SPATIAL-TEMPORAL DATA AUGMENTATION BASED ON LSTM AUTOENCODER NETWORK FOR SKELETON-BASED HUMAN ACTION RECOGNITION

Juanhui Tu[1], *Hong Liu*[1], *Fanyang Meng*[1,2], *Mengyuan Liu*[3], *Runwei Ding*[1]

[1]Key Laboratory of Machine Perception, Peking University, Shenzhen Graduate School
[2]Shenzhen Institute of Information Technology, China
[3]School of Electrical and Electronic Engineering, Nanyang Technological University
{juanhuitu@pku.edu.cn, hongliu@pku.edu.cn, fymeng@pku.edu.cn, liumengyuan@ntu.edu.sg, dingrunwei@pku.edu.cn}

ABSTRACT

Data augmentation is known to be of crucial importance for the generalization of RNN-based methods of skeleton-based human action recognition. Traditional data augmentation methods artificially adopt various transformations merely in spatial domain, which lack effective temporal representation. This paper extends traditional Long Short-Term Memory (LSTM) and presents a novel LSTM autoencoder network (LSTM-AE) for spatial-temporal data augmentation. In the LSTM-AE, the LSTM network preserves the temporal information of skeleton sequences, and the autoencoder architecture can automatically eliminate irrelevant and redundant information. Meanwhile, a regularized cross-entropy loss is defined to guide the LSTM-AE to learn more suitable representations of skeleton data. Experimental results on the currently largest NTU RGB+D dataset and public SmartHome dataset verify that the proposed model outperforms the state-of-the-art methods, and can be integrated with most of the RNN-based action recognition models easily.

Index Terms— 3D Action Recognition, Long Short-Term Memory, Data Augmentation, Autoencoder

1. INTRODUCTION

Human action recognition has been used in a wide range of applications, such as video surveillance [1], human-machine interaction [2], and video analysis [3]. With the wide spread of depth sensors such as Microsoft Kinect, action recognition using 3D skeleton sequences has attracted a lot of research attention. Lots of advanced approaches have been proposed [4–6], especially deep learning methods like Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM). Despite significant progress, the generalization ability of RNN models is still a research focus.

This work is supported by National Natural Science Foundation of China (NSFC, No.U1613209,61340046,61673030), Natural Science Foundation of Guangdong Province (No.2015A030311034), Scientific Research Project of Guangdong Province (No.2015B010919004), Specialized Research Fund for Strategic and Prospective Industrial Development of Shenzhen City (No.ZLZBCXLJZI20160729020003), Scientific Research Project of Shenzhen City (No.JCYJ20170306164738129), Shenzhen Key Laboratory for Intelligent Multimedia and Virtual Reality (No.ZDSYS201703031405467).

Relation to prior work: As neural networks often require a lot of data to improve generalization and reduce the risk of over-fitting, data augmentation is an explicit form of regularization that is widely used during the training of deep neural networks [7–10]. It aims at enlarging the training dataset from existing data using various translations. Wang *et al.* [7] proposed rotation, scaling and shear transformation as data augmentation techniques based on 3D transformation to make better use of limited supply of training data. Ke *et al.* [8] employed cropping technique to increase the number of samples. Yang *et al.* [9] exploited horizontal flip as data augmentation method without losing any information. Li *et al.* [10] designed different data augmentation strategies, such as 3D coordinate random rotation, Gaussian noise and video crop to augment the scale of original dataset.

However, the aforementioned data augmentation methods only leverage various transformations in spatial domain, which ignore the effective representation in temporal domain. For instance, the method horizontal flip confuses the temporal information of skeleton sequences. Different from previous works, our proposed LSTM autoencoder network (LSTM-AE) can retain temporal representation of skeleton sequences. In essence, the above methods add interference information unrelated to classification to expand the dataset. And then deep neural networks are utilized to learn suitable features related to classification. In contrast, with the characteristic of autoencoder, our LSTM-AE can eliminate irrelevant information such as noise. In consequence, based on samples generated from LSTM-AE, deep neural networks can directly learn discriminative features related to classification. Moreover, the proposed regularized cross-entropy loss enables original samples to be consistent with generated samples at semantic level.

Our main contributions are as following: (1) A novel spatial-temporal data augmentation network (LSTM-AE) is designed to generate samples which reserve both spatial and temporal representation of skeleton sequences, and can be integrated with various RNN-based models. (2) A regularized cross-entropy loss is defined to guide LSTM-AE to learn more suitable representations of skeleton sequences.

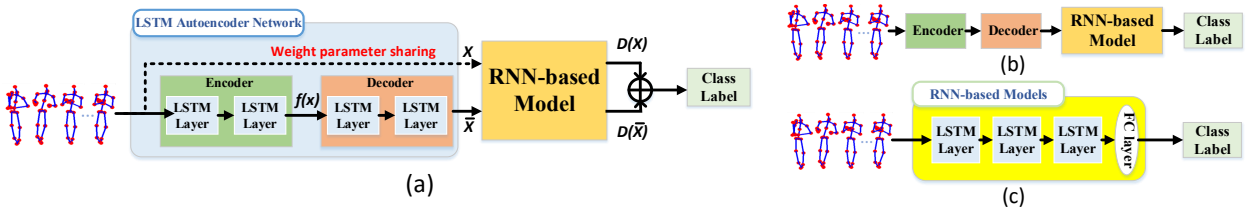

Fig. 1. (a) Overall framework of the end-to-end RNN-based method, which consists of the LSTM autoencoder network and RNN-based Models. (b) The contrastive network with LSTM-AE. (c) The baseline LSTM network (RNN-based method without LSTM-AE).

2. THE PROPOSED METHOD

In this section, overall framework of the end-to-end RNN-based method for skeleton-based human action recognition is illustrated in Fig.1(a). It consists of LSTM autoencoder network (LSTM-AE) and RNN-based models. Fig.1(b) and (c) are listed for comparison with our proposed method. The remainder of this section is organized as follows: we first describe the LSTM-AE, then introduce three RNN-based models that we adopt in our experimental section. Finally, a regularized cross-entropy loss function of LSTM-AE is introduced.

2.1. LSTM Autoencoder Network

RNN is a powerful model for sequential data modeling and feature extraction [11], which is designed to preserve temporal information. Due to the vanishing gradient and error blowing up problems [12, 13], the standard RNN can barely store information for long periods of time. The advanced RNN architecture LSTM [13] mitigates this problem. An LSTM neuron contains a memory cell c_t which has a self-connected recurrent edge of weight 1. 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:

$$\begin{aligned} i_t, f_t, o_t &= \sigma(W_x x_t + W_h h_{t-1} + b) \\ g_t &= tanh(W_{xg} x_t + W_{hg} h_{t-1} + b_g) \\ c_t &= f_t * c_{t-1} + i_t * g_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{1}$$

We employ LSTM neuron to build the proposed LSTM-AE. The network is capable of retaining the effective temporal information of skeleton sequences, which is different from the traditional data transformations in spatial domain. As shown in Fig.1(a), for a skeleton sequence as the input, the input data $\boldsymbol{X}$ and the reconstruction data $\overline{\boldsymbol{X}}$ through the autoencoder architecture are input to RNN-based models in parallel. In this way, they share the weight parameters of RNN-based models in the process of network training. $D(\boldsymbol{X})$ and $D(\overline{\boldsymbol{X}})$ are the output of RNN-based models respectively, and the final output of RNN-based models of LSTM-AE is represented as $D(\boldsymbol{X}) + D(\overline{\boldsymbol{X}})$. Fig.1(b) shows the contrastive network with LSTM-AE, it does not have original data $\boldsymbol{X}$ as additional input to RNN-based models. The contrastive network is utilized to demonstrate the validity of LSTM-AE architecture.

For the autoencoder architecture of LSTM-AE, it comprises encoder and decoder. The encoder and decoder share similar structure, i.e., stacking several LSTM layers. The number of LSTM layers to construct the autoencoder architecture is flexible. Suppose both encoder and decoder contain two layers of LSTM respectively as shown in Fig.1(a), the neurons of the second LSTM layer in encoder is equal to that of the first LSTM layer in decoder, which is corresponding to the compression dimensions K. Especially, different compression dimensions affect the data reconstruction capability. To be more specific, for the encoder step, the input data $\boldsymbol{X}$ are mapped to a compressed data representation $f(x)$ in a low-dimensional subspace. For the decoder step, the compressed data representation $f(x)$ is mapped to a vector $\overline{\boldsymbol{X}}$ in the original data space.

2.2. RNN-based Models

Since the LSTM network is capable of modeling long-term temporal dynamics and automatically learning feature representations, many recent works widely leverage LSTM neurons as basic units to build deep architectures to recognize human actions from raw skeleton inputs. The compared LSTM architectures are introduced as follows:

Deep LSTM network (baseline): According to [14, 15], as shown in Fig.1(c), we build the baseline LSTM network by stacking three LSTM layers called deep LSTM network, followed by one full-connected layer.

Deep Bidirectional LSTM (BLSTM) network: The idea of BLSTM is derived from bidirectional RNN [16], which processes sequence data in both forward and backward directions with two separate hidden layers. We use BLSTM instead of LSTM to implement the baseline, which generates a new BLSTM network.

Deep LSTM-zoneout (LSTMZ) network: Zoneout [17] is a new method for regularizing RNNs. Instead of discarding (setting to zero) the output of each hidden neuron with a probability during the training like dropout, zoneout stochastically forces some hidden units to maintain their previous values at each timestep. Hence, the computation of c_t and h_t are changed as follows:

$$c_t = d_t^c * c_{t-1} + (1 - d_t^c) * (f_t * c_{t-1} + i_t * g_t) \tag{2}$$

$$h_t = d_t^h * h_{t-1} + (1 - d_t^h) * (o_t * tanh(f_t * c_{t-1} + i_t * g_t) \tag{3}$$

where d_t^c and d_t^h are the zoneout masks. Based on zoneout, we build deep LSTM-zonout network. The architecture of deep LSTM-zoneout networks is similar to that of deep LSTM network, including three LSTM-zoneout layers and one fully-connected layer.

2.3. Regularized Loss Function

To guide the proposed LSTM-AE to learn more discriminative and suitable representations of skeleton sequences, we formulate a regularized loss function with cross-entropy loss for a sequence as:

$$Loss = loss(\hat{\boldsymbol{y}}_c, \boldsymbol{y}) + loss(\hat{\boldsymbol{y}}_r, \boldsymbol{y}) + \lambda \cdot \left\| \boldsymbol{X} - \overline{\boldsymbol{X}} \right\|_2 \quad (4)$$

where $\boldsymbol{y} = (y_1, y_2, ..., y_C)^{\mathrm{T}}$ denotes the groundtruth label for each skeleton sequence. and C represents the total number of classes. If a skeleton sequence belongs to the i^{th} class, y_i equals one; otherwise, y_i equals zero. $loss(\hat{\boldsymbol{y}}_c, \boldsymbol{y})$ denotes classification loss and $loss(\hat{\boldsymbol{y}}_r, \boldsymbol{y})$ denotes the reconstruction loss. The cross-entropy loss is utilized to formulate the two losses as:

$$loss(\hat{\boldsymbol{y}}_c, \boldsymbol{y}) = -\sum_{i=1}^{C} y_i log \hat{\boldsymbol{y}}_c^i, \quad loss(\hat{\boldsymbol{y}}_r, \boldsymbol{y}) = -\sum_{i=1}^{C} y_i log \hat{\boldsymbol{y}}_r^i \quad (5)$$

Specifically, as shown in Fig. 1(a), for $\boldsymbol{X}$ as the input to the RNN-based models, $\hat{\boldsymbol{y}}_c^i$ in the first item indicates the probability that the sequence is predicted as the i^{th} class. For $\overline{\boldsymbol{X}}$ as input to the RNN-based models, $\hat{\boldsymbol{y}}_r^i$ in the second item indicates the probability that the sequence is predicted as the i^{th} class. The scalar λ balances the significance between the reconstruction loss and the classification loss. The regularization is to minimize the difference between $\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$ under control with $l2$ norm. The regularized loss function is designed to ensure the consistency at the semantic level between $\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$.

3. EXPERIMENTS

Experiments are conducted on two challenging 3D action datasets which have limited training data. We introduce NTU RGB+D dataset [18] and SmartHome dataset [19], and then present and analyze the experimental results.

3.1. Datasets and Protocols

NTU RGB+D dataset [18] contains 60 actions performed by 40 subjects from various views, generating 56880 skeleton sequences. This dataset contains noisy skeleton joints (see Fig. 2(c)), which bring extra challenge for recognition. Following the cross-subject protocol in [18], we split the dataset into 40,320 training samples and 16,560 test samples. Following the cross-view protocol in [18], the training and test sets have 37,920 and 18,960 samples respectively. There is a great difference between the training set and test set for the same kind of action (see Fig. 2(a)(b)). We use this dataset to show that our spatial temporal data augmentation method can alleviate this type of difference.

SmartHome dataset [19] contains 6 types of actions performed 6 times by 9 subjects in 5 situations from single view, resulting in 1620 sequences. Due to occlusions and the unconstrained poses of action performers, skeleton joints contain much noises. The noisy skeleton snaps of action "wave" are illustrated in Fig. 3. Following the cross-subject protocol in [19], we use subjects #1, 3, 5, 7, 9 for training and subjects #2, 4, 6, 8 for testing.

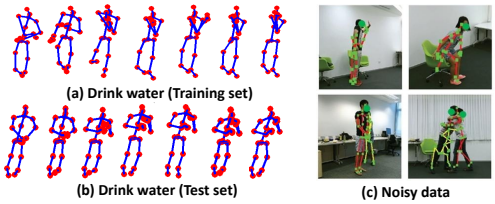

Fig. 2. Snaps from NTU RGB+D dataset [18].

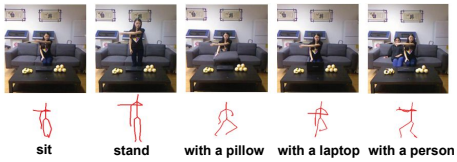

Fig. 3. Skeletons of action "wave" from SmartHome dataset [19].

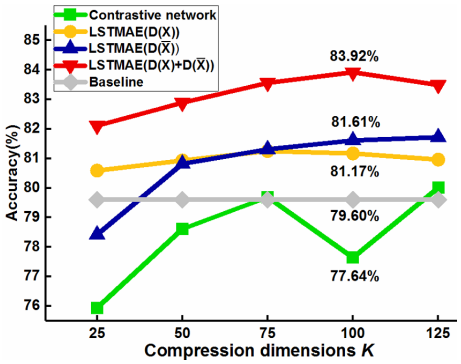

Fig. 4. Visualization of recognition accuracies among different K on NTU RGB+D (cross-view protocol).

3.2. Implementation Details

The implementation is derived from Pytorch toolbox based on one NVIDIA GeForce GTX 1080 GPU and our codes are open source[1]. For the LSTM-AE, the layers of LSTM in encoder and decoder are set as two. For the RNN-based models, each LSTM layer is composed of 100 LSTM neurons, and the number of neurons of the FC layer is equal to the number of action classes. Dropout [20] with a probability of 0.5 is used to alleviate overfitting. Adam [21] is adapted to train all the networks, and the initial learning rate is set as 0.001. For deep LSTM-zoneout network, the value of zoneout is set as 0.1. The batch sizes for the SmartHome dataset and NTU RGB+D dataset are 32 and 256 respectively.

3.3. Experimental Results

Evaluation of LSTM-AE network. Fig. 4 shows the visualization of recognition accuracies of RNN-based methods with LSTM-AE at different compression dimensions K on NTU RGB+D (cross-view protocol). Owing to the dimension of each frame is 150 for NTU RGB+D dataset, we change the value of K from 25 to 125. As shown in Fig. 4, our **LSTM-AE**($D(\boldsymbol{X}) + D(\overline{\boldsymbol{X}})$) performs better than the baseline (deep LSTM network), especially improves the accuracy by 4.32% when K equals 100. When K is small, the reconstruction data lacks sufficient information of original data for classification.

[1] https://github.com/Damilytutu/LSTMAE

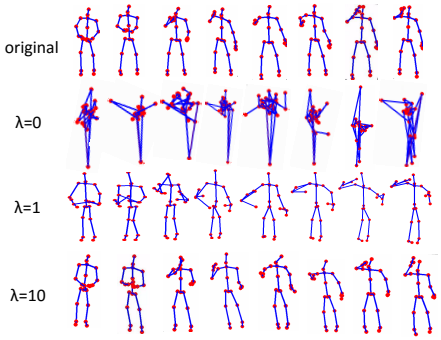

Fig. 5. Visualization of sequence among different λ for action making a phone call on NTU RGB+D dataset.

Table 1. Comparison of recognition accuracies among different RNN-based models on NTU RGB+D dataset. Legend: w/o L is short for "without LSTM-AE".

Models	NTU RGB+D (CS)		NTU RGB+D (CV)	
	w/o L	LSTM-AE	w/o L	LSTM-AE
Deep LSTM	70.81%	73.31%	79.60%	83.92%
Deep BLSTM	72.03%	74.66%	80.91%	84.78%
Deep LSTMZ	78.14%	80.63%	85.73%	88.56%

When K is large, the reconstruction data still involves irrelevant information. Therefore, the value of K affects the quality of reconstruction data. In addition, LSTM-AE($D(\boldsymbol{X})$) and LSTM-AE($D(\overline{\boldsymbol{X}})$) gains 1.57% and 2.01% respectively than the baseline at $K = 100$, which validates the effectiveness of spatial-temporal data augmentation. Especially, the accuracy of contrastive network (77.64%) is worse than the baseline, which further indicates the effectiveness of the LSTM-AE architecture.

Evaluation of regularized loss function. Fig. 5 shows the visualization of action making a phone call among different λ. The first row represents the original skeleton sequence. When $\lambda = 0$, generated data and original data are merely similar at the feature level. When $\lambda = 10$, we can see generated data are very similar to original data, which still reserve irrelevant information consequently. While when $\lambda = 1$, compared with original data, the generated data enlarge the movement of the hand joints and meanwhile weaken the foot movement. Since the movement of action making a phone call focus on the hands, the generated data are more beneficial for classification than original data.

Evaluation of different RNN-based models. Table 1 shows the recognition accuracies of applying LSTM-AE on NTU RGB+D dataset with different RNN-based models. For cross-view protocol, our method outperforms deep LSTM network (baseline), deep BLSTM network and deep LSTMZ network by 4.32%, 3.87% and 2.83%, respectively. These results indicate that models trained with LSTM-AE have significant improvement, validating our method is applicable to various LSTM architectures.

- **Comparing with data augmentation methods.** Fig.6 shows the performance among Scale, Rotation and our LSTM-AE on the NTU(cross-subject), NTU(cross-view) and SmartHome dataset respectively. Our LSTM-AE outperforms the other two methods. Especially in comparison with the baseline without any augmentation, our LSTM-AE brings up

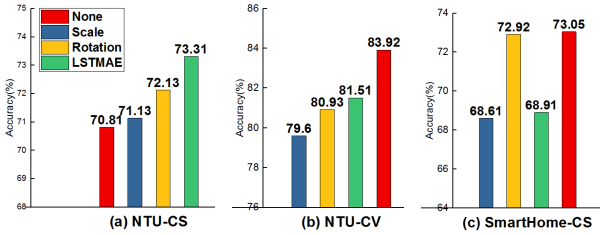

Fig. 6. Performance evaluation among different data augmentation methods on NTU RGB+D and SmartHome datasets in accuracy(%).

Table 2. Comparison of our method with the state-of-the-art methods on NTU RGB+D dataset (cross-subject and cross-view protocols).

Methods	CS	CV
HBRNN-L [22]	59.07%	63.97%
Part-aware LSTM [18]	62.93%	70.27%
ST-LSTM+Trust Gate [23]	69.20%	75.70%
Geomeric Features [24]	70.26%	82.39%
GCA-LSTM [25]	74.40%	82.80%
Skeleton Visualization [3]	77.69%	83.67%
Clips+CNN+MTLN [8]	79.57%	84.83%
Deep LSTMZ	**78.14%**	**85.73%**
Deep LSTMZ + LSTM-AE(Ours)	**80.63%**	**88.56%**

Table 3. Comparison of our method with the state-of-the-art methods on SmartHome dataset (cross-subject protocol).

Methods	CS
ConvNets [26]	67.22%
JTM [27]	71.11%
SM+MM [19]	77.92%
Skeleton Visualization [3]	78.61%
Deep LSTMZ	**75.74%**
Deep LSTMZ + LSTM-AE(Ours)	**78.82%**

to 4.32% accuracy improvement for cross-view protocol on NTU RGB+D. We can see the performance of the method "Scale" and "Rotation" is affected by the datasets. Since the main problem of NTU RGB+D is multi-view, the method "Rotation" performs better than "Scale". Using cross-subject protocol on SmartHome, the method Scale gains 4.01% than Rotation. Our method works well on both datasets, since the viewpoint changes and scale variations are automatically handled by our spatial temporal data augmentation method.

Comparison with the state-of-the-art methods. In Table 2 and 3, we compare our method with the state-of-the-art methods on NTU RGB+D and SmartHome. Our method outperforms the state of the art. Specifically, based on deep LSTMZ, our method achieves the highest accuracy of 80.63% and 88.56% using cross-subject and cross-view protocol respectively on NTU RGB+D, and obtain 78.82% using cross-subject protocol on SmartHome.

4. CONLUSIONS

This paper presents a novel spatial-temporal data augmentation network (LSTM-AE), generating samples which reserve the spatial and temporal information of skeleton sequences. Meanwhile, the architecture of autoencoder can eliminate the irrelevant information. Besides, a regularized cross-entropy loss is designed to guide the LSTM-AE to learn more underlying and discriminative representations. Experiments conducted on public SmartHome and NTU RGB+D datasets demonstrate that our method outperforms the state-of-the-art methods, and can be integrated with most of the RNN-based models.

5. REFERENCES

[1] J. Zheng, Z. Jiang, and R. Chellappa, "Cross-view action recognition via transferable dictionary learning," *IEEE TIP*, vol. 25, no. 6, pp. 2542–2556, 2016.

[2] H. Liu, Q. He, and M. Liu, "Human action recognition using adaptive hierarchical depth motion maps and gabor filter," *in Proc. ICASSP*, pp. 1847–1851, 2017.

[3] M. Liu, H. Liu, and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition," *PR*, vol. 68, pp. 346–362, 2017.

[4] J. Aggarwal and X. Lu, "Human activity recognition from 3D data: A review," *PRL*, vol. 48, pp. 70–80, 2014.

[5] L. Presti and M. La Cascia, "3D skeleton-based human action classification: A survey," *PR*, vol. 53, pp. 130–147, 2016.

[6] J. Zhang, W. Li, P. Ogunbona, P. Wang, and C. Tang, "RGB-D-based action recognition datasets: A survey," *PR*, vol. 60, pp. 86–105, 2016.

[7] H. Wang and L. Wang, "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks," *arXiv preprint arXiv:1704.02581*, 2017.

[8] Q.H. Ke, M. Bennamoun, S.J. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3d action recognition," *in Proc. CVPR*, 2017.

[9] W. Yang, T. Lyons, H. Ni, C. Schmid, L. Jin, and J. Chang, "Leveraging the path signature for skeleton-based human action recognition," *arXiv preprint arXiv:1707.03993*, 2017.

[10] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin, and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep cnn," *arXiv preprint arXiv:1704.05645*, 2017.

[11] A. Graves, "Supervised sequence labelling with recurrent neural networks," *Springer*, vol. 385, 2012.

[12] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] 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 Proc. AAAI*, vol. 2, pp. 3697–3703, 2016.

[15] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," *in Proc. AAAI*, pp. 4263–4270, 2017.

[16] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE TSP*, vol. 45, no. 11, pp. 2673–2681, 1997.

[17] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, et al., "Zoneout: Regularizing rnns by randomly preserving hidden activations," *arXiv preprint arXiv:1606.01305*, 2016.

[18] A. Shahroudy, J. Liu, T. Ng, and G. Wang, "NTU RGB+D: A large scale dataset for 3D human activity analysis," *in Proc. CVPR*, pp. 1010–1019, 2016.

[19] M. Liu, Q. He, and H. Liu, "Fusing shape and motion matrices for view invariant action recognition using 3D skeletons," *in Proc. ICIP*, 2017.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint*, 2014.

[22] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," *in Proc. CVPR*, pp. 1110–1118, 2015.

[23] J. Liu, A. Shahroudy, D. Xu, and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition," *in Proc. ECCV*, pp. 816–833, 2016.

[24] S. Zhang, X. Liu, and J. Xiao, "On geometric features for skeleton-based action recognition using multilayer LSTM networks," *in Proc. WACV*, pp. 148–157, 2017.

[25] J. Liu, G. Wang, P. Hu, L. Duan, and A. Kot, "Global context-aware attention LSTM networks for 3D action recognition," *in Proc. CVPR*, vol. 7, pp. 1647–1656, 2017.

[26] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," *in Proc. ACPR*, pp. 579–583, 2015.

[27] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," *in Proc. ACM Multimedia*, pp. 102–106, 2016.