SKELETON-BASED ACTION RECOGNITION USING FEATURE FUSION FOR SPATIAL-TEMPORAL GRAPH CONVOLUTIONAL NETWORKS

Dinh Tan Pham[1,3], *Thi Phuong Dang* [2], *Duc Quang Nguyen*[2],
Thi Lan Le[2,3], *Hai Vu*[2,3]

Abstract

Human action recognition (HAR) has been used in a variety of applications such as gaming, healthcare, surveillance, and robotics. Research on utilizing data such as color, depth, and skeletal data has been extensively conducted to achieve high-performance HAR. Compared with color and depth data, skeletal data are more compact, therefore, they are more efficient for computation and storage. Moreover, skeletal data are invariant to clothing textures, background, and lighting conditions. With the booming of deep learning, HAR has received a lot of attention. Spatial-Temporal Graph Convolution Networks (ST-GCN) have proved to be state-of-the-art architecture for HAR using skeleton data. However, this does not hold when working with challenging datasets that contain incomplete and noisy skeletal data. In this paper, a new method is proposed for HAR by adding a Feature Fusion module and applying hyperparameter optimization. The performance of the proposed method is evaluated on the challenging dataset CMDFALL and the newly-built MICA-Action3D dataset. Experimental results show that the proposed method significantly improves the performance of ST-GCN on these challenging datasets.

Index terms

Action recognition, graph convolutional network, skeletal data, feature fusion.

1. Introduction

Research on human action recognition (HAR) has been actively conducted in recent years. HAR is used in a wide number of applications such as robotics, gaming, surveillance, and healthcare. HAR focuses on predicting which action is being done by a person using data collected by sensors. Sensors can be either wearable sensors or ambient sensors. Wearable sensors include accelerometers and body-mounted cameras. Ambient sensors could be microphones and surveillance cameras. Input data for HAR may be color, depth, optical flows, or skeletal data. Research on utilizing these types

[1] Faculty of Information Technology, Hanoi University of Mining and Geology, Hanoi, Vietnam
[2] School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam
[3] Computer Vision Department, MICA Institute, Hanoi University of Science and Technology, Hanoi, Vietnam

of data is investigated in relevant works [1], [2]. However, HAR is still a difficult task due to the diversity of actions, inter-class similarity, and intra-class variance.

Human actions are naturally performed in a 3D space, which generates 3D skeleton data. The human skeleton can be modeled by joints connected in a certain order. Each action can be represented as the evolution of joints over time. Therefore, skeletal data are sequences of joint coordinates. Action representation using skeletal data is more compact when compared with color or depth data so HAR using skeletal data offers computation and storage efficiency. Moreover, skeletal data are reliable for HAR since they are invariant to the subject's appearance, background, and lighting. Early research of psychologists shows that skeleton data are informative to represent certain action classes [3]. Nowadays, skeletal data can be acquired directly by motion-capture/depth sensors or indirectly via applying pose estimation on videos. Skeletal data has become more accessible thanks to the adoption of low-cost depth sensors like Microsoft Kinect, as well as other efficient pose estimation methods. This makes skeleton-based HAR a popular approach. However, a problem that exists is inherent noise and incompleteness in skeletal data. The impact of this problem to HAR can be reduced by deploying data pre-processing techniques.

In terms of skeleton-based HAR, different methods have been proposed with promising results. Early methods on HAR use handcraft feature extraction, which results in limited performance and makes it difficult to generalize since those methods do not fully exploit spatial and temporal relationships among joints. Recently, Spatial-Temporal Graph Convolution Networks (ST-GCN) architecture is proposed to capture the spatial connections as well as the temporal evolution of joints [4]. In ST-GCN, each node in the graph maps to a joint in the skeletal model. There are two kinds of edges in this graph. Spatial edges are edges between naturally connected joints in the skeletal model. Temporal edges are edges that connect the same joints in different frames. Therefore, ST-GCN encodes relations between joints over spatial and temporal dimensions as graphs. The advantage of ST-GCN is that the relations between joints are naturally represented as graphs. Multiple graph convolutional layers are applied to extract the feature maps. The dynamics of human actions which are represented by the motion pattern of joints in the temporal dimension can be learned by ST-GCN. However, original works of ST-GCN [4] did not attempt to tackle the limitation of skeletal data such as noise and incompleteness. There exists inherent noise in skeletal data in most datasets in practice. ST-GCN comes with promising results on large datasets such as Kinetics and NTU-RGBD but this does not hold for challenging datasets with noise and incomplete skeletal data. Furthermore, absolute joint positions are deployed in the original ST-GCN framework. We realize that utilizing spatial and temporal joint offsets extracted from joint data can improve HAR performance in our recent work [5]. HAR using joint offsets is more robust to noise and incompleteness in skeletal data.

To handle the above issues, in this paper, a new framework is proposed to improve the performance of ST-GCN on challenging datasets using a Feature Fusion module and hyperparameter optimization. The Feature Fusion module combines joint offsets in both spatial and temporal dimensions. Hyperparameter optimization, which is the same as in [6], is applied to ST-GCN using a stochastic gradient descent (SGD) algorithm with optimized Nesterov momentum. The performance of the proposed method is evaluated on challenge datasets such as CMDFALL [7] and MICA-Action3D [8]. Experimental results indicate that the proposed method significantly improves the performance of the ST-GCN on the evaluation datasets. The paper is organized as follows. Section 2 reviews related works in HAR. Section 3 describes the proposed system, whereas experimental results are discussed in Section 4. Section 5 provides concluding remarks.

2. Related work

Different methods for feature engineering have been proposed in the literature for skeleton-based HAR. These methods can be divided into handcraft-based and deep learning-based methods.

In the handcraft-based approach, feature engineering is designed empirically to capture the evolution of joint motion. In [9], a graph is designed to encode the kinetics of actions. A bag of 3D points with equal distance on projection contours is utilized to model the human pose for each frame. An ensemble model is introduced in [10] by grouping joints into different subsets. Relative Joint Positions (RJP) are defined as the offsets from the joint positions to the center joint. Fourier Temporal Pyramid (FTP) is designed to handle noise in skeletal data. In [11], features are concatenated to create feature vectors. Principal Component Analysis is then implemented to extract EigenJoints. In [12], the Cov3DJ captures dependence between joints using a covariance matrix of joint positions. Covariance matrices are calculated over temporal windows of different sizes, which creates a temporal hierarchy. In [13], joints are grouped into five groups (spine, two arms, and two legs) and covariance matrices are computed on 3D joint coordinates (CovP3DJ). The CovP3DJ is reported to be efficient in both spatial and temporal dimensions. In [14], the Lie group theory is applied for skeleton-based pose representation. The human pose is structured as the combination of relative positions among bones. For each pair of bones, there exists a transformation matrix to convert from one bone segment to the other. Transformations between bones are represented as elements of $SE(3) \times SE(3) \times ... \times SE(3)$ group. The human pose in each frame is an item of a Lie group. Due to the manifold nature of the Lie group, interpolation between elements can not be directly applied. The logarithmic function is applied to map the Lie group to its algebra $se(3) \times se(3) \times ... \times se(3)$. In [15] and [16], kinetic features are concatenated to form feature vectors. In a different scheme, HAR methods using joint subsets are proposed. Joints are selected manually or automatically. In [17], the Sequence of Most Informative Joints (SMIJ) use joints that engage most in actions. Five joints with the largest variance of joint angles are automatically selected for each action. In [18], CovMIJ is proposed to take advantage of SMIJ and Cov3DJ. Most informative joints (MIJ) are automatically selected using the position variance as the statistical measure. This offers efficient computation and reduces the impact of noise in skeletal data. In [8], the Covariance Descriptor-based Adaptive Most Information

Joints (CovAMIJ) method is proposed. The method defines the selection of joints using covariance matrices. Joint position and velocity are combined for action representation.

In the rise of representation learning, different network architectures have been proposed for HAR such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), and Graph Convolutional Network (GCN). Features are learned directedly from input data. RNN proves itself as a good representation in the time domain while LSTM can explore long-term dependency in time sequences [19]. CNN is a classic method in image classification with efficient grid modeling. In [20], a two-stream ConvNet architecture is proposed as a two-stream network. In [21], an architecture combining LSTM and CNN is proposed for HAR. The standard backpropagation algorithm is applied to tackle gradient descent. In [22], Temporal Convolutional Neural Network with residuals (Res-TCN) is proposed, which can explicitly learn interpretable spatio-temporal representations for HAR. In [23], a CNN-LSTM network is proposed to utilize CNN and LSTM to exploit spatial and temporal information, respectively. In [7], HAR is preliminarily evaluated on the CMDFALL dataset using Res-TCN deep learning architecture. The CMDFALL dataset focuses on human falling actions to simulate scenes for elderly monitoring in healthcare. In [24], a Richly Activated - Graph Convolutional Network (RA-GCN) is proposed to enhance the robustness of action recognition models on incomplete and noisy skeletal data. Each stream learns features from currently unactivated joints, which are masked by class activation maps obtained by preceding streams. In [25], joint and bone information is utilized in a two-stream framework. The framework is proposed to learn graph topology adaptively for different GCN layers. In [26], graph convolution is performed on graph edges to explore relations between different bones, as well as temporal neighboring edges. Two neural networks are constructed to handle graph nodes and graph edges using a shared dense layer. In [27], graph regression is proposed to learn the graph from different observations. Optimization is performed on graph structure over consecutive frames using spatio-temporal modeling of skeletons. This helps enforce the sparsity of graphs for simple representation. In [6], a multi-stream adaptive graph convolutional neural network is proposed using joint, bone, and motion information. A directed graph neural network is designed specially to extract features for prediction. Graph topology is made adaptive based on the training process. Motion information is exploited and combined with spatial information to enhance the performance of a two-stream framework. Hyperparameter optimization is introduced using stochastic gradient descent (SGD) with Nesterov momentum.

3. Proposed system

ST-GCN achieves good performance on large datasets such as Kinetics and NTU-RGBD [4]. However, this does not hold true for incomplete and noisy datasets [24]. ST-GCN uses joint positions as input data. A diagram of the proposed system is shown in Fig. 1. A Feature Fusion module is added to generate new input data to ST-GCN using RJP and joint velocity. As each sample has a different number of frames, all samples are normalized to the same length by zero paddings.

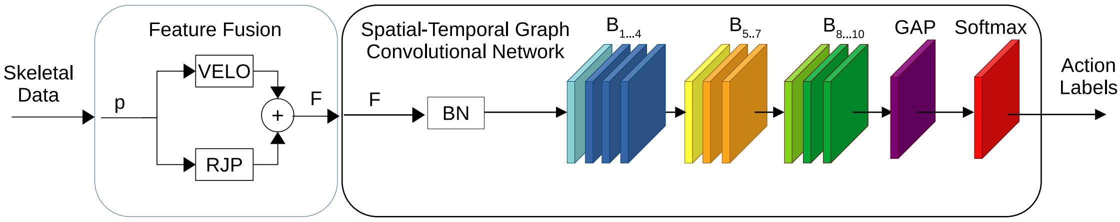


Fig. 1. The proposed system.

A Batch Normalization (BN) layer is applied at the beginning of ST-GCN to normalize data. A stack of basic blocks with the same order as in [25] is added. There are ten basic blocks in the proposed system, namely $B_1, B_2, \ldots, B_{10}$. The first four blocks have 64 channels, the next three blocks have 128 channels and the last three blocks have 256 channels. A global average pooling (GAP) layer is added to form feature maps into the same size. Data are finally passed through a softmax classifier.

A diagram of each basic block is shown in Fig. 2. The basic block consists of a Spatial GCN (Convs), a BN, a ReLU, a Dropout, a temporal GCN (Convt), a BN, and a ReLU layer. Random dropout with a drop rate of 0.5 is applied to prevent overfitting. A residual link is added to stabilize the training.

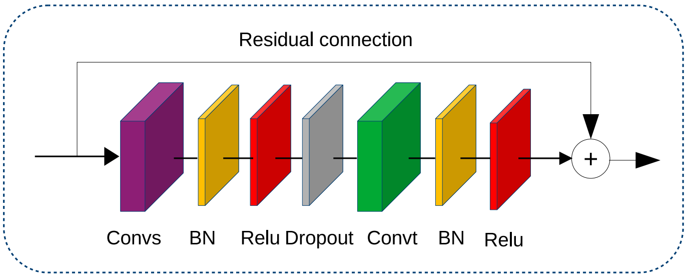


Fig. 2. Spatial-Temporal basic block.

3.1. Feature Fusion

RJP and joint velocity, as well as their combination, are well-established features for HAR using handcraft feature extraction [10], [16]. Joint offsets are invariant to translation and view changes, so HAR using joint offsets is more robust than using the absolute joint positions. For instance, the *high arm wave* action is better described by relative positions from joints to the center joint in the skeletal model rather than the absolute positions of joints in 3D space. For actions such as *hammer* and *hand catch* in the MICA-Action3D dataset, there are many similar poses so these actions are of high similarity. It is worth noticing that the velocity of the right hand in the *hammer* action is faster than in the *hand catch* action when moving down. So joint velocity is also an important feature. In this work, RJP and joint velocity are combined in the Feature Fusion module for data pre-processing before feeding the data to ST-GCN.

Coordinates of the i^{th} joint in frame t can be expressed as:

$$p_{\mathrm{i}}(t) = [x_{\mathrm{i}}(t), y_{\mathrm{i}}(t), z_{\mathrm{i}}(t)] \tag{1}$$

The human skeleton at time frame t composes of N joints:

$$p(t) = [p_0(t), p_1(t), \ldots, p_{\mathrm{N-1}}(t)] \tag{2}$$

Relative Joint Positions (RJP) are defined as the offsets between joints to the center joint p_c of the skeletal model as shown in Fig. 3. The middle spine joint in the skeletal model is selected as the center joint p_c. RJP can be mathematically expressed as:

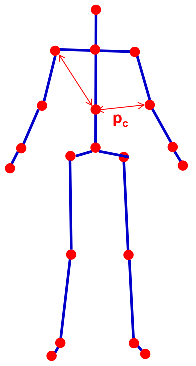


Fig. 3. RJPs are defined as spatial offsets between joints to the center joint.

$$RJP_i(t) = p_i(t) - p_c(t) \tag{3}$$

with $i = 0, 1, \ldots, N-1$.

Motivated by the bio-mechanic-based method in [16], we take joint velocity $VELO(t)$ as a feature to represent human actions. These can be seen as the first-order derivatives of joint positions. The velocity of the i^{th} joint at frame t is defined as:

$$VELO_i(t) = p_i(t+2) - p_i(t) \tag{4}$$

with $i = 0, 1, \ldots, N-1$.

Joint velocity in the last two frames is set equal to their neighboring frame. Feature vectors F are created by combining $RJP(t)$ with $VELO(t)$:

$$F(t) = [RJP(t), VELO(t)] \tag{5}$$

Output data of feature fusion module are matrices of size $2C \times T \times N \times M$, whereas:

- C is the number of joint coordinate dimensions (aka. the number of channels). The value is three for dimensions x, y, z.
- T is the maximum number of frames for action representation in each dataset. $T = 600$ for CMDFALL and $T = 175$ for MICA-Action3D. Skeletal sequences that are shorter than T are padded by zeros to the same length T for each dataset.
- N is the joint quantity in the skeletal model. $N = 20$ for both CMDFALL and MICA-Action3D.
- M is the maximum number of persons in each frame. $M = 1$ for both CMDFALL and MICA-Action3D.

In our proposed framework, input data to ST-GCN are the combination of RJP and joint velocity, not absolute joint positions as in the original ST-GCN scheme.

3.2. Spatial-Temporal Graph Convolutional Network

ST-GCN is a deep learning network that processes graph-structured data to output labels. The graph is used as a representation of the skeletal sequences. An undirectional graph $G = (V, E)$ is constructed on a skeletal sequence with N joints and T frames to represent both intra-frame and inter-frame links. In this graph, joints in each skeletal sequence are included in the node set $V = \{v_{ti} \mid t = 1, \ldots, T, i = 1, \ldots, N\}$. Inputs to ST-GCN are absolute joint coordinates. The graph is naturally defined in two steps. Firstly, joints in each frame are connected to form spatial edges according to the natural connectivity of the skeletal model. Denote natural intra-frame connections in each frame as $E_S = \{v_{ti}v_{tj} \mid (i, j) \in H\}$, where H is the set of connected joint pairs. Secondly, each joint is connected to the same joint in consecutive frames to form temporal edges $E_T = \left\{v_{ti}v_{(t+1)i}\right\}$. This graph enables ST-GCN to work with various skeletal models.

The convolution operation for the graph is extended from the convolution operation for 2D images. Output feature in image convolution is a 2D grid with the same size as the input using stride one and padding. For a convolutional with a kernel size of $K \times K$, input feature maps f_{in} and number of channel C, output value at spatial position $\mathbf{x}$ can be expressed as [4]:

$$f_{\text{out}}(\mathbf{x}) = \sum_{h=1}^{K} \sum_{w=1}^{K} f_{\text{in}}(\mathbf{p}(\mathbf{x}, h, w)) \cdot \mathbf{w}(h, w) \tag{6}$$

where p is the sampling function and w is the weight function. For image convolution, sampling function can be written as $\mathbf{p}(\mathbf{x}, h, w) = \mathbf{x} + \mathbf{p}'(h, w)$. As the input location $\mathbf{x}$ has no bearing on the weight function, weights are applied to all pixels on the image. Convolution operation on graphs is defined by extending the above formulation on feature maps of the graph V_t. The feature map is denoted as $f_{in}^{t} : V_t \to R^c$.

For images, the sampling function $\mathbf{p}(h, w)$ for each pixel is defined based on its neighboring pixels. For graphs, the sampling function of a node v_{ti} is defined on the neighbor set $B\left(v_{ti}\right)=\left\{v_{tj} \mid d\left(v_{tj}, v_{ti}\right) \leq D\right\}$ where $d\left(v_{tj}, v_{ti}\right)$ denotes the shortest path from v_{tj} to v_{ti}. The sampling function can be written as:

$$\mathbf{p}\left(v_{ti}, v_{tj}\right)=v_{tj} \tag{7}$$

In image convolution, a natural grid exists for each pixel. Neighboring pixels have a fixed spatial order. The weight function is a tensor of dimensions (c, K, K). There is no such implicit arrangement for graphs. In [28], the order is assigned by labeling around the root node to construct the weight function. Instead of unique labeling to all neighbor nodes, the neighbor set $B(v_{ti})$ of v_{ti} is partitioned into K subsets. A numeric label is assigned to each subset. Weight function can be expressed as:

$$\mathbf{w}\left(v_{ti}, v_{tj}\right)=\mathbf{w}^{\prime}\left(l_{ti}\left(v_{tj}\right)\right) \tag{8}$$

With the above sampling function and weight function, spatial graph convolution can be expressed as:

$$f_{\text{out}}\left(v_{ti}\right)=\sum_{v_{tj} \in B\left(v_{ti}\right)} \frac{1}{Z_{ti}\left(v_{tj}\right)} f_{in}\left(\mathbf{p}\left(v_{ti}, v_{tj}\right)\right) \cdot \mathbf{w}\left(v_{ti}, v_{tj}\right) \tag{9}$$

where $Z_{ti}\left(v_{tj}\right)$ is the normalizing term included to balance the contribution of different subsets to output. For temporal graph modeling, convolution is computed for every single joint along with all frames in skeletal sequence same as in [4]. Let Γ be the temporal kernel size, temporally connected joints are in the neighbor node set:

$$B\left(v_{ti}\right)=\left\{v_{qj}\left|d\left(v_{tj}, v_{ti}\right) \leq K,\right| q-t \mid \leq\lfloor\Gamma / 2\rfloor\right\} \tag{10}$$

Label map for the neighborhood of v_{ti} is defined as:

$$l_{ST}\left(v_{qj}\right)=l_{ti}\left(v_{tj}\right)+(q-t+\lfloor\Gamma / 2\rfloor) \times K \tag{11}$$

where $l_{ti}\left(v_{tj}\right)$ is label mapping for v_{ti}. Spatial edges are represented by adjacent matrix $\mathbf{A}$. ST-GCN can be implemented using the formula [29]:

$$\mathbf{f}_{\text{out}}=\mathbf{\Lambda}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I}) \mathbf{\Lambda}^{-\frac{1}{2}} \mathbf{f}_{\text{in}} \mathbf{W} \tag{12}$$

where $\Lambda^{ii}=\sum_{j}\left(A^{ij}+I^{ij}\right)$. Identity matrix I is added to the adjacent matrix to include self joint connections.

3.3. Hyperparameter optimization

Experiments are conducted using the PyTorch framework. In this paper, hyperparameter optimization same as in [6] is applied. A stochastic gradient descent algorithm with Nesterov momentum is used as the optimization algorithm. Loss function using cross-entropy is used to back-propagate gradients. The learning rate is 0.1 and it is divided by 10 at the thirtieth and fortieth epoch. The setting value of weight decay is 0.0001. Learning rate warm-up is applied for the first five epochs. The model is trained in 50 epochs. Details on hyperparameter optimization are summarized in Table 1.

Table 1. Hyperparameter optimization

No.	Hyperparameter	ST-GCN	Proposed
1	adjust_lr	[10, 30]	[30, 40]
2	warm_up_epoch	None	5

4. Experiment and evaluation

4.1. Evaluation datasets

The proposed method is evaluated on the challenging dataset CMDFALL [7] and our newly-built dataset MICA-Action3D [8]. On both datasets, half subjects are used for training, while the other half is used for testing.

CMDFALL dataset: The dataset is built to evaluate algorithms to detect human falling action in healthcare applications such as elderly monitoring [7]. Elderly falling detection may help alert family members or medical staff to offer medical treatment in time. There are seven Kinect sensors used as ambient sensors. Each subject wears two wireless accelerometers. There are 20 action classes, performed by 50 subjects with ages ranging from 21 to 40. The subjects include 30 males and 20 females. The list of actions in the CMDFALL dataset is shown in Table 2. In this paper, evaluation is performed on data from Kinect view 3 as recommended by the authors of the dataset [7]. Skeletal data from Kinect view 3 contains 1,963 action samples. As the CMDFALL dataset focuses on falling actions, there exists serious noise in skeletal data. The reason is that the Kinect sensor is designed for video gaming so its skeleton tracking algorithm only works well for subjects in standing poses. CMDFALL focuses on falling actions with many different non-standing poses.

Table 2. List of actions in CMDFALL dataset

Action ID	Action name	Action ID	Action name
1	walk	11	right fall
2	run slowly	12	crawl
3	static jump	13	sit on chair then stand up
4	move hand and leg	14	move chair
5	left hand pick up	15	sit on chair then fall left
6	right hand pick up	16	sit on chair then fall right
7	stagger	17	sit on bed and stand up
8	front fall	18	lie on bed and sit up
9	back fall	19	lie on bed and fall left
10	left fall	20	lie on bed and fall right

MICA-Action3D dataset: The dataset data are collected by a Kinect sensor [8]. The dataset is built by ourselves based on the list of 20 action classes in MSR-Action3D [9] as shown in Table 3. These actions are the interactions between humans with game consoles. There are 20 subjects, each subject performs one action two or three times.

There are 1,196 action samples in total. A sample frame from the MICA-Action3D dataset is shown in Fig. 4 with color, depth, and skeletal data.

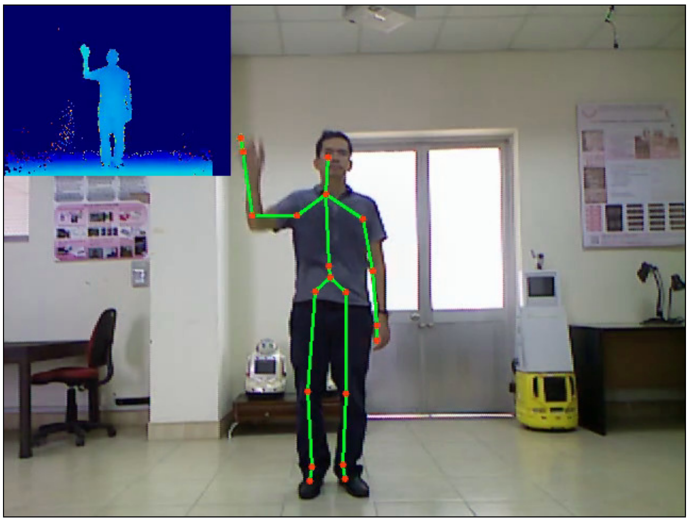


Fig. 4. Sample frame in MICA-Action3D.

Table 3. List of actions in MICA-Action3D dataset

Action ID	Action name	Action ID	Action name
1	high arm wave	11	two-hand wave
2	horizontal arm wave	12	side boxing
3	hammer	13	bend
4	hand catch	14	forward kick
5	forward punch	15	side kick
6	high throw	16	jogging
7	draw X	17	tennis swing
8	draw tick	18	tennis serve
9	draw circle	19	golf swing
10	hand clap	20	pick-up and throw

4.2. Experimental results

As shown in Table 4, performance is significantly improved by introducing a Feature Fusion module for data pre-processing. Our proposed method achieves an F1-score of up to 70.68% on the CMDFALL dataset while the F1-score of ST-GCN is only 51.16%. The confusion matrix on the CMDFALL dataset is shown in Fig. 5. Confusion on the CMDFALL dataset mainly happens to *right fall* and *left fall* actions. In these actions, subjects lie on the ground after falling so serious noise occurs in skeletal data. As shown in Fig. 7, skeletal data contains serious noise inherently for non-standing human poses. Such serious noise degrades action recognition performance. Besides, an action can be observed from different views and directions to the Kinect sensor. For instance,

Table 4. Performance evaluation on CMDFALL dataset

No.	Method	Precision (%)	Recall (%)	F1 (%)
1	Res-TCN, CVPRW 2017 [22]	-	-	39.38
2	CNN, 2019 [19]	48.68	41.78	40.34
3	CNN-LSTM, 2018 [23]	45.24	40.58	39.24
4	CNN-Velocity, MAPR 2019 [19]	49.97	47.89	46.13
5	CNN-LSTM-Velocity, MAPR 2019 [19]	47.64	46.51	45.23
6	RA-GCN, ICIP 2019 [24]	61.18	59.28	58.63
7	CovMIJ, KSE 2018 [18]	-	-	62.5
8	CovAMIJ, MTAP 2021 [8]	-	-	64
9	ST-GCN, AAAI 2018 [4]	52.33	53.99	51.16
10	**Proposed**	**72.05**	**70.57**	**70.68**

Table 5. Performance evaluation on MICA-Action3D dataset

No.	Method	Precision (%)	Recall (%)	F1 (%)
1	ST-GCN, AAAI 2018 [4]	83.64	83.41	82.82
2	**Proposed**	**96.70**	**96.65**	**96.62**

recognizing action by left hand or right hand can easily be confused. Similarly, the definition of *left/right* in *right fall* and *left fall* is to the subject, not to Kinect sensor's viewpoint. It can be seen that the CNN-LSTM-Velocity method only achieves an F1 score of 45.23% on CMDFALL. To apply CNN to HAR, the skeletal data are converted into images. Some relations between joints in the skeletal structure are lost during this conversion process. ST-GCN-based methods show superior performance to CNN-based methods since ST-GCN can represent the skeletal structure as graphs, not as images for CNN-based methods. To further confirm the robustness of the proposed method, evaluation is performed on our newly-built dataset MICA-Action3D as shown in Table 5. MICA-Action3D is constructed based on the actions defined in MSR-Action3D [9]. The confusion matrix on the MICA-Action3D dataset is shown in Fig. 6. F1-scores of ST-GCN and the proposed method are 82.82% and 96.62%, respectively. As can be seen in the confusion matrix of MICA-Action3D, recognition results are comparatively high among all action classes. The reason is that action classes in MICA-Action3D are more discriminate than in CMDFALL. By using the proposed method, the inter-class similarity can be resolved. As shown in Fig. 9, similar actions such as *high arm wave*, *draw X*, *draw circle* can be separated. Also, some actions with noise in MICA-Action3D as shown in Fig. 8 still achieves good performance. As shown from the confusion matrix for MICA-Action3D, the accuracy rate is 100% for *bend* and *tennis swing* actions, while it is 92% for *high arm wave*. In terms of handling noise issues on skeletal data, the CMDFALL dataset mainly focuses on falling actions so noise in skeletal data is very serious, which leads to poor classification results. It requires more investigation on correcting skeletal data on actions with the complex shape of the skeleton model. This suggests future research direction. The proposed method includes feature fusion between joint velocity and RJP with hyperparameter optimization to ST-GCN. To examine the contribution of every single step in the proposed method, evaluation is performed on

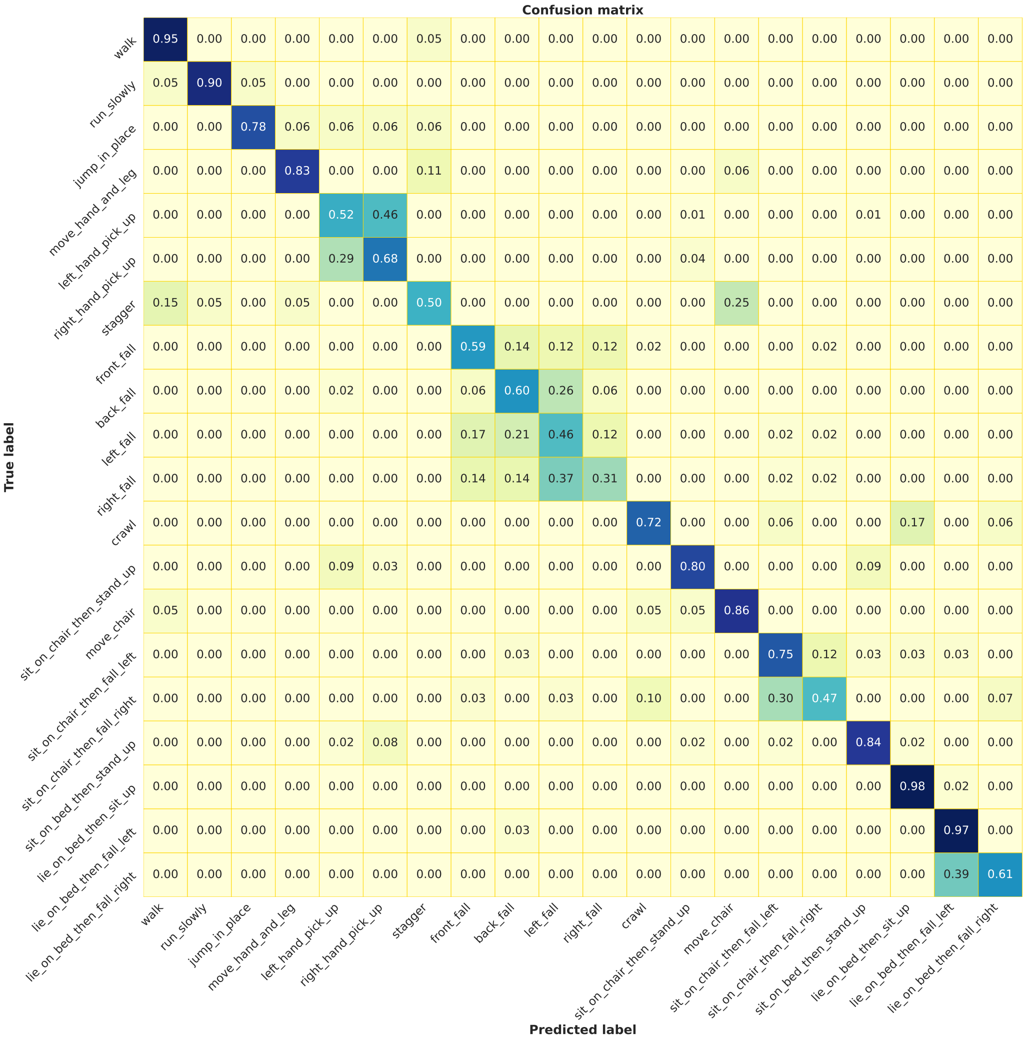
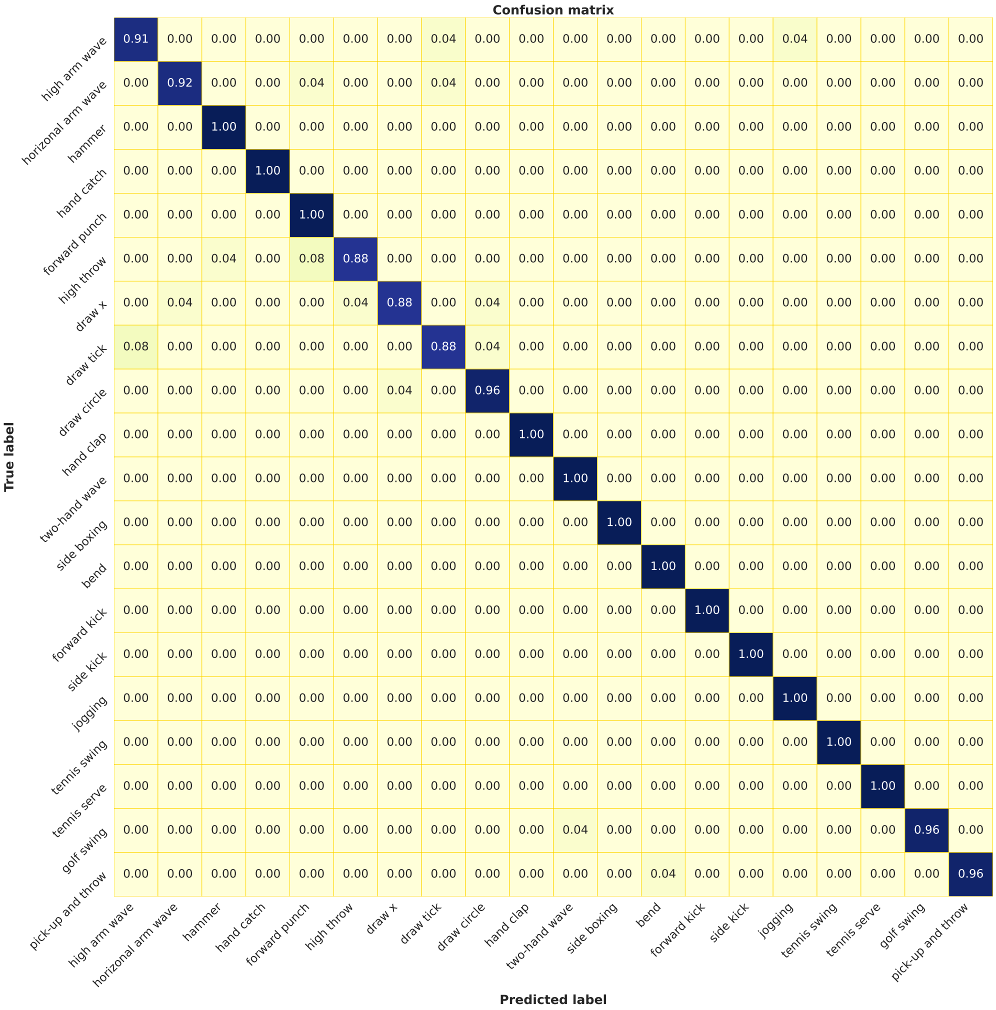


Fig. 5. Confusion matrix on CMDFALL dataset.

the CMDFALL dataset by applying every single step separately as shown in Table 6. For the original ST-GCN using absolute joint positions, the F1 score is 51.16%. By using RJP and joint velocity separately instead of absolute joint positions, the F1 scores are 53.05% and 54.87%, respectively. By fusing RJP and joint velocity, the system achieves an F1 score of 59.47%. It can be seen that hyperparameter optimization also plays an important role in the proposed method. However, introducing hyperparameter optimization alone to ST-GCN achieves an F1-score of 55.05% while combining with feature fusion achieves an F1-score of up to 70.68%.

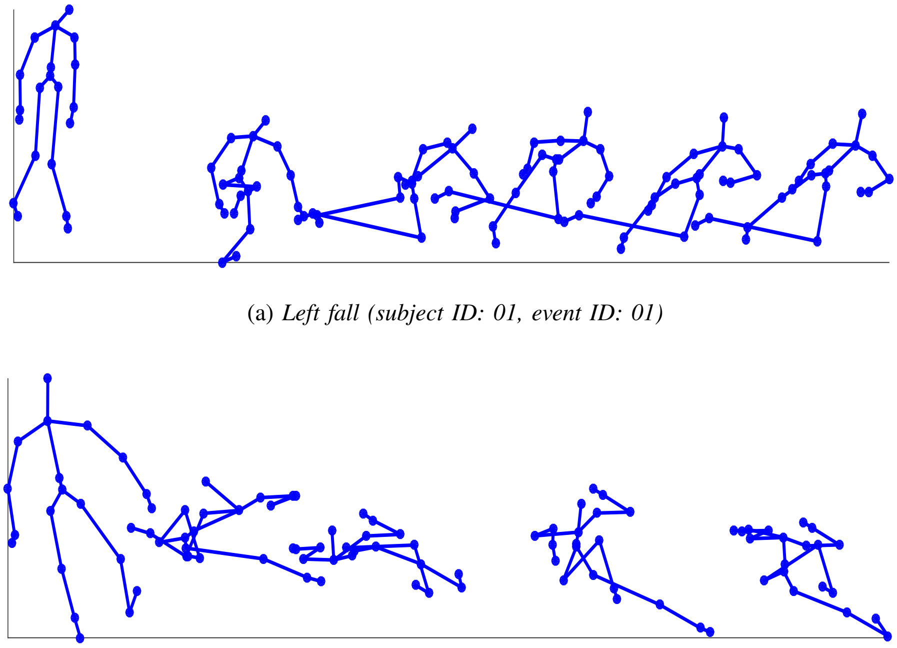


	high arm wave	horizonal arm wave	hammer	hand catch	forward punch	high throw	draw x	draw tick	draw circle	hand clap	two-hand wave	side boxing	bend	forward kick	side kick	jogging	tennis swing	tennis serve	golf swing	pick-up and throw
high arm wave	0.91	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00
horizonal arm wave	0.00	0.92	0.00	0.00	0.04	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hammer	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hand catch	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
forward punch	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
high throw	0.00	0.00	0.04	0.00	0.08	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
draw x	0.00	0.04	0.00	0.00	0.00	0.04	0.88	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
draw tick	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.88	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
draw circle	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
hand clap	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
two-hand wave	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
side boxing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bend	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
forward kick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
side kick	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
jogging	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
tennis swing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
tennis serve	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
golf swing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00
pick-up and throw	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.96

Fig. 6. Confusion matrix on MICA-Action3D dataset.

Table 6. Performance when applying every single step in the proposed method on CMDFALL

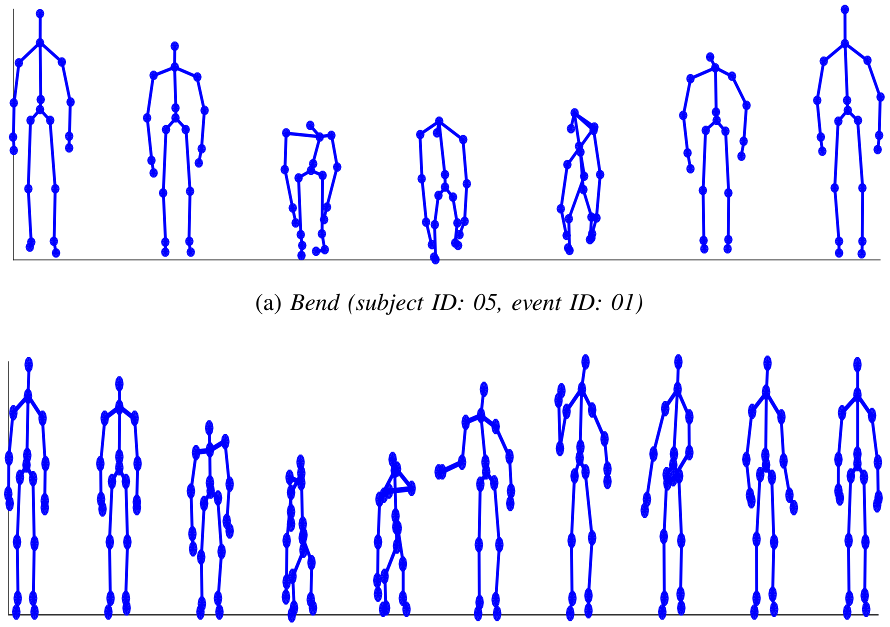
No.	Method	Precision (%)	Recall (%)	F1 (%)
1	ST-GCN + Absolute joint positions	52.33	53.99	51.16
2	ST-GCN + Joint velocity	58.27	54.13	54.87
3	ST-GCN + RJP	55.76	54.22	53.05
4	ST-GCN + Hyperparameter optimization	56.56	57.08	55.05
5	ST-GCN + RJP + Joint velocity	61.23	60.35	59.47
6	**Proposed**	**72.05**	**70.57**	**70.68**



(a) *Left fall (subject ID: 01, event ID: 01)*

(b) *Right fall (subject ID: 08, event ID: 01)*

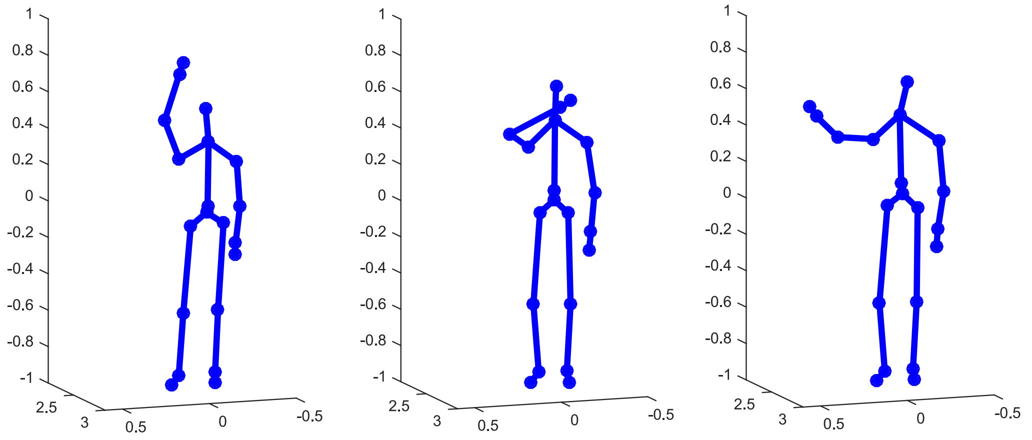
Fig. 7. Serious noise in left fall and right fall actions in CMDFALL.



(a) *Bend (subject ID: 05, event ID: 01)*

(b) *Pick-up and throw (subject ID: 06, event ID: 01)*

Fig. 8. Noise in different actions in MICA-Action3D.



(a) *High arm wave*

(b) *Draw X*

(c) *Draw circle*

Fig. 9. Sample frame of (a) high arm wave, (b) draw X, and (c) draw circle in MICA-Action3D.

Experiments are implemented on a server with an Intel i7-8700 CPU, 32 GB memory, and a GeForce GTX 1080Ti GPU. Time consumption for training and testing is shown in Table 7. The proposed method outperforms the baseline method ST-GCN while time consumption for training/testing is of the same order. The reason is that by using RJP and joint velocity features, computation is mainly performed on sparse matrices so the computation is even more efficient than using the absolute joint positions. The training time required for ST-GCN on CMDFALL is 1,067 seconds while it is only 1,033 seconds for the proposed method.

Table 7. Time consumption for training and testing

Dataset	**Training time (s)**		**Testing time (s)**		**Number of samples**	**Testing time/sample (ms)**	
	ST-GCN	Proposed	ST-GCN	Proposed		ST-GCN	Proposed
CMDFALL	1,067	1,033	7.4	7.2	792	9	9
MICA-Action3D	244	217	4.4	4.4	478	9	9

5. Conclusions and future works

In this paper, an improved architecture of ST-GCN was proposed by introducing a Feature Fusion module for data pre-processing and applying hyperparameter optimization. The Feature Fusion module combines RJP and joint velocity as input data. Experiments on two evaluation datasets show that the proposed method achieves better performance than ST-GCN on challenging datasets. Future work might focus on noise reduction and self-correction in data pre-processing. The proposed framework will also be evaluated on other challenging datasets. We will further investigate the performance of the proposed method with skeleton data estimated from RGB image sequences.

Acknowledgment

This research is funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2017.12.

References

[1] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey," *Image and vision computing*, vol. 60, pp. 4–21, 2017.

[2] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *preprint arXiv:1806.11230*, 2018.

[3] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[4] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *arXiv preprint arXiv:1801.07455*, 2018.

[5] D.-T. Pham, T.-N. Nguyen, T.-L. Le, and H. Vu, "Spatio-temporal representation for skeleton-based human action recognition," in *2020 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6, IEEE, 2020.

[6] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with multi-stream adaptive graph convolutional networks," *arXiv preprint arXiv:1912.06971*, 2019.

[7] T.-H. Tran, T.-L. Le, D.-T. Pham, V.-N. Hoang, V.-M. Khong, Q.-T. Tran, T.-S. Nguyen, and C. Pham, "A multi-modal multi-view dataset for human fall analysis and preliminary investigation on modality," in *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1947–1952, IEEE, 2018.

[8] V.-T. Nguyen, T.-N. Nguyen, T.-L. Le, D.-T. Pham, and H. Vu, "Adaptive most joint selection and covariance descriptions for a robust skeleton-based human action recognition," *Multimedia Tools and Applications (MTAP)*, pp. 1–27, 2021.

[9] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3D points," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 9–14, IEEE, 2010.

[10] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1290–1297, IEEE, 2012.

[11] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using Naive-Bayes-Nearest-Neighbor," in *Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 14–19, IEEE, 2012.

[12] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pp. 2466–2472, AAAI Press, 2013.

[13] H. A. El-Ghaish, A. Shoukry, and M. E. Hussein, "CovP3DJ: Skeleton-parts-based-covariance descriptor for human action recognition.," in *VISIGRAPP (5: VISAPP)*, pp. 343–350, 2018.

[14] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a Lie group," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 588–595, 2014.

[15] L. Zhou, W. Li, Y. Zhang, P. Ogunbona, D. T. Nguyen, and H. Zhang, "Discriminative key pose extraction using extended LC-KSVD for action recognition," in *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2014.

[16] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche, "3D real-time human action recognition using a spline interpolation approach," in *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 61–66, IEEE, 2015.

[17] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 8–13, June 2012.

[18] T.-N. Nguyen, D.-T. Pham, T.-L. Le, H. Vu, and T.-H. Tran, "Novel skeleton-based action recognition using covariance descriptors on most informative joints," in *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 50–55, IEEE, 2018.

[19] V.-N. Hoang, T.-L. Le, T.-H. Tran, V.-T. Nguyen, *et al.*, "3D skeleton-based action recognition with convolutional neural networks," in *2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, pp. 1–6, IEEE, 2019.

[20] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, pp. 568–576, 2014.

[21] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3054–3062, 2016.

[22] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in *Conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1623–1631, IEEE, 2017.

[23] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio-temporal visual attention on skeleton image sequences," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[24] Y.-F. Song, Z. Zhang, and L. Wang, "Richly activated graph convolutional network for action recognition with incomplete skeletons," in *International Conference on Image Processing (ICIP)*, pp. 1–5, IEEE, 2019.

[25] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, 2019.

[26] X. Zhang, C. Xu, X. Tian, and D. Tao, "Graph edge convolutional neural networks for skeleton-based action recognition," *IEEE transactions on neural networks and learning systems*, 2019.

[27] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo, "Optimized skeleton-based action recognition via sparsified graph regression," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 601–610, 2019.

[28] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*, pp. 2014–2023, 2016.

[29] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

Manuscript received 06-10-2020; Accepted 09-05-2021.

■

Dinh Tan Pham received M.Eng. degree in Electrical Engineering from Chulalongkorn University in 2005 and B.Eng. degree in Electronics and Telecommunications from Hanoi University of Science and Technology (HUST) in 2003. He is a lecturer in the Faculty of IT, Hanoi University of Mining and Geology. He is doing a Ph.D. at HUST. His research interests include computer vision, deep learning, and robotics. E-mail: phamdinhtan@humg.edu.vn

Thi Phuong Dang graduated from HUST, Hanoi, Vietnam in 2020 with a B.Eng in Electronics and Telecommunications. She is currently an integration engineer at Ericsson. E-mail: dangphuong13997@gmail.com

Duc Quang Nguyen received B.Eng. degree in Electronics and Telecommunications from HUST, Hanoi, Vietnam in 2020. He is currently a Digital Integrated Circuit Design engineer at Dolphin Technology Vietnam Center. E-mail: nguyenquang.hy.97@gmail.com

Thi Lan Le received her Ph.D. degree at INRIA Sophia Antipolis, France in video retrieval in 2009. She is currently a lecturer/researcher in HUST, Hanoi, Vietnam. Her research interests include computer vision, content-based indexing and retrieval, video understanding, and human-robot interaction. E-mail: thi-lan.le@mica.edu.vn

Hai Vu received BE degree in Electronics and Telecommunications in 1999 and ME degree in Information Processing and Communication in 2002, both from HUST, Hanoi, Vietnam. He received PhD degree in Computer Science from Osaka University, Japan, in 2009. He has been a lecturer/researcher in HUST, Hanoi, Vietnam since 2012. His current research interests are in Computer Vision, Pattern Recognition, particularly, applying these techniques in Agricultural Engineering, Medical Imaging and Human-Computer Interactions. E-mail: hai.vu@mica.edu.vn

NHẬN DẠNG HOẠT ĐỘNG DỰA TRÊN KHUNG XƯƠNG SỬ DỤNG KẾT HỢP CÁC ĐẶC TRƯNG CHO MẠNG TÍCH CHẬP ĐỒ THỊ KHÔNG GIAN-THỜI GIAN

Phạm Đình Tân, Đặng Thị Phượng, Nguyễn Đức Quang, Lê Thị Lan, Vũ Hải

Tóm tắt

Kỹ thuật nhận dạng hoạt động người (HAR) là một bài toán được ứng dụng rộng rãi trong nhiều lĩnh vực như trò chơi, y tế, giám sát và điều khiển rô-bốt. Nhiều nghiên cứu về nhận dạng hoạt động người đã được đề xuất. Các phương pháp này tập trung vào khai thác dữ liệu ảnh màu, ảnh độ sâu và khung xương nhằm nâng cao hiệu năng nhận dạng hoạt động. So với ảnh màu và ảnh độ sâu, dữ liệu khung xương thường nhỏ gọn, do đó hiệu quả hơn trong tính toán và lưu trữ. Ngoài ra, dữ liệu khung xương bất biến với sự thay đổi về trang phục của người thực hiện hoạt động, môi trường xung quanh và điều kiện chiếu sáng. Cùng với sự bùng nổ của kỹ thuật học sâu, các mạng tích chập đồ thị không gian-thời gian (ST-GCN) cho thấy hiệu quả trong biểu diễn và nhận dạng hoạt động dựa trên khớp xương. Tuy nhiên, khi làm việc trên các dữ liệu thách thức như chứa nhiều nhiễu, thiếu thông tin, hiệu quả của phương pháp ST-GCN giảm đi đáng kể. Trong bài báo này, một phương pháp mới được đề xuất dựa trên ST-GCN cho nhận dạng hoạt động sử dụng kết hợp các đặc trưng và tối ưu các siêu tham số. Hiệu năng của phương pháp đề xuất được đánh giá trên tập dữ liệu có nhiều nhiễu là CMDFALL và tập dữ liệu MICA-Action3D. Kết quả cho thấy phương pháp đề xuất có hiệu năng tốt hơn ST-GCN trên các tập dữ liệu thử nghiệm.