Paper:

An abnormal behavior detection method for elderly people at home

Zhipeng Cheng, Kaoru Hirota, Yaping Dai, Zhiyang Jia*

School of Automatic, Beijing Institute of Technology
No.5 Zhongguancun South Street, Haidian District, Beijing 100081, P.R.China
E-mail: zp_cheng@bit.edu.cn

Abstract. With the aggravation of the aging problem and empty-nester problem, caring for the health problems of the elderly has become an important field. Compared with some available behavior detection technologies, an abnormal behavior detection method for elderly people at home with Kinect v2 is proposed. This method obtains skeleton sequences with Kinect v2 and constructs a graph convolutional network to classify actions. The skeleton sequences are treated as a graph and separated into 4 partition graphs as the input of a spatial-temporal graph convolutional netwrok(ST-GCN). The effectiveness of this method is verified on filtered NTU RGB+D datasets including 10 actions. The accuracy rate of our method achieves 90.33%, and is 4.98% higher than traditional ST-GCN.

Keywords: Behavior detection; Skeleton sequences; Graph convolutional network; Partition graph

1. Introduction

The population aging has become a common problem worldwide. According to the *China National population development plan (2016 - 2030)*, Chinese elderly population over 60 will account for 25% in 2030. Due to the social pressure, younger people have to work or live apart from family, which causes the empty-nester problem. Therefore, caring for the health problem of the elderly, especially for the elderly living alone, has become a field increasing importance[1].

As aging and weakened muscle strength all increase the risk of falls. Some studies have indicated that fall in elderly people is a very dangerous situation especially when they live alone[1,2]. And fall is one of the leading causes of death in the elderly[3]. Thus, it is necessary to identify the fall and inform the guardian or medical staff of the elderly. Some diseases can also be detected by camera through behavior, such as cough, vomiting and so on. We define those behaviors as abnormal behavior at home.

In recent years, a variety of methods have proposed to recognize human actions. A first distinction can be made between wearable and non-wearable techniques[4]. The formers use some kinds of wearable device to capture body movements. Although wearable device can be accurate and suitable even in outdoor conditions, it limits body movements and causes physical discomfort. In addition, the elderly is easier to forget wearing those devices, which results in device out of action.

Compared with wearable devices, non-wearable techniques using camera have less interference to the elderly. At present, vision-based action recognition has become the main method in action recognition. Extracting the shape of the human body for behavior analysis is a common method. Improved dense trajectories(IDT)[5] is a traditional but useful method without deep learning. With the development of deep learning, more and more action recognition methods based on computer vision have been proposed, which has become a research hotspot[6]. Meanwhile, there are many network models used to recognize action with computer vision, such as CNN, RNN, LSTM, GCN et al.[7]. RGB video, depth video and skeleton data can be the input of models.

The aim of this paper is to develop anabnormal detect method for elderly people at home with Kinect v2 camera, which can accurately recognize 10 human poses, including *pick up*, *sit down*, *sit up*, *cough*, *falling down*, *headache*, *chest pain*, *back pain*, *neck pain* and *vomiting*. We adopt a spatial-temporal graph convolutional network(ST-GCN) to classify action from skeleton data. The skeleton sequences are treated as graph and separated into 4 partition graphs as input of model. It aims to capture high-level properties of each part and adopt an aggregation across subgraphs to learn the relations between them. Finally, we validate our work on a public action datasets: NTU RGB+D.

The rest of this paper is organized as follows: some related work about action recognition will be introduced in Section 2. Section 3 gives a detailed introduction to our method, and Section 4 shows the experimental result. The conclusion is presented in Section 5.

2. Related work

2.1. Input of action recognition networks

In the past few decades, the RGB and depth videos have been widely studied as input for human action recognition[7,8,9]. The video based action recognition methods mainly focus on modeling spatial and temporal represen-

tations from frames[10]. Although video based methods are end-to-end and have achieved promising results, those methods rely on lots of calculations and has less robust when facing complicated background as well as changing conditions involving in body scales, viewpoints and motion speed[7].

In 2017, Cao Z et al.[11] present an algorithm about human pose estimation from RGB images. Then, the CMU team releases an open source human pose estimation algorithm named OpenPose, which can detect human joints from RGB or RGB+D videos and form an sequence of 2D or 3D joints coordinates. Also, Microsoft proposes a Kinect 2 camera that can get 3D joints in 30 FPS. Thanks to those advanced human pose estimation algorithms and devices, it is easier to gain 3D skeleton data and make action recognition based on skeleton become an active research topic[10,12].

3D skeleton data represents the body structure with a set of 3D coordinate positions of key joints. There are three main notable characteristics in skeleton sequences[7] : 1) Spatial information. Each node and its adjacent nodes have strong correlations and contain abundant body structural information. 2) Temporal information. Temporal continuity exists not only in the same joints, but also in the body structure. 3) The co-occurrence relationship between spatial and temporal domains. Compared with video data, skeleton sequences peform more robust to complicated backgroud and easy to obtain action features.

2.2. Deep learning methods using skeleton data

The main methods using deep learning with skeleton data can be divided into three parts: RNN-based method, CNN-based method and GCN-based method[7].

Recurrent neural network(RNN) is an effective way for processing sequential data by taking the output of previous time step as the input of the current time step[12]. It represents the skeleton data as a sequence of joints based on the designed traversal strategy, which is then modeled with RNN-based architectures[13]. Long short term memory(LSTM) and Gated recurrent unit(GRU) are variants of RNN, which introduce gates and linear memory units inside RNN. Those variants are proposed to make up the shortages such as vanishing gradient and long-term temporal modeling problems of the standard RNN. However, RNNs are lack of the spatial modeling ability and need a long time to train models. Meanwhile, RNNs still need to solve the gradient attenuation between layers when use activation funtion like sigmoid and tanh.

Convolutional neural network(CNN) has been applied to the skeleton-based action recognition successfully. Different from RNNs, CNN models can efficiently learn spatial information. Generally, to stasfy the need of CNNs' input, skeleton sequence data is transfromed from vector sequence to a pseudo-image[13]. It represents the skeleton sequences as an image by encoding temporal dynamics and skeleton joints simply as rows and columns respectively. However, it just focuses on the connection of each node and its adjacent nodes. That will ignore some latent correlation and loss some useful action features.

Recently the Graph convolution network(GCN) has been adopted in skeleton-based action recognition because of the effective representation for the graph structure[14]. Instead of representing skeleton data as sequences or pseudo-images, GCN-based methods define the skeleton as a graph with joints and bones, which can be considered as vertexes and edges. Compared with the other two methods, the GCN-based methods are more intuitive to reflect human body.

2.3. Spatial-temporal graph convolutional network

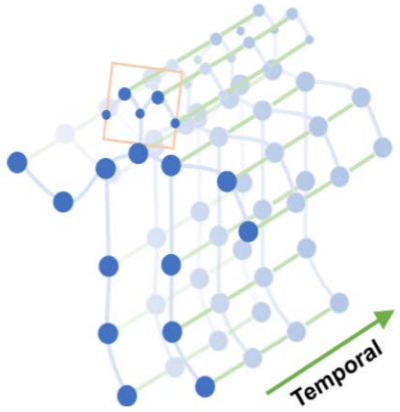

Fig. 1. The spatial temporal graph of a skeleton sequence

Yan et al.[14] propose ST-GCN to capture the patterns embedded in the spatial configuration of the joints as well as their temporal dynamics. Human skeleton is intuitively represented as a sparse graph with joints as nodes and natural connections between them as edges. As shown in **Fig.1**, the skeleton sequences contain many frames formed by nodes, and each node corresponds to a joint of the human body. It constructs an undirected spatial temporal graph $G = (V, E)$ on a skeleton sequence with N joints and T frames.

In this graph, the node set $V = \{v_{ti} \mid t = 1, \ldots, T, i = 1, \ldots, N\}$ includes the all joints in a skeleton sequence. The edge set E is composed of two subsects as $E_S = \left\{v_{ti}v_{tj} \mid (i, j) \in H\right\}$ and $E_T = \{v_{ti}v_{(t+1)i}\}$, where H is the set of naturally connected human body joints. E_S is defined as the intra-skeleton between two nearest joints according to the set of naturally connected human body joints. E_T is defined as the same joints in consecutive frames, which can represent a node's trajectory over time.

The main idea of ST-GCN is using a spatial convolution and temporal convolution to learn about action feature from the spatial temporal graph. The graph convolution operation on node v_{ti} is written as:

$$f_{\text{out}}(v_{ti}) = \sum_{v_{tj} \in N(v_{ti})} \frac{1}{Z_{ti}\left(v_{tj}\right)} f_{in}\left(v_{tj}\right) \cdot \mathbf{w}\left(l_{ti}\left(v_{tj}\right)\right), \qquad (1)$$

where the normalizing term $Z_{ti}\left(v_{tj}\right) = |\{v_{tk} \mid l_{ti}(v_{tk}) = l_{ti}\left(v_{tj}\right)\}$ equals the cardinality of the corresponding subset, which is added to balance the contributions of different subsets to the output. f is the feature map. $\mathbf{w}$ is the weighting function similar to the original convolution operation, which provides a weight vector based on input feature. $N(v_{ti})$ is the neighbor set of a joint node v_{ti}, and defined as:

$$N(v_{ti}) = \left\{v_{qj} | d\left(v_{tj}, v_{ti}\right) \leq K, |q-t| \leq \lfloor \tau/2 \rfloor\right\}, \quad (2)$$

where K controls the spatial range to be included in the neighbor nodes and is referred to the spatial kernel size. Similarly, τ controls the temporal range to be included in the neighbor graph and is reffered to the temporal kernel size. The mapping function l_{ti} maps a node in the neighborhood to its subset label. It also defines a labeling map l_{ST} to complete the convolution operation on the spatial temporal graph as:

$$l_{ST}\left(v_{qj}\right) = l_{ti}\left(v_{tj}\right) + (q - t + \lfloor \tau/2 \rfloor) \times K. \quad (3)$$

[14] provides 3 partition strategies to define the spatial label map l_{ti} and points out that a more advanced partitioning strategy for label map l_{ti} will lead to better modeling capacity and recognition performance.

3. Methodology

3.1. Obtain skeleton data

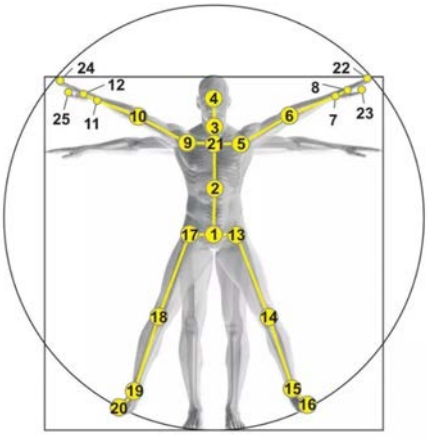

Fig. 2. Configuration of 25 joints by Kinect 2.The label of the joints are: 1.base of the spine; 2.middle of the spine; 3.neck; 4.head; 5.left shoulder; 6.left elbow; 7.left wrist; 8.left hand; 9.right shoulder; 10.right elbow; 11.right wrist; 12.right hand; 13.left hip; 14.left knee; 15.left ankle; 16.left foot; 17.right hip; 18.right knee; 19.right ankle; 20.right foot; 21.spine; 22.tip of the left hand; 23.left thumb; 24.tip of the right hand; 25.right thumb.

Kinect v2 has a RGB sensor and an infrared sensor that senses the outside world, which can output RGB and depth image. The depth image mainly perceives the environment through black and white spectroscopy, and uses different gray levels corresponds to the physical distance between objects and sensor. The distance is farther as the gray value increase. Kinect v2 obtains 3D joints coordinates through built-in skeleton tracking system from the captured images. It can generate skeleton sequences at a rate of 30 frames per second.

In home behavior detection, we can use the above device to extract skeleton sequences of the elderly people, and the Kinect v2 camera can be mounted on a mobile robot or in a fixed position. The configuration of 25 joints by Kinect v2 is shown in **Fig.2**, which can describe the spatial information of the human body briefly.

3.2. A ST-GCN model with partition graph

In traditional ST-GCN models[14], the skeleton graph is heuristically predefined and represents only the physical structure of the human body. It is difficult for the model to capture the dependency between two joints, which are located far away from each other on the predefined human-body-base graphs. However, human body can be visualized as connected rigid parts, much like a deformable part-based model[15]. It is natural to think human body as a combination of multiple body parts, such as head, torso, hands, arms, legs and feet. What's more, the human action can be considered as composed of each trunk movement. For example, walk involves the movement of arms and legs mainly. Therefore, the graph of human body skeleton can be divided into subgraphs, where each subgraph represents a part of human body. And the action can be detected according to the combination of each part movement.

3.2.1. Partitioning strategy

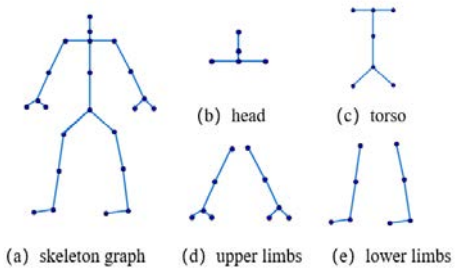

Fig. 3. Human body structure

Considering the structure of human body, Thakkar et al.[15] propose that separate the graph into 2 parts with axial and appendicular, 4 parts as shown in **Fig.3** and 6 parts that divide limbs by left and right. It proves that 4 parts perform best in those partitioning strategies. Thus, we separated the skeleton graph into 4 parts including head, torso, upper limbs and lower limbs.

An overlap of at least one joint between two adjacent parts is defined to cover all natural connections between parts in the skeleton graph. For example, two shoulder joints are shared between head and upper limbs, which represent the link between them. So the skeleton graph G

transforms into :

$$G = \bigcup_{p\in\{1,2,3,4\}} G_p \mid G_p = \left(V_p, \mathrm{E}_p\right). \tag{4}$$

Due to the skeleton graph has little nodes, we use 1-neighborhood($K = 1$) for spatial dimension, Equation (2) can be transfomed into spatial neighborhood N_{1p} and temporal neighborhood N_τ :

$$N_{1p}(v_i) = \left\{v_j \mid \mathbf{d}\left(v_i, v_j\right) \le 1, v_i, v_j \in V_p\right\}, \tag{5}$$

$$N_\tau(v_{it_a}) = \left\{v_{it_b} \mid \mathbf{d}\left(v_{it_a}, v_{it_b}\right) \le \left\lfloor \frac{\tau}{2} \right\rfloor\right\}. \tag{6}$$

For ordering vertices in the neighborhoods, we adopt a uni-labeling partitioning strategy for label map with a spatial label from [14]. The spatial label to weigh vertices in N_{1p} of each vertex eaually is defined as:

$$l_S\left(v_{jt}\right) = \left\{1 \mid v_{jt} \in N_{1p}(v_{it})\right\}, \tag{7}$$

and the temporal label to weigh vertices acroos frames in N_τ differently is defined as:

$$l_T\left(v_{it_b}\right) = \left\{\left((t_b - t_a) + \left\lfloor \frac{\tau}{2} \right\rfloor\right) \mid v_{it_b} \in N_\tau(v_{it_a})\right\} \tag{8}$$

3.2.2. Spatial-temporal graph convolution with partition graph

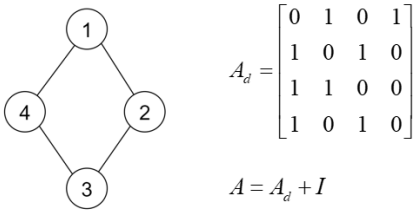

Fig. 4. The connection matrix of a graph. The matrix A_d is the adjacency matrix of graph, and an identity matrix I represents self-connections. A is the connection matrix.

The aim of graph convolutions over parts is to capture high-level properties of parts and adopt an aggregation across subgraphs to learn the relations between them. According to [15], convolutions for each part are performed separately and the results are combined using an aggregation function F_{agg}. As shown in **Fig.4**, the connection matrix A represents the intra-body connections of joints. In a single frame, using the labeled spatial receptive fields from Equation (7) and the connection matrix, the spatial convolutions can be defined as:

$$\mathbf{Y}_p(v_{it}) = \sum_{v_{jt}\in N_{1p}(v_{it})} \mathbf{A}_p(i,j)\mathbf{W}_p\left(l_S\left(v_{jt}\right)\right)\mathbf{X}_p\left(v_{jt}\right), \quad p \in \{1,\ldots,4\}, \tag{9}$$

where A_p is a normalized connection matrix A of subgraph G_p. X_p is the input feature and Y_p is the output feature of subgraph. The convolution parameters W_p is a learnable spatial convolution kernel of subgraph, while the neighbors of v_i only in that part(N_{1p}) are considered. It can be shared across parts or kept separate.

This graph convolutions cross parts is aiming to learn rich representations from nodes in parts and the connnection between parts. In order to combine the feature across parts, the function F_{agg} combines the imformation at shared vertices is defined as:

$$\begin{aligned}\mathbf{Y}_S(v_{it}) &= F_{agg}(\{\mathbf{Y}_1(v_{it}), \ldots, \mathbf{Y}_4(v_{it})\}) \\ &= \sum_i^4 \mathbf{W}_{agg}(i)\mathbf{Y}_i\left(v_{it_b}\right),\end{aligned} \tag{10}$$

where v_{it} is the joints between two adjacent parts. The aggregation function F_{agg} is defined as a weighted sum fusion.

To learn about the temporal feature, a temporal convolution is adopted with the temporal label according to Equation (8):

$$\mathbf{Y}_T(v_{i_a}) = \sum_{v_{j_b}\in N_\tau(v_{i_a})} \mathbf{W}_T\left(l_T\left(v_{it_b}\right)\right)\mathbf{Y}\left(v_{it_b}\right), \tag{11}$$

where Y_T is the output obtained after applying temporal convolution on Y of the τ frame. To the shared joints, Y is the output Y_S obtained after aggregating at single frame. And to the other joints, Y is Y_p that from each subgraph. $\mathbf{W}_T$ is a part graph convolution kernel.

3.2.3. Implementation model

As shown in **Fig.5**, the main part of a ST-GCN unit[14] is two convolution layers that implement spatial convolution and temporal convolution. To avoid overfitting, the residual mechanism and dropout are applied on each ST-GCN unit. The batch normalization layer is applied to normalize data.

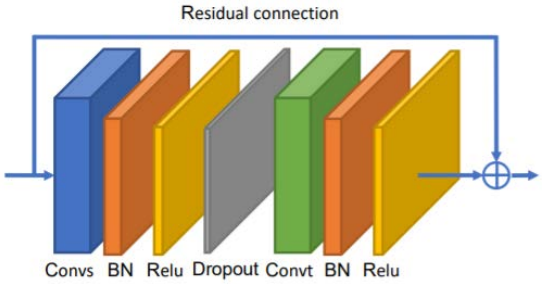

Fig. 5. The structure of a ST-GCN unit with residual

Our model is similar to the model proposed by YAN et al.[14]. As shown in **Fig.6**, the model takes the input as a tensor having features for each vertex in spatial-temporal graph of human skeleton seqeunces and outputs a vector of class scores by softmax function. It mainly consists of 9 ST-GCN units(each units with the four part spatial convolution $\mathbf{W}_S$ kenerls, one temporal convolution $\mathbf{W}_T$ kernel). The ST-GCN unit is defined as a tensor as [in-

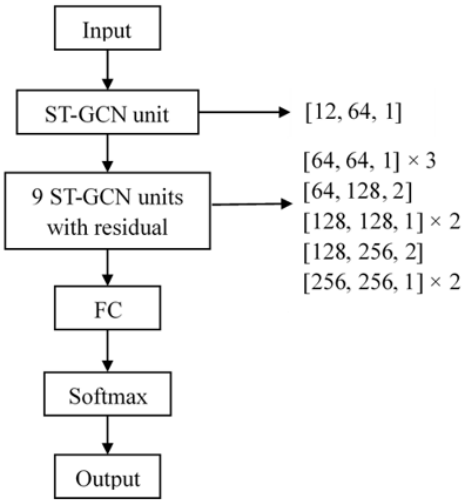

Fig. 6. The structure of model

put channels, output channels, stride]. The skeleton sequences contains 3 channels including x, y and depth, and there are 4 subgraphs. So the head of model is a ST-GCN unit with [12,64,1]. In the 9 units with residual, the first 3 units have 64 output channels, the next 3 units have 128 output channels, and the last 3 have 256 output channels. When the stride is 2, the main aim is similar to a pool layer to keep the main features while reducing the parameters and feature dimensions.

4. Experiment

4.1. Dataset

NTU RGB+D dataset[16] is a classical dataset about action recognition in recent years. It contains 60 different human action classes that are divided into three major group: daily actions, mutual actions, and health-related actions. There are 56,880 action samples in total which are performed by 40 distinct subjects. It provides RGB video, depth map sequence, 3D skeleton data and infrared video captured by there kinect v2 cameras concurrently in the same height with 3 different angles: $-45°$, $0°$ and $45°$.

We choose 10 classes as our experiment data, including: pick up, sit down, sit up, sneeze/cough, falling down, headache, chest pain, back pain, neck pain, nausea/vomiting. The actions except pick up, sit down, sit up are defined as abnormal behavior. The train set and validation set are separated by cross subjects where subject 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38 are train set and others are validation set. The size of train set is 6698 and the size of validation set is 2751.

4.2. Experiment setup and results

The experimental environment is under Unbuntu 16.0.4 with i9 7900x and a GTX 1080Ti GPU. The algorithm is conducted on the Pytorch deep learning framework. Crossentropy is selected as the loss function to backpropagate gradients and Stochastic Gradient Descent(SGD) is used as the optimizer and run the training for 70 epochs. The initial learning rate is set to 0.1 and decays by 0.1 at epochs 20, 40 and 60. The dropout rate is set to 0.5 and the batch size is set to 16.

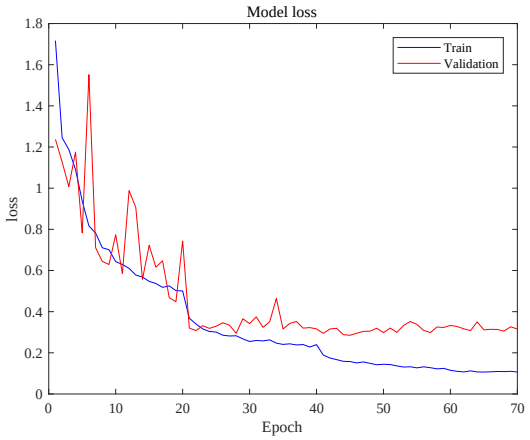

Fig. 7. Loss curve

The filtered datasets with 10 action classes is used to verify our model. We plot the obtained train loss and validation loss to observe the training process during epochs in **Fig.7**. As shown in **Fig.7**, the loss curve of the training process decreased steadily, while the validation losse curve fluctuated in the early epochs but tended to decline steadily. The training loss is approximately 0.2 less than the validation loss in the last epoch. It causes by the setting of NTU dataset, that not each person performs each action in each view. The actions exist difference between different people, and even the same action performed by a person exists difference in different views.

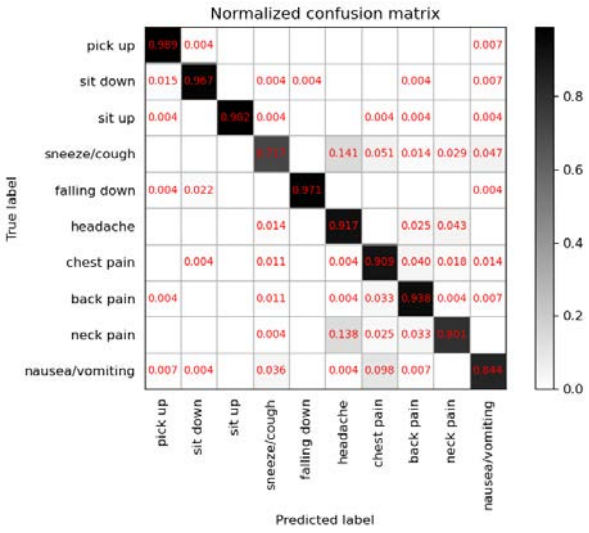

Fig. 8. Confusion matrix

To evaluate the overall performance, the confusion matrix of our methods on validation sets are illustrated in **Fig.8**. This method achives good perfomance on classfying 7 action classes. However, we can also observe

that the perfomance on classfying sneeze/cough, neck pain and nausea/vomiting is not good. Sneeze/cough and neck pain are more likely to classify as headache wrongly, while nausea/vomiting is mostly mistaken for chest pain. This is because some features of these actions are similar. For example, both of cough and headache have minor movement of head and the contacts between hands and head. It's easy to misunderstand people when classfy some fuzzy actions in vision without other modal imformation like sounds, which is likely to be confused by the trained networks. To distinguish the similar actions and improve the accuracy rate, the distance between hands and other joints like head, neck, spine and so on should be considered in the further work. For example, compared with headache, the hands are closer to neck than it to head when neck pain.

Table 1. Comparison results obtained on filtered dataset.

Methods	Accuracy(%)
ST-GCN[14]	85.35
ST-GCN with bones	87.51
Our method	**90.33**

For the filtered dataset, we respectively completed comparison experiments with ST-GCN, which is a state-of-the-art methods. The baseline of ST-GCN[14] is 81.53% on NTU RGB+D dataset with 60 classes. To match our experiment setting, we reproduce a ST-GCN model proposed in [14], which foucuses on the joint coordinates. And a ST-GCN model with bone input, where bone contains two joints information, is done to compare with our method too. The final result is shown in **Table 1**. It can be seen that bone input ST-GCN performs better than traditional ST-GCN, since bone information contains more feature than joints information. Also, our method has an improved effect on behavior detection than traditional ST-GCN. It captures high-level properties of each part and learn the relations between each subgragh.

5. Conclusion

In this paper, we propose an abnormal behavior detection for elderly people in home using skeleton-based action recognition with Kinect v2 and ST-GCN with part graph input. Firstly, this method is based on skeleton data from Kinect v2 and transform human skeleton as a graph with 25 joints according to human natural structure. Then, separate the skeleton graph into four parts graph to learn about the relation between two distant nodes in the skeleton graph by using partition graph convolutions in the spatial neighborhood and aggregate function. Finally, use temporal convolutions to obtain the temporal features of action, and classify the action.

According to the comparison experiments, this method is more accurate than traditional ST-GCN. It can be more effective to detect whether the abnormal behavior has happened, so as to respond quickly to provide help by notifying guardian or call for help. The proposed method is based on human joints information, which has more robust to the environment changes. Therefore, this method can be applied on a mobile robot with Kinect v2 to provide home care for elderly people in practice. In the future work, we would try to combine object detection and decision technology to form a system that can detect the dangerous situation from behavior for elderly people and respond to it.

Acknowledgements

This work was supported by the Beijing Municipal Natural Science Foundation under Grant No. 3192028.

References:

[1] Yao, C., Hu, J., Min, W. et al. A novel real-time fall detection method based on head segmentation and convolutional neural network. J Real-Time Image Proc, 2020.

[2] X. Kong, L. Meng and H. Tomiyama, Fall detection for elderly persons using a depth camera. 2017 International Conference on Advanced Mechatronic Systems (ICAMechS), Xiamen, 2017, pp. 269-273.

[3] A. F. Ambrose, G. Paul, and J. M. Hausdorff, Risk factors for falls among older adults: A review of the literature. Maturitas, vol. 75, no. 1,pp. 51-61, 2013.

[4] Y. Xu, J. Chen, Q. Yang and Q. Guo, Human Posture Recognition and fall detection Using Kinect V2 Camera. 2019 Chinese Control Conference (CCC), Guangzhou, China, 2019, pp. 8488-8493.

[5] Wang, Heng, and Cordelia Schmid. Action recognition with improved trajectories. Proceedings of the IEEE international conference on computer vision, 2013.

[6] G. Sun and Z. Wang, Fall detection algorithm for the elderly based on human posture estimation. 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2020, pp. 172-176.

[7] Ren B, Liu M, Ding R, et al. A Survey on 3D Skeleton-Based Action Recognition Using Learning Method. arXiv preprint arXiv:2002.05907, 2020.

[8] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[9] Hu, Jian-Fang,Zheng W S, Lai J et al. Jointly learning heterogeneous features for RGB-D activity recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

[10] C. Si, W. Chen, W. Wang, L. Wang and T. Tan. An Attention Enhanced Graph Convolutional LSTM Network for Skeleton-Based Action Recognition. CVPR2019, Long Beach, CA, USA, 2019, pp. 1227-1236.

[11] Cao Z, Simon T, Wei S E, et al. Realtime multi-person 2d pose estimation using part affinity fields. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017: 7291-7299.

[12] Li, Chuankun, et al. Skeleton-based action recognition using LSTM and CNN. 2017 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), 2017.

[13] Shi L, Zhang Y, Cheng J, et al. Skeleton-based action recognition with directed graph neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019: 7912-7921.

[14] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. Thirty-second AAAI conference on artificial intelligence, 2018.

[15] Thakkar, Kalpit, and P. J. Narayanan. Part-based graph convolutional network for action recognition. arXiv preprint arXiv:1809.04983, 2018.

[16] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, Gang Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016