

An Improved Feature-Based Method for Fall Detection

Leiyue YAO, Wei YANG, Wei HUANG

Abstract: Aiming at improving the efficiency and accuracy of fall detection, this paper fuses traditional feature-based methods and Support Vector Machine (SVM). The proposed method provides two major improvements. Firstly, the classic features were adopted and together with machine learning technology form an improved and efficient fall detection method. Secondly, the definition of a threshold which needs massive experiments was now learned by the program itself. Compared with the current popular end-to-end deep learning methods, the improved feature-based method fusing machine learning technology shows great advantages in time efficiency because of the significant reduction of the input parameters. Additionally, with the help of SVM, the thresholds need no manual definition, which saves a lot of time and makes it more precise. Our approach is evaluated on a public dataset, TST fall detection dataset v2. The results show that our approach achieves an accuracy of 93.56%, which is better than other typical methods. Furthermore, the approach can be used in real-time video surveillance because of its time efficiency and robustness.

Keywords: computer vision; fall detection; feature-based method; handcrafted feature; Support Vector Machine (SVM)

1 INTRODUCTION

According to the report of the World Health Organization, approximately 28~35% of people aged 65 and over fall each year and 32~42% of those over 70 years of age. In fact, falls exponentially increase due to age-related biological changes, which lead to a high incidence of falls and fall-related injuries in ageing societies [1]. Falls sustained by subjects can have severe consequences and may be fatal, especially for elderly persons who live alone [2, 3]. If a falling person cannot get help in a short period of time, the situation will be even worse. For these and many other reasons, the number of research works on fall detection has increased dramatically over recent years. Automatic detection of human falls provides help to reduce the time between the fall and the arrival of medical attention. [4, 5]

In [6], fall detection methods were classified into two major categories: handcrafted action representations and learning-based action representations.

Earlier works paid attention to recognizing actions from RGB data. In these research works, various handcrafted action representations have been raised along with the developments of action recognition. Literature [7-10] introduced 4 typical vision-based fall detection methods, in which 4 classic features, such as Head position, human shape, centroid height, and height velocity, are respectively used. With the advent of depth camera, images with depth information (RGB-D image) which is also called depth images are more and more widely used in human action recognition because of 3D information provided by them. In [11], a 3D bounding box of height and composition of width and depth was calculated frame by frame. When the shape of the 3D bounding box exceeds a threshold value, a fall can be detected. However, these shape-based methods perform poorly in fall-like activity discrimination. In [12], two classic features, head height and centroid height of the subject, were used to detect falls. In [13], researchers raised a new feature for fall detection, which is named body major orientation. Our previous work [14] proposed a feature, torso angle, to discriminate fall and fall like activities. In this paper, torso angle is used to identify starting key frame (SKF) of a possible fall video.

Different from handcrafted features, deep learning method is designed to mimic the way of how humans observe the world from a biological perspective. These kinds of method always contain hierarchical layers and much more trainable parameters than shallow architectures. Recurrent Neural Network (RNN) is one of the most used networks. However, the short-term memory makes it impossible to be applied in real world. Based on this, Long Short-Term Memory (LSTM) which is usually used as a hidden layer of RNN was proposed to model temporal evolutions. Zhu, W. et al. [15] took the skeleton of each frame as the input and adopted a novel regularization scheme to learn the co-occurrence features of skeleton joints. Du, Y. et al. [16] proposed an end-to-end hierarchical RNN for skeleton based action recognition. Aiming at handling the noise and occlusion in 3D skeleton data, Liu, J. et al. [17] introduced new gating mechanism within LSTM to learn the reliability of the sequential input data and accordingly adjust its effect on updating the long-term context information stored in the memory cell. It cannot be denied that RNN-based methods achieve great performance in action recognition. However, most of this kind of methods is end-to-end contracture which tends to overstress the temporal information [18].

Recent years, RGB-D image based action recognition becomes a new research hotspot. Zhu et al. [19] represented actions as a sequence of key poses. However, the method may become defective in fall-like activities discrimination because key poses do not contain temporal information. In [20], researchers adopted dynamic clustering method together with multiclass SVM to automatic classified 12 daily activities. However, sliding window used in dynamic clustering method is not time-efficient. Furthermore, selection of the sliding window which has strong effect on detection accuracy is a tricky work. Rodrigo et al. [21] used multi-class recognition method to classify different kinds of gesture by encoding the motion data of human joints as a one-dimensional array. However, the identification of the SKF and ending key frame (EKF) for a continuous action is not well addressed in their work. Since its simplification and over-average performance, similar methods by using different kinds of devices emerged. In [22], wearable sEMG sensors were used to extract 15 features. Then, these features were

encoded as an array and fed into a classifier to recognize daily activities and detect falls. Nowadays, with the popularization of mobile network, many researchers are trying to use mobile phone instead of traditional sensors to collect motion information of the subject [23, 24]. However, wearable sensors, especially those precise sensors, are always brittle and uncomfortable for people to wear. Moreover, wearable device based methods will not work if the devices were not well equipped. Therefore, wearable device based method is not a suitable choice for action recognition.

In this paper, we proposed an effective way to improve feature-based method by combining classical features with machine learning technology. Inspired by [21], we also represent action as sequences of features. Different from key poses used in [19], all the characters used in our method contain temporal information. Additionally, thresholds were defined by program itself because of the usage of machine learning technology. More introduction of our method is elaborated in section 3.

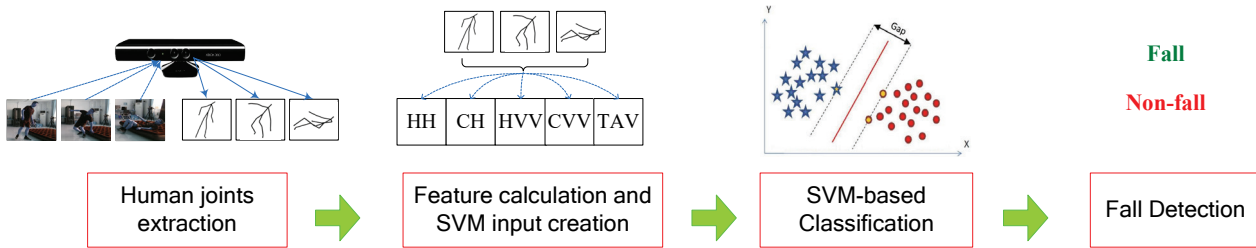


Figure 1 The pipe line of our method

The remainder of this paper is organized as follows. Section 3 introduces our model and algorithm. Section 4 indicates our experiments and the results. Section 5 concludes the paper and discusses future work.

3 OUR METHOD

Our proposed method utilizes four classic features and one self-defined feature to form a one-dimensional array as the input of SVM. Kinect sensor v2.0 is used to capture depth images and human skeleton is extracted from depth image for feature calculation. An SVM is followed to train/predict fall or non-fall actions. Fig.1 shows the pipeline of our method. Instead of creating an end-to-end action recognition model, our method combines classic features with machine learning technology. On the one hand, the input parameters were dramatically reduced. On the other hand, the strong representativeness for fall action of classic features were fully utilized.

3.1 Depth Image and Human Skeleton

Although there are a lot of fall detection methods based on monocular camera, depth image provides 3D information and many other conveniences for researchers to trace monitored person and calculate features. In our method, Kinect sensor v2.0 was adopted for depth image and human skeleton collection. Fig. 2 shows the human skeleton provided by Kinect.

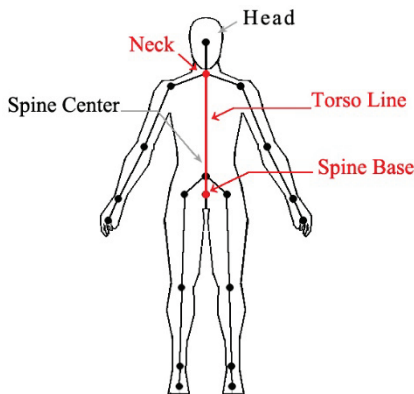


Figure 2 Human Joints captured by Kinect sensor

In the best situation, Kinect v2.0 can accurately capture 25 joints of the subject. In our method, to avoid complex computation and improve the efficiency of the detection program, only 4 joints are used for feature calculation, which are labelled in Fig. 2. The red line, so called torso line, is used for a self-defined feature extraction, which is named as torso angle [14].

3.2 Torso Angle and Balance Estimation

Because gravity line is vertical to the ground, all lines parallel with y-axis in depth image can be seen as gravity lines when Kinect is installed horizontally. The angle form by gravity line and torso line is the self-defined feature—torso angle (α). Fig. 3 is a depth image captured by Kinect, which clearly shows the torso angle.

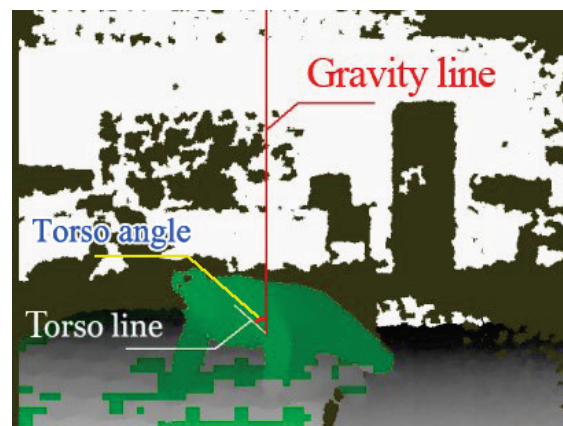


Figure 3 Illustration of torso angle

Torso angle is a strong representative feature to discriminate fall and fall-like activities. Compared with those classic height based features, torso angle lays a clear line between balance and unbalance. Formula (1) is used to calculate its value:

$$\cos(\alpha) = \cos(\overrightarrow{NS}, \overrightarrow{SG}) = \frac{\overrightarrow{NS} \cdot \overrightarrow{SG}}{|\overrightarrow{NS}| \times |\overrightarrow{SG}|} \quad (1)$$

where \overrightarrow{NS} means the vector from joint neck to joint spine base, and \overrightarrow{SG} denotes the vector from joint spine base to any point on the gravity line.

As for almost all kinds of falls, torso angle is increasing during a fall and reaches its peak value when body touches the ground. The limit of stability test (LOST) shows that an adult will lose his/her balance when the body is leaning forward/backward over 12.5 degrees or leaning left/right over 16 degrees [25]. According to this result, we take a frame as the SKF of a possible fall when torso angle exceeds 12.5 degrees. Additionally, the 33th frame after SKF can be taken as the EKF, which has already been proved in our previous work [14]. Therefore, the features used in our method are calculated in the area of SKF to EKF.

3.3 Feature Definition and Calculation

There are five features in total to form the input of an SVM. A possible fall can be represented as a one-dimensional array by these features. All the features were extracted from the depth image as there are 30 frames per second when we use Kinect to capture depth image. Thus, the given period of time from SKF to EKF can be represented as $T = \{t_1, t_2, \dots, t_n\}$. Then, features can be defined as follows.

Definition 1: Minimum Head Height (HH)

HH means the minimum distance from head to ground during the time period from SKF to EKF. As for some serious falls, head may bounce once it hits bottom, so HH cannot be simply extracted from EKF. The head height in the m -th frame, hh_m , can be calculated by:

$$hh_m = \frac{|Ax+By+Cz+D|}{\sqrt{A^2+B^2+C^2}} \quad (2)$$

where $\{A, B, C, D\}$ is the ground coefficients which can be fetched from the API of Kinect sensor, and $\{x, y, z\}$ is the 3D position of head joint in a certain frame.

Definition 2: Minimum Centroid Height (CH)

Like feature HH, CH denotes the minimum distance from centroid of the monitored person to ground. To simplify the computation work, spine center joint is approximately considered as the centroid of the human beings. Formula (2) can also be used to calculate feature CH.

Definition 3: Maximum Head Vertical Velocity (Max(HVV))

Max(HVV) stands for the maximum head position change in the vertical direction. Suppose $HH = \{hh_1, hh_2, \dots, hh_3\}$ is the discrete representation of monitored person's head height corresponding to $T = \{t_1, t_2, \dots, t_n\}$. Then, $HVV = \{hvv_1, hvv_2, \dots, hvv_n\}$ can be obtained by:

$$hvv_m = \frac{hh_m - hh_1}{t_m - t_1} \quad (3)$$

Finally, the maximum value in $\{hvv_1, hvv_2, \dots, hvv_n\}$ is selected as a feature.

Definition 4: Maximum Centroid Vertical Velocity (Max(CVV))

The same as Max(HVV), Max(CVV) indicates maximum centroid vertical velocity. Calculation of this feature is just like Max(HVV).

Definition 5: Maximum Torso Angle Increasing Velocity (Max(TAV))

Different from features 1-4, torso angle is a self-defined feature in our previous work, which is used to discriminate fall and fall-like activities. Compared to those sliding window based SKF detection methods, the efficiency of the program can be improved dramatically because of the only calculation of an angle in a single frame. According to formula (4), torso angle can be traced frame by frame. Once its value exceeds the threshold (12.5 degree), the current frame is considered as the SKF of a possible fall.

In our program, torso angle is not only used as a feature, but also as a trigger to judge whether the current action should be quantified as sequence of characters or not.

Assume $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$, then $TAV = \{tav_1, tav_2, \dots, tav_n\}$ can be fetched by:

$$tav_m = \frac{\alpha_m - \alpha_1}{t_m - t_1} \quad (4)$$

3.4 The Improved Method

The kernel of the proposed method is using handcrafted features to encode a possible fall action into a one-dimensional float array with 5 items. Then, an SVM is adopted to judge whether the possible action is fall or not. Compared to traditional feature-based methods, our improved method uses multiple features to improve the accuracy and robustness of the detection program. Additionally, SVM is adopted to automatically learn the threshold of the fusing features, which avoids manual and tricky threshold definitions. Furthermore, a more efficient feature, torso angle, is used instead of the commonly used sliding window to dramatically improve the efficiency of the program.

Fig. 4 shows the general block of our improved method. There are 3 steps in total, and steps 2 and 3 will not be triggered unless torso angle exceeds its threshold value.

The most challenging task in deep-based human action recognition is to deal with the temporal dimension. However, those classic handcrafted features were well designed and paid enough attention to temporal issues. By fusing the advantages of machine learning technologies, traditional feature-based methods can be enhanced dramatically.

Especially worth emphasizing is the feature-based SKF detection method. Compared with the sliding window based method, the former one lets the accuracy and efficiency be improved by an order of magnitude. According to our previous work, most frames can be processed in 0.1~0.8 ms [14], which can be ignored because the time interval between two frames is far larger than the time consumption of torso angle calculation. The pseudo-code is listed as follows.

According to the pseudo-code, only step 1 works unless torso angle exceeds its threshold. This strategy effectively reduces the computational load because fall is a small probability event.

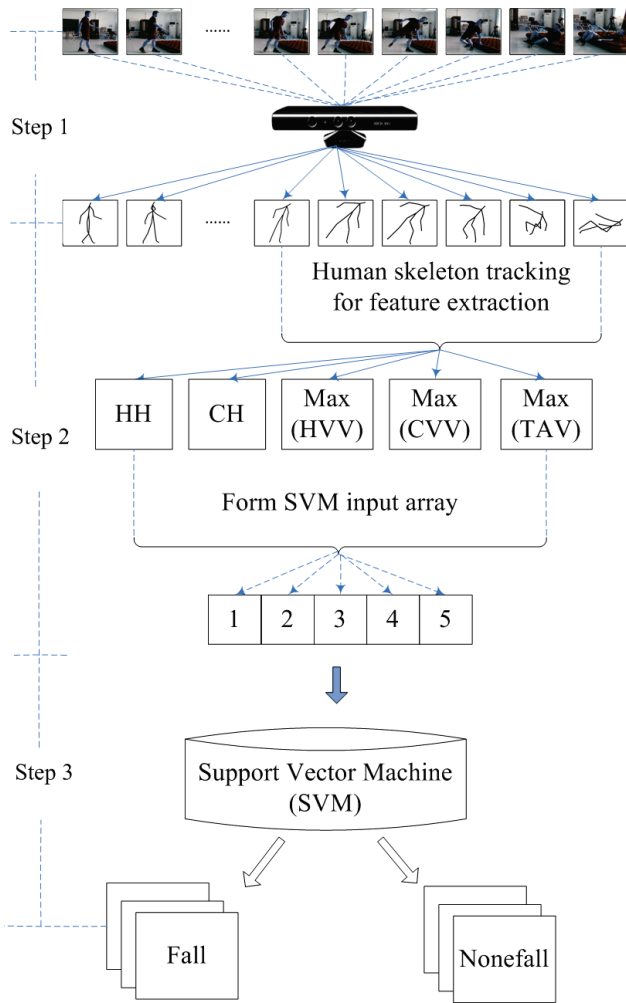


Figure 4 The general block of the improved fall detection method

Algorithm 1: Activity quantification and fall detection

Input: depth images captured by Kinect sensor

Initialisation: SKF = 0;
featureArray = new float[5];

Output: isFall

```

svm = initSVM();
Loop1: torsoAngle = calculateTorsoAngleOfCurrentFrame();
    if (torsoAngle > 12.5) {
SKF = getCurrentFrame();
        goto Loop 2;
    }
Loop 2: trackedFrame = getCurrentFrame() - SKF;
    if (trackedFrame ≤ (EKF - SKF)) {
        featureArray[0] = calHH(Head);
        featureArray[1] = calCH(SpinCenter);
        featureArray[2] = calMaxHV(Head);
        featureArray[3] = calMaxCV(SpinCenter);
        featureArray[4] = calMaxAV(SpinBase, Neck);
    }
    else {
        isFall = svm.predict(featureArray);
ResetAllparas();
        goto Loop1;
    }

```

4 EXPERIMENTS AND RESULTS

We have used a Kinect sensor V2.0 to collect motion data of joints of the subject, and adopted Visual Studio 2013 and emgu.cv 3.1 to implement our method. For the SVM, we use an "Inter" function kernel and keep default values of other parameters. The experiments were run on a desktop computer with Intel Core i7-4790 3.60 GHz processor, NVidia GeForce GTX 1060 GPU with 6 GB video memory and 16 GB of RAM clocked at 1333 MHz.

4.1 Method Evaluation on TST V2

We systematically evaluate the effectiveness of the proposed method and compare it with other classic fall detection methods on a publicly available datasets: TST V2 [27]. The dataset contains 264 video sequences simulated by 11 volunteers with 4 activity daily living (ADL) classes and 4 fall classes. The videos were collected by Microsoft Kinect v2 and Inertial Measurement Unit (IMU). Each volunteer repeats each activity 3 times. In the experiments, we only used the motion data of human joints captured by Kinect, and the data from IMU was not adopted.

In all experiments, 70% of the samples of TST V2 were used as the training set, and 30% were used as the testing set. In the first step, the program ran on the training set for feature vectors calculation. After feature matrix is formed, program runs on the testing set to evaluate the accuracy of the method. Finally, the program runs on the whole dataset. Tab. 1 records the experiment results.

Table 1 The performance of our proposed method on the TST V2 dataset.

Category	Activity	Sample number	Result	
			√	×
ADL	Sit on a chair	33	33	
	Walk and grasp	33	33	
	Walk back and forth	33	33	
	Lay down	33	30	3
Fall	Frontal fall	33	30	3
	Backward fall	33	30	3
	Side fall	33	32	1
	Backward fall and sit	33	26	7

Table 2 The performance of our improved proposed method on the TST V2 dataset.

Category	Activity	Sample number	Result	
			√	×
ADL	Sit on a chair	33	33	
	Walk and grasp	33	33	
	Walk back and forth	33	33	
	Lay down	33	30	3
Fall	Frontal fall	33	30	3
	Backward fall	33	30	3
	Side fall	33	32	1
	Backward fall and sit	33	33	

As shown in Tab. 1, our program did not receive a good performance in the group of "backward fall and sit". After carefully reviewing the wrongly detected samples, we found out that the 7 wrongly detected samples were performed too similar to "sit". Thus, they were not detected correctly. However, we do not think it is an unsolvable problem. Since centroid height is quite different between "backward fall and sit" and "sit on a chair", this problem can be solved by adding weight on the feature CH.

Tab. 2 shows the result after centroid height is added to the program as a judging condition (the selection of

threshold follows the rule that is used in [8]). However, we abandon this improvement in our final program, because we do not want to add any manual threshold selection to our method.

4.2 Comparison with Other Methods

Our proposed method was compared with 4 classic feature-based methods: the head height velocity and head height based method [7], head height based method [8], 3D bounding box based method [11], head height and centroid height based method [12].

Fig. 5 shows the statistics of the comparison test result. As shown in the graph, although our method did not receive the best score in the term of *TNR*, it obtained the best results in *TPR* and *ACC*, which are more important evaluation criteria than *TNR*.

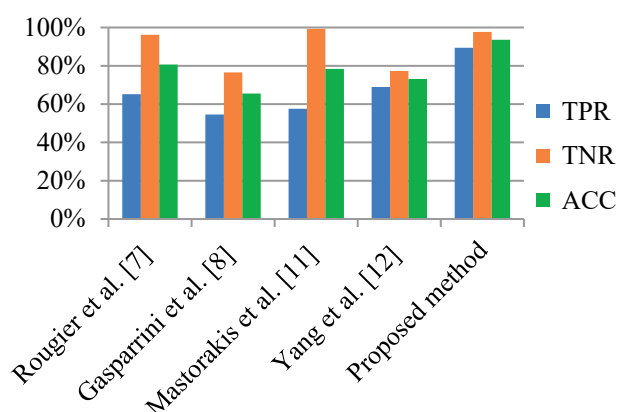


Figure 5 The statistics of 4 classic feature-based methods and our proposed method where $TPR = \frac{TP}{TP + FN}$, $TNR = \frac{TN}{TN + FP}$ and $ACC = \frac{TP + TN}{TP + TN + FP + FN}$.

Tab. 3 records the detail information of the experiment. As shown in the table, all methods can perfectly differentiate ‘walk’ activities from ‘fall’ activities. However, height-based methods [8, 12], no matter head height based method or centroid height based method, failed to distinguish between fall and fall-like activities. The 3D bounding box method [11] and head-tracking method [7] made the same performance in the experiments. Although these two methods performed much better than the height based methods in the group of ‘lie down’, they all worked poorly in the group of ‘backward falling and sitting’.

The 3D bounding box method received the second-best result in the test. Especially in the terms of “lie down”, only 1 sample was wrongly detected. After reviewing its algorithms, we believe that the accuracy of this method can be improved by adding feature “centroid height” to the algorithm. However, it is impossible for other 3 feature-based methods to improve the accuracy in the term of ‘backward falling and sit’, because head height will be always higher than its threshold value. Additionally, although head height velocity together with head height form a stronger feature and perform better than only head height based method, it is still not an effective way to discriminate fall and “lie down”. Moreover, it can be ascertained that the result will be even worse if the subject acts ‘lying down’ more quickly than he did in the videos of the dataset.

Table 3 Comparisons of the fall detection and fall discrimination capability on the TST v2 dataset with other Kinect-based fall-detection approaches.

Approach	Fall				ADL			
	Frontal fall	Backward fall	Side fall	Backward fall and sit	Sit on chair	Walk and grasp	Walk back and froth	Lie down
Rougier et al. [7]	25	25	33	3		1		4
Gasparini et al. [8]	25	31	33	8				31
Mastorakis et al. [11]	19	28	25	4	1			1
Yang et al. [12]	25	29	33	4				30
Proposed method	30	30	32	26				3

5 CONCLUSION

In this paper, we raised a new feature, torso angle, for efficient SKF identification of a possible fall. Then, a fall detection method based on classic features and machine learning method was proposed. With the help of torso angle, balance and unbalance can be effectively and efficiently identified by program. Whenever the torso angle exceeds 12.5 degrees, the current action is considered as a possible fall and encoded as a one-dimensional float array by 5 hand-crafted features. Then, an SVM is adopted to judge whether the action is a fall or not. Compare with classic feature based fall detection methods, our method has advantages in automatically thresholds discovering. While compared with deep learning based method, our method dramatically reduces the input parameters.

According to the introduction of Kinect, the best distances from Kinect to the monitored person should be in the range of 0.4 to 3 meters. Beyond this distance, the skeleton data become unreliable. Although the narrow field-of-view of the Kinect sensor may limit its usage in a relatively bigger area, it can be solved by adding more devices.

Experiment results showed that the accuracy of our method reaches 93.56% when it is tested on the TST v2 dataset. It is much better than other 4 classic feature-based methods, and the new feature, torso angle, can let our method discriminate fall-like activity more effectively and efficiently than sliding window. Moreover, compared with monocular camera, depth image and the only use of joint motion information provide a guarantee for people’s privacy

Acknowledgment

This research was supported by the Science and technology research project of Jiang Xi science and technology agency under Grant 20171BBE50060 and the science and technology planning project of Nan Chang science and technology bureau under Grant 2016-ZCJHCXY-013 and 2016-ZCJHCXY-016.

6 REFERENCES

- [1] Ozcan, A., Donat, H., Gelecek, N., Ozdirenc, M., & Karadibak, D. (2005). The relationship between risk factors for falling and the quality of life in older adults. *Bmc Public Health*, 5(1), 90-100.

- <https://doi.org/10.1186/1471-2458-5-90>
- [2] Yardley, L. & Smith, H. (2002). A prospective study of the relationship between feared consequences of falling and avoidance of activity in community-living older people. *The Gerontologist*, 42(1), 17-23. <https://doi.org/10.1093/geront/42.1.17>.
- [3] Salvà, A., Bolívar, I., Pera, G., & Arias, C. (2004). Incidence and consequences of falls among elderly people living in the community. *Medicina Clínica*, 122(5), 172-176. [https://doi.org/10.1016/S0025-7753\(04\)74184-6](https://doi.org/10.1016/S0025-7753(04)74184-6)
- [4] Doughty, K., Lewis, R., & McIntosh, A. (2000). The design of a practical and reliable fall detector for community and institutional telecare. *Journal of Telemedicine and Telecare*, 6(suppl 1), 150-154. <https://doi.org/10.1258/1357633001934483>
- [5] Huang, C. N., Chiang, C. Y., Chen, G. C., Hsu, S. J., Chu, W. C., & Chan, C. T. (2010). Fall detection system for healthcare quality improvement in residential care facilities. *Journal of Medical and Biological Engineering*, 30(4):247-252. <https://doi.org/10.1007/s10856-009-3866-2>
- [6] Zhen, X., Shao, L., Maybank, S. J., & Chellappa, R. (2016). Handcrafted vs. learned representations for human action recognition. *Image and Vision Computing*, 55, 39-41. <https://doi.org/10.1016/j.imavis.2016.10.002>
- [7] Rougier, C., Meunier, J., Starnaud, A., & Rousseau, J. (2013). 3D head tracking for fall detection using a single calibrated camera. *Image & Vision Computing*, 31(3), 246-254. <https://doi.org/10.1016/j.imavis.2012.11.003>
- [8] Gasparrini, S., Cipitelli, E., Spinsante, S., & Gambi, E. (2014). A depth-based fall detection system using a kinect? *Sensors*, 14(2), 2756-2775. <https://doi.org/10.3390/s140202756>
- [9] Rami, A., Momani, M., & Daoud, M. I. (2017). Fall Detection for Elderly from Partially Observed Depth-Map Video Sequences Based on View-Invariant Human Activity Representation. *Applied Sciences*, 7(4): 316-335. <https://doi.org/10.3390/app7040316>
- [10] Rougier, C., Auvinet, E., Rousseau, J., Mignotte, M., & Meunier, J. (2011). Fall Detection from Depth Map Video Sequences. *International Conference on Smart Homes and Health Telematics*. Springer, Berlin, Heidelberg. 121-128. https://doi.org/10.1007/978-3-642-21535-3_16
- [11] Mastorakis, G. & Makris, D. (2014). Fall detection system using kinect's infrared sensor. *Journal of Real-Time Image Processing*, 9(4), 635-646. <https://doi.org/10.1007/s11554-012-0246-9>
- [12] Lei, Y., Yanyun, R., Huosheng, H., & Bo, T. (2015). New fast fall detection method based on spatio-temporal context tracking of head by using depth images. *Sensors*, 15(9), 23004-23019. <https://doi.org/10.3390/s150923004>
- [13] Planinc, R. & Kampel, M. (2013). Introducing the use of depth data for fall detection. *Personal and Ubiquitous Computing*, 17(6), 1063-1072. <https://doi.org/10.1007/s00779-012-0552-z>
- [14] Yao, Leiyue, et al. (2017). A New Approach to Fall Detection Based on the Human Torso Motion Model. *Applied Sciences*, 7(10), 993-1100. <https://doi.org/10.3390/app7100993>
- [15] Zhu, Wentao, et al. (2016). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. *Thirtieth AAAI Conference on Artificial Intelligence AAAI Press*, 3697-3703.
- [16] Yong, D., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. *Computer Vision and Pattern Recognition IEEE*, 1110-1118. <https://doi.org/10.1109/CVPR.2015.7298714>
- [17] Liu, Jun, et al. (2016). Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition. *European Conference on Computer Vision Springer*, Cham, 816-833. https://doi.org/10.1007/978-3-319-46487-9_50
- [18] Wang, P., Li, Z., Hou, Y., & Li, W. (2016). Action recognition based on joint trajectory maps using convolutional neural networks. *Proceedings of the 24th ACM international conference on Multimedia (MM '16)*, 102-106. <https://doi.org/10.1145/2964284.2967191>
- [19] Guangming, Z., Liang, Z., Peiyi, S., & Juan, S. (2016). An online continuous human action recognition algorithm based on the kinect sensor. *Sensors*, 16(2), 161. <https://doi.org/10.3390/s16020161>
- [20] Manzi, A., Dario, P., & Cavallo, F. (2017). A Human Activity Recognition System Based on Dynamic Clustering of Skeleton Data. *Sensors*, 17(5), 1100. <https://doi.org/10.3390/s17051100>
- [21] Rodrigo Ibañez, Álvaro Soria, Teyseyre, A., Rodriguez, G., & Campo, M. (2017). Approximate string matching: a lightweight approach to recognize gestures with kinect. *Pattern Recognition*, 62, 73-86. <https://doi.org/10.1016/j.patcog.2016.08.022>
- [22] Xugang, X., Minyan, T., Miran, S. M., & Zhizeng, L. (2017). Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable sEMG sensors. *Sensors*, 17(6), 1229. <https://doi.org/10.3390/s17061229>
- [23] Tong, C., Lian, Y., Zhang, Y., Xie, Z., Long, X., & Niu, J. (2016). A novel real-time fall detection system based on real-time video and mobile phones. *Journal of Circuits, Systems and Computers*, 26(04), 1-16. <https://doi.org/10.1142/S0218126617500566>
- [24] Álvarez de la Concepción, Miguel Ángel, Soria Morillo, L. M., Álvarez García, Juan Antonio, & González-Abril, Luis. (2016). Mobile activity recognition and fall detection system for elderly people using Ameva algorithm. *Pervasive and Mobile Computing*, S1574119216300505. <https://doi.org/10.1016/j.pmcj.2016.05.002>
- [25] Melzer, I., Benjuya, N., Kaplanski, J., & Alexander, N. (2008). Association between ankle muscle strength and limit of stability in older adults. *Age and Ageing*, 38(1), 119-123. <https://doi.org/10.1093/ageing/afn249>
- [26] Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68, 346-362. <https://doi.org/10.1016/j.patcog.2017.02.030>
- [27] Gasparrini, Samuele, et al. (2016). *Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion*. ICT Innovations 2015. Springer International Publishing, 99-108. https://doi.org/10.1007/978-3-319-25733-4_11

Contact information:

Leiyue YAO, Professor
School of Information Engineering,
Nanchang University,
Nanchang, China, 330031
Leiyue_yao@163.com

Wei YANG, Lecturer
Jiangxi University of Technology,
The Center of Collaboration and Innovation,
Nanchang, China, 330098
Wei.yang@163.com

Wei HUANG, Associate Professor
(Corresponding author)
School of Information Engineering,
Nanchang University,
Nanchang, China, 330031
huangwei@ncu.edu.cn