Optimizing Features Quality: A Normalized Covariance Fusion Framework for Skeleton Action Recognition

GUAN HUANG AND QIUYAN YAN
School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
Corresponding author: Qiuyan Yan (yanqy@cumt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61977061 and Grant 51934007.

ABSTRACT Action recognition based on 3D skeleton sequences has gained considerable attention in recent years. Due to effectively representing the spatial and the temporal characters of skeleton sequences, the Covariance Matrix (CM) features combined with the Long Short-Term Memory (LSTM) network is an effective and reasonable roadmap to enhance the action recognition accuracy. However, the CM features in the existing recognition models are computed from the raw data without normalization or with static normalization. Moreover, a CM feature is calculated from all coordinates in one frame, treating all coordinates in three axes identically and neglecting the relationship of the coordinates in the same axe. In this paper, an end to end deep learning framework is proposed that includes a normalization layer dynamically adapting to data distribution and inference procedure. After normalization, the three covariance feature sequences from the coordinates in three axes are produced from the sliding windows and are fused into one fusion matrix using a convolution layer. Finally, the fusion matrix is sequentially fed into an LSTM network to recognize skeleton action. The novelty of the proposed framework is combining the adaptive preprocessing and the features fusion to the LSTM network and improving the recognition accuracy by optimizing the quality of the features rather than network construction. In the experiments, the proposed framework is verified on the public datasets and one student action dataset collected from a real classroom. The experimental results demonstrate that the proposed method achieves a significant improvement in accuracy compared to the state-of-the-art methods. It can be concluded that the proposed framework can not only accurately capture the correlation of joints in the same frame but can also effectively express the dependences of sequential frames.

INDEX TERMS Action recognition, covariance, dynamic normalization, feature fusion, skeleton data.

I. INTRODUCTION

Human action recognition has a wide range of applications, including video surveillance, human-machine interaction, interactive entertainment and multimedia information retrieval [1]. Nowadays, the prevalence of highly-accurate and affordable depth devices has increased the use of depth videos to achieve computer vision tasks in various applications. Action recognition based on 3D skeleton sequences has attracted widespread attention from both academia and industry. Compared to the RGB videos, the skeleton data is more robust to cluttered backgrounds and illumination changes, with human actions described as movements of the skeleton joints [2]. Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM) units [3], [4] have been widely used to model the spatial-temporal information of the skeleton sequences for action recognition. However, it is difficult to construct deep LSTM networks to learn high-level features of skeleton data. Covariance Matrix (CM) can be considered as high-level feature. It has been a general representation method that has been successfully applied on image, vision and text processing [5]. For the skeleton data, the CM is computed on the coordinates of skeleton joints in a time-series manner. It has been proved that the CM combined with the time features can effectively describe the spatial characters of action sequences. Combining the CM and the LSTM network is the effective and reasonable roadmap to enhance the accuracy of action recognition. To date, several
LSTM models with CM features as inputs have been proposed. However, all these models have two limitations as follows. On the one side, normalization is a data preprocessing technique that transforms the data with attributes of different units into a uniform scale. After normalization, all attributes become relatively independent and better conditioned for convergence. Due to different body shapes and deviations in the movements from different subjects, the performance of deep learning models may degenerate rapidly if the skeleton data is not appropriately normalized. The traditional normalization methods include z-score, min-max and decimal scaling normalization [6]. There are also several deep learning normalization approaches available that include batch normalization [7], layer normalization [8] and group normalization [9]. The traditional normalization techniques can easily be applied to static data. However, it is still difficult for the existing approaches to handle evolving data, i.e., continuously change in range values and classes. Although the performance can slightly be improved through a deep learning schema, the space complexity is usually high. Moreover, from tasks, such as action recognition based on skeletal data, some important information may be lost when normalized in the traditional manner [10]. Therefore, to dynamically normalize the skeleton data, this paper proposes a dynamic normalization module that is capable of normalizing the data adaptively during inference according to the distribution of the measurements of the current skeletal data. On the other side, the traditional methods use all coordinates of all joints in one frame to compute the CM feature, which is not reasonable. It is meaningless to compare the x coordinate of one joint with the y coordinate of the other joint. Nevertheless, paying more attention to the spatial and the temporal relationship between the coordinates from the same axis may discover more valuable information for motion recognition. To encode the temporal dependency between the same axes of intra- and inter-frames, each skeleton sequence is transformed into three parts in the proposed framework and each part consists of one axis of the skeleton sequence data. Then all three skeleton sequences are fed into a sliding window with variable window length. The advantage of the proposed novel representation is that the deep learning models can be leveraged to learn fine-grained features of intra- and inter-frames. Generally, the limitation of the existing methods includes: Firstly, CM features are computed from raw data without normalization, or with static normalization. Secondly, the existing methods mainly compute the CM feature of all coordinates in one frame and neglect the relation of the coordinates in the same axis. The main contributions of this paper are summarized as follows:

1) A dynamic normalization module is proposed that learns to normalize the data according to their distribution instead of using fixed normalization schemes.

2) A covariance fusion module (CFM) is proposed to learn fine-grained features of intra- and inter-frames. The CFM first generates the CM features on three axes through the sliding window that can ensure the nonsingularity of CM. Then, the three CMs are fused using a convolution operation and a fusion CM is generated as the input of the LSTM network.

3) A lightweight strong baseline is developed on several datasets, which is more powerful than most the state-of-the-art methods.

The rest of the paper is organized as follows. Section II reviews the related works. Section III elaborates the normalized covariance fusion framework, which includes normalization layer design, CM fusion Layer design and loss function design. Section IV introduces the utilized dataset and the experimental results. Finally, Section V concludes the article.

II. RELATED WORKS

This section briefly reviews the relevant literature on normalization, data fusion and skeleton-based action recognition methods using deep learning networks.

A. NORMALIZATION ON SKELETON DATA

In 3D skeleton-based action classification, an action is described as a collection of time series of 3D positions of the joints in the skeleton. Generally, there are three paradigms for the normalization, which have been investigated in skeleton-based action classification. First, recognition by indicating the joint angles. The representation depends on the selection of the reference coordinate system, which may be different according to the recording environment and biometric differences. To deal with these issues, a collection of coordinate system transform methods have been proposed. [11] Considered the joint angles between any two connected limbs and represented an action as a time series of joint angles. Pham et al. calculated angles between the joint positions for the normalization. Dynamic Time Wrapping (DTW) was the obeyed classification method and resulted in an accuracy of over 90% for the recognition of five gestures such as hand waves and greetings [12]. In [13], data has been initially registered into a common coordinate system to make the joint coordinates comparable to each other. Second, recognition by setting the skeleton to a fixed position. In [2], human poses have been normalized by aligning the torsos and the shoulders. In [14] and [15], all the 3D joint coordinates have been transformed from the world coordinate system to the person-centric coordinate system placing by the Hip Center. In [16], the skeletons have been aligned based on the head location. Next, all the other coordinates in the skeleton were normalized by the head length [17]. Gaglio et al. [18] and Cippitelli et al. [19] normalized the 3D-skeleton-data by traversing the positions to a new coordinate system that was fixed at the torso. They achieved accuracy rates above 90% using Support Vector Machines (SVMs) on the KARD dataset containing strong specifiable gestures like hand waves and sidekick. Laraba at al. [20] excluded the arm joints from the dataset and normalized the other point positions by setting the distances to the pelvis joint. They recognized the dance steps with an accuracy of above 70%. Third, transforming the data into a range. In [5], the coordinates of the 3D joints have
been normalized to a range of \([0, 1]\) in all the dimensions over the sequence. Each coordinate of the normalized pose vector has been smoothed across time with a 5 by 1 Gaussian filter. The poses were normalized to compensate for biometric differences. A reference skeleton was learned from the training data. The length of skeleton limbs in the reference skeleton was adjusted to unit norm. All the skeletons in the test and the training data were transformed to impose the same limb segment length as in the reference skeleton while preserving the direction vector. A similar normalization has been applied in [21], where the inter-frame linear interpolation has also been applied to account for missing values in the skeletal data.

**B. DATA FUSION METHODS**

Data fusion has a long history and is of great significance in mining the data value. Early data fusion methods transformed the data into a single feature-based data and treated the transformed data as a single dataset. Nowadays, the common data fusion methods study the value of data from different perspectives. For example, the simplest data fusion method is to directly combine two one-dimensional datasets that have the same meaning in this dimension. Moreover, data fusion can be performed from the perspective of extracting the features of different dimensions. Different data fusion methods can provide different optimization results to the machine learning model. The representative deep learning architectures of the data fusion models include deep belief net, stacked autoencoder, convolutional neural network and recurrent neural network. The restricted Boltzmann machine is the basic block of the deep belief net [22]. Srivastava and Salakhutdinov [23] proposed a multimodal generative model based on the deep Boltzmann learning model and learned multimodal representations by fitting the joint distributions of multimodal data over the various modalities, such as image, text, and audio. Suk et al. and the Alzheimer’s Disease Neuroimaging Initiative [24] proposed a multimodal Boltzmann model that could fuse the complementary knowledge from the multimodal data to effectively diagnose Alzheimer at an early phase. To accurately estimate human poses, Ouyang et al. [25] designed a multisource deep learning model that learned multimodal representation from mixture type, appearance score, and deformation modalities by extracting the joint distribution of the body pattern in high-order space. To generate visually and semantically effective human skeletons from a series of images, especially videos, Hong et al. [26] proposed a multimodal deep autoencoder to capture the fusion relationship between the images and the poses. Wang et al. [27] designed a multimodal stacked autoencoder for feature learning of words in which the association and the gating mechanisms were adopted to improve the word features. Khattar et al. [28] designed a multimodal variational framework based on encoder-decoder architecture. To model the semantic mapping distribution between the images and the sentences, Ma et al. [29] proposed a multimodal convolutional neural network. In order to fully capture the semantic correlations, a three-level fusion strategy—the word level, the phrase level, and the sentence level, has been proposed. Frome et al. [30] presented a multimodal convolutional neural network with leveraging the semantic information from the text data to scale the visual recognition system to an unlimited number of discrete categories. There are some new RNN-based multimodal deep learning methods. For example, Abdulnabi et al. [31] designed a multimodal RNN to label the indoor scenes in which the intermodality feature and the cross-modality feature are learned by the RNN and the transform layers. Narayanan et al. [32] designed the gate recurrent cell with the multimodal sensor data to model driver behaviors. Sano et al. [33] proposed a multimodal BiLSTM to detect ambulatory sleep in which the BiLSTM was used to extract the features of the data collected from wearable devices. Then each intermodality feature was concatenated by a fully connected network. The methods of producing covariance descriptors for classification tasks have been intensively studied. In particular, [34] proposed the patch-specific covariance descriptors, efficiently computed with the integral images. Relying on systematically encoding mutual relationships inside the data, the covariance descriptors have been applied to many different applications such as face recognition [35], person identification [36] and more general classification tasks [37]. Furthermore, the covariance features have been proposed to measure the similarities across the data samples [38]. This latter direction grounds on the mathematical properties of positive definite matrices, exploiting the Riemannian metrics on the manifold for image classification. Once moved from a finite to an infinite-dimensional space, the performance enhances [39], [40] and only recently, the deep learning approaches have shown to be superior. However, one of the main limitation related to the covariance matrix is that it only enables to capture the linear inter-relationships [41]. For instance, the principal component analysis exploits a covariance matrix to remove the linear correlation of data points [42]. Among the attempts for modeling more complicated relationships, the additional statistics, such as entropy and mutual information [43], and kernels [44] have been adopted. A different approach is to model the non-linear behaviors by preliminary applying a preprocessing step and encode the raw data utilizing a transformation, which increases the feature space. Such approach has been used for spatial and temporal derivatives for gesture recognition in [45], where both different color spaces and edge detectors were considered for image classification, and filter bank responses were used as features to estimate the head orientation.

**C. SKELETON-BASED ACTION RECOGNITION**

The RNNs with LSTM units have been used to model the spatial-temporal information of the skeleton sequences for action recognition. Du et al. [46] divided the skeleton joints into five sets, and fed them into five LSTMs for feature fusion and classification. Du et al. [47] used random rotation and scale transformations during the training process to improve the robustness of the models. Shahroudy et al. [48]
the skeleton joints into five parts and fed them to a part-aware LSTM, which consisted of five sets of input and forget gates for the five body parts. Veeriah et al. [49] designed a differential RNN (dRNN) model and computed different orders of Derivative of States to capture the salient motions of actions. Zhu et al. [50] fed the skeleton joints to a deep LSTM at each time slot to learn the inherent co-occurrence features of the skeleton joints. Liu et al. [51] designed a spatial-temporal LSTM with a Trust Gate to jointly learn both the spatial and the temporal information of skeleton sequences and to automatically remove the noisy joints. Although the LSTM networks are designed to explore the long-term temporal dependencies, it is still difficult for the LSTM to memorize the information of the entire sequence with many time steps [49], [52]. Furthermore, it is also difficult to construct a deep LSTM to extract the high-level features [53], [54]. In contrast to the LSTM, the CNNs can extract such high-level features, but cannot model the long-term temporal dependency of the entire video [55].

III. PROPOSED METHOD

Fig. 1 shows the overall architecture of the proposed method. A skeleton sequence is first dynamically normalized. The three axes covariance matrices corresponding to the 3D coordinates of skeleton joints are generated with a sliding window. After that, a Covariance Fusion Module is used to fuse the three covariance matrices. The fused matrices are then fed to an LSTM network for action recognition. This section describes the details of the proposed framework.

A. DYNAMIC NORMALIZATION MODULE DESIGN

As shown in the Fig. 2, let \( X^i \in \mathbb{R}^{T_i \times d} ; i = 1, 2, \ldots, N \) be a collection of \( N \) skeletal sequence data, each of them composed of \( T_i \) d-dimensional measurements (or features). The notation \( x^i_j \in \mathbb{R}^d ; j = 1, 2, \ldots, T_i \) is used to refer to the \( d \) features observed at a time point \( j \) in the skeletal sequence data \( i \). The target of the proposed method is to learn to normalize the measurements \( x^i_j \) using the following formula:

\[
\tilde{x}^i_j = \left( x^i_j - \mu^i \right) \otimes \sigma^i \tag{1}
\]

where \( \otimes \) is the Hadamard division operator. A Hadamard division is defined as:

\[
\tilde{x}^i_j = \frac{x^i_{j,k} - \mu^i_{k}}{\sigma^i_{k}} \tag{2}
\]

The global z-score normalization is \( \mu = [\mu_1, \mu_2, \ldots, \mu_d] \), \( \sigma = [\sigma_1, \sigma_2, \ldots, \sigma_d] \), where \( \mu_k \) and \( \sigma_k \) refer to the global average and the standard deviation of the k-th feature, respectively:

\[
\mu_k = \frac{1}{NT} \sum_{i=1}^{N} \sum_{j=1}^{T_i} x^i_{j,k} \tag{3}
\]

\[
\sigma_k = \sqrt{\frac{1}{NT} \sum_{i=1}^{N} \sum_{j=1}^{T_i} (x^i_{j,k} - \mu_k)^2} \tag{4}
\]

As discussed above, \( \mu \) and \( \sigma \) might not be optimal for normalizing every possible measurement vector.
Therefore, this work proposes to normalize each skeletal sequence so that \( \mu \) and \( \sigma \) are learned and depend on the current input of each sliding window, instead of being the global averages calculated using the entire dataset. The initial estimation for the mean of the current skeletal sequence \( X^i \) is (5).

\[
\mu^i = \frac{1}{T_i} \sum_{j=1}^{T_i} x^j
\]

where \( x^j \in \mathbb{R}^d \).

A matrix \( W_\mu \) is defined that represents the deep learning model weight parameters:

\[
da^i = W_\mu \mu^i + b_\mu
\]

where \( a^i \in \mathbb{R}^d \), indicates the dynamic mean of the current skeletal sequence data. The magnitude of the variation in the adaptive mean of the skeletal sequence data can be calculated as (7).

\[
c^k_i = \sqrt{\frac{1}{T_i} \sum_{j=1}^{T_i} (x^j - \mu^i)^2}, \quad k = 1, 2, \ldots, d
\]

The dynamic standard deviation can be calculated as (8).

\[
d^i = W_c c^i + b_c
\]

For \( x^j \), adaptive regularization can be performed as (9).

\[
\tilde{x}^j = \text{sigmoid}\left((x^j - a^i) \odot d^i\right)
\]

### TABLE 1. Variables notation.

| Variable | Notation | Variable | Notation |
|----------|----------|----------|----------|
| \( X^i \) | A skeletal sequence data | \( W_\mu \) | Number of subsequences |
| \( T_i \) | Length of a skeletal sequence data | \( m \) | Number of appended frames |
| \( x^j \) | Features observed at a time point \( j \) in skeletal sequence data | \( \sigma^c, \sigma^p, \sigma^q \) | The \( \sigma \)th skeletal point values on the 3 axes in frame \( j \) respectively |
| \( \overline{x}^j \) | Normalized features observed at a time point \( j \) in skeletal sequence data | \( s \) | Step size |
| \( \mu^i \) | Initial estimation for the mean of the skeletal sequence | \( S \) | Value of the bone sequence at the same coordinate |
| \( \sigma^i \) | Initial estimation for the standard deviation of skeletal sequence | \( M \) | Difference of time index |
| \( N \) | Size of skeletal sequence data | \( f \) | Fusion function |
| \( T \) | Total length of skeletal sequence dataset | \( G_i, G_0 \) | Coordinates of the torso joint and coordinates of the \( i \)-th joint |
| \( d \) | Feature dimension | \( j_i \) | Distance vector between \( G_i \) and \( G_0 \) |
| \( a^i \) | Dynamic mean of the skeletal sequence | \( d^i \) | Dynamic standard deviation of the skeletal sequence |
| \( D \) | Window size | \( F \) | Pose feature vector |
| \( y_n \) | Real label | \( y_n^p \) | Predicted label |
| \( \overline{y}_q \) | Distance between the \( q \)-th skeleton of the normalized data and the reference skeleton | \( \overline{y}_q^p \) | Distance between the \( q \)-th skeleton of the raw data and the reference skeleton |
| \( l_1, l_2 \) | Two parts of the objective function | \( \omega_1, \omega_2 \) | Weights of the objective function |
| \( L \) | Final objective function | \( H \) | A filter |

### FIGURE 2. Dynamic normalization module.
possibly-overlapping sub-sequences of lengths \( d, d/2, d/3, \ldots \). Generally, the \( i \)-th level of the pyramid contains sub-sequence of length \( d/2^i \) frames, where \( d \) is the feature dimension. The consecutive sub-sequences in the level are separated by a constant step \( s \). For both the non-overlapping and the overlapping modes, one issue to be considered is that some datasets have only a few data frames for an action, either less than the length of the sub-sequence (sliding window), or the last remaining frames after sliding is less than the length of the sub-sequence (sliding window). The method used in the proposed framework is circular selection.

The descriptor has a fixed length independent from the length of the sub-sequence (sliding window). The method used in the last remaining frames after sliding is less than the length of the sub-sequence (sliding window), or the last remaining frames after sliding is less than the length of the sub-sequence (sliding window). The method used in the proposed framework is circular selection.

\[
W_i = \left\lceil \frac{T_i - D}{s} \right\rceil + 1 \tag{10}
\]

For example, if the sequence length is 10, the step size is 2 and the window size is 3, then a \( \left\lceil \frac{10 - 3}{2} \right\rceil + 1 = 4 + 1 = 5 \) sub-sequence can be generated.

C. COVARIANCE FUSION MODULE DESIGN

Next, the covariance matrix is used for skeleton joint locations over time as a discriminative descriptor for a sequence. The descriptor has a fixed length independent from the length of the original sequence. As shown in Fig. 4, the descriptor is constructed by computing the covariance matrix on the coordinates of body skeleton joints over time. In order to encode the short-term temporal dependency of joint locations from a different view, multiple covariance matrices are used with each covering a sliding window of the normalized input sequence at one axis.

Suppose the body has \( d \) skeleton points and the action has \( T_i \) data frames. Let \( \hat{c}_i, p_i, \) and \( q_i \) represent the \( c \)th skeletal point values on the 3 axes in frame \( j \), respectively. Let \( O = \{o_1, o_2, \ldots, o_T\} \), \( P = \{p_1, p_2, \ldots, p_d\} \) and \( Q = \{q_1, q_2, \ldots, q_d\} \) represent the sequence of skeletal point values on the 3 axes at a given frame, respectively. Since the probabilities of \( O, P, Q \) are usually unknown, the sample covariance is used as (11).

\[
C(S) = \frac{1}{T_i - 1} \sum_{j=1}^{T_i} (S - \bar{S})(S - \bar{S})^T \tag{11}
\]

where \( S \) represents the value of the bone sequence at the same coordinate. The covariance formula for data frame from \( t_1 + 1 \) to \( t_2 \) can be obtained as (12).

\[
C^{(t_1, t_2)}(S) = \frac{1}{M - 1} \left( U^{(t_1, t_2)} - \frac{1}{M} V^{(t_1, t_2)} \left( V^{(t_1, t_2)} \right)^T \right) \tag{12}
\]

\[
V^{(i)} = \sum_{i=1}^{t} S^{(i)} \tag{13}
\]

\[
U^{(i)} = \sum_{i=1}^{t} S^{(i)} \left( S^{(i)} \right)^T \tag{14}
\]

where \( M = t_2 - t_1 \), \( U^{(t_1, t_2)} = U(t_2) - U(t_1) \), \( V^{(t_1, t_2)} = V(t_2) - V(t_1) \). A fusion function \( f: o_i, p_i, q_i \rightarrow z_i \) fuses the three axes covariance matrices \( o_i \in \mathbb{R}^{d \times d}, p_i \in \mathbb{R}^{d \times d} \) and \( q_i \in \mathbb{R}^{d \times d}, \) at time \( t \), to produce an output map \( z_t \in \mathbb{R}^{d \times d} \), where \( d \) is the number of skeleton points. When applied to feedforward convolutional network architectures, consisting of convolutional, fully-connected, pooling and nonlinearity layers, \( f \) can be applied at different points in the network to implement various fusion functions. In this paper, convolution fusion is used. \( z_t \rightarrow \) \( f(o_i, p_i, q_i) \) stacks the three feature maps at the same spatial location \( i, j \) across the feature channels, as shown in the Fig. 5.

\[
z_{stack}^{stack} = o_{ij}, z_{stack} = p_{ij}, z_{stack} = q_{ij} \tag{15}
\]

where \( z_{stack} \in \mathbb{R}^{d \times d \times 3} \). Then, the stacked data is convolved with a bank of filters \( H \) and biases \( b_{stack} \) as:

\[
z_{conv} = z_{stack} \times H + b_{stack} \tag{16}
\]

Here, the filter \( H \) is used to model the weighted combinations of the three feature maps \( o_i, p_i, q_i \) at the same spatial location. When used as a trainable filter kernel in the network, \( H \) can learn the correspondences of the three feature maps that minimize a joint loss function. Given that there should be no significant relationship between the different skeletal points, most of the values between all the skeleton points should be 0 or close to 0. In order to filter out this small value information, \( H \) is used here, which is meant to extract the locally available information for integration.

The LSTM represents an improved RNN architecture, which mitigates the vanishing gradient effect of the RNN. An LSTM neuron contains a memory cell \( c_t \), which has...
a self-connected recurrent edge of weight. At each time step $t$, the neuron can choose to write, reset, or read the memory cell governed by the input gate $i_t$, forget gate $f_t$, and output gate $o_t$, as shown in (17)-(22), respectively.

$$i_t = \sigma_g \left( W_i x_t + U_i h_{t-1} + b_i \right) \quad (17)$$

$$f_t = \sigma_g \left( W_f x_t + U_f h_{t-1} + b_f \right) \quad (18)$$

$$\tilde{c}_t = \sigma_h \left( W_c x_t + U_c h_{t-1} + b_c \right) \quad (19)$$

$$o_t = \sigma_g \left( W_o x_t + U_o h_{t-1} + b_o \right) \quad (20)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (21)$$

$$h_t = o_t \odot \sigma_h(c_t) \quad (22)$$

where the initial values are $c_0 = 0$ and $h_0 = 0$ and the operator $\odot$ denotes the Hadamard product (element-wise product). The subscript $t$ indexes the time step. The input vector to the LSTM unit is $x_t$. The activation vectors of forget, input and output gates are $f_t, i_t, o_t \in R^h$, respectively. The hidden state vector also known as the output vector of the LSTM unit is $h_t \in R^h$. $\tilde{c}_t \in R^h$ is the cell input activation vector and $c_t \in R^h$ is the cell state vector. $W \in R^{h \times d}$, $U \in R^{h \times h}$ and $b \in R^h$ refer to weight matrices and bias vector parameters that need to be learned during training. The superscripts $d$ and $h$ refer to the numbers of input features and hidden units, respectively. $\sigma_g$ is the sigmoid function. $\sigma_c$ and $\sigma_h$ are the hyperbolic tangent functions.

A straightforward solution is to compensate the position of the skeleton by centering the coordinate space in one skeleton joint. Considering a skeleton composed of $P$ joints, $G_0$ and $G_1$...
being the coordinates of the torso and i-th joints, the i-th joint feature $j_i$ is the distance vector between $G_0$ and $G_i$.

$$j_i = \|G_i - G_0\|^2$$  (23)

These features may be seen as a set of distance vectors connecting each joint of the torso. A posture feature vector $F$ is created for each skeleton frame can be expressed as:

$$F = [j_1, j_2, \ldots, j_d]$$  (24)

In order to find the optimal solution, it is essential to measure the quality of a solution. This is done using an objective function. For each frame of the sequence, the distance from the reference skeleton for each skeleton point is calculated.

$$l_1 = \frac{1}{N} \sum_{i=1}^{N} \left[ \sum_{p=1}^{T} \left( \frac{1}{d} \sum_{q=1}^{d} J_q^i - \frac{1}{d} \sum_{q=1}^{d} J_q^n \right) \right]$$  (25)

Equation (25) is the objective function of the dynamic normalization part, where $J_q^i$ is the distance between the qth joint of the raw data and the reference skeleton and $J_q^n$ is the distance between the qth skeleton of the normalized data and the reference skeleton. For each classification result, the information entropy is calculated according to (26).

$$l_2 = -\frac{1}{N} \sum_{n=1}^{N} [y_n \log \tilde{y}_n + (1 - y_n) \log (1 - \tilde{y}_n)]$$  (26)

This part of the objective function is defined to ensure the validity of the dynamic normalization. The final objective function is a combination of the weights of these two parts of the error function and defined as (27).

$$L = \omega_1 l_1 + \omega_2 l_2$$  (27)

where $\omega_1$ and $\omega_2$ are the weights of these two parts of the objective function.

IV. EXPERIMENTAL RESULTS

In order to examine the effectiveness of the proposed method, MSR-Action3D, UTKinect-Action3D, UTD-MHAD, NTU RGB+D, SBU Kinect Interaction, PKU-MMD, Kinetics Human Action Video, G3D and Classroom datasets are used. The details of these datasets are explained in the next section by modeling the actions. The experiments are conducted on the PyTorch and TensorFlow platforms with NVIDIA GeForce RTX 2070 GPU cards.

A. DATASETS

The MSR-Action3D dataset (MSR) consists of 3D positions of 20 skeletal joints, including 20 types of actions. Each action is conducted three times by ten subjects, and there are 557 action sequences. The UTKinect-Action3D dataset consists of 20 skeletal joints, including ten types of actions. Each action is performed twice by ten subjects, and there are a total of 199 action sequences. The Florence 3D Action dataset (Florence) consists of 15 skeletal joints, including nine types of actions. Each action is conducted two to three times by ten subjects, and there are 215 action sequences in total. The UTD-MHAD dataset (UTD) consists of 20 skeletal joints and contains 27 types of actions. Each action is conducted four times by eight subjects, and there is a total of 816 action sequences. The NTU RGB+D dataset (NTU) is so far the largest skeleton-based human action recognition dataset. It contains 56680 skeleton sequences and 60 annotated action classes. There are two recommended evaluation protocols: Cross-Subject (CS) and Cross-View (CV). In the cross-subject setting, sequences of 20 subjects are used for training and the sequences of the rest of the 20 subjects are used for validation. In the cross-view setting, the samples are split by camera views. The samples from two camera views are used for training and the rest are used for testing. The SBU Kinect Interaction dataset (SBU) is a Kinect captured human activity recognition dataset depicting two-person interaction. It contains 282 skeleton sequences and 6822 frames of 8 classes. There are 15 joints for each skeleton. The PKU-MMD dataset (PKU) is currently the largest skeleton-based action detection dataset. It contains 1076 long untrimmed video sequences performed by 66 subjects in three camera views and 51 annotated action categories, resulting in almost 20,000 action instances and 5.4 million frames in total. Similar to NTU RGB+D, there are also two recommended evaluation protocols: cross-subject and cross-view. The Kinetics Human Action Video dataset (Kinetics) contains 400 human action classes, with at least 400 video clips for each action. Each clip lasts around 10s and is taken from a different YouTube video. The actions are human-focused and cover a broad range of classes including human-object interactions such as playing instruments, as well as human-human interactions such as shaking hands. The Game 3D dataset contains a range of gaming actions captured with Microsoft Kinect. The Kinect enabled us to record synchronized video, depth and skeleton data. The Game 3D dataset contains 10 subjects performing 20 gaming actions, punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw a bowling ball, aim and fire a gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap. The 20 gaming actions were recorded in 7 action sequences. Most sequences contain multiple actions in a controlled indoor environment with a fixed camera, a typical setup for gesture-based gaming. In order to test the effect of the proposed method in real scenes, experimental data on the action of students in the classroom was collected and the classroom dataset (Classroom) was obtained, as shown in the Fig. 6. The classroom data set provides 3D positions of 25 skeletal joints, including 7 actions. Each action was performed 3 times and lasted about 13 to 15 seconds. There were 10 experimental subjects.

B. EFFECTIVENESS OF COVARIANCE FUSION MODULE AND DYNAMIC NORMALIZATION MODULE

A covariance matrix contains the important spatial information of a skeleton sequence which is important for skeleton-based action recognition. In order to demonstrate the
TABLE 2. Classification accuracies of different models on different datasets.

| Model          | LSTM | CM+LSTM(without FL) | CPM+LSTM | CFM+DNM+LSTM |
|----------------|------|---------------------|----------|--------------|
| Florence       | 58.23| 65.39               | 84.02    | 93.25        |
| UTD            | 45.31| 48.50               | 77.68    | 90.25        |
| UTK            | 36.25| 53.21               | 79.92    | 94.36        |
| MSR            | 42.26| 40.15               | 66.61    | 91.47        |
| Classroom      | 36.57| 40.29               | 56.53    | 82.84        |
| NTU (CV)       | 46.39| 68.62               | 84.36    | 92.34        |
| NTU (CS)       | 43.28| 62.37               | 79.15    | 87.29        |
| Game 3D        | 56.32| 73.29               | 81.25    | 91.72        |
| Kinetics       | 21.38| 26.45               | 31.28    | 35.80        |
| PKU (CV)       | 56.29| 73.26               | 89.26    | 92.60        |
| PKU (CS)       | 49.27| 70.14               | 86.28    | 90.40        |
| SBU            | 61.25| 78.29               | 90.21    | 94.32        |

FIGURE 6. Classroom dataset.

effectiveness of covariance matrix layer, various experiments on different datasets were performed. Table 2 and Fig. 7 compares the accuracies of different models. In the following, CFM, FL, DNM and LSTMs denote the covariance fusion module, the fusion layer, the dynamic normalization module and the LSTM networks, respectively. Four designed models (Columns 1 to 4 in Table. 2) are investigated to validate the effectiveness of the CFM, FL and DNM. The three main observations are as follows.

1) It is difficult to construct deep LSTM networks to learn high-level features. For the learning of spatial and temporal information of skeleton sequence, the covariance matrix of the skeleton sequence is used. Given the different sequence lengths of different actions, a sliding window is used to split the long sequences of different lengths into fixed-length sub-sequences. Compared with the LSTM network (Column 1 in Table. 2) without covariance matrix and sliding window, the accuracy is improved by about 3% to 15% (Column 2 in Table. 2). The effectiveness of using the covariance matrix and sliding window is verified.

2) In the CFM, the skeletal sequence is decomposed on 3 axes to compute the covariance matrices. The matrices on each axis do not contain a lot of information that can be used for action recognition. To efficiently fuse the locally valid information on the 3 axes, the FL (Column 3 in Table. 2) is used. The FL improves the accuracy of the overall model by about 15%-30% compared with the method without the FL.

3) Data normalization is a fundamental preprocessing step for mining and learning from the data. The normal normalization may cause loss of available information in a sliding window. Therefore, to enhance the characteristics of the actions in the window, the DNM is used to adjust the normalization strategy through an adaptive approach. The experiments show that the DNM is effective and improves the accuracy by about 4% to 27%.

In summary, the explicit modeling of the covariance matrix, fusion, dynamic normalization and LSTM network benefit the learning of spatial and temporal information. The information enables the model to efficiently exploit the information of sequence order.

C. SLIDING WINDOW SIZE

The rank of the covariance matrix obeys the rule that \( \text{rank} (C) \leq \min (d, n - 1) \). When \( C \) is used as region descriptor, the number of feature vectors extracted from an image region. \( n \) is usually much larger than the dimensions \( d \). This ensures \( C \) to be nonsingular and allows it to be reliably estimated [44]. In order to examine the effect of different sliding window lengths on the proposed model, experiments
FIGURE 7. Comparisons of different methods on different datasets in terms of accuracy and the method.

FIGURE 8. Accuracy comparison of the proposed model with different sliding window sizes on different datasets.

were conducted using different lengths of sliding windows, and the results are shown in Table. 3 and Fig. 8.

The proposed method models the correlations of the joints using the sliding window and the size is $d$. To demonstrate its effectiveness, the performance of the proposed model is compared with four different window sizes in Table. 3. It can be observed that the classification accuracy changes with different sliding window sizes. Columns 1 to 2 in Table. 3 are less effective in terms of classification accuracy than the models with a window size of less than $d$. The effect of window size of less than $d$ is shown in Table. 3. The result occurs because when the window length is too small, some useful information that can be used for action recognition is lost. For example, for the action of standing up from a chair, the sequence of actions of leaving the chair may be cut into two separate sub-sequences, each of which cannot determine the current job, making the recognition more difficult. Columns 4 to 5 in Table. 3 have window sizes greater than $d$. 

VOLUME 8, 2020
The effect of the model with window size > $d$ is greater than the effect of the model with window size = $d$ in terms of classification accuracy. It can be seen that the accuracy of the corresponding model decreases instead of increasing when the size of the window band is larger than $d$. This is presumably due to the fact that the increase in the window size leads to a decrease in the number of sub-sequences generated for skeletal sequences of the same size, i.e. a decrease in the total training data. At the same time, a longer window can lead to some interfering frames, e.g. for the action of eating, the sub-sequence of bringing the food into the mouth is more important, while the earlier hand-raising action is equivalent to an interference in its recognition.

**D. COMPARISON WITH THE STATE-OF-THE-ARTS**

The performance of the proposed method is compared with other state-of-the-art methods on NTU RGB+D, Florence, UTD, MSR, Game, Kinetics-Skeleton, Classroom, PKU-MMD and SBU datasets. The comparative results are shown in Table. 4 and Fig. 9.

As shown in Table. 4 and Fig. 9, the proposed method brings performance improvement of 1.95%, 1.10% and 6.33% on Florence, MSR and Classroom datasets, respectively. The Co-LSTM, STA-LSTM and IndRNN are RNN-based methods. Compared to these RNN-based models, the accuracy of the proposed model is improved on almost all datasets. To better explore the structural information of skeleton, some models used GCN, which is the most highly regarded search method currently in skeletal data-based action recognition. To further investigate the performance of the model proposed in this paper, experiments were conducted on a large dataset NTU RGB+D 120. The highest accuracy on this dataset was the GCN-based model PA-ResGCN-B19 with 87.3% accuracy (CROSS-SUBJECT).

**TABLE 4.** Classification accuracy of the proposed model with different sliding window sizes on different datasets.

|                  | Florence | UTD | UTK | MSR | Classroom | NTU (CV) | NTU (CS) | Game 3D | Kinetics | PKU (CV) | PKU (CS) | SBU |
|------------------|-----------|-----|-----|-----|-----------|----------|----------|---------|----------|----------|----------|-----|
| RA-GCN[56]       | 81.35     | 79.25 | 89.23 | 56.74 | 76.51     | 93.60    | 87.3     | 78.60   | 25.60    | 85.60    | 79.2     | 75.30|
| SPDNet[57]       | 49.30     | 50.62 | 67.74 | 47.75 | 32.26     | 46.35    | 42.59    | 61.35   | 15.90    | 42.60    | 37.8     | 77.60|
| Lie Group[14]    | 91.40     | 92.36 | 97.20 | 90.37 | 56.40     | 52.80    | 50.1     | 80.60   | 30.30    | 82.30    | 76.7     | 85.70|
| Co-LSTM[50]      | 78.24     | 76.35 | 80.21 | 77.32 | 62.58     | 85.32    | 81.7     | 78.30   | 27.10    | 67.30    | 62.8     | 90.41|
| STA-LSTM[58]     | 80.29     | 79.56 | 79.26 | 81.16 | 68.37     | 81.20    | 73.4     | 75.60   | 28.40    | 92.60    | 86.9     | 91.50|
| ST-TR[39]        | 85.20     | 84.60 | 86.30 | 75.60 | 69.30     | 96.10    | 89.9     | 79.5    | 37.4     | 90.7     | 83.6     | 82.4  |
| HCN[60]          | 89.3      | 84.3  | 90.1  | 83.4  | 71.4      | 91.1     | 86.5     | 83.5    | 28.9     | 94.2     | 92.6     | 98.6 |
| HDM-BC[61]       | 91.3      | 92.8  | 87.5  | 86.1  | 64.8      | 77.6     | 71.2     | 92.0    | 30.6     | 86.6     | 84.5     | 90.4 |
| IndRNN[62]       | 85.6      | 79.8  | 82.4  | 79.2  | 76.5      | 88.0     | 81.8     | 86.3    | 32.8     | 85.4     | 82.9     | 89.6 |
| Ours method      | 93.25     | 90.25 | 94.36 | 91.47 | 82.84     | 92.34    | 87.29    | 91.72   | 35.80    | 92.6     | 90.4     | 94.32|

**FIGURE 9.** Accuracy comparison of the proposed method with state-of-the-art methods on different datasets.
The highest RNN-based model, Logsig-RNN, had an accuracy of 68.3% (CROSS-SUBJECT). The model in this paper has an accuracy of 81.4% (CROSS-SUBJECT) on this dataset. Compared with the GCN-based models, the proposed model does not perform better on all datasets. However, even if the accuracy gap is somewhat less, it is still within a certain range, which validates the effectiveness of the proposed model.

V. CONCLUSION
In this work, an end-to-end framework for skeleton-based action recognition is presented. In order to model the correlations of joints, the covariance fusion module and LSTM network are proposed for capturing the correlations of joints in the same frame and a window of frames is proposed for modeling the dependencies of frames. To enhance the characteristics of the actions in the window, the dynamic normalization module is used to adjust the normalization strategy through an adaptive approach. The experimental results demonstrate that the proposed method achieves a significant improvement in accuracy compared with the state-of-the-art methods.

ACKNOWLEDGMENT
This work was partially supported by the National Natural Science Foundation of China (Grant No. 61977061 and 51934007).

REFERENCES
[1] X. Yang and Y. Tian, “Super normal vector for activity recognition using depth sequences,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 804–811.
[2] F. Han, B. Reilly, W. Hoff, and H. Zhang, “Space-time representation of people based on 3D skeletal data: A review,” Comput. Vis. Image Understand., vol. 158, pp. 85–105, May 2017.
[3] A. Graves, “Supervised sequence labelling,” in Supervised Sequence Labelling With Recurrent Neural Networks. Berlin, Germany: Springer, 2012, pp. 3–13.
[4] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., May 2013, pp. 6645–6649.
[5] M. E. Hussein, M. Torki, M. A. Gowayed, and M. El-Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations,” in Proc. Int. Joint Conf. Artif. Intell., 2013, pp. 1–5.
[6] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, “A comparison of normalization methods for high density oligonucleotide array data based on variance and bias,” Bioinformatics, vol. 19, no. 2, pp. 185–193, Jan. 2003.
[7] S. Loffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” 2015, arXiv:1502.03167. [Online]. Available: http://arxiv.org/abs/1502.03167
[8] J. Lei Ba, J. Ryan Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450. [Online]. Available: http://arxiv.org/abs/1607.06450
[9] Y. Wu and K. He, “Group normalization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 3–19.
[10] E. Ogasawara, L. C. Martinez, D. de Oliveira, G. Zimbraz, G. L. Pap, and M. Mattoso, “Adaptive normalization: A novel data normalization approach for non-stationary time series,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2010, pp. 1–8.
[11] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, “Sequence of the most informative joints (SMJ): A new representation for human skeletal action recognition,” J. Vis. Commun. Image Represent., vol. 25, no. 1, pp. 24–38, Jan. 2014.
[12] C. H. Pham, Q. K. Le, and T. H. Le, “Human action recognition using dynamic time warping and voting algorithm,” VNU J. Sci., Comput. Sci. Commun. Eng., vol. 30, no. 3, p. 15, 2014.
[13] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu, “Concurrent action detection with structural prediction,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 3136–3143.
[14] R. Venulapallli, F. Arrate, and R. Chellappa, “Human action recognition by representing 3D skeletons as points in a lie group,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 588–595.
[15] D. Wu and L. Shao, “Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 724–731.
[16] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, “Bio-inspired dynamic 3D discriminative skeletal features for human action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2013, pp. 471–478.
[17] C. Wang, Y. Wang, and A. L. Yuille, “An approach to pose-based action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2013, pp. 915–922.
[18] S. Gagliolo, G. L. Re, and M. Morana, “Human activity recognition process using 3-D posture data,” IEEE Trans. Human-Machine Syst., vol. 45, no. 5, pp. 586–597, Oct. 2015.
[19] E. Cipitelli, S. Gasparriini, E. Gambi, and S. Spinansre, “A human activity recognition system using skeleton data from RGBD sensors,” Comput. Intell. Neurosci., vol. 2016, pp. 1–14, 2016.
[20] S. Laraba, J. Tilmanne, and T. Dutoit, “Adaptation procedure for HMM-based sensor-dependent gesture recognition,” in Proc. 8th ACM SIGGRAPH Conf. Motion Games, 2015, pp. 17–22.
[21] M. Zanfir, M. Leordeanu, and C. Sminchiscu, “The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 2752–2759.
[22] N. Zhang, S. Ding, J. Zhang, and Y. Xue, “An overview on restricted Boltzmann machines,” Neurocomputing, vol. 275, pp. 1186–1199, Jan. 2018.
[23] N. Srivastava and R. R. Salakhutdinov, “Multimodal learning with deep Boltzmann machines,” in Proc. Adv. neural Inf. Process. Syst., Dec. 2012, pp. 2222–2230.
[24] H.-I. Suk, S.-W. Lee, and D. Shen, “Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis,” Neuroimage, vol. 101, p. 569–582, Nov. 2014.
[25] W. Ouyang, X. Chu, and X. Wang, “Multi-source deep learning for human pose estimation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 2329–2336.
[26] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, “Multimodal deep autoencoder for human pose recovery,” IEEE Trans. Image Process., vol. 24, no. 12, pp. 5659–5670, Dec. 2015.
[27] S. Wang, J. Zhang, and C. Zong, “Associative multichannel autoencoder for multimodal word representation,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 115–124.
[28] D. Khattar, J. S. Goud, M. Gupta, and V. Varma, “Mvae: Multimodal variational autoencoder for fake news detection,” in Proc. World Wide Web Conf., 2019, pp. 2915–2921.
[29] L. Ma, Z. Lu, L. Shang, and H. Li, “Multimodal convolutional neural networks for matching image and sentence,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 2623–2631.
[30] A. Frome, “4Devise: A deep visual-semantic embedding model,” in Proc. NIPS, vol. 2013, pp. 2121–2129.
[31] A. Abdulnabi, B. Sbiti, Z. Zuo, L.-P. Chau, and G. Wang, “Multimodal recurrent neural networks with information transfer layers for indoor scene labeling,” IEEE Trans. Multimedia, vol. 20, no. 7, pp. 1656–1671, Jul. 2018.
[32] A. Narayanan, A. Siravuru, and B. Dariush, “Sensor fusion: Gated recurrent fusion to learn driving behavior from temporal multimodal data,” 2019, arXiv:1910.00628. [Online]. Available: http://arxiv.org/abs/1910.00628
[33] A. Sano, W. Chen, D. Lopez-Martinez, S. Taylor, and R. W. Picard, “Multimodal ambulatory sleep detection using LSTM recurrent neural networks,” IEEE J. Biomed. Health Informat., vol. 23, no. 4, pp. 1607–1617, Jul. 2019.
[34] T. Kerola, N. Inoue, and K. Shinoda, “Spectral graph skeletons for 3D action recognition,” in Proc. Asian Conf. Comput. Vis., 2014, pp. 417–432.
[35] X. Yu, S. Xiong, Y. Gao, and X. Yuan, “Contour covariance: A fast descriptor for classification,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 589–600.
M. H. Quang, M. San Biagio, and V. Murino, “Log-Hilbert-Schmidt information for classification,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 809–816.

M. San Biagio, S. Martelli, M. Crocco, M. Cristani, and V. Murino, “Encoding classes of unaligned objects using structural similarity cross-covariance tensors,” in Proc. Iberoamerican Congr. Pattern Recognit., 2013, pp. 133–140.

M. H. Quang, M. San Biagio, and V. Murino, “Log-Hilbert-Schmidt metric between positive definite operators on Hilbert spaces,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 388–396.

M. Harandi, M. Salzmann, and F. Porikli, “Bregman divergences for infinite dimensional covariance matrices,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1003–1010.

J. D. Hamilton, Time Series Analysis. Princeton, NJ, USA: Princeton Univ. Press, 2020.

M. Harandi, M. Cristani, and V. Murino, “Heterogeneous auto-similarities of characteristics (HASC): Exploiting relational information for classification,” in Proc. IEEE Int. Conf. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1972–1984, Aug. 2013.

C. M. Bishop, Pattern Recognition Machine Learning. Cham, Switzerland: Springer, 2006.

D. Tosato, M. Serra, M. Cristani, and V. Murino, “Characterizing humans on Riemannian manifolds,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 10, pp. 2042–2056, Oct. 2013.

Y.-F. Song, Z. Zhang, and L. Wang, “Richly activated graph convolutional network for action recognition with incomplete skeletons,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 1–5.

J. Gu, G. Wang, J. Cai, and T. Chen, “An empirical study of language model learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7733–7742.

J. Liu, A. Shahroudy, D. Xu, and G. Wang, “NTU RGB+D: A large scale dataset for 3D human activity analysis,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 1010–1019.

Y. Du, W. Wang, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1110–1118.

A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell, “Spatio-temporal covariance descriptors for action and gesture recognition,” in Proc. IEEE Workshop Appl. Comput. Vis. (WACV), Jan. 2015, pp. 103–110.

W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks,” 2016, arXiv:1603.07772. [Online]. Available: http://arxiv.org/abs/1603.07772.

J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal lstm with trust gates for 3d human action recognition,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 816–833.

T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, long short-term memory, fully connected deep neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2015, pp. 4580–4584.

L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, “Temporal segment networks: Towards good practices for deep action recognition,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 20–36.

Y. Du, Y. Fu, and L. Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4041–4049.

C. Li, Q. Zhong, D. Xie, and S. Pu, “Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation,” 2018, arXiv:1804.06055. [Online]. Available: http://arxiv.org/abs/1804.06055.

R. Zhao, W. Xu, H. Su, and Q. Ji, “Bayesian hierarchical dynamic model for human action recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7733–7742.

S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, “Independently recurrent neural network (IndRNN): Building a longer and deeper RNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5457–5466.

GUAN HUANG received the B.S. degree in computer science and technology from Nanchang University, Nanchang, China, in 2017. He is currently pursuing the master’s degree with the China University of Mining and Technology, Xuzhou, China. His research interests include deep learning for action recognition.

QIUYAN YAN received the Ph.D. degree in computer applications from the China University of Mining and Technology in 2010. She is currently an Associate Professor with the China University of Mining and Technology. Her current research interests include multimodal image action recognition, big data analytics for education, and temporal data mining.