Ensemble Deep Learning for Skeleton-based Action Recognition using Temporal Sliding LSTM networks

Inwoong Lee, Doyoung Kim, Seoungyoon Kang, Sanghoon Lee*
Department of Electrical and Electronic Engineering, Yonsei University, Seoul, Korea
{mayddb100, tnyffx, auffallig, slee}@yonsei.ac.kr

Abstract

This paper addresses the problems of feature representation of skeleton joints and the modeling of temporal dynamics to recognize human actions. Traditional methods generally use relative coordinate systems dependent on some joints, and model only the long-term dependency, while excluding short-term and medium term dependencies. Instead of taking raw skeletons as the input, we transform the skeletons into another coordinate system to obtain the robustness to scale, rotation and translation, and then extract salient motion features from them. Considering that Long Short-term Memory (LSTM) networks with various time-step sizes can model various attributes well, we propose novel ensemble Temporal Sliding LSTM (TS-LSTM) networks for skeleton-based action recognition. The proposed network is composed of multiple parts containing short-term, medium-term and long-term TS-LSTM networks, respectively. In our network, we utilize an average ensemble among multiple parts as a final feature to capture various temporal dependencies. We evaluate the proposed networks and the additional other architectures to verify the effectiveness of the proposed networks, and also compare them with several other methods on five challenging datasets. The experimental results demonstrate that our network models achieve the state-of-the-art performance through various temporal features. Additionally, we analyze a relation between the recognized actions and the multi-term TS-LSTM features by visualizing the softmax features of multiple parts.

1. Introduction

Human action recognition is one of many challenging tasks targeted by computer vision researchers. It has many important applications including video surveillance, human-computer interaction, game control, sports video analysis, etc. Although traditional studies about action recognition have been focused on recognizing actions from the monocular RGB video sequences, it is hard to fully capture the human action in 3D space by using monocular video sensors. With a rapid development of 3D data acquisition over the past few decades, lots of researches on human activity recognition from 3D data can have been actively performed [2].

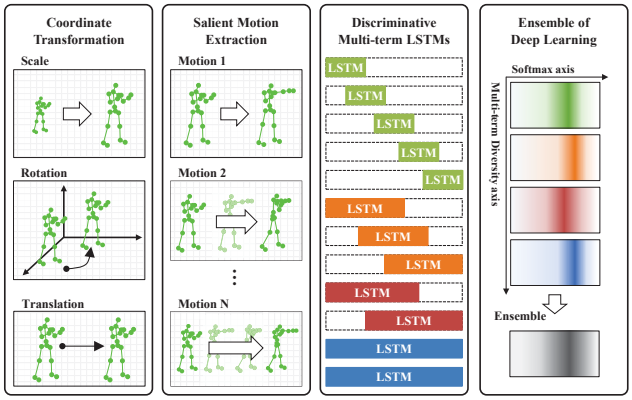


Figure 1: System overview of the proposed deep learning network. The main four phases of the system are composed of coordinate transformation, motion feature extraction, multi-term LSTMs and ensemble deep learning.

A human body can be represented by a stick figure called human skeleton, which consists of line segments linked by joints, and the motion of joints can provide the key to motion estimation and recognition of the whole figure [1]. Hence, if we can reliably extract and track a human skeleton in 3D space, action recognition can be performed by classifying the temporal movement of the skeleton. Currently, reliable joint coordinates can be obtained from the depth sensor using the real-time skeleton estimation algorithms [15, 22]. These kinds of effective pose estimation technologies have been facilitating studies on skeleton-based action recognition.

There are two related issues for human skeleton-based action recognition. The first one is a problem for input data

*Corresponding author (e-mail: slee@yonsei.ac.kr).

variations such as scale, rotation and translation, and the other is the modeling of the human actions that are variable, dynamic and similar with each other. Most of the existing skeleton-based action recognition methods use relative joint coordinates [17, 16, 6], which can overlook absolute movements of skeleton joints. For the modeling of human actions, recent researches show that Long Short-Term Memory (LSTM) networks [6, 24, 10] are superior to temporal pyramids [17, 12, 16] and hidden markov models [21, 20]. Nevertheless, these kinds of LSTM networks just model the overall temporal dynamics of skeleton joints without considering the detailed temporal dynamics of them.

In this paper, we propose novel ensemble temporal sliding LSTM networks for action recognition, in which the ensemble means a combination of various action attributes. Fig. 1 gives an overview of our model. Firstly, we transform the coordinates of input skeleton sequences so that the data can be robust to scale, rotation and translation. Secondly, instead of using the simple joint positions, we employ the motion features in terms of temporal differences, which help our networks to be focused on the actual skeleton movements. Thirdly, the motion features are processed with multi-term LSTMs containing short-term, medium-term and long-term LSTMs, which allow robustness to variable temporal dynamics. Finally, the multi-term LSTMs capture a variety of action dynamics through ensemble.

1.1. Related Works

In this subsection, we briefly review the existing literature closely related to the proposed model of dealing with the two main issues on human skeleton-based action recognition. The first is feature representation about the skeleton input sequences, and the other is modeling of the temporal dynamics for action recognition. Wang *et al*. [17] represented the human movement by means of the pairwise relative positions of the joints for more discriminative features. Cho *et al*. [4] normalized the orientation of skeletons so that each and every skeleton could have its root at the origin. Using the relative geometry between all pairs of body parts, Vemulapalli *et al*. [16] represented the 3D geometric relation of the body parts in Lie group. Du *et al*. [6] utilized the center among hip center, hip left and hip right joint coordinates as the origin of the coordinate system. These kinds of relative coordinate systems can misinterpret the actions when classifying the absolute movements of skeleton joints. Wang *et al*. [17] extracted the 3D joint position and the local occupancy pattern, and then they were processed with Fourier Temporal Pyramid (FTP) to represent temporal dynamics of the actions. Vemulapalli *et al*. [16] employed Dynamic Time Warping (DTW) and FTP to handle the issues such as rate variations, temporal misalignment, noise, etc. Instead of modeling temporal evolution of features, Luo *et al*. [12] proposed a new dictionary learning method with temporal pyramid matching for keeping the temporal information. Xia *et al*. [21] employed the histogram based representation of 3D human posture, and then recognized the actions using discrete Hidden Markov Model (HMM). Wu and Shao [20] extracted high level skeletal joint features, and then used them for estimating the emission probability of HMM to infer the action sequences.

Even though the methods of DTW, FTP and HMM are useful when dealing with temporal dynamics, the recent utilization of LSTM networks has been showing the superior performance to model the temporal dynamics than the traditional methods. Du *et al*. [6] proposed a hierarchical recurrent neural network, in which the temporal representations of low-level body parts were modeled and combined into the representations of high-level parts. Zhu *et al*. [24] developed an end-to-end fully connected deep LSTM network with the novel regularization to learn the co-occurrence features of skeleton joints. Liu *et al*. [10] introduced a new gating mechanism within LSTM to learn the reliability of sequential data and accordingly adjusted its effect on updating the long-term context information stored in the memory cell. Since all these researches generally observed only the long-term memory of human actions, it can be difficult to completely model various temporal dynamics including short-term, medium-term actions, etc.

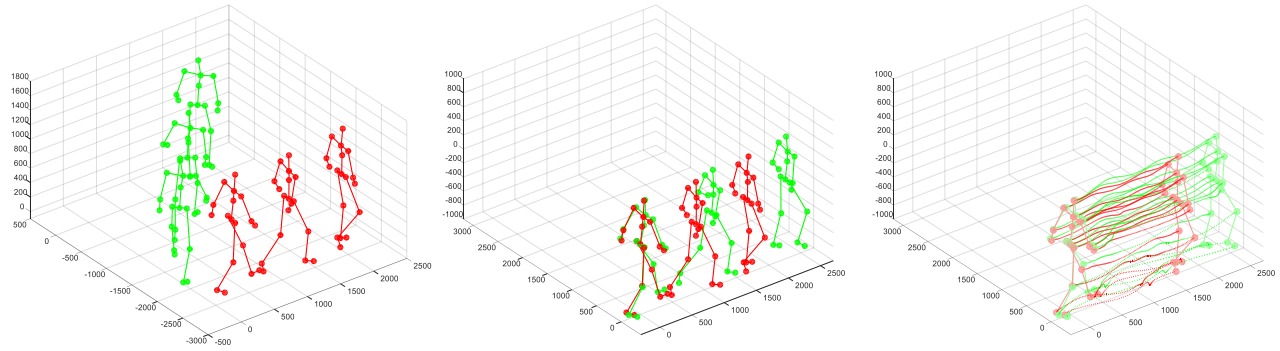
1.2. Contributions

We arrange the main contributions as follows:

- We investigate feature representation for human skeleton in order to obtain the robustness to various variations and extracting salient motions. Experimentally, it is demonstrated that the feature representation dramatically enhances the performance of action recognition.
- We utilize an ensemble of multi-term temporal sliding LSTM networks, which can capture short-term, medium-term, long-term temporal dependencies and even spatial skeleton pose dependency, separately. Unlike traditional ensemble studies, our models effectively learn various spatial and temporal dynamics in terms of different action attributes.
- We conduct comprehensive evaluations on the MSR Action3D dataset [9], UTKinect-Action dataset [21], NTU RGB+D dataset [14], Northwestern-UCLA dataset [19] and UWA3DII dataset [13]. The experimental results demonstrate that our network model significantly outperforms previously developed methods for skeleton-based action recognition.

2. System Model

In this section, initially, we introduce the feature representation of the proposed system, including a transforma-



(a) (b) (c)

Figure 2: Feature representation processes. (a) Original input skeleton frames ($\mathbf{s}_t$). (b) Transformed input skeleton frames ($\hat{\mathbf{s}}_t$). (c) Extracted salient motion features ($\mathbf{x}_t$).

tion of the input skeletons and an extraction of motion features. Next, we present the temporal sliding LSTM used as a specific module of the system. Finally, we explain the whole architecture including the training and testing processes.

2.1. Feature Representation

As shown in Fig. 2(a), the original input skeletons can go through an orientation misalignment when the skeletons are obtained. In other words, even though the skeletons are included in the same action category, the movements of the skeletons can have a different attribute due to the orientation misalignment. In order to solve this problem, we need to transform the original coordinate system into a human cognitive coordinate system, which can have an orientation consistency as depicted in Fig. 2(b).

Let $\mathbf{s}_t^i \in \mathbb{R}^{3\times 1}$ be the coordinates of the i^{th} joint of the t^{th} skeleton frame. The transformed skeleton joint coordinates are then given by

$$\hat{\mathbf{s}}_t^i = \mathbf{R}^{-1}(\mathbf{s}_t^i - \mathbf{o}_{\mathrm{R}}), \ \ \forall i \in J, \ \ \forall t \in T \tag{1}$$

where J and T denote the sets of the skeleton joint and frame indexes, respectively. In (1), the rotation matrix $\mathbf{R}$ and the origin of rotation $\mathbf{o}_{\mathrm{R}}$ are obtained as

$$\mathbf{R} = \left[\frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \middle| \frac{\mathbf{v}_2 - \mathrm{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)}{\|\mathbf{v}_2 - \mathrm{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)\|} \middle| \frac{\mathbf{v}_1 \times \mathbf{v}_2}{\|\mathbf{v}_1 \times \mathbf{v}_2\|} \right], \tag{2}$$

$$\mathbf{o}_{\mathrm{R}} = \left(\mathbf{s}_{t=0}^{\mathrm{H\_L}} + \mathbf{s}_{t=0}^{\mathrm{H\_R}}\right)/2, \tag{3}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$[1] are the vector vertical to the ground and to the difference vector between the hip left joint and the hip right joint of the initial skeleton in each sequence, respectively. In (2), $\mathrm{Proj}_{\mathbf{v}_1}(\mathbf{v}_2)$ and $\mathbf{v}_1 \times \mathbf{v}_2$ denote the vector projection of $\mathbf{v}_2$ onto $\mathbf{v}_1$ and the cross product of the two vectors, respectively. In (3), $\mathbf{s}_{t=0}^{\mathrm{H\_L}}$ and $\mathbf{s}_{t=0}^{\mathrm{H\_R}}$ denote the coordinates of the hip left and right joints of the initial skeleton of each sequence, respectively.

Fig. 2(c) shows the extraction process of the salient motion features. Instead of using the skeleton joint coordinates, we use the temporal differences between the two frames. While the skeleton joint coordinates just focus on current locations, the motion features can capture the actual movements of the skeleton joints [8]. Based on this insight, we additionally utilize the motion features as input features of the proposed architecture.

Let $\hat{\mathbf{s}}_t \in \mathbb{R}^{\mathrm{S_{IN}}\times 1}$ be the transformed skeleton coordinates of the t^{th} frame and $\mathrm{S_{IN}}$ be the input dimension size of the proposed system. The transformed skeleton coordinates are then obtained by

$$\hat{\mathbf{s}}_t = \mathrm{concat}\left(\left[\hat{\mathbf{s}}_t^0, \ \hat{\mathbf{s}}_t^1, ..., \ \hat{\mathbf{s}}_t^{|J|-1}\right], \ 0\right), \ \ \forall t \in T \tag{4}$$

where concat([elements], 0) and $|J|$ denote the concatenation along the 0^{th} axis of the elements and the number of elements of set J, respectively. The motion features are then obtained by

$$\mathbf{x}_t = \hat{\mathbf{s}}_t - \hat{\mathbf{s}}_{t-D}, \ \ \forall t \in T(t \geq D) \tag{5}$$

where D is the temporal difference offset. These motion features can become various forms according to D and are normalized through dividing them by $(D+1)$. We can use both the transformed skeleton coordinates and motion features as input features. They are scaled into the unit of centimeter, which makes our model perform well.

2.2. Temporal Sliding LSTM

Generally, LSTM networks have been used to model temporal dynamics [7]. Although the forget gates of LSTM

[1] In order to obtain a vector vertical to $\mathbf{v}_1$ on the plane of containing $\mathbf{v}_1$ and $\mathbf{v}_2$, we use the GramSchmidt process.

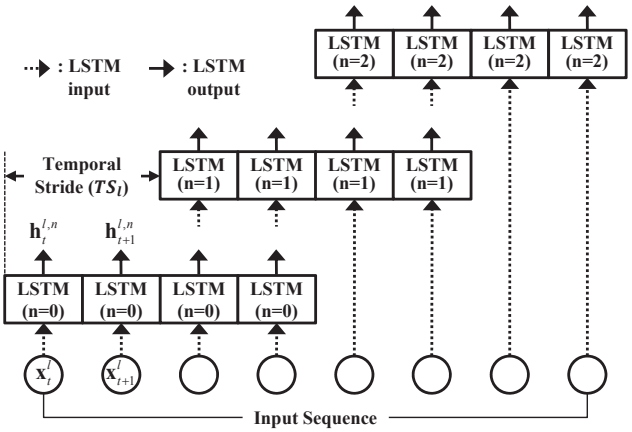


Figure 3: Conceptual diagram of the proposed l^{th} TS-LSTM module when $N_l = 3$, $W_l = 4$, and $TS_l = 2$.

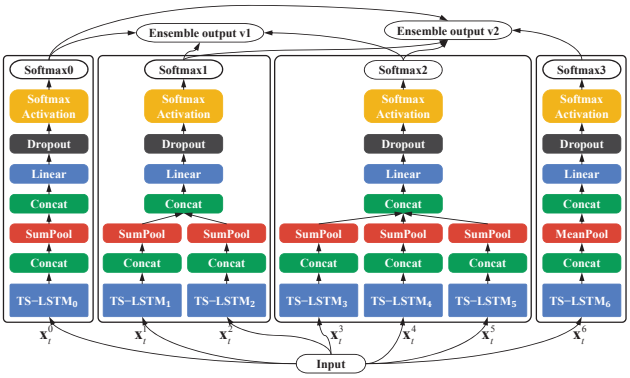


Figure 4: Whole architecture composed of short-term, medium-term, long-term and pose TS-LSTM modules where $l = 0$ for short-term, $l = 1, 2$ for medium-term, $l = 3, 4, 5$ for long-term and $l = 6$ for pose feature.

networks may help the short-term and medium-term dependencies to be acquired, it is actually almost impossible to completely forget the memories of LSTM cells. In order to model these various dependencies, we propose the Temporal Sliding LSTM (TS-LSTM) module. As shown in Fig. 3, since the l^{th} TS-LSTM module can have various LSTM network numbers (N_l), LSTM window sizes (W_l), and temporal strides (TS_l), it is very useful when classifying the actions with variable temporal dynamics. In other words, in case of recognizing the actions with variable sequence length, we only have to adjust the window size and the stride of TS-LSTM.

Let $\mathbf{x}_t^l$ be the input of the l^{th} TS-LSTM and D_l be the difference offset of it. Substituting D_l into D of (5), the input of the l^{th} TS-LSTM is selected as $\mathbf{x}_t^l = \hat{\mathbf{s}}_t - \hat{\mathbf{s}}_{t-D_l}$ ($0 \le l \le 5$) and $\mathbf{x}_t^l = \hat{\mathbf{s}}_t$ ($l = 6$) as shown in Fig. 4. The memory cell, three gates and the output of the t^{th} frame of the n^{th} LSTM of the l^{th} TS-LSTM are then obtained as

$$\mathbf{i}_t^{l,n} = \sigma\left(\mathbf{w}_{\mathbf{ix}}^{l,n}\mathbf{x}_t^{l,n} + \mathbf{w}_{\mathbf{ih}}^{l,n}\mathbf{h}_{t-1}^{l,n} + \mathbf{w}_{\mathbf{ic}}^{l,n}\mathbf{c}_{t-1}^{l,n} + \mathbf{b}_{\mathbf{i}}^{l,n}\right) \quad (6)$$

$$\mathbf{f}_t^{l,n} = \sigma\left(\mathbf{w}_{\mathbf{fx}}^{l,n}\mathbf{x}_t^{l,n} + \mathbf{w}_{\mathbf{fh}}^{l,n}\mathbf{h}_{t-1}^{l,n} + \mathbf{w}_{\mathbf{fc}}^{l,n}\mathbf{c}_{t-1}^{l,n} + \mathbf{b}_{\mathbf{f}}^{l,n}\right) \quad (7)$$

$$\mathbf{c}_t^{l,n} = \mathbf{f}_t^{l,n}\mathbf{c}_{t-1}^{l,n} + \mathbf{i}_t^{l,n}\tanh\left(\mathbf{w}_{\mathbf{cx}}^{l,n}\mathbf{x}_t^{l,n} + \mathbf{w}_{\mathbf{ch}}^{l,n}\mathbf{h}_{t-1}^{l,n} + \mathbf{b}_{\mathbf{c}}^{l,n}\right) \quad (8)$$

$$\mathbf{o}_t^{l,n} = \sigma\left(\mathbf{w}_{\mathbf{ox}}^{l,n}\mathbf{x}_t^{l,n} + \mathbf{w}_{\mathbf{oh}}^{l,n}\mathbf{h}_{t-1}^{l,n} + \mathbf{w}_{\mathbf{oc}}^{l,n}\mathbf{c}_{t-1}^{l,n} + \mathbf{b}_{\mathbf{o}}^{l,n}\right) \quad (9)$$

$$\mathbf{h}_t^{l,n} = \mathbf{o}_t^{l,n}\tanh\left(\mathbf{c}_t^{l,n}\right) \quad (10)$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathbf{i}_t^{l,n}$, $\mathbf{f}_t^{l,n}$, $\mathbf{c}_t^{l,n}$, $\mathbf{o}_t^{l,n}$ and $\mathbf{h}_t^{l,n}$ are respectively the input gate, forget gate, cell activation, output gate and output vectors of the t^{th} frame of the n^{th} LSTM of the l^{th} TS-LSTM. In (6)-(9), all the matrices $\mathbf{w}_{\mathbf{mn}}^{l,n}$ are the connection weights from $\mathbf{n}$ to $\mathbf{m}$ of the n^{th} LSTM of the l^{th} TS-LSTM.

2.3. Proposed Network Architecture

As shown in Fig. 4, the proposed architecture consists of multiple TS-LSTM modules determined by D_l, W_l, and TS_l. The architecture has different structures according to the type of the process such as training and testing. In the training process, each part is mainly composed of the TS-LSTM layer, the sum pooling (SumPool) layer, the linear (LN) layer, the softmax activation layer and the cross-entropy layer to calculate the cost function. In the testing process, instead of using the cross-entropy layer, we utilize the average ensemble of the various softmax outputs as the final output inspired by GoogLeNet [5].

Let $\mathbf{x}_t^{l,m}$ be $\mathbf{x}_t^l$ of the m^{th} sequence. Substituting $\mathbf{x}_t^{l,m}$ into $\mathbf{x}_t^l$ of (6)-(9), $\mathbf{h}_t^{l,n}$ of (10) can be written as $\mathbf{h}_t^{l,n,m}$. The SumPool and MeanPool values of the m^{th} sequence of the n^{th} LSTM of the l^{th} TS-LSTM are then obtained as

$$\mathbf{q}_S^{l,m} = \sum_{t=0}^{W_l-1} \text{concat}\left(\left[\mathbf{h}_{n\cdot TS_l+t}^{l,n,m}\right]_{n=0}^{n=N_l-1}, 0\right) \quad (11)$$

$$\mathbf{q}_M^{l,m} = \mathbf{q}_S^{l,m}/W_l \quad (12)$$

where $[(\cdot)_n]_{n=0}^{n=N_l-1} = [(\cdot)_0, (\cdot)_1, ..., (\cdot)_{N_l-1}]$. The MeanPool value of the m^{th} sequence of the n^{th} LSTM of the l^{th} TS-LSTM, $\mathbf{q}_M^{l,m} = \mathbf{q}_S^{l,m}/W_l$. In each part, the concatenation of the SumPool and MeanPool values of the m^{th} sequence is written as

$$\mathbf{r}_{\mathrm{S}}^m = \left[\mathbf{q}_S^{0,m}\right]^T,\ \mathbf{r}_{\mathrm{M}}^m = \left[\text{concat}\left(\left[\mathbf{q}_S^{1,m}, \mathbf{q}_S^{2,m}\right], 1\right)\right]^T,$$

$$\mathbf{r}_{\mathrm{L}}^m = \left[\text{concat}\left(\left[\mathbf{q}_S^{3,m}, \mathbf{q}_S^{4,m}, \mathbf{q}_S^{5,m}\right], 1\right)\right]^T,\ \mathbf{r}_{\mathrm{P}}^m = \left[\mathbf{q}_M^{6,m}\right]^T \quad (13)$$

where $[\cdot]^T$ means the transpose operation. The linear activation of each part is then obtained as

$$\mathbf{a}_{\mathrm{S}}^{m} = \mathbf{w}_{\mathrm{S}} \cdot \mathbf{r}_{\mathrm{S}}^{m} + \mathbf{b}_{\mathrm{S}} \tag{14}$$

$$\mathbf{a}_{\mathrm{M}}^{m} = \mathbf{w}_{\mathrm{M}} \cdot \mathbf{r}_{\mathrm{M}}^{m} + \mathbf{b}_{\mathrm{M}} \tag{15}$$

$$\mathbf{a}_{\mathrm{L}}^{m} = \mathbf{w}_{\mathrm{L}} \cdot \mathbf{r}_{\mathrm{L}}^{m} + \mathbf{b}_{\mathrm{L}} \tag{16}$$

$$\mathbf{a}_{\mathrm{P}}^{m} = \mathbf{w}_{\mathrm{P}} \cdot \mathbf{r}_{\mathrm{P}}^{m} + \mathbf{b}_{\mathrm{P}} \tag{17}$$

where $\mathbf{w}$ and $\mathbf{b}$[2] are the weight and bias terms of the LN layer, respectively. Let $\mathbf{a}_{\mathrm{S}}^{m,k}$, $\mathbf{a}_{\mathrm{M}}^{m,k}$, $\mathbf{a}_{\mathrm{L}}^{m,k}$ and $\mathbf{a}_{\mathrm{P}}^{m,k}$ be the k^{th} action class values of $\mathbf{a}_{\mathrm{S}}^{m}$, $\mathbf{a}_{\mathrm{M}}^{m}$, $\mathbf{a}_{\mathrm{L}}^{m}$ and $\mathbf{a}_{\mathrm{P}}^{m}$, respectively. The linear activation values of the m^{th} sequence are then normalized with the softmax function:

$$\Pr\left(c|\mathbf{a}_{\mathrm{S}}^{m}\right) = \frac{\exp\left(\mathbf{a}_{\mathrm{S}}^{m,c}\right)}{\sum_{k=0}^{N_C-1}\exp\left(\mathbf{a}_{\mathrm{S}}^{m,k}\right)} \tag{18}$$

$$\Pr\left(c|\mathbf{a}_{\mathrm{M}}^{m}\right) = \frac{\exp\left(\mathbf{a}_{\mathrm{M}}^{m,c}\right)}{\sum_{k=0}^{N_C-1}\exp\left(\mathbf{a}_{\mathrm{M}}^{m,k}\right)} \tag{19}$$

$$\Pr\left(c|\mathbf{a}_{\mathrm{L}}^{m}\right) = \frac{\exp\left(\mathbf{a}_{\mathrm{L}}^{m,c}\right)}{\sum_{k=0}^{N_C-1}\exp\left(\mathbf{a}_{\mathrm{L}}^{m,k}\right)} \tag{20}$$

$$\Pr\left(c|\mathbf{a}_{\mathrm{P}}^{m}\right) = \frac{\exp\left(\mathbf{a}_{\mathrm{P}}^{m,c}\right)}{\sum_{k=0}^{N_C-1}\exp\left(\mathbf{a}_{\mathrm{P}}^{m,k}\right)} \tag{21}$$

where c and N_C are the corresponding class index and the total number of action classes, respectively.

In order to find the maximum likelihood of all the training samples, we apply the cross-entropy function into two objective functions:

$$e_1 = -\sum_{m=0}^{N_M-1}\sum_{c=0}^{N_C-1} y_c^m \ln\{\Pr\left(c|\mathbf{a}_{\mathrm{S}}^{m}\right)\Pr\left(c|\mathbf{a}_{\mathrm{M}}^{m}\right)\Pr\left(c|\mathbf{a}_{\mathrm{L}}^{m}\right)\} \tag{22}$$

$$e_2 = e_1 - \sum_{m=0}^{N_M-1}\sum_{c=0}^{N_C-1} y_c^m \ln\{\Pr\left(c|\mathbf{a}_{\mathrm{P}}^{m}\right)\} \tag{23}$$

where y_c^m and N_M are the ground-truth label of the m^{th} sample and the total number of training samples. We train the models by minimizing the two objective functions, separately.

In the testing process, the ensemble output v1 is obtained with average ensemble among the three linear activation values such as $\Pr\left(c|\mathbf{a}_{\mathrm{S}}^{m}\right), \Pr\left(c|\mathbf{a}_{\mathrm{M}}^{m}\right)$ and $\Pr\left(c|\mathbf{a}_{\mathrm{L}}^{m}\right)$ and the ensemble output v2 is obtained with average ensemble among the four linear activation values ($\Pr\left(c|\mathbf{a}_{\mathrm{S}}^{m}\right), \Pr\left(c|\mathbf{a}_{\mathrm{M}}^{m}\right), \Pr\left(c|\mathbf{a}_{\mathrm{L}}^{m}\right)$ and $\Pr\left(c|\mathbf{a}_{\mathrm{P}}^{m}\right)$), separately.

[2]The subscripts of $\mathbf{w}$ and $\mathbf{b}$ such as S, M, L, and P denote the short-term, medium-term, long-term and pose parts.

3. Experiments

In this section, we evaluate the proposed model and compare with several recent methods on the five benchmark datasets: MSR Action3D dataset [9], UTKinect-Action dataset [21], NTU RGB+D dataset [14], Northwestern-UCLA dataset [19] and UWA3DII dataset [13]. We also analyze a relation between the recognized actions and the multi-term TS-LSTM features.

In order to show the effects of the proposed techniques, we conduct experiments under five different architectures. The first one is the simple LSTM used as the baseline of LSTM for skeleton-based action recognition. The second one applies the Human Cognitive Coordinate (HCC) into the first one, which shows the effect of the HCC. The third one additionally applies the Salient Motion Feature (SMF) into the second one, which shows the effect of the SMF. The fourth one is the proposed ensemble TS-LSTM v1 using the cost of (22). The final one is the proposed ensemble TS-LSTM v2 using the cost of (23).

3.1. Datasets and Parameter Settings

MSR Action3D dataset: This dataset was captured using a depth sensor like Kinect. It consists of 20 actions performed by 10 subjects for two or three times. Altogether, there are 557 valid action sequences, and each frame in a sequence is composed of 20 skeleton joints.

UTKinect-Action dataset: This dataset was captured using a single stationary Kinect. It consists of 10 actions performed by 10 different subjects, and each subject performed every action twice. Altogether, there are 199 action sequences, and the 3D locations of 20 joints are given. This is regarded as a challenging dataset because of variations in the view point and high intra-class variations.

NTU RGB+D dataset: This dataset was captured by 3 Microsoft Kinect v2 cameras. It contains 60 action classes in total, which are divided into three major groups: 40 daily actions, 9 health-related actions and 11 mutual actions. Each sequence contains the 3D locations of 25 skeleton joints. It is very challenging due to the large intra-class and view point variations.

Northwestern-UCLA dataset: This dataset was captured simultaneously by 3 Microsoft Kinect v1 cameras. It contains 1494 sequences covering 10 action categories. Each action is performed one to six times by ten subjects. This dataset contains data taken from a variety of viewpoints.

UWA3DII dataset: This dataset was captured by 4 Microsoft Kinect v1 cameras. It contains 30 human actions performed four times by ten subjects. Each action is observed from front view, left and right side views, and top view. The dataset is challenging because of varying viewpoints, self-occlusion and high similarity among actions.

Table 1: Parameter settings of the proposed models. TS-LSTM$_l$ is the l^{th} TS-LSTM within the proposed models, the parameters of which are (D_l, W_l, TS_l). LN means the number of hidden units of the TS-LSTM concatenation of each part, and the subscripts of LN such as S, M, L and P denote the short-term, medium-term, long-term and pose parts, respectively. N_T is a maximal skeleton frame length of all sample data.

	TS-LSTM$_0$	TS-LSTM$_1$	TS-LSTM$_2$	TS-LSTM$_3$	TS-LSTM$_4$	TS-LSTM$_5$	TS-LSTM$_6$	LN$_\mathrm{S}$	LN$_\mathrm{M}$	LN$_\mathrm{L}$	LN$_\mathrm{P}$	N_T
MSR	(1, 15, 15)	(1, 40, 35)	(5, 36, 35)	(1, 75, –)	(5, 71, –)	(10, 66, –)	(0, 45, 22)	100	80	60	80	76
UTKi	(1, 27, 20)	(1, 63, 50)	(5, 59, 50)	(1, 113, –)	(5, 109, –)	(10, 104, –)	(0, 100, 25)	100	80	60	20	114
NTU	(1, 77, 74)	(1, 103, 98)	(5, 99, 98)	(1, 154, 145)	(5, 150, 145)	(10, 145, 145)	(0, 77, 74)	500	400	300	200	300
UCLA	(1, 40, 25)	(1, 100, 50)	(5, 70, 45)	(1, 198, –)	(5, 170, –)	(10, 150, –)	(0, 100, 50)	240	340	518	200	201
UWA	(2, 42, 20)	(1, 88, 40)	(5, 84, 40)	(1, 164, –)	(5, 155, –)	(10, 145, –)	(0, 42, 20)	294	344	464	294	167

Table 1 shows the parameter setting of our main proposed model. We use all the skeleton joints as the input of each dataset. All the sequences on the MSR Action3D dataset (MSR) and the UTKinect-Action dataset (UTKi) are used for the experiments. We exclude invalid sequences from the NTU RGB+D dataset (NTU), the Northwestern-UCLA dataset (UCLA) and the UWA3DII dataset (UWA) because they have shorter sequence length than only 10 frames. In different architectures, the LSTMs perform sum pooling or mean pooling according to the input feature type and the probability to keep hidden units of each LN layer has a value of 0.4.

3.2. Results and Comparisons

MSR Action3D dataset: We follow the standard protocol provided in [9]. In this standard protocol, the dataset is divided into three action sets such as Action Set1 (AS1), Action Set2 (AS2) and Action Set3 (AS3). We use the samples of subjects 1, 3, 5, 7, 9 for training and the samples of subjects 2, 4, 6, 8, 10 for testing. As shown in Table 2, the proposed ensemble TS-LSTM v1 and v2 achieve the significantly enhanced average accuracies (96.63 %) and (97.22 %) compared with the previous methods, respectively.

Table 2: Experimental result comparison on the MSR Action3D dataset.

Method	AS1	AS2	AS3	Ave.
Bag of 3d points [9]	72.9	71.9	79.2	74.7
Lie group [16]	**95.29**	83.87	98.22	92.46
HBRNN [6]	93.33	94.64	95.50	94.49
ST-LSTM + Trust Gate [10]	N/A	N/A	N/A	94.8
LSTM	70.48	71.43	72.07	71.33
LSTM + HCC	76.19	74.11	81.98	77.43
LSTM + HCC + SMF	92.38	90.18	92.79	91.78
Ensemble TS-LSTM v1	95.24	95.54	99.10	96.63
Ensemble TS-LSTM v2	95.24	**96.43**	**100**	**97.22**

As shown in Table 2, the addition of HCC and SMF into LSTM makes the average accuracy increase by 6.1 % and 14.35 %, respectively, which indicates that our feature representation is very useful on this dataset. The ensemble TS-LSTM models are around 2 % higher than the previous best method [10]. Moreover, our network models are superior to the other methods on almost every action set, including AS2 and AS3, which means that the proposed models are more robust to various actions than the other methods.

UTKinect-Action dataset: We follow the protocol [25], in which half of the subjects are used for training and the remaining are used for testing. The first 5 subjects are used for training while the last 5 subjects are used for testing. As shown in Table 3, our models achieve the higher results (95.96 %) and (96.97 %) compared with the previous best model [23] (95.96 %), respectively.

Table 3: Experimental result comparison on the UTKinect-Action dataset.

Method	Acc.	Method	Acc.
Skeleton joint features [25]	87.9	LSTM	60.61
Histograms of 3D joints [21]	90.92	LSTM + HCC	72.73
Elastic functional coding [3]	94.87	LSTM + HCC + SMF	93.94
ST-LSTM + Trust Gate [10]	95.0	Ensemble TS-LSTM v1	95.96
Geometric features [23]	95.96	Ensemble TS-LSTM v2	**96.97**

Different from the MSR Action3D dataset, it should be noted that the addition of HCC and SMF into LSTM makes the average accuracy increase by 12.12 % and 21.21 %, respectively, which indicates that our feature representation on this dataset is more effective than that on the MSR Action3D dataset. Both ensemble TS-LSTM v1 and v2 are around 1 % higher than the previous best method [10].

NTU RGB+D dataset: We follow two standard evaluation protocols [14]. One is cross-subject (CS) evaluation, where half of the subjects are used for training, and the remaining is used for testing. The second is cross-view (CV) evaluation where two viewpoints are used for training, and one is used for testing. Since the original basis of HCC can be different due to the different viewpoints, we use the trunk of initial skeleton of each sequence as the basis of HCC instead of the vertical vector. As shown in Table 4, our models achieve the competitive results compared with the previous methods, which indicates that it can be better to model various temporal dynamics even for this challenging dataset.

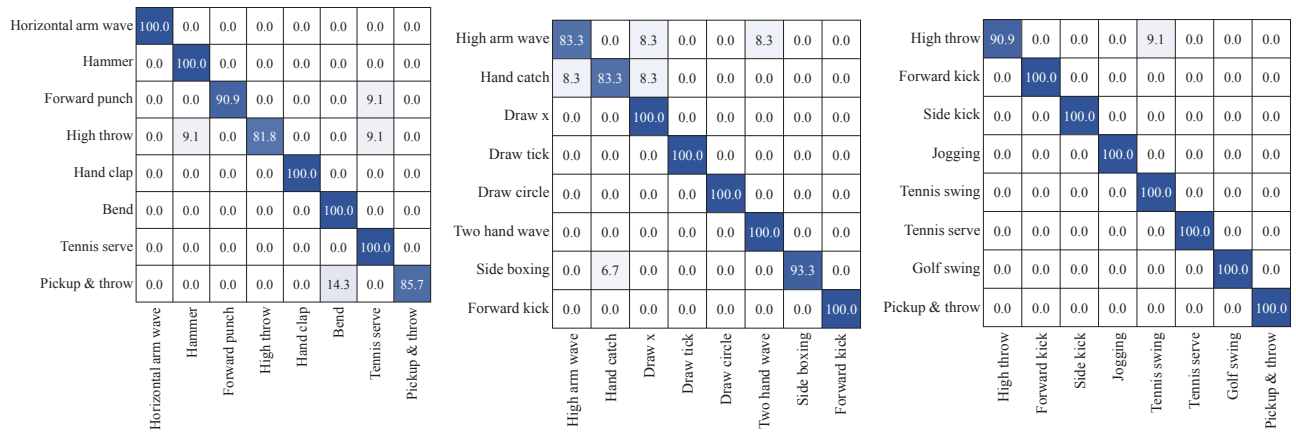


Figure 5: Confusion matrices of the ensemble TS-LSTM v1 according to action set on the MSR Action3D dataset. The row and column of each confusion matrix denote the ground truth and the prediction, respectively. (a) AS1. (b) AS2. (c) AS3.

Table 4: Experimental results on the NTU RGB+D dataset.

Method	CS Acc.	CV Acc.
HBRNN [6] (reported by [14])	59.07	63.97
Part-aware LSTM [14]	62.93	70.27
ST-LSTM + Trust Gate [10]	69.2	77.7
Geometric features [23]	70.26	82.39
Enhanced skeleton visualization [11]	**75.97**	**82.56**
Ensemble TS-LSTM v1	73.88	80.40
Ensemble TS-LSTM v2	74.60	81.25

Northwestern-UCLA dataset: We follow the evaluation protocol [19]. We use samples from the first two cameras as training data, and the samples from the third camera as test data. As shown in Table 5, the proposed ensemble TS-LSTM v1 and v2 achieve the competitive results (85.99 %) and (89.22 %) compared with the previous best model [11] (86.09 %) on the Northwestern-UCLA dataset.

Table 5: Experimental results on the Northwestern-UCLA dataset.

Method	Accuracy (%)
Lie group [16] (reported by [11])	74.20
Actionlet ensemble [18] (reported by [11])	76.00
HBRNN-L [6] (reported by [11])	78.52
Enhanced skeleton visualization [11]	86.09
Ensemble TS-LSTM v1	85.99
Ensemble TS-LSTM v2	**89.22**

UWA3DII dataset: We follow the cross view protocol [13]. We use samples from two views as training data, and samples from the two remaining views as test data. As shown in Table 6, our models achieve the significantly enhanced mean accuracies (72.4 %) and (75.6 %) compared with the previous best method [11] (66.0 %) on the UWA3DII dataset.

3.3. Result Analysis

When analyzing the experimental results, we use the MSR Action3D dataset due to its comprehensive composition of action sets. As shown in Fig. 5(c), almost every action on AS3 is correctly classified except only one case of the action 'High throw", which is very similar to the action "Tennis swing" even for the human perception. In Fig. 5(b), the action "Side boxing" is misclassified to "Hand catch" while the action "Hand catch" is miscategorized to "High arm wave" or "Draw x". In Fig. 5(a), the actions "Forward punch" and "Tennis serve" are overlapped quite a lot in the sequences. Similar to this, the actions such as "Bend", "Pickup & throw", "High throw" and "Hammer" also share quite a large overlap in the sequences. Nevertheless, the proposed ensemble TS-LSTM v1[3] classifies these similar actions to some degree by using the multiple TS-LSTM networks.

In order to analyze the proposed ensemble TS-LSTM v1 in detail, we visualize the softmax outputs of the three parts containing the TS-LSTMs on the AS1 test data of the MSR Action3D dataset as depicted in Fig. 6. Overall, the diagonal probabilities of Softmax2 with long-term LSTMs are higher than those of Softmax0 with short-term LSTMs and

[3]Except that there is a little more performance enhancement, confusion matrices of the proposed ensemble TS-LSTM v2 are similar to those of the proposed ensemble TS-LSTM v1. Thus, we analyze the effect of ensemble focusing on the proposed ensemble TS-LSTM v1.

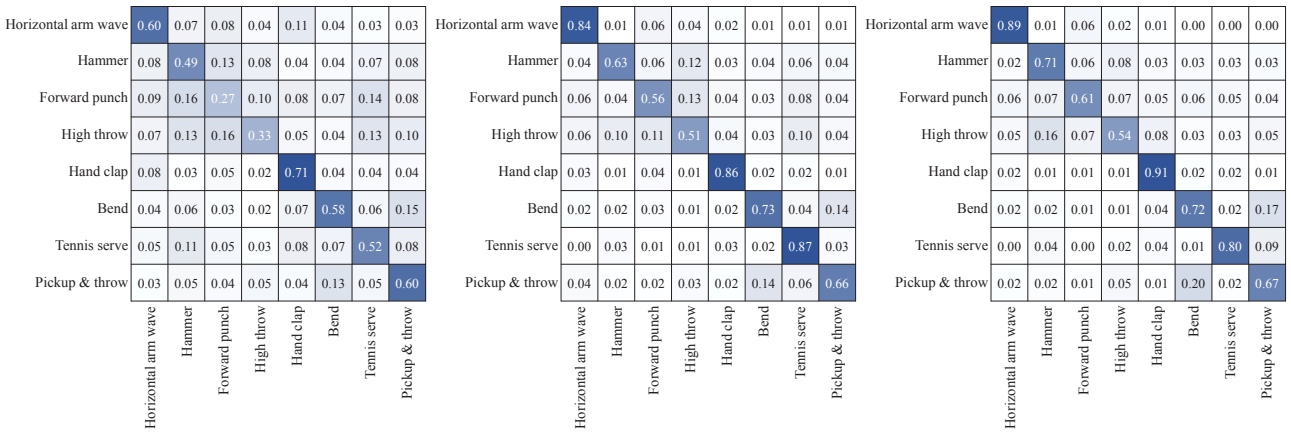


Figure 6: Softmax average probabilities of the ensemble TS-LSTM v1 according to action category on the AS1 test data of the MSR Action3D dataset. The row and the column of each confusion matrix denote the ground truth and the prediction, respectively. (a) Softmax0. (b) Softmax1. (c) Softmax2.

Table 6: Experimental results on the UWA3DII dataset.

Training views	V_1 & V_2		V_1 & V_3		V_1 & V_4		V_2 & V_3		V_2 & V_4		V_3 & V_4		Mean
Test view	V_3	V_4	V_2	V_4	V_2	V_3	V_1	V_4	V_1	V_3	V_1	V_2	
Actionlet ensemble [18] (reported by [11])	45.0	40.4	35.1	36.9	34.7	36.0	49.5	29.3	57.1	35.4	49.0	29.3	39.8
Lie group [16] (reported by [11])	49.4	42.8	34.6	39.7	38.1	44.8	53.3	33.5	53.6	41.2	56.7	32.6	43.4
Enhanced skeleton visualization [11]	66.4	68.1	56.8	66.1	58.8	66.2	74.2	67.0	76.9	64.8	72.2	54.0	66.0
Ensemble TS-LSTM v1	64.9	76.8	69.3	**78.3**	67.7	66.5	76.1	78.3	77.6	65.0	**81.2**	66.5	72.4
Ensemble TS-LSTM v2	**72.1**	**79.1**	**74.0**	77.6	**75.6**	**70.1**	**79.6**	**79.9**	**83.9**	**66.1**	79.2	**69.7**	**75.6**

Softmax1 with medium-term LSTMs, which indicates that the global temporal features have relatively more influence on the performance than the local temporal features. However, Softmax0 and Softmax1 somtimes produce lower misclassification rates compared with Softmax2, which makes the model less prone to overfitting to some certain actions. For example, Softmax0 and Softmax1 have lower misclassification probabilities of the action "Pickup & throw" to the action "Bend" than Softmax2, which indicates that Softmax0 and Softmax1 can compensate the weakness of Softmax2. Consequently, our ensemble TS-LSTM v1 can distinguish even the very closely similar actions by using various dicriminative features such as the short-term, medium-term and long-term temporal features.

4. Conclusion

Initially, we have transformed a human skeleton into the human cognitive coordinate system by using the Gram-Schmidt process, and extracted the pose and motion features to capture various spatial and temporal dynamics. After that, we have presented the novel utilization method of massive LSTMs according to time-step size, including training and testing processes. We have experimentally showed that the proposed networks outperform various state-of-the-art action recognition methods on the five different datasets.

As future work, we will investigate solutions for the failure cases on the datasets. A possible direction will be the analysis of TS-LSTM features for the proposed model to perform well on the failure cases. Another direction is to adjust the parameters of the proposed ensemble TS-LSTM networks to capture various spatial and temporal dynamics. Other directions include the application of various data augmentation techniques into the proposed models.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.R7120-17-1007, SIAT CCTV Cloud Platform).

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. *Proc. IEEE Workshop Non-Rigid and Articulated Motion*, pages 90–102, 1997.

[2] J. K. Aggarwal and L. Xia. Human activity recognition from 3d data: A review. *Pattern Recognition Letters*, 48:70–80, 2014.

[3] R. Anirudh, P. Turaga, J. Su, and A. Srivastava. Elastic functional coding of human actions: From vector-fields to latent variables. *In CVPR*, pages 3147–3155, 2015.

[4] K. Cho and X. Chen. Classifying and visualizing motion capture sequences using deep neural networks. *In Conference on Computer Vision Theory and Applications (VISASPP)*, 2014.

[5] C.Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *In CVPR*, pages 1–9, 2015.

[6] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. *In CVPR*, pages 1110–1118, 2015.

[7] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[8] H. Kim, S. Lee, and A. C. Bovik. Saliency prediction on stereoscopic videos. *IEEE Transactions on Image Processing*, 23(4):1476–1490, 2014.

[9] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. *In IEEE International Workshop on CVPR*, pages 9–14, 2010.

[10] J. Liu, A. Shahroudy, D. Xu, and G. Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. *In ECCV*, pages 816–833, 2016.

[11] M. Liu, H. Liu, and C. Chen. Enhanced skeleton visualization for view invariant human action recognition. *In PR*, 68:346–362, 2017.

[12] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. *In ICCV*, pages 1809–1816, 2013.

[13] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian. Histogram of oriented principal components for cross-view action recognition. *In IEEE TPAMI*, 38(12):2430–2443, 2016.

[14] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. *In CVPR*, pages 1010–1019, 2016.

[15] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[16] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. *In CVPR*, pages 588–595, 2014.

[17] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. *In CVPR*, pages 1290–1297, 2012.

[18] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3d human action recognition. *In IEEE TPAMI*, 36(5):914–927, 2014.

[19] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. Zhu. Super normal vector for activity recognition using depth sequences. *In CVPR*, pages 2649–2659, 2014.

[20] D. Wu and L. Shao. Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. *In CVPR*, pages 724–731, 2014.

[21] L. Xia, C. C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27, 2012.

[22] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. *In CVPR*, pages 2345–2352, 2014.

[23] S. Zhang, X. Liu, and J. Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. *In WACV*, pages 148–157, 2017.

[24] 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. *In AAAI*, pages 3697–3703, 2016.

[25] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. *In CVPR*, pages 486–491, 2013.