Activity Recognition from Skeleton and Acceleration Data Using CNN and GCN

Donghui Mao, Xinyu Lin, Yiyun Liu, Mingrui Xu, Guoxiang Wang,
Jiaming Chen, and Wei Zhang

Abstract Most of the existing methods of activity recognition are based on single-label classification, however, these methods can not be used in this challenge which focuses on multi-label classification based micro-activity recognition. To address this, we propose a GCN model using the binary cross entropy loss function, which enables multi-label classification and achieves average accuracy of 83.1% on the Cooking Activity Dataset. In addition, to utilize the advantages of multi-modal data, we propose a joint training CNN model that combines the acceleration and skeleton data together. Finally, the proposed CNN model achieves a average accuracy of 82.8% for macro-activity recognition on Cooking Activity Dataset.

1 Introduction

Activity recognition is a widely studied research topic in the computer vision community, which has two main lines of methods, skeleton-based activity recognition and sensor-based activity recognition. With the rapid developments of deep learning, activity recognition has made significant progress in recent years.

Skeleton data gives an abstract description of human activity, which contains abundant spatio-temporal information. With the increasing availability of sensors and devices, such as the Microsoft Kinect [21] and the motion capture system, it is more convenient to collect the skeleton data. Thus, a large body of skeleton-based methods have been presented and achieved good performance [22, 8, 11, 18, 20, 16, 12, 13]. In addition, the acceleration data has also been used in activity recognition and proved to be effective [2, 5, 4, 7].

Although the above methods have achieved impressive performance on activity recognition, there are two main problems need to be addressed.

The authors are with the School of Control Science and Engineering, Shandong University, China.

1. Existing skeleton-based activity recognition methods are only applicable for single-label and short sequence classification (e.g., NTU dataset [15]), not for multi-label and long sequence classification.
2. Using only single-modal data can not provide effective representations of human activity. Specifically, the acceleration data is sensitive to the action that changes rapidly while slightly. On the other hand, the skeleton data contains the key information of action that change obviously in space.

To address these problems, in this challenge[1], we propose a deep learning method to perform activity recognition from skeleton and acceleration data using CNN and GCN. The main contributions of this paper are summarized as follows.

1. We propose a GCN model to perform activity recognition using skeleton data, which is illustrated in Fig.1. To enable multi-label classification for micro-activity recognition, the *binary cross entropy* (BEC) loss function is introduced with several additional training techniques.
2. To fuse the skeleton data and acceleration data for more effective activity recognition, we develop a joint-training CNN model based on [11], as shown in Fig.2.

Extensive experiments are conducted and the results indicate that the proposed models are effective for micro-activity and macro-activity recognition.

2 Related Work

Numbers of studies have been conducted in activity recognition. In this section, we briefly review the related work.

The recurrent neural network (RNN) is usually combined with Long Short-Term Memory (LSTM) neurons, which is widely used in time sequences processing for their learning ability for temporal data [17]. For example, Zhu et al. [22] proposed a LSTM based network to automatically learn co-occurrence features of different joints for activity recognition. These methods have achieved good performance in short-sequence activity recognition, however, they are not suitable to encode long-sequence data for the recursive structure. Instead, CNN is more applicable for long-sequence activity recognition for its powerful learning ability.

Recently, many researchers used the CNN to encode the input data for activity recognition. Ke et al. [8] transformed the skeleton data into images and then fed them into a multi-task CNN for activity recognition. Zhang et al. [20] proposed a view-adaptive network based on CNN to improve the recognition performance on multi-view data. With the powerful feature learning ability of the CNN, these CNN based recognition methods have achieved impressive results.

There are also some works proposed GCN based model for activity recognition. The GCN can address the problems caused by non-euclidean data structure, which is useful in skeleton-based recognition. These mainly follow two streams.

1. Spectral Perspective: Graph data are translated into spectrum, where CNN are used for spectral analysis [3, 6]. However, such translation is extremely difficult in most cases due to the high requirement of prior-knowledge in modeling.
2. Spatial Perspective: The convolution operations are applied on the spatial domain which make this stream widely used in activity recognition. The ST-GCN [18] and many other reinforcement learning models [12, 13] can achieve excellent result in large dataset.

In addition, some works attempt to introduce acceleration data for activity recognition and achieved desired performance [19, 7]. Pourbabaee et al. [14] fed the ECG data, a kind of time series data, into the CNN to get temporal features. Acceleration data is a typical time series data, similar to prior work, we adopt a acceleration-to-image translation strategy to process the acceleration data which is then fed into the CNN for high-level feature extraction. These studies have proved the powerful ability and feasibility of CNN to encode acceleration data.

Our approach is markedly different from the existing works, that is, we present a novel GCN model with BCE loss function to solve the micro-activity recognition and a CNN model to combine skeleton data and acceleration data to solve macro-activity recognition.

3 Method

In this competition, the major challenge is the classification of multiple action in long sequence. To address this, we propose the GCN model with the BCE loss function for micro-activity recognition. And for macro-activity classification, we propose a joint-training CNN model to fuse the acceleration data and the skeleton data together to utilize the advantages of multi-modal data.

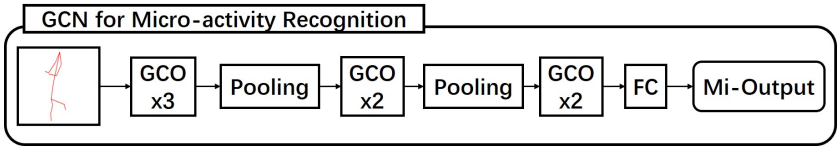

Fig. 1 Illustration of GCN structure for micro-activity classification. Graph convolution operation is represented as GCO and the fully connected layer is represented as FC.

3.1 GCN for Micro-activity Recognition

The proposed GCN model consists of several layers of graph convolution unit, as shown in Fig. 1. The GCN model is built upon the network in [18], while we

introduce the BCE loss function to enable the multi-label classification for micro-activity recognition. Here we give the definition of the BCE loss function:

$$L(\hat{y_j}, y_j) = -y_j \log \hat{y_j} - (1 - y_j) \log(1 - \hat{y_j}) \quad j = 1, 2, ..., 10 \tag{1}$$

where $\hat{y}$ is the output of GCN, y is the label of sample. There are 10 channels, j is corresponding to micro-activity j of our dataset. And we define the output of GCN as $\hat{\mathbf{y}}$:

$$\hat{\mathbf{y}} = [\hat{y_{ij}}]_{n \times m} \tag{2}$$

where n is the number of samples in forward propagation and m is 10, corresponding to 10 micro-activities in dataset. Each dimension is 1 or 0. The output 1 indicates that this action exists in the segment, the probability is:

$$P_{ij} = P(\hat{y_{ij}} = 1 | x_i, \omega_i) = O_{\omega_i}(x_i) \quad i = 1, 2, ..., n, \quad j = 1, 2, ..., 10 \tag{3}$$

If $P_{ij} > t$, the predicted $\hat{y_{ij}}$ is assigned to 1, which indicates action j exists in sample i, otherwise 0. The 10 channels' probability are computed separately to generate multiple peaks in 10 dimensions, which can be used for multi-label classification. x_i is input and t is fixed as threshold, ranging from 0 to 1. The O_ω is the graph convolution network with an output understanding as the probability. The model weights of ω are pre-trained on NTU dataset and then finetuned on our own dataset.

3.2 CNN for Macro-activity Recognition

The architecture of our designed CNN is shown in Fig. 2. Similar to [19, 11, 8], we map the skeleton data D_1, the difference data D_2 between adjacent frames of skeleton data and the acceleration data D_3 to images $I_i, i = 1, 2, 3$, then

$$I_{ij} = M(D_{ij}) \quad i = 1, 2, 3; j = 1, 2, ..., n \tag{4}$$

where M is the mapping operation and n is the number of samples. As shown in Fig. 3, in one frame, one point in the coordinate is correspond to one pixel in the image, the values of x, y and z directions are mapped to the value of r, g and b channels in the image. Before images I_1, I_2 are fed into convolution layers, a transformation is applied:

$$T_{ij} = TF(I_{ij}), \quad i = 1, 2; j = 1, 2, ..., n \tag{5}$$

where TF represents for transformation operation. Then, the transformed images $T_i, i = 1, 2$ and I_3 are fed into three streams of CNN to extract high-level features separately. For a unified form, we denote three images as $U_i, i = 1, 2, 3$.

$$C_{ij} = U_{ij} * W_{ij} \emptyset \tag{6}$$

where C_{ij} is the output of convolution layer in i sub-nets and $\emptyset$ is the rectified linear unit(ReLU) layer. Then C_3 is flatted to 1D vector $\mathbf{V_1}$ after a flatten transformation represented as Fl.

$$\mathbf{V_1} = Fl(C_3) \tag{7}$$

C_1, C_2 are concatenated and flattened to 1D vertor $\mathbf{V_2}$ after a fully connected layer, represented as F.

$$\mathbf{V_2} = F\{[C_1, C_2]\} \tag{8}$$

Then $\mathbf{V_1}$ and $\mathbf{V_2}$ are concatenated to get $\mathbf{V_3}$, where the skeleton data and acceleration data are jointed together.

$$\mathbf{V_3} = [\mathbf{V_1}, \mathbf{V_2}] \tag{9}$$

Finally the vector $\mathbf{V_3}$ is sent to one FC layer, and get the macro-activity classification result with three dimensions.

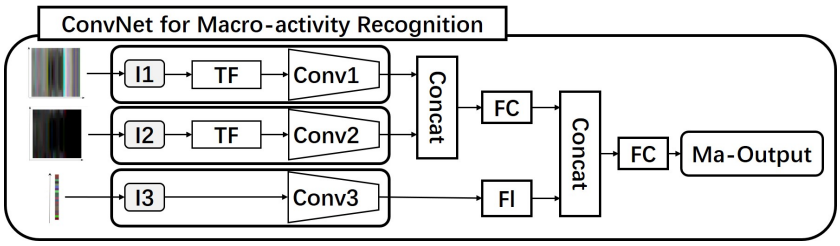

Fig. 2 Illustration of CNN structure for macro-activity classification. $I_i, (i = 1, 2, 3)$ represents input, TF represents transform layer, Conv represents convluation layer. FC and Fl are correspond to fully connected layer and flatten operation.

4 Experimental Results

The proposed GCN model and CNN model are both pre-trained on NTU dataset and fine-tuned on Cooking Activity Dataset (CAD) [9, 10]. The two datasets are introduced in Sect.4.1 and the data processing in Sect.4.2. In the following, results are presented in Sect.4.3, which show the effectiveness of the proposed methods.

4.1 Dataset

Cooking Activity Dataset CAD has two types of data, acceleration and skeleton data. The former is collected by four accelerators, from four wearable devices (on right arm, left hip, right wrist and left wrist) and the latter is from one motion capture system. Four subjects who prepared 3 recipes(sandwich, fruit salad, cereal) 5 times each. Each recoding (1 subject prepared 1 recipe 1 time) is segmented into 30 seconds segments. The training data has 288 segments, collected from 3 subjects, and the test data has 180 segments, collected from the fourth subject. It contains 3 different macro-activity classes and 10 different micro-activity classes. Each acceleration data contains data in x, y and z directions and motion caption data records 29 joints data in x, y and z directions. It is challenging because the set is small and segments are messed up in no order.

NTU RGBD Dataset The dataset is one of the largest set for human action recognition, which has 60 different classes collected using the Kinect Camera. It has RGBD data and skeleton data from 40 subjects. The skeleton data contains 25 joints data for each subjects. We utilize the NTU dataset to warm the model, although it has different number joints and is collected by different system, compared with the CAD.

4.2 Data Processing

We process the data in different ways for GCN and CNN. For GCN, we use the skeleton data after sorting in some specific order. For CNN, we utilize both skeleton and acceleration data, which are mapped into images. The transformed features are shown in Fig. 3. The strategy of missing data padding will be detailed in the following.

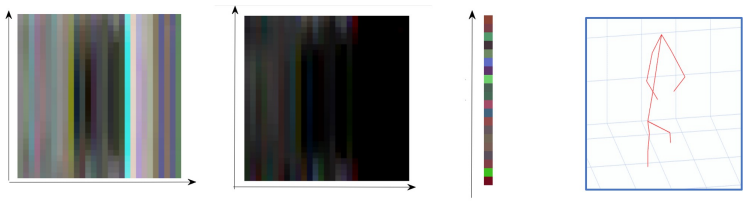

Fig. 3 Illustration of mapping images from skeleton and acceleration data. The left one is the raw skeleton image mapping from skeleton data, The second one is the difference skeleton image corresponding to difference data of adjacent frames. The third one is correspond to acceleration data. Pixels on the horizontal axis represent the joint points, and pixels on the vertical axis represent the frames, which is slicing from 1 to 29 frames in these images for view. We only use right arm aceleration data, thus there only one pixel on horizontal axis on the acceleration image. And the fourth image is the skeleton data after visualization, which is fed into GCN model.

Processing Missing Data There are two types of data missing in CAD. One is that almost all the data miss in one file, the other is that the data miss in one or several block but with complete neighbour information. We discard the files that miss too much information. To address the latter missing problem, the missing data is fixed with 0 or nearest frame value.

Because the skeleton data has more information than acceleration data, we align the acceleration data with the skeleton data to keep more information when joint training for CNN model. Finally, we choose to utilize right arm data for its completeness. Otherwise we have to cut down the skeleton data file number, because there are 127 missing files for left wrist, 19 missing files for right arm and 10 missing files for left hip.

Processing for CNN We map the values in (x, y, z) directions of skeleton and accelerator data to (r, g, b) channels in images like [8], finally, we get three images for each segment, as shown in Fig.3. Each skeleton data has approximately 3000 frames and the accelerator data has about 1500 frames except for blank files. Instead of using all the data, we only sample some frames of skeleton data and cut the accelerator data to 1024 frames. The shape of raw and difference skeleton image are $(s, 29, 3)$, the first dimension is corresponding to the sample number, the second dimension represents the joint numbers and the third dimension contains the information from (x, y, z) directions. The shape of acceleration image is (1024, 3), the dimensions are corresponding to frames and (x, y, z). It is worth noting that the NTU dataset is extended to 29 joints by copying the latest four joints to keep the data structure consistent with CAD.

Processing for GCN The NTU data for warming the the GCN is processed as mentioned above. The skeleton data is processed by the way illustrated in Fig.3 for the GCN's input structure requirement. We conduct experiments to indicate how the sample number affects the result. We get the input with the shape of $(3, s, 29, 1)$, the dimensions are corresponding to (x, y, z), sampling numbers, 29 joints and 1 subject. Then we convert the input sequence into a spatial-temporal graph, which will be computed by successive convolution operations.

4.3 Results

We conduct extensive experiments to demonstrate that the proposed models are effective and study how the parameters affect the performance of the models. The evaluation criterion is 10 folders cross-validation where the top accuracy for each folder is recorded and average accuracy is computed.

4.3.1 Study for CNN

For macro-activity recognition, we conduct experiments to evaluate the effect of the sampling numbers of skeleton data, the scale of training and validation dataset, the training and validation sets dividing based on subjects, the joint training and the methods to fill the missing data.

We conduct 3 groups experiments (E.1-3 in Table.1) to show the performance of joint training. Acceleration model (AM) and skeleton model (SM) experiments are carried out to evaluate the performance of single modal data training. Both networks are jointed together to show the effects of the joint training for multi-modal data. In these experiments, we set the CAD 80% for training and 20% for validation, sample 256 frames and pad missing values with the nearest frame values. As shown in Table.1, either the AM and SM can't achieve a higher accuracy than AM+SM. The results indicate that the proposed joint training framework can achieve better results.

Compared to E.3, E.4-11 in Table.1 are conducted. E.4 is conducted to show the influence of the missing data processing strategy, compared with the baseline E.3 which pads missing data with the nearest frame values. Padding with nearest values can achieve higher accuracy. E.5 and E.6 are focusing on the influence of sample number. Sampling 256 frames for skeleton data in E.3 performs best. Sampling less maybe loss some useful features and sampling more maybe increase some noise. E.7 and E.8 are developed to indicate the affects of scale of training and validation dataset. Dividing the set 90% for training and 10% for validation can get a higher result. E.9-E.11 are developed to show whether the model have the generalization ability for every subject. Similarly, the accuracy shows the learning ability for each people, despite the low accuracy for a small training set.

Table 1 Experiments(E for short in Table) for macro-activity recognition, E.1-3 are developed for joint training. E.4-6 is developed for data processing and E.7-11 are developed for set spliting.

E	Topic	Setting	Accuracy
1	Model	AM	0.460
2	Model	SM	0.776
3	Model	AM + SM	0.828
4	padding method	padding with 0	0.760
5	sample-wise	sampling 128 frames	0.768
6	sample-wise	sampling 320 frames	0.772
7	scale-wise	80% for training, 20% for validation	0.701
8	scale-wise	70% for training, 30% for validation	0.716
9	subject-wise	s1,2 for training, s3 for validation	0.537
10	subject-wise	s1,3 for training, s2 for validation	0.566
11	subject-wise	s2,3 for training, s1 for validation	0.525

4.3.2 Study for CGN

For micro-activity recognition, comparative experiments are conducted to show how these parameters affect the result, and which is the best setting. The E.2 in Table.2 is the baseline of E.1 and E.3-7. In E.2, we sample 300 frames from skeleton data, set 80% CAD for training and 20% for validation and pad the missing data with 0.

We carry out experiments based on various sample numbers ranging from 200 to 600 frames, the results are shown in Table.2 E.1-5. The experiment results indicate that sampling 300 frames perform the best. The analysis is same to that mentioned in Sect.4.3.1. Furthermore, E.6 is developed to evaluate the padding methods, which shows that padding with 0 in E.2 can achieve a better performance. We also evaluate the proposed GCN model by spliting CAD in different scales and different subjects. The comparative results between E.7 and E.2 indicate that the spliting strategy of 80% for training and 20% for validation performs better than that of 70% for training and 30% for validation. And E.8-10 are conducted to evaluate the subject-wise spliting, which shows that the proposed GCN model is more effective in distinguishing the actions of the second people.

Table 2 Experiments for micro-activity recognition, E.1-6 are set for data processing, E.7-10 are set for set spliting.

E	Topic	Setting	Accuracy
1	sample-wise	sampling 200 frames	0.489
2	sample-wise	sampling 300 frames	0.831
3	sample-wise	sampling 400 frames	0.489
4	sample-wise	sampling 500 frames	0.589
5	sample-wise	sampling 600 frames	0.578
6	padding method	padding nearest values	0.740
7	scale-wise	70% for training, 30% for validation	0.761
8	subject-wise	s1,2 for training, s3 for validation	0.479
9	subject-wise	s1,3 for training, s2 for validation	0.601
10	subject-wise	s2,3 for training, s1 for validation	0.560

5 Conclusions

In this paper, we presented a joint-training framework to fuse skeleton and acceleration data for activity recognition using CNN. To enable the mutil-label recognition, the the binary cross-entropy (BCE) loss function is introduced to our designed GCN model. The experimental results show that the proposed GCN model achieve average accuracy of 82.8% for micro-activity recognition on the CAD. By combining the acceleration and skeleton data together, our CNN model obtains the accuracy of 83.1% for macro-activity recognition.

During the course of our experiments we found that the accuracy of the macro-activity recognition is unstable. Several possible reasons are given here based on our studies: i) the dataset scale is too small; ii) we don't utilize the time stamp features. This may cause the mismatch of the features extracted by AM and SM. These problem may be solved when expanding the dataset, utilizing the time features or combining SM+AM with LSTM.

Appendix

The training details are given in Table.3.

Table 3 The summary of the resources used

Items	Details
Sensor modalities	Motion capture system and right arm accelerometer
Features uesd	As described in Sect.4.2 and shown in Fig. 3
Programming language	Python
Packages used	pytorch
Machine specification	Four pieces of GPU:GeForce GTX 1080 Ti, 12 Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 30 Gibs of RAM
Training and testing time	Macro-recognition: 6 hours for pre-training on NTU data set, and 2 hours for once on challenge data set. Micro-activity recogniton: 20 hours for pre-training on NTU, and 12 minutes once on challenge data set.

References

1. Alia S, S., Lago, P., Takeda, S., Adachi, K., Benaissa, B., Ahad, M., Inoue, S.: Summary of the cooking activity recognition challenge. In: Human Activity Recognition Challenge, Smart Innovation, Systems and Technologies. Springer Nature (2020)
2. Bayat, A., Pomplun, M., Tran, D.A.: A study on human activity recognition using accelerometer data from smartphones. Procedia Computer Science **34**, 450–457 (2014)
3. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
4. Catal, C., Tufekci, S., Pirmit, E., Kocabag, G.: On the use of ensemble of classifiers for accelerometer-based activity recognition. Applied Soft Computing **37**, 1018–1022 (2015)
5. Chen, Y., Xue, Y.: A deep learning approach to human activity recognition based on single accelerometer. In: 2015 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1488–1492. IEEE (2015)
6. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
7. Ignatov, A.: Real-time human activity recognition from accelerometer data using convolutional neural networks. Applied Soft Computing **62**, 915–922 (2018)
8. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3288–3297 (2017)
9. Lago, P., Takeda, S., Adachi, K., Alia S, S., Matsuki, M., Benaissa, B., Inoue, S., Charpillet, C.: Cooking activity dataset with macro and micro activities (2020). DOI 10.21227/hyzg-9m49
10. Lago, P., Takeda, S., Alia S, S., Adachi, K., Benaissa, B., Charpillet, F., Inoue, S.: A dataset for complex activity recognition with micro and macro activities in a cooking scenario. Preprint (2020)

11. Li, C., Zhong, Q., Xie, D., Pu, S.: Skeleton-based action recognition with convolutional neural networks. In: 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 597–600. IEEE (2017)
12. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3595–3603 (2019)
13. Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Symbiotic graph neural networks for 3d skeleton-based human action recognition and motion prediction. arXiv preprint arXiv:1910.02212 (2019)
14. Pourbabaee, B., Roshtkhari, M.J., Khorasani, K.: Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients. IEEE Transactions on Systems, Man, and Cybernetics: Systems **48**(12), 2095–2104 (2018)
15. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1010–1019 (2016)
16. Shi, L., Zhang, Y., Cheng, J., LU, H.: Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. arXiv preprint arXiv:1912.06971 (2019)
17. Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L.: Deep learning for sensor-based activity recognition: A survey. Pattern Recognition Letters **119**, 3–11 (2019)
18. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence (2018)
19. Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J.: Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International Conference on Mobile Computing, Applications and Services, pp. 197–205. IEEE (2014)
20. Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., Zheng, N.: View adaptive neural networks for high performance skeleton-based human action recognition. IEEE transactions on pattern analysis and machine intelligence **41**(8), 1963–1978 (2019)
21. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE multimedia **19**(2), 4–10 (2012)
22. Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., Xie, X.: Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)