GRAPH CONVOLUTIONAL LSTM MODEL FOR SKELETON-BASED ACTION RECOGNITION

Han Zhang[1,2] *Yonghong Song*[2] *Yuanlin Zhang*[2]

[1]School of Software Engineering, Xi'an Jiaotong University, Xi'an, China
[2]Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China

ABSTRACT

Skeleton-based action recognition has made impressive progress these years. Yet few methods consider spatial configuration of joints and temporal correlation meanwhile as a unity. To model action sequences in a way which regard both two dimensions, a Graph Convolutional Long Short Term Memory Networks (GC-LSTM) model is proposed in this paper, which automatically learns spatiotemporal features to model the action. Our model introduces the GCN operation into conventional RNN unit including graph convolution at each time step for input-to-state and state-to-state transition. Plenty of experiment analyses show that the proposed GC-LSTM model strives (1) to focus more on discriminative parts at discriminative frames and (2) to be insensitive to the redundant parts which are irrelevant for recognition. Moreover, several methods are compared with ours on two publicly available datasets and experimental results demonstrate that our model achieves the state-of-the-art performance.

Index Terms— Spatiotemporal feature, GC-LSTM, skeleton, action recognition

1. INTRODUCTION

As a significant and challenging direction in computer vision research, human action recognition plays a role in many applications, such as video understanding, human-computer interaction, intelligent video surveillance and robot vision [1][2]. In recent years, there are many effective approaches to recognize human actions, and the input data modality can be roughly categorized into RGB videos [3] and 3D skeleton sequences [4]. Furthermore, the skeleton sequences can be obtained by the Microsoft Kinect [5] and human pose estimation algorithms [6]. In this paper, we focus on recognizing human actions from 3D skeleton sequences (Fig. 1).

There are roughly two categories for skeleton based action recognition methods: RNN-based methods and CNN-based methods.

RNN-based methods simply employ the joint coordinates at individual time steps to form feature vectors to which temporal analysis is applied thereon. However, the capability of these methods is limited as they do not explicitly exploit the spatial relationships among the joints, which is crucial for understanding human actions. Hierarchical RNN [4] is proposed to learn hierarchical motion representations from skeleton sequences. [7] designs a fully connected deep LSTM network with a regularization scheme to learn the co-occurrence features of skeleton joints. Zhang et al. [8] exploit a view adaptive LSTM model, which enables the network itself to adapt to the most suitable observation viewpoints from end to end.

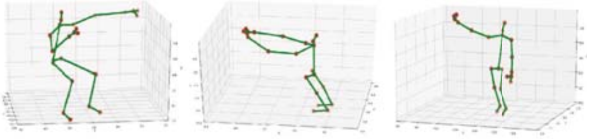

Fig. 1. Skeleton representations of the action "taking a selfie" from NTU dataset captured from different viewpoints and performed by different subjects.

[3][9][10] exploit CNNs for skeleton based action recognition, which can capture the relationship of the neighboring frames and relationship of neighboring joints. CNN-based methods focus more on local features extraction while less on the temporal dynamics, making no discrimination on all frames. A common baseline means the action sequence is first mapped into an image, and then CNNs are used to recognize the category of the still image [3].

In recent years, great progress has been made in GCN research [11][12], which deals with the graphs of non euclidean data. In the field of action recognition, new methods attempting to take advantage of the natural connections relationship among joints have been developed [13][14]. [13] proposes a ST-GCN to learn both the spatial and temporal patterns from data. [14] models the frame selection as a progressive process through deep reinforcement learning and employs the graph-based convolutional neural network to capture the dependency among the joints for action recognition. These methods show encouraging improvement; nevertheless, few method further focuses on extracting spatiotemporal features.

To move beyond such limitations, we need a new method that can automatically capture the patterns embedded in the spatial configuration of the joints as well as their temporal

978-1-5386-9552-4/19/$31.00 ©2019 IEEE
DOI 10.1109/ICME.2019.00078

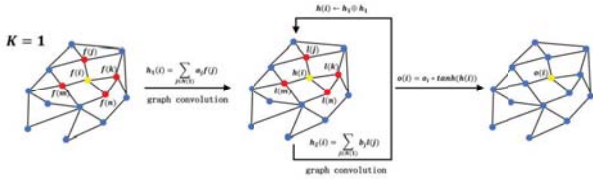

Fig. 2. The overview of GC-LSTM. The hidden state H_t will be generated through the way of input-to-state graph convolution and state-to-state graph convolution at each time step, which considers both spatial and temporal information. K=1 indicates that one-hop neighborhood's features are used to calculate the node feature.

dynamics. In the course that how an observer judges the action of the subject, the observer combines fragmented information of the body poses of a subject at certain time slices together to recognize the action category. For instance, when an observer finds that a subject raises his arm(s) at t_1 and looks at his raised arm(s) with hand(s) holding something at $t_2 (t_2 > t_1)$, he will know the subject is "taking a selfie", no matter the subject takes a selfie with one hand or two hands, sitting or standing. Therefore, it is crucial to combine spatial and temporal information together for action recognition. As mentioned, the skeletons are in the form of graphs instead of a 2D or 3D grids. Furthermore, it's necessary to select critical body part and discard the redundancy part. So it is intuitive to use Graph Neural networks (GCNs) to deal with. And RNNs have an advantage over temporal series analysis task. Therefore, we hope to combine the strengths of them two.

In this paper, we propose to extend graph convolutional neural networks and recurrent neural networks to the graph convolutional long short term memory networks (GC-LSTM) to extract spatial-temporal features for action recognition. The main contributions of this work lie in two aspects: 1) We propose a GC-LSTM model, a generic graph-based RNN formulation for modeling dynamic skeletons, to synthesize the strengths of graph convolutional networks and recurrent neural networks. 2) On two benchmark datasets for skeleton-based action recognition, the proposed model achieves superior performance compared to the state-of-the-art methods.

2. METHODOLOGY

Existing approaches have verified the effectiveness of CNNs for static image recognition. In tasks such as precipitation nowcasting, [15] proposes a convolutional LSTM (ConvLSTM) can simultaneously learn spatial and temporal dynamic features, and it has a competitive performance for the problem. We are motivated to be the first to introduce the appealing property of GCNs to RNN-based methods for skeleton based action recognition. This attempt finally contributes to the GC-LSTM model.

2.1. Pipeline overview

Skeleton based data is usually represented by a sequence of frames, each of which will have a set of joint coordinates. Given the sequences of body joints in the form of 2D or 3D coordinates, we construct a graph with joints as graph nodes and natural connectivities in human body structures as graph edges. The input to the GC-LSTM is therefore the joint coordinate vectors on the graph nodes. This can be considered as an analog to image based CNNs where the input is formed by pixel intensity vectors residing on the 2D image grid. Multiple layers of GC-LSTM operations (Fig. 2) will be applied on the input data, generating high-level feature maps as a form of graph. The operations allow discriminative body part information at discriminative frame to be integrated along both the spatial and the temporal dimension. Then the feature maps will be sent to a conventional CNN to be classified into the corresponding action category. The whole model is trained in an end-to-end manner with backpropgation. We will now detail how we construct the skeleton graph and go over the components in the GC-LSTM model.

2.2. Skeleton Graph Construction

Since the human body can be considered as an articulated system consisting of hinged joints and rigid bones, which inherently lies in a graphed-based structure, we construct a graph $\mathcal{G}(x, W)$ to model the human body for each single frame as illustrated in Fig. 1, where $x \in \mathbb{R}^{N\times 3}$ contains the 3D coordinates of the N joints and W is a $N \times N$ weighted adjacency matrix:

$$\omega_{ij} = \begin{cases} \alpha, & \text{if } i = j \\ \beta, & \text{if joint i and joint j are connected} \\ \gamma, & \text{if joint i and joint j are disconnected} \end{cases} \tag{1}$$

Here we set $\omega_{ii} = \alpha$ to indicate the influence of joints between consecutive frames. Moreover, we set $\omega_{ij} = \beta$ to express the relationship between joints as intrinsic dependency and $\omega_{ij} = \gamma$ shows the disconnected joints' dependency between each other. Therefore, Given a video with T frames, we first construct each frame into a graph according to Eqn.1 as $G = [\mathcal{G}_1, \mathcal{G}_2, ..., \mathcal{G}_T]$, where G, a 3D tensor, will be sent into GC-LSTM for action recognition.

2.3. Graph Convolutional Long-Short Term Memory Neural Network

We now present our GC-LSTM network. Although the FC-LSTM layer has proven powerful for handling temporal correlation for action recognition [4][8] [16][17], it contains too much redundancy for spatial data. In terms of still image recognition task, CNN-based methods are much better than full-connected neural networks, because the former (1) have much less parameters to be trained and (2) consider more about spatial correlation. To address this problem, we propose an extension of FC-LSTM which has graph convolutional structures [11] in both input-to-state and state-to-state transitions as shown in Fig. 3 (b). By stacking multiple GC-LSTM layers, we are able to build a network model not only for temporal dynamics analysis but also for more general spatio-temporal skeleton sequences action recognition.

The major drawback of FC-LSTM in handling spatiotemporal data is its usage of full connections in input-to-state and state-to-state transitions in which no spatial representation is encoded. To overcome this problem, a distinguishing feature of our design is that all the inputs x_t, cell outputs C_t, hidden states H_t, and gates i_t, f_t, o_t of the GC-LSTM are 3D tensors whose last two dimensions are spatial dimensions. To get a better picture of the inputs and states at each time step, we may imagine them as vectors standing on a graph node. The GC-LSTM determines the future state of a certain node in the graph by the inputs and past states of its local neighbors as shown in Fig. 2. This can be achieved by using a graph convolution operator in the state-to-state and input-to-state transitions. The key equations of GC-LSTM are shown in Eqn. 2, where W_{--} and '*' denotes the graph convolution kernel and operator [11] which will be detailed later and '$\circ$' denotes the Hadamard product:

$$\begin{aligned} i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + b_f) \\ C_t &= f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + b_o) \\ H_t &= o_t \circ tanh(C_t) \end{aligned} \tag{2}$$

If we view the states as the hidden representations of moving objects, a GC-LSTM with a larger transitional kernel should be able to capture longer global motions while one with a smaller kernel can capture shorter local motions. Also, if the inputs, cell outputs and hidden states of the traditional FC-LSTM are represented by original LSTM equations, they can be seen as 3D tensors with the last two dimensions being one. In this sense, FC-LSTM is actually a special case of GC-LSTM with all features standing on a single cell.

Flowing through multiple layers of GC-LSTM, the input G transforms into G', which is also a 3D tensor and will be finally sent into a conventional CNN for action recognition, the output of CNNs is

$$o = \{o_1, o_2, ..., o_C\} \tag{3}$$

where C is the number of action categories. We adopt the categorical cross-entropy loss to train the GC-LSTM. The predicted probability being the i^{th} class given a sequence G is

$$p(C_i|G) = \frac{e^{o_i}}{\sum_{j=1}^{C} e^{o_j}}, k = 1, ..., C. \tag{4}$$

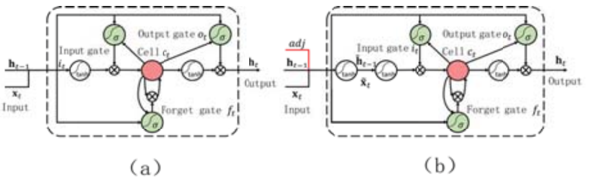

Fig. 3. The structure of neurons. (a) LSTM, (b) GC-LSTM. Compared to LSTM, GC-LSTM need a more input adjacency matrix of the graph as illustrated in red.

2.4. Graph-based Convolution

The graph-based convolutional layer is the basic module in this network. We consider the graph Laplacian [12] on the spectral domain with the normalized definition: $L = I_n - D^{-1/2} W D^{-1/2}$, where D is the diagonal degree matrix with $d_{ii} = \sum_j \omega_{ij}$. We scale L as $\widetilde{L} = 2L/\lambda_{max} - I_n$ and denote $\overline{x}_k = T_k(\widetilde{L}) * x$, where λ_{max} is the maximised eigen value of L and T_k is the Chebyshev polynomial as [14]. Then, the convolutional operator can be formulated as [11]:

$$y(\eta, W) * x = \eta[\overline{x}_0, \overline{x}_1, ..., \overline{x}_{K-1}]^T \tag{5}$$

Here, $\eta \in [\eta_0, \eta_1, ..., \eta_{K-1}]$ are the parameters to be trained, K is the size of the graph-based convolutional kernel and x is the inputs or hidden states of the GC-LSTM in our network.

3. EXPERIMENTS

In this section we evaluate the performance of our network in skeleton based action recognition experiment. We experiment on two benchmark datasets: the UT-Kinect dataset (Yun et al. 2012) [18] and the largest RGB+D dataset of NTU (Shahroudy et al. 2016) [17].

3.1. Datasets and Settings

NTU RGB+D Dataset (NTU) This is the current largest action recognition dataset with joints annotations collected by Microsoft Kinect v2 (Shahroudy et al. 2016) with 56880 video samples. It contains 60 different action classes including daily actions, mutual and health-related actions, performed by 40 distinct subjects and captured by three cameras from different viewpoints in the meantime. [17] defines two standard evaluation protocols for this dataset: for the Cross-Subject (CS) evaluation, 40 subjects are split into training and testing groups; for Cross-View (CV) evaluation, all the samples of cameras 2 and 3 are used for training while the samples of camera 1 for testing.

UT-Kinect Dataset (UT) This dataset captured by Kinect contains 10 actions. Each action is performed by 10 subjects twice. It has 200 skeleton sequences (6027 frames). Each

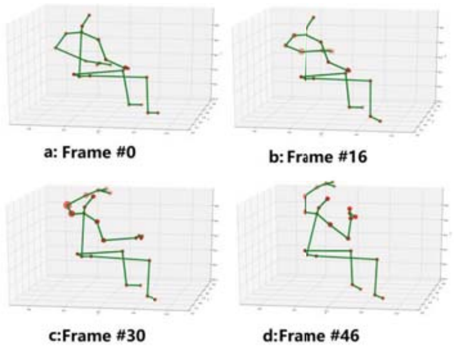

Fig. 4. Visualization of the neural response magnitude of each node in the last layer of GC-LSTM for action “taking a selfie” from NTU dataset.

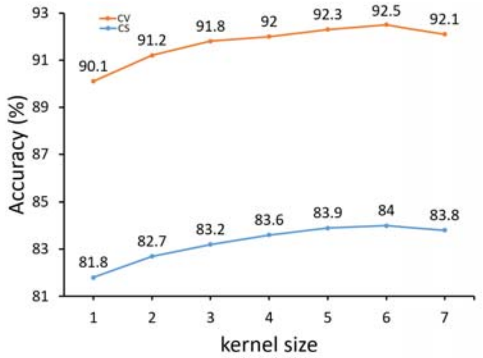

Fig. 5. Performance comparisons on different kernel size for graph convolution on NTU RGB+D dataset with Cross-Subject and Cross-View setting in accuracy $(\%)$

subject has 20 joints. We evaluate performance using the leave-one-out cross-validation protocol of this dataset [18].

Implementation Details Our proposed method is implemented with Tensorflow [19] with Keras [20] toolbox. The kernel size of the graph convolutional lstm layer is set to be 5. For simplicity, we construct a three-layer GC-LSTM model. α and β are set to one and γ is set to zero in skeleton graph construction. We set the dropout rate to 0.5 to alleviate overfitting. Adam [21] is adapted to train all the networks, and the initial learning rate is set to 0.0001. We set batch size for the NTU and UT dataset to 32 and 8, respectively.

3.2. Visualization from the Learned Response Magnitude

The visualization of the neural response magnitude of each node in the last layer of GC-LSTM is illustrated in Fig. 4. To visualize how GC-LSTM exploit local correlation and local pattern, the feature vector magnitude of each node in the final GC-LSTM layer is computed. The response magnitude of each node is shown by the size of the red circle. For action “taking a selfie”, at discriminative frames, the response magnitude on the raised arm and head is large, meanwhile, the response on legs and trunk is small; on the other hand, the responses on nodes at indistinctive frames are all small, which is consistent with human perception. The results show that the proposed GC-LSTM model strives to focus more on discriminative parts at discriminative frames rather than the redundant parts which are irrelevant for recognition.

3.3. Efficiency of the proposed method

To validate the effectiveness of the proposed model, we make comparisons between our baseline schemes. We evaluate the influence of the kernel size of input-to-state graph convolution and state-to-state graph convolution and show the comparisons in Fig. 5.

We make comparisons on our baseline whose kernel size equals to one. As the increase of kernel size, recognition accuracy increases since larger kernel has a higher capability of modeling the evolution of action dynamics as well as spatial structure. The increase in the size of kernel from one to five brings an improvement of about 2% accuracy. When the kernel size increases from five to six, the performance further increase, but it takes much longer time to train. Further increase of the size of kernel not only drop the performance, but significantly boost the training period.

For action recognition, the kernel means the neighbourhood of node. The bigger the kernel is, the more neighbor nodes will be considered for the certain node. When human is standing up from a chair, the hip, knee and ankle position will influence each other.

In our experiments, we make kernel size equal to five by default to balance performance, speed and complexity, which conforms to human perception that the kernel size almost equals to number of joints in a body part, the category of action is usually judged by the motion of body part by human.

3.4. Comparisons to Other State-of-the-Art

We compare the performance of our proposed architecture (GC-LSTM) aganist other state-of-the-art approaches on the NTU dataset and UT dataset in Table 1 and Table 2.

As shown in Table 1, with the effective spatiotemporal features extracted by gclstm networks for recognition, we can see that the method achieves the competitive performances of 83.9% and 92.3% on the current largest NTU dataset. For the ablation study, the baseline-CNN is a $ResNet$ (we transform a skeleton sequence into an image similar to that in [22]) and the baseline-LSTM is a 3 layer LSTM network, our model achieves 1.6% (CS) and 1.5% (CV) improvement over the Baseline-CNN and 9.6% (CS) and 7.7% (CV) improvement over the Baseline-LSTM, which shows the effectiveness of our proposed method. ST-GCN[13] is a similar work to ours following the spatial perspective to construct the GCN-

Table 1. The comparison results on NTU RGB+D dataset with Cross-Subject and Cross-View setting in accuracy (%)

Method	CS	CV
HBRNN-L[23]	59.1	64.0
Part-aware LSTM[17]	59.1	70.3
Trust Gate ST-LSTM[24]	69.2	77.7
STA-LSTM[16]	73.4	81.2
Ensemble TS-LSTM[25]	74.6	81.3
VA-LSTM[8]	79.4	87.6
ST-GCN[13]	81.5	88.3
DPRL+GCNN[14]	83.5	89.8
Baseline-CNN	82.3	90.8
Baseline-LSTM	74.3	84.6
GC-LSTM(Ours)	**83.9**	**92.3**

s on graph. The results of our model outperform by 2.4% and 4.0% compared with [13] in cross-subject evaluation and cross-view evaluation, respectively. And our method construct frame-level graph while [13] constructs sequence-level graph. Experiments are conducted with 8 TITANX GPUs in [13] and we proceed experiments on only one TITANX GPU. [14] also uses the GCN operation on the single frame skeleton graph. The difference between ours and [14] is that [14] first do the frame selection as a progressive process. Then [14] leverages graph convolution on each selected skeleton graph and simply concatenate them to generate a series of feature maps for classification. while ours considering spatiotemporal local correlation outperforms by 0.4 % and 2.5% compared with [14].

As shown in Table 2, our proposed model achieves the state-of-the-art performance of 98.5% on UT dataset. Furthermore, we discover that our proposed method outperforms most of the other state-of-the-art methods except VA-LSTM [8]. The reason is that [8] benefits a lot from the view adaptation sub-network, which is specially designed to recognize actions in variant views. In the UT dataset, such conditions are common. On the other hand, ours achieves a competitive performance over graphical model compared to [14].

Table 2. The comparison results on UT dataset in accuracy (%)

Method	Acc.(%)
Grassmann Manifold[26]	88.5
Histogram of 3D Joints[18]	90.9
Riemannian Manifold[27]	91.5
STA-LSTM[16]	97.0
VA-LSTM[8]	99.5
DPRL+GCNN[14]	98.5
GC-LSTM	**98.5**

4. CONCLUSIONS

In this paper, we propose a novel method called GC-LSTM, which achieves competitive results compared with the current state-of-the-art methods. Our method overcome weaknesses of the RNN-based methods and CNN-based methods through combining GCN with LSTM together. With the help of multiple layers of GC-LSTM, a high level feature representation of action sequence consisting of discriminative spatial and temporal information is generated, which is used for category classification. With extensive experiments on the current largest NTU RGB+D dataset and UT dataset, we verify the effectiveness of our model for the skeleton based action recognition.

In the future, we will consider more about the graph adjacency matrix which employs hand-crafted parameters. It is desirable to explore self-learning methods to determine the weights among nodes.

Acknowledgments This work is partially supported by the Natural Science Basic Research Plan in Shaanxi Province of China (Program No.2018JM6104), and the National Key Research and Development Project of China 017YFB1301101.

5. REFERENCES

[1] Ronald Poppe, “A survey on vision-based human action recognition,” *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[2] Daniel Weinland, Remi Ronfard, and Edmond Boyer, “A survey of vision-based methods for action representation, segmentation and recognition,” *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[3] Pichao Wang, Wanqing Li, Zhimin Gao, Jing Zhang, Chang Tang, and Philip Ogunbona, “Deep convolutional neural networks for action recognition using depth map sequences,” *arXiv: Computer Vision and Pattern Recognition*, 2015.

[4] Yong Du, Wei Wang, and Liang Wang, “Hierarchical recurrent neural network for skeleton based action recognition,” pp. 1110–1118, 2015.

[5] Zhengyou Zhang, “Microsoft kinect sensor and its effect,” *IEEE MultiMedia*, vol. 19, no. 2, pp. 4–10, 2012.

[6] Zhe Cao, Tomas Simon, Shihen Wei, and Yaser Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” *computer vision and pattern recognition*, pp. 1302–1310, 2017.

[7] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie, “Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks,” *national conference on artificial intelligence*, pp. 3697–3704, 2016.

[8] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng, “View adaptive recurrent neural networks for high performance human action recognition from skeleton data,” pp. 2117–2126, 2017.

[9] Qiuhong Ke, Mohammed Bennamoun, Senjian An, Ferdous Ahmed Sohel, and Farid Boussaid, “A new representation of skeleton sequences for 3d action recognition,” *computer vision and pattern recognition*, pp. 4570–4579, 2017.

[10] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu, “3d convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] Michael Defferrard, Xavier Bresson, and Pierre Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” *neural information processing systems*, pp. 3844–3852, 2016.

[12] Thomas N Kipf and Max Welling, “Semi-supervised classification with graph convolutional networks,” *international conference on learning representations*, 2017.

[13] Sijie Yan, Yuanjun Xiong, Dahua Lin, and Xiaoou Tang, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” *national conference on artificial intelligence*, 2018.

[14] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou, “Deep progressive reinforcement learning for skeleton-based action recognition,” 2018.

[15] Xingjian Shi, Zhourong Chen, Hao Wang, Dit Yan Yeung, Wai Kin Wong, and Wangchun Woo, “Convolutional lstm network: a machine learning approach for precipitation nowcasting,” *neural information processing systems*, pp. 802–810, 2015.

[16] Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu, “An end-to-end spatio-temporal attention model for human action recognition from skeleton data,” *national conference on artificial intelligence*, pp. 4263–4270, 2017.

[17] Amir Shahroudy, Jun Liu, Tiantsong Ng, and Gang Wang, “Ntu rgb+d: A large scale dataset for 3d human activity analysis,” *computer vision and pattern recognition*, pp. 1010–1019, 2016.

[18] L. Xia, C.C. Chen, and JK Aggarwal, “View invariant human action recognition using histograms of 3d joints,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[19] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., “Tensorflow: a system for large-scale machine learning,” *operating systems design and implementation*, pp. 265–283, 2016.

[20] Franois Chollet, “Keras,” `http://github.com/fchollet/keras`, 2015.

[21] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *international conference on learning representations*, 2015.

[22] Yong Du, Yun Fu, and Liang Wang, “Skeleton based action recognition with convolutional neural network,” *asian conference on pattern recognition*, pp. 579–583, 2015.

[23] Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations,” in *International Joint Conference on Artificial Intelligence*, 2013, pp. 639–44.

[24] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang, “Spatio-temporal lstm with trust gates for 3d human action recognition,” *european conference on computer vision*, pp. 816–833, 2016.

[25] Inwoong Lee, Do Young Kim, Seoungyoon Kang, and Sanghoon Lee, “Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks,” pp. 1012–1020, 2017.

[26] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava, “Accurate 3d action recognition using learning on the grassmann manifold,” *Pattern Recognition*, vol. 48, no. 2, pp. 556–567, 2015.

[27] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo, “3-d human action recognition by shape analysis of motion trajectories on riemannian manifold,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 45, no. 7, pp. 1340–1352, 2015.