The Study of Graph Convolutional Network for Activity Recognition

Yanrui Zhu¹, Yuxin Ren²*

¹ Inspur Group Co., Ltd, Jinan, Shandong Province, 250000, China
² School of Control Science and Engineering, Shandong University, Jinan, Shandong Province, 250000, China
³ Corresponding author’s e-mail: renyuxin@sdu.edu.cn

Abstract. GCN is an excellent single-modal machine learning model for processing non-Euclidean data, such as skeleton data. Conventional GCN only uses single-modal data as its input. When we have multiple modal data, the application of GCN will be limited. To utilize multi-modal data advantages, we propose a method to combine GCN and LSTM to extract skeleton feature and acceleration feature, respectively. To test our method, we did a lot of experiments on Cooking Activity Dataset.

1. Introduction

Action recognition has been an active and widely studied research topic in the computer vision community recently. With the rapid development of deep learning, many efficient methods have emerged in the field of action recognition. Most of the methods are video-based [3, 11, 10]. They have proposed many effective feature extraction methods and achieved good results on the action recognition dataset. But skeleton-based methods usually have a wider range of applications, because the skeleton data can be obtained from videos or sensors. Skeleton data is a relatively intuitive feature, which contains rich temporal and spatial information. Due to the advancement of sensors and various pose estimation algorithms, such as the Microsoft Kinect [16] and OpenPose [2], many skeleton-based action recognition algorithms have been proposed and achieved good performance [17, 4, 7, 14, 15, 13, 8, 9, 8]. Among them the GCN model is a very effective algorithm, because it can address the problems caused by the non-euclidean data structure. Although the above methods have achieved good performance, most of them only use single-modal data, limiting the improvement of algorithm performance. To address this problem, we propose a novel model based on AS-GCN [8] to fuse skeleton data and acceleration data.

2. Method

2.1. AS-GCN with Multimodal Data Fusion

Nine AS-GCN blocks stack the backbone of the AS-GCN network, and each AS-GCN block is composed of ASGC and T-CN in series, as shown in Figure 1. ASGC is used to extract spatial features of the skeleton, as TCN is used to extract features in the time dimension. The input features is a skeleton including the connection between the joints and each joint’s position sequence. After ASGC, the coordinate information of each joint is mapped to a high-dimensional feature space. Our main idea is to use LSTM to map four acceleration samples into a high-dimensional feature and add a joint to the...
skeleton data by copy the joint data at the four acceleration sensors’ geometric centre to fuse this feature.

\[ X_{\text{sin}} = RT \times n_{\text{joint}} \times din \]

\[ X_{\text{din}} = RT \times (n_{\text{acc}} \times din) \]

The input data is divided into two parts, including acceleration data and skeleton data. Let \( X_{\text{sin}} \) be the input skeleton feature and \( X_{\text{din}} \) be the acceleration data, where \( T \) represents the total number of frames, \( n_{\text{joint}} \) and \( n_{\text{acc}} \) represent the number of mocap and acceleration sensors, respectively. Let \( X_{\text{out}} \) be the output features obtained from ASGC, and \( X_{\text{out}} \) be the output features of LSTM layers, where \( d_{\text{out}} \) is the dimension of out-out features. The ASGC operation is,

\[ X_{\text{out}} = \text{ASGC}(X_{\text{sin}}) \]  

(1)

and the LSTM operation is,

\[ X_{\text{out}} = \text{LSTM}(X_{\text{in}}) \]  

(2)

To perform data fusion, we reshape the shape of \( X_{\text{out}} \) to \( T \times d_{\text{out}} \) and then apply linear weighting to \( X_{\text{out}} \) and the newly added joint; that is,

\[ X_{\text{inh}} = \alpha_1 X_{\text{out}} + \alpha_2 X_{\text{inh}} \]  

(3)

where \( X_{\text{inh}} = X_{\text{out}} \) represents the features of added joints of all frames, and \( \alpha_1 \) and \( \alpha_2 \) represent the Weight coefficient.

2.2. AS-GCN with BCE Loss

In order to adapt the model to the multi-label samples of Cooking Activity Dataset, we adopted the BCE loss function to enable the multi-label classification. Here is the definition of BCE loss function:

\[ L \left( y_{\hat{i}}, y_i \right) = -y_{\hat{i}} \log y_{\hat{i}} - (1 - y_{\hat{i}}) \log (1 - y_{\hat{i}}) \]  

(4)

where \( y_{\hat{i}} \) is the output of AS-GCN, \( y_i \) is the label of sample. There are 10 categories, \( i \) is corresponding to micro-activity \( i \) of CAD dataset. And the output of GCN is \( \hat{y} \).

3. Experimental Results

3.1. Dataset

**Cooking Activity Dataset** CAD has two types of data, acceleration and skeleton data. The acceleration data is collected by four accelerators from four wearable devices on the right arm, left hip,
right wrist and left wrist. At the same time, the skeleton data comes from a motion capture system. Four subjects prepared 3 recipes including sandwiches, fruit salads, and cereals, 5 times each. Each complete action is divided into 30-second segments. The training data has 288 samples collected from 3 subjects, and the test data has 180 samples, all from the fourth subject. The dataset has 10 different micro-activity classes and 3 different macro-activity classes. Each acceleration sample contains data in $x$, $y$ and $z$ directions and mocap data records 29 joints position information in $x$, $y$ and $z$ directions.

### 3.2. Data Processing

#### Processing Missing Data

There are three types of data missing in CAD. Almost all the data miss one file; another is that the data miss one or several blocks but with complete neighbor information. The third is the missing frames. We discard the files that miss too much information. For the second case, the lost data is fixed to the average value of 5 frames before and after the current frame. As for the third one, We pad the missing frames with zeros and put them at the end of the file. The above operations ensure the continuity of the action as much as possible. Since the skeleton data has more valid frames than the acceleration data, we align the acceleration data with the skeleton data according to their timestamp when training the GCN model and the LSTM model to preserve more information. Finally, we utilize the mocap data of 29 joints and the acceleration data of 4 joints.

#### Processing for Pre-train

CAD only provides 468 samples, of which 224 samples are used as the training set, 57 samples are used as the validation set, 105 samples are used as the test set, and the remaining samples are discarded due to excessive data missing. Therefore, we put the model on the NTU [12] for pre-training to improve the model’s generalization ability. Therefore, we put the model on the NTU for pre-training to improve the model’s generalization ability. To achieve this, we must ensure the similarity of the input skeleton structure (including node number and location). Here we solve this problem by adding new nodes for NTU and sorting CAD nodes. For adding nodes to the NTU sample, we copy the NTU sample’s nodes according to the joint distribution of the CAD sample. As for the joint ordering, we sort the joint order of CAD samples according to the NTU’s joint number.

#### Data Alignment

Data alignment is one of the critical challenges of multimodal machine learning [1], and more useful information can be obtained by implementing this method. Here we use timestamps to align the acceleration data and mocap data. Specifically, due to the severe lack of acceleration data, we take mocap data as the central part of the input data and supplement the corresponding acceleration data based on the time stamp information of the mocap data.

### 3.3. Results

The GCN model is pre-trained on NTU dataset and fine-tuned on Cooking Activity Dataset (CAD) [5, 6]. The dataset and the data processing are introduced in Sect.3.2. In the following, results are presented in Sect.3.3.

The hyperparameters $\alpha_1$ and $\alpha_2$ control the proportion of acceleration feature in the skeleton feature. We conduct extensive experiments to test the model performance on the test set. The E.1 in the Table.1 is the baseline of our experiments which only uses the skeleton data. E.2-5 are sets with different degrees of data fusion controlled by the relative size of $\alpha_1$ and $\alpha_2$.

| E  | $\alpha_1$ | $\alpha_2$ | Accuracy |
|----|------------|------------|----------|
| 1  | 1          | 0          | 0.594    |
| 2  | 1          | 0.1        | 0.590    |
| 3  | 1          | 0.4        | 0.567    |
| 4  | 1          | 0.5        | 0.583    |
| 5  | 1          | 0.6        | 0.560    |
4. Conclusions
In this paper, we present a novel GCN model with multimodal data fusion in Cooking Activity Dataset. BCE loss function is applied in this model to enable the mutil-label classification. From the results, we can know when the acceleration data is introduced, the performance of the algorithm has dropped slightly. In AS-GCN, additional links, action-links and structural-links, are taken into account. A-Link and S-link can provide the more implicit correlation, which may be defused if we add the newly acceleration joint in a wrong place. Besides, the feature extraction network uses the simplest LSTM, which may not extract acceleration information well. Another reason is the lack of many acceleration data, which may become an interference during the testing phase.

References
[1] Baltrusaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence 41(2), 423–443 (2018)
[2] Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., Sheikh, Y.A.: Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence (2019)
[3] Fan, H., Li, Y., Xiong, B., Lo, W.Y., Feichtenhofer, C.: Pyslowfast. https://github.com/facebookresearch/slowfast (2020)
[4] 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)
[5] 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
[6] 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)
[7] 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)
[8] Li, M., Chen, S., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Actional-structural graph convolutional networks or skeleton-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3595–3603 (2019)
[9] 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)
[10] Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 9–14. IEEE (2010)
[11] Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L.: Tea: Temporal excitation and aggregation for action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 909–918 (2020)
[12] Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
[13] 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)
[14] 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)
[15] 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)
[16] Zhang, Z.: Microsoft kinect sensor and its effect. IEEE multimedia 19(2), 4–10 (2012)
[17] 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)