel Setup	All Channels	Counts Only	Time Only	None Polarity
Accuracy	86.6±0.4%	87.3±0.9%	52.3±2.5%	86.0±1.2%

TABLE 6: Recognition accuracy of EV-Gait-IMG with different representation setups.

The average and standard deviation of recognition accuracy of each representation setup is computed from 30 repeated experiments and is in Table 6. From the results, we can see, the channels holding the event-distribution characteristics only (counts of events) produce significantly higher accuracy than those holding the temporal characteristics only (87.3% v.s. 52.3%) and including temporal channels cannot guarantee better performance (86.6%).

4.3.2 Evaluation on ResBlocks

Kernel Size	No ResBlock	K=2	K=3	K=4	K=5
Accuracy	85.8±0.5%	86.4±0.7%	87.3±0.9%	86.6±0.6%	86.9±0.5%

TABLE 7: Recognition accuracy of EV-Gait-IMG with different convolution kernel sizes

In this section, we evaluate the impact of the ResBlock on the recognition accuracy of EV-Gait-IMG when different convolution kernel sizes are applied. We repeat the training and inference process 30 times for each kernel size and compute their average and standard deviation of recognition accuracy. The simplified network structure without ResBlock components is also considered in this part of evaluation. From the results shown in Table 7, we can find the ResBlocks are able to improve the recognition accuracy by up to 1.5% (87.3% v.s. 85.8%) and the change of kernel size only has trivial impact on the recognition accuracy.

4.3.3 Evaluation on Network Complexity of CNN

Operation	Remove one FC layer	Remove one ResBlock	Remove one ConvNet	Original
Accuracy	87.0±0.9%	85.2±0.9%	85.5±0.9%	87.3±0.9%

TABLE 8: The impact on recognition accuracy when removing one of the FC/ convolution/Resblock layers.

We also study the impact of the network complexity of EV-Gait-IMG on the recognition accuracy by removing different types of layers. We remove one of the FC, ResBlock or ConvNet layers to form the simplified models. Then the new models are retrained and the corresponding accuracy is reported in Table 8. From the results we can observe, our original network achieves the highest recognition accuracy

and one of the FC layers can be removed when implemented on the resource-constrained platform as it only brings trivial impact on the accuracy.

4.4 Comparison with Different EV-based Methods

Finally, we compare the recognition accuracy of different EV-based deep recognition networks with either graph-based or image-like representations. We also include supportive vector machine (SVM) as a benchmark to determine if deep neural networks are necessary for the EV-based gait recognition task. Besides EV-Gait-3DGraph and EV-Gait-IMG proposed in this paper, the other competing approaches are as follows:

2DGraph-3DCNN [28] was proposed to extract the spatio-temporal feature from event-streams. It splits each event-stream into multiple slices over time. For each slice, a very short-term of period (e.g., 30ms) is picked to construct a 2D-Graph and spatial features are extracted through graph-based convolution with a B-spline kernel [33]. Then the 2D-Graphs are transferred to a grid representation through Graph2Grid operations [28]. Finally, 3DCNN [58] is applied to extract the spatio-temporal features for actions recognition. We optimize the network structure of 2DGraph-3DCNN for the gait recognition task, and the final detailed network settings are GC_0(5, 64) - MP_0(2) - GC_1(5, 128) - MP_1(4) - Graph2Grid(8, 32, 128) - $3DConv_0$(3, 128) - $3DMP_0$(2) - $3DConv_1$(3, 256) - $3DMP_1$(2) - $3DConv_2$(3, 512) - $3DMP_2$(2) - $3DConv_3$(3, 512) - $3DMP_3$(2) - GA(512) - FC(256) - Dropout(0.5). GC(5, 64) is a graph-based convolution with kernel size 5 and output feature size 64. MP(2) and 3DMP(2) are two or three dimensional MaxPooling layer with pooling size 2 for each dimension. Graph2Grid(8, 32, 128) converts a stack of graphs constructed from 8 slices of event-stream to eight 32×32 matrices and the output feature size (depth) is 128. 3DConv(3, 512) is 3D-convolution layer with kernel size 3 and output feature size 512. GA(512) merges multiple features from previous layer into a one-dimensional global feature. Finally, FC(256) is fully-connected layer with 256 nodes and Dropout(0.5) randomly throws half of the coefficients to alleviate the problem of overfitting.

LSTM-CNN is based on EV-Gait-IMG but considers the variance of human gait through time. It splits the whole event-stream into multiple slices and we set the number of slices as 8 which produces the highest accuracy. Each short-term slice is converted to the image-like representation. The CNN-based network inheriting from EV-Gait-IMG is applied to extract spatial feature from each slice. Then the sequence of feature vectors are taken as the input of a Long Short-Time Memory (LSTM) network with 100 hidden states to recognize gaits from the event-streams.

SVM-PCA is a benchmark method to determine if the deep neural networks are necessary for our EV-based gait recognition task. SVM-PCA adopts the same event images as EV-Gait-IMG and concatenates the event images by columns to form high-dimensional vectors. Principal Component Analysis (PCA) is applied on the high-dimensional vectors to extract features (we set the output dimension of PCA as 500) to train SVM-based classifier for gait recognition.

4.4.1 Comparison on Best Accuracy

We compute the average and standard deviation of the recognition accuracy of the five competing approaches over 30 independent training and inference trials. The parameters of the five approaches are all carefully tuned and the best averaged accuracy is reported in Table 9. The results show that, the two approaches with graph-based representations achieves significantly higher recognition accuracy than those with image-like representations and the gap is up to 8.4% (94.9% v.s. 86.5%). By further comparing the two graph-based approaches, we find the 3D-Graph representation produces higher recognition accuracy than 2D-Graph as it can better preserve the spatio-temporal information of the asynchronous event-streams than a sequence of discrete 2D-Graphs in a human gait recognition task. It is worth noting that, the accuracy of SVM-based classifier cannot compete with the four deep learning approaches and the difference is up to 16.5%.

Methods	EV-Gait -3DGraph	2DGraph -3DCNN	EV-Gait -IMG	LSTM -CNN	SVM -PCA
Accuracy	94.9±1.5%	92.2±2.1%	87.3±0.9%	86.5±0.8%	78.05%

TABLE 9: Recognition accuracy of different EV-based gait recognition approaches.

There are a number of reasons that 3D-Graph representation generates the highest recognition accuracy among the competing approaches for event-based gait recognition. First, image-like representations suffer from misalignment and noisy background issues, the recognition accuracy cannot be guaranteed if the distance between the walking subject and the camera is not well-controlled, which leads to various scales of the recorded subject. However, event-stream alignment is challenging and still remains unsolved. On the contrary, graph-based representation focuses on the moving subject in the view directly, therefore alleviates the influence of misalignment and background noises. 2DGraph-3DCNN employs 2D-Graphs for spatial feature extraction, however, the 3DCNN component requires careful alignment when mapping the 2D-Graphs to grid representation. Finally, 2DGraph-3DCNN converts the event-stream to discrete 2D-Graphs by picking up very short-term period from the slices of event-stream and the information in between is discarded. While EV-Gait-3DGraph takes the whole event-stream as an entirety and preserves most of the shape when constructing the 3D-Graph.

4.4.2 Comparison on Number of Training Samples

We then compare the recognition accuracy of the event-based approaches with respect of the amount of training samples per subject. The amount of samples per subject required for training is important as few shot learning can save significant training efforts especially when registering new subjects. It has significant impact on user experience. In particular, we randomly select different number of training samples from each subject, varying from 5 to 100. For each case, we retrain all the four event-based approaches for 30 times and report the average and standard deviation of recognition accuracy. Table 10 shows the results, and we see that as more samples are used in training, the recognition accuracy of all approaches grows, but with different growth

rates. By comparing the results across different approaches, EV-Gait-IMG produces significantly higher recognition accuracy than graph-based approaches when number of training samples is low and becomes almost level after 10 or more training samples are used. The accuracy of EV-Gait-3DGraph surpasses other approaches when sufficient training samples (over 50 in the table) are used. This indicates that EV-Gait-IMG doesn't require massive training data to converge so the image-like representation is the choice when only limited number of training samples are available. In contrast, EV-Gait-3DGraph shows significantly higher performance cap than those using image-like representation, therefore, is more preferable when sufficient training data could be sourced.

Samples\Methods	EV-Gait -3DGraph	2DGraph -3DCNN	EV-Gait -IMG	LSTM -CNN
5 Samples	36.3±2.2%	6.1±1.3%	**79.9**±2.0%	33.1±6.3%
10 Samples	61.7±4.1%	14.5±2.7%	**85.9**±1.8%	57.6±8.6%
20 Samples	78.6±1.1%	48.2±7.3%	**86.5**±0.7%	76.0±4.2%
50 Samples	**90.0**±1.2%	82.9±3.2%	87.2±0.8%	85.5%±1.4
100 Samples	**94.9**±1.5%	92.2±2.1%	87.3±0.9%	86.5±0.8%

TABLE 10: The recognition accuracy of EV-based deep recognition networks with different number of training samples per subject (highest accuracy in each line is highlighted).

4.4.3 Comparison on Length of Event-stream

In EV-based gait recognition, the event-streams are generated while the subjects are walking. The subjects are different from their figures and pattern of walking. In this section, we evaluate the recognition accuracy on different length of event-streams to show that the gait (walking pattern) plays a significant role in subject identification using event camera. We split each long event-stream into multiple slices according to predefined length of time from 50ms to 1500ms. The slices with the same length are gathered for training and inference with EV-Gait-3DGraph and EV-Gait-IMG. The average and standard deviation of recognition accuracy on different length of event-streams is reported in Table 11. From the results we can find, the recognition accuracy of the both approaches increases with the growth of the length. Specifically, when the length of the event-streams is short (where the "gait" cannot be well observed), the recognition accuracy is much lower than that with full length event-streams. The comparison indicates the "gait" plays significant role in recognizing the walking subjects.

Length \ Methods	EV-Gait-3DGraph	EV-Gait-IMG
50ms	14.4±0.5%	**41.8±1.5%**
100ms	26.1±0.6%	**55.8±0.8%**
200ms	38.5±0.7%	**56.5±1.1%**
500 ms	**67.4±1.3%**	59.6±2.6%
1000ms	**82.1±0.8%**	63.4±2.7%
1500ms	**86.0±1.8%**	76.6±1.9%
full length	**94.9±1.5%**	87.3±0.9%

TABLE 11: The recognition accuracy of event-based deep recognition networks with different length of event-streams.

4.4.4 Comparison on Different Lighting Conditions

To investigate if the event camera is able to capture human gaits in low-light condition, we collected the dataset DVS128-Gait-Night during night without dedicated lighting. We again recruited 20 volunteers for data collection and each volunteer contributed 200 samples of gait in front of event camera. We evaluate recognition accuracy of the EV-Gait-3DGraph and EV-Gait-IMG on DVS128-Gait-Night and include the results from daytime dataset as benchmark to demonstrate their performance in low-light condition. When computing the recognition accuracy, half of the samples are randomly selected for training and the rest for testing. The results shown in Table 12 are average and standard deviation of the results from 30 independent trials in which the training and testing samples are re-selected randomly. From the results we can observe, dataset collected from low-light condition is more challenging than that from daytime and the recognition accuracy of EV-Gait-3DGraph and EV-Gait-IMG drops by 3.7% and 24.6% respectively. Meanwhile, EV-Gait-3DGraph demonstrates significantly superior performance than EV-Gait-IMG and the difference is over 24% on low-light dataset.

Light\Methods	EV-Gait-3DGraph	EV-Gait-IMG
Daytime	99.7±0.1%	96.1±0.4%
Nighttime	96.0±1.6%	71.5±4.3%

TABLE 12: Recognition accuracy of EV-Gait on datasets collected from different lighting conditions.

4.4.5 Comparison of Resources Consumption

Apart from the recognition accuracy, the resources consumption of the EV-Gait approaches are also important for practical use. We implement both EV-Gait-3DGraph and EV-Gait-IMG on Intel UP Board [59] with a Quad-core 1.44Ghz Intel Atom x5-Z8350 microprocessor on board. The RAM of the board is 1G and ROM is 16G. The operating system is Ubuntu 16.04. After implementation, we profile the resources consumption of the total number of coefficients, averaged inference time, memory usage and energy consumption of the proposed EV-Gait approaches. The number of coefficients can be conveniently obtained from Pytorch API. Average inference time and memory usage can be drawn from the system when running the programs. We use external tool to monitor the power consumption (current and voltage) of the board when running the inference of different EV-Gait approaches.

Methods	EV-Gait-3DGraph	EV-Gait-IMG
Number of Coefficients	7.15 M	64.61 M
Average Inference Time	436.23 ms	238.43 ms
Memory Usage	410.87 Mbytes	413.05 Mbytes
Energy Consumption	0.876 J	0.238 J

TABLE 13: Resources consumption of EV-Gait on UP board.

The resources consumption of gait recognition on UP board are shown in Table 13. First of all, the average inference time is acceptable on the resource-constrained platform; the inference can made within half second with the slower approach (EV-Gait-3DGraph). Then by comparing the resources consumption of different EV-Gait approaches, we can observe the number of coefficients of GCN-based approach is only about one ninth of CNN-based approach (7.15 million v.s. 64.61 million), however, it requires almost the

same running memory (410.87Mbytes v.s. 413.05Mbytes), 1.8 times inference time and 3.7 times energy consumption compared with CNN-based approach. Our conjecture is because the graph-based convolution is based on an extension library for Pytorch (Pytorch Geometric [60]) which is implemented by third-party and not well optimized. Therefore, we can claim that, with the popularity of GCNs, the resources consumption of the GCN-based approach can be significantly reduced when proper optimization on the implementation of graph convolution are available in the future.

4.5 Comparison with RGB-based Benchmarks

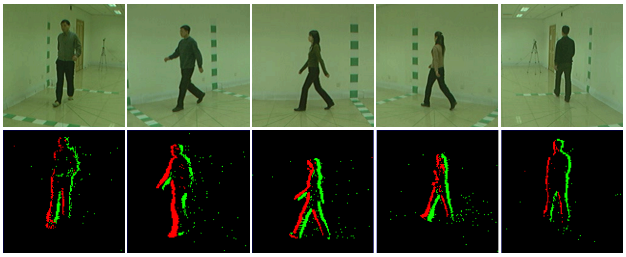


Fig. 5: Examples from the original CASIA-B dataset (top row) and visualization of the corresponding event streams (accumulated over 20ms) in our converted EV-CASIA-B dataset (bottom row).

We have showed that event-based approaches perform well in data collected from real-world settings, and we now show that it could also achieve comparable performance with the state-of-the-art gait recognition approaches that are designed for RGB images. Since those approaches do not work on event-streams, for fair comparison, we convert the widely used CASIA-B [61] benchmark into its event version EV-CASIA-B. Then we run event-based approaches on the converted EV-CASIA-B dataset, and compare the resulting recognition accuracy with that of the state-of-the-art approaches on the original CASIA-B dataset.

4.5.1 Data Collection and Implementation Details

CASIA-B is one of the most popular benchmark for RGB camera-based gait recognition methods [62], [63], [64], [65]. It contains data from 124 subjects, each of which has 66 video clips recorded by RGB camera from 11 different view angles (0° to 180°), i.e., 6 clips for each angle. The view angle is the relative angle between the view of the camera and walking direction of the subjects. To convert the CASIA-B dataset to event format, we use a similar approach as in [66] and use a DVS128 sensor to record the playbacks of the video clips on screen. In particular, we use a Dell 23 inch monitor with resolution 1920×1080 at 60Hz. Fig. 5 shows some examples from the original CASIA-B dataset (top row) and the visualization of the corresponding event streams in our converted EV-CASIA-B dataset.

We consider the same deep network structure as in the previous experiments on the DVS128-Gait-Day dataset. For training, we use the data of the first 74 subjects to pre-train the network. Then for the other 50 subjects, for each viewing angle we use the first 4 out of 6 clips to fine-tune the network, and the rest 2 clips are used for testing. We implement two competing approaches that work on RGB images: i) **3D-CNN** [18] and ii) **Ensemble-CNN** [18], which can achieve state-of-the-art gait recognition performance on the original CASIA-B benchmark.

4.5.2 Evaluation on Different View Angles

Table 14 shows the gait recognition accuracy of the proposed EV-Gait-3DGraph and EV-Gait-IMG with the competing approaches 3D-CNN and Ensemble-CNN. It is worth pointing out that the frame rate of the video clips in CASIA-B dataset is only 25 FPS, with a low resolution at 320×240. As a result when converting such data into event format via playback on the screen, the DVS sensor will inevitably pick up lots of noise. In addition, unlike the original RGB data, the event-streams inherently contain much less information (see Fig. 5). However, as we can see from Table 14, the proposed EV-Gait-IMG can still achieve comparable gait recognition accuracy (89.9%) with the competing RGB camera based approaches overall (94.1%). For some viewing angles, especially when the walking directions of the subjects are perpendicular with the camera optical axis (e.g. around 90°), the proposed EV-Gait-IMG even outperforms the state-of-the-art 3D-CNN and Ensemble-CNN (96.2% vs. 88.3% and 91.5%). EV-Gait-IMG achieves the highest accuracy when the view angle is 90° because in such settings the event streams captured by the DVS sensor can preserve most of the motion features. On the other hand, for the viewing angles that the subjects walk towards/away from the camera (e.g. 0° or 162°), the accuracy of EV-Gait-IMG is slightly inferior to the RGB-based approaches. This is expected, since in those cases compared to RGB images, the event-streams contain fewer informative features on the subjects' motion patterns, and thus struggle to extract their identities. It is worth noting that the recognition accuracy of EV-Gait-3DGraph shown in the first row of Table 14 is much worse than any other competing approaches because only 4 video clips per subject are used for training which is not sufficient to obtain a well-trained model of EV-Gait-3DGraph. This observation is consistent with our evaluation results on the amount of training data.

5 Conclusion

In this paper, we investigate the optimal representation of asynchronous event-streams by proposing EV-Gait-3DGraph and EV-Gait-IMG, new approaches for gait recognition with 3DGraph and image-like representations using DVS sensors. EV-Gait-3DGraph and EV-Gait-IMG constructs either 3D-Graphs or image-like representations from asynchronous event-streams. Then corresponding and graph-based and CNN-based deep neural networks are designed for recognizing gait from event-streams. We collect multiple event-based gait datasets from both real-world experiments and RGB-based benchmark. According to the evaluations on the dataset collected from practical setting, EV-Gait-3DGraph and EV-Gait-IMG achieve up to 94.9% and 87.3% accuracy respectively when amount of training data is sufficient for networks to converge. Finally, we evaluate the event-based approaches on a dataset converted from

Methods\Angle	0°	18°	36°	54°	72°	90°	108°	126°	144°	162°	180°	mean
EV-Gait-3DGraph	58.6%	81.8%	92.9%	86.6%	89.2%	92.1%	89.8%	87.5%	86.7%	78.5%	60%	82.2%
EV-Gait-IMG	77.3%	89.3%	94.0%	91.8%	**92.3**%	**96.2**%	**91.8**%	91.8%	91.4%	87.8%	**85.7%**	89.9%
3D-CNN	87.1%	93.2%	97.0%	94.6%	90.2%	88.3%	91.1%	93.8%	96.5%	96%	85.7%	92.1%
Ensemble-CNN	88.7%	95.1%	98.2%	96.4%	94.1%	91.5%	93.9%	97.5%	98.4%	95.8%	85.6%	94.1%

TABLE 14: Gait recognition accuracy of EV-Gait-3DGraph, EV-Gait-IMG (evaluated on EV-CASIA-B dataset) and two competing RGB based approaches (evaluated on CASIA-B dataset). Note that for viewing angles 72°, 90° and 108°, EV-Gait-IMG even performs better than the RGB based approaches.

video-based gait dataset and the results are comparable with state-of-the-art RGB-based approaches on the benchmark.

Acknowledgment This work is partially supported by National Natural Science Foundation of China under Grant 61702133, 91846205 and the National Key R&D Program under Grant 2017YFB1400100. The authors would like to thank NVIDIA for GPU donations.

REFERENCES

[1] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15μ s latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[2] R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, "A 240× 180 10mw 12us latency sparse-output vision sensor for mobile applications," in *VLSI Circuits (VLSIC), 2013 Symposium on*. IEEE, 2013, pp. C186–C187.

[3] 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 Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, 2011.

[4] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 16–23.

[5] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3d reconstruction and 6-dof tracking with an event camera," in *European Conference on Computer Vision*. Springer, 2016, pp. 349–364.

[6] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visualinertial odometry for event cameras using keyframe-based nonlinear optimization," in *British Machine Vis. Conf.(BMVC)*, vol. 3, 2017.

[7] T. Delbruck, M. Pfeiffer, R. Juston, G. Orchard, E. Müggler, A. Linares-Barranco, and M. Tilden, "Human vs. computer slot car racing using an event and frame-based davis vision sensor," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 2409–2412.

[8] E. Mueggler, N. Baumli, F. Fontana, and D. Scaramuzza, "Towards evasive maneuvers with quadrotors using dynamic vision sensors." in *ECMR*, 2015, pp. 1–8.

[9] T. Delbruck and M. Lang, "Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor," *Frontiers in neuroscience*, vol. 7, p. 223, 2013.

[10] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of aer dynamic vision sensors," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 781–784.

[11] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "Hats: Histograms of averaged time surfaces for robust event-based object classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740.

[12] 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 transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1346–1359, 2017.

[13] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current opinion in neurobiology*, vol. 20, no. 3, pp. 288–295, 2010.

[14] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 316–322, 2006.

[15] L. Lee and W. E. L. Grimson, "Gait analysis for recognition and classification," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 155–162.

[16] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.

[17] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 10, 2007.

[18] Z. Wu, Y. Huang, L. Wang, X. Wang, and T. Tan, "A comprehensive study on cross-view gait based human identification with deep cnns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1.

[19] K. Shiraga, Y. Makihara, D. Muramatsu, T. Echigo, and Y. Yagi, "Geinet: View-invariant gait recognition using a convolutional neural network," in *Biometrics (ICB), 2016 International Conference on*. IEEE, 2016, pp. 1–8.

[20] T. Wolf, M. Babaee, and G. Rigoll, "Multi-view gait recognition using 3d convolutional neural networks," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4165–4169.

[21] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Computer Vision and Image Understanding*, vol. 164, pp. 103–110, 2017.

[22] J. Han and B. Bhanu, "Individual recognition using gait energy image," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 316–322, 2006.

[23] Z. Liu and S. Sarkar, "Simplest representation yet for gait recognition: Averaged silhouette," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 4. IEEE, 2004, pp. 211–214.

[24] L. Wang, T. Tan, W. Hu, H. Ning *et al.*, "Automatic gait recognition based on statistical shape analysis," *IEEE transactions on image processing*, vol. 12, no. 9, pp. 1120–1131, 2003.

[25] A. Amir, B. Taba, D. J. Berg, T. Melano, J. L. McKinstry, C. Di Nolfo, T. K. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza *et al.*, "A low power, fully event-based gesture recognition system." in *CVPR*, 2017, pp. 7388–7397.

[26] P. K. Park, K. Lee, J. H. Lee, B. Kang, C.-W. Shin, J. Woo, J.-S. Kim, Y. Suh, S. Kim, S. Moradi *et al.*, "Computationally efficient, real-time motion recognition based on bio-inspired visual and cognitive processing," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 932–935.

[27] Y. Bi, A. Chadha, A. Abbas, E. Bourtsoulatze, and Y. Andreopoulos, "Graph-based object classification for neuromorphic vision sensing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 491–501.

[28] ——, "Graph-based spatio-temporal feature learning for neuromorphic vision sensing," *IEEE Transactions on Image Processing*, vol. 29, pp. 9084–9098, 2020.

[29] G. Gao, M. Kyrarini, M. Razavi, X. Wang, and A. Gräser, "Comparison of dynamic vision sensor-based and imu-based systems for ankle joint angle gait analysis," in *Frontiers of Signal Processing (ICFSP), International Conference on*. IEEE, 2016, pp. 93–98.

[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.

[31] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[32] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering,"

in *Advances in neural information processing systems*, 2016, pp. 3844–3852.
[33] M. Fey, J. Eric Lenssen, F. Weichert, and H. Müller, "Splinecnn: Fast geometric deep learning with continuous b-spline kernels," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 869–877.
[34] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5115–5124.
[35] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.
[36] A. Micheli, "Neural network for graphs: A contextual constructive approach," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 498–511, 2009.
[37] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 1993–2001.
[38] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *Acm Transactions On Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.
[39] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5199–5208.
[40] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1711–1719.
[41] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao, "Zero-shot video object segmentation via attentive graph neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 9236–9245.
[42] Y. Wang, X. Weng, and K. Kitani, "Joint detection and multi-object tracking with graph neural networks," *arXiv preprint arXiv:2006.13164*, 2020.
[43] X. Weng, Y. Wang, Y. Man, and K. Kitani, "Gnn3dmot: Graph neural network for 3d multi-object tracking with multi-feature learning," *arXiv preprint arXiv:2006.07327*, 2020.
[44] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912–7921.
[45] K. Lee, H. Woo, and T. Suk, "Point data reduction using 3d grids," *The International Journal of Advanced Manufacturing Technology*, vol. 18, no. 3, pp. 201–210, 2001.
[46] "Pointmatcher library tutorial," https://libpointmatcher.readthedocs.io/en/latest/#tutorials.
[47] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3693–3702.
[48] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.
[49] Y. Wang, B. Du, Y. Shen, K. Wu, G. Zhao, J. Sun, and H. Wen, "Ev-gait: Event-based robust gait recognition using dynamic vision sensors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6358–6367.
[50] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Ev-flownet: Self-supervised optical flow estimation for event-based cameras," *arXiv preprint arXiv:1802.06898*, 2018.
[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
[52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
[53] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[56] "https://inivation.com/support/hardware/dvs128/," *DVS128, Inivation*.
[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.
[58] L. Zhang, G. Zhu, P. Shen, J. Song, S. Afaq Shah, and M. Bennamoun, "Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 3120–3128.
[59] "Intel up board," https://up-board.org/.
[60] "Pytorch geometric documentation," https://pytorch-geometric.readthedocs.io/en/latest/.
[61] S. Yu, D. Tan, and T. Tan, "A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4. IEEE, 2006, pp. 441–444.
[62] T. H. Lam, K. H. Cheung, and J. N. Liu, "Gait flow image: A silhouette-based gait representation for human identification," *Pattern recognition*, vol. 44, no. 4, pp. 973–987, 2011.
[63] M. Goffredo, I. Bouchrika, J. N. Carter, and M. S. Nixon, "Self-calibrating view-invariant gait biometrics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 997–1008, 2010.
[64] K. Bashir, T. Xiang, and S. Gong, "Gait recognition using gait entropy image," 2009.
[65] W. Kusakunniran, Q. Wu, J. Zhang, and H. Li, "Gait recognition under various viewing angles based on correlated motion regression," *IEEE transactions on circuits and systems for video technology*, vol. 22, no. 6, pp. 966–980, 2012.
[66] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "Dvs benchmark datasets for object tracking, action recognition, and object recognition," *Frontiers in neuroscience*, vol. 10, p. 405, 2016.

Yanxiang Wang received the BS in Microelectronics from Jilin University, China and the MEng degrees in Computer Technology from Harbin Engineering University, China. His research interests are Moblie computing, wireless sensor networks and computer vision.

Xian Zhang received the bachelor of software engineering from Nanchang University, China, in 2013. He is a graduate student in College of Computer Science and Technology, Harbin Engineering University. Much of his research has focused on the Internet of Things (IoT), computer vision.

Yiran Shen is professor in School of Software, Shandong University. He received his BE in communication engineering from Shandong University, China and his PhD degree in computer science and engineering from University of New South Wales. He published regularly at top-tier conferences and journals. Generally speaking, his research interest is the merging area of Internet-of-Things (IoTs) and artificial intelligence. He is a senior member of IEEE.

Bowen Du received the BE and ME degrees in software engineering from Tongji University, Shanghai, China, in 2013 and 2016 respectively. He is currently working toward the PhD degree in computer science at the University of Warwick,Coventry, United Kingdom. His research interests include cyber physical systems, mobile computing, and artificial intelligence in sensor systems.

Guangrong Zhao received his bachelor degree from Wuhan University of Science and Technology. He is the author and co-author of several top papers on wireless sensor networks and computer vision, such as ACM TOSN, CVPR. His research interests include wireless sensor networks, federated learning, mobile computing, etc.

Lizhen Cui received Ph.D. MSc, and Bachelor degree from Shandong University in 2005, 2002, and 1999 respectively. He is a professor in the School of Software and Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR) of Shandong University, and also a visiting professor at Nanyang Technological University Singapore. He published over 100 papers in journals and refereed conference proceedings. His research interests include big data management and analysis.

Hongkai Wen is an Associate Professor in Department of Computer Science, University of Warwick. Before that he obtained his D.Phil at the University of Oxford, and became a post-doctoral researcher in a joint project between Oxford Computer Science and Robotics Institute. Broadly speaking, his research belongs to the area of Cyber-Physical Systems, which use networked smart devices to sense and interactive with the physical world.