IET Image Processing

Special issue

Call for Papers

Be Seen. Be Cited.
Submit your work to a new IET special issue

"Advancements in Fine Art Pattern Extraction and Recognition"

Guest Editors: Fabio Bellavia, Gennaro Vessio, Giovanna Castellano and Sinem Aslan

Read more

The Institution of Engineering and Technology

Received: 29 June 2020 | Revised: 29 December 2020 | Accepted: 25 January 2021

IET Image Processing

DOI: 10.1049/ipr2.12208

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH

Falling motion detection algorithm based on deep learning

Na Zhu[1] | Guangzhe Zhao[1] | Xiaolong Zhang[2] | Zhexue Jin[3]

[1] College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China

[2] Beijing Academy of Artifivial Intellgence, Zhongguancun South Street, Haidian District, Beijing, China

[3] Department of Computer Science and Technology, Yanbian University, No 977, Gongyuan Road, Yanji, China

Correspondence
Zhexue Jin, Department of Computer Science and Technology, Yanbian University, No 977, Gongyuan Road, Yanji, China.
Email: jinzhexue@ybu.edu.cn

Funding information
National Aerospace Science Foundation of China, Grant/Award Number: 61871021

Abstract

Falling is a significant cause of injuries and even death in the elderly. The timely detection of the fall action helps to rescue people who may have physical health problems due to the fall, so fall detection is necessary. The traditional fall detection methods are mostly based on wearable devices, which need to be worn all the time, and the cost of the device is high. In recent years, the fall detection method based on computer vision has become a research hot spot. This paper proposes a framework for falling motion detection based on deep learning. To quickly and accurately classify human movements, a method using bone key points as the feature descriptors of human movements is proposed. The OpenPose algorithm is used to extract the human skeleton point information as the primary human body feature, and then use the deep learning method to classify further and recognise our action features. In this paper, four types of daily actions, such as falling and walking, are classified and recognised. The results show that the algorithm achieves an accuracy of 99.4% on our dataset. Simultaneously, 86.1% accuracy is reached in the public dataset fall detection dataset.

1 | INTRODUCTION

Unexpected fall movements have a high probability of causing injuries and death to the elderly and patients with poor health. According to a report published by the World Health Organization in 2008 about preventing the elderly from falling [1], it is estimated that 646,000 people fall to death worldwide each year and the largest proportion of people over 65 years old. There were 3730 hospitalisations due to falls each year. In recent years, the aging of the population in various countries has deepened, and these data have increased year by year.

When a fall event occurs, timely and accurate detection of dangerous actions and alarms can reduce panic in the heart of the fallen people and allow patients to get timely help from family members or doctors. Therefore, the detection of falling motion has great practical significance.

Falling motion recognition methods are divided into sensor-based methods, environmental sensor-based methods, and vision-based methods. Sensor-based methods [2–4] use external sensors such as accelerometers, three-axis magnetometers, and gyroscopes to collect human posture and posture changes. Monitor changes in acceleration or Euler angles by setting a threshold to determine whether the wearer is falling. The accuracy and sensitivity of this method generally depend on the type of sensor selected and the location where the device is installed. As a limitation of the external device, the monitoring object is required to wear the device at all times. These sensors will fail if the user forgets to wear them or fails to charge them in time. Environmental sensor-based methods generally use infrared sensors or other sensors to detect the fall of the elderly environment. Feng et al. [5] detected fall events by placing pressure-sensitive fibre optic sensors on the floor. Taramasco et al. [6] installed infrared sensors at the height of 10 cm and 1 m above the ground to monitor the human body heat change at different heights to determine whether a fall event occurred. This method of installing special sensors in an indoor environment is complicated in equipment installation and expensive, difficult to promote.

According to the different types of data collected by cameras and the number of cameras, the fall detection method based on the vision can be divided into a depth camera-based method, a multi-camera method and a single camera method. Depth camera usually refers to the Kinect camera, which can collect depth information and image information simultaneously, and its data

type is known as RGB-D. Adhikari et al. [7] used the background subtraction method to extract human contour information and then used the convolutional neural networks (CNN) convolutional network to perform further feature extraction and classification. Xu et al. [8] used a neural network back propagation (BP) algorithm to classify poses collected by Kinect camera, for posture classification to detect human falls. The multi-camera method collects multi-view image information by setting multiple cameras or panoramic cameras. This method solves the problem of blur and occlusion caused by a single perspective. Manekar et al. [9] used a 360-degree panoramic camera to collect a full range of indoor activity scenes and used 3D convolution or CNN + long short-term memory (LSTM) methods to identify continuously picture actions. Espinosa et al. [10] used the optical flow algorithm to extract motion features for the image information of fixed window size and time collected by the two cameras and then used CNN convolutional network for motion classification and recognition. The method of monocular vision [11] detected falls based on a skeletal image. The image information collected by the ordinary camera is processed. Compared with the high equipment cost of professional depth cameras and the complicated process of installing multiple cameras in an indoor environment, the single cameras fall detection the method based on the motion recognition method has better practical application significance.

Traditional sensor-based fall recognition relies on the recognition sensitivity of the sensor. This method can accurately detect falls, but the sensor is expensive and requires the subject to wear a detection device at all times. The fall detection method based on the Kinect camera and multi-camera can conveniently obtain the three-dimensional information of the human body and effectively solve human body occlusion. However, such cameras are costly and difficult to enter the home of ordinary people. In response to the above problems, we propose a method based on an ordinary camera to detect human falls.

This paper proposes a method of falling action recognition based on deep learning. The key points of human bones are extracted through the body posture estimation algorithm. The information on bone joints can significantly exclude the effects of light changes, background fluctuations and appearance changes and improve the robustness of human motion recognition. We have defined four movements, all of which are general indoor human behaviours, including falls. Through the OpenPose algorithm, we obtained the key point data of the four actions. Besides, deep learning classification training is used to classify human bone data. We compared existing methods based on contour features on public dataset FDD [7], and the results show that our algorithm is effective and feasible.

The contributions of this article are as follows: (i) Our method has better overall accuracy than other algorithms on the basic dataset FDD. (ii) Our method can identify other actions that are harmful to human health in addition to falling actions. (iii) Our method can be directly applied to the existing home camera monitoring system to achieve real-time monitoring through data upload.

2 | RELATED WORK

A complete fall motion recognition system must include three steps: human detection, motion feature extraction, and motion classification. For different types of data, the action features we extract and the classification methods we adopt are also different. At present, the data used in the field of motion recognition research include signal data collected by sensors, RGB+D (DepthMap) image data collected by depth cameras, RGB image data collected by ordinary cameras. Because the first two kinds of data have higher requirements for hardware devices, considering the practical application significance, this article mainly studies RGB image data.

2.1 | Object detection

In the related research of vision-based fall detection, the detection of human targets generally uses traditional manual features to detect human targets. For example, Nguyen et al. [12] used background subtraction between consecutive video frames to extract the silhouette of the human body then used an approximate ellipse and a circumscribed rectangle to calibrate the target object, and finally calculated the deformation of the human silhouette to determine whether a fall has occurred. Xiang et al. [13] proposed a technology based on motion history or energy image (MHOEI), through the combination of motion history image (MHI) algorithm and motion energy image (MEI) to detect and track moving human targets that appear in the panoramic camera. Then, the detected posture changes and threshold settings of the human target are used to detect whether there are abnormal actions such as falling. The human target features used in this type of target detection method are all artificial and low-level, which is not very expressive for a large number of and multiple types of targets.

The target detection algorithm based on deep learning has greatly improved the detection accuracy and detection speed compared with the traditional methods. The detection methods of deep learning are divided into classification-based detection algorithms and regression-based detection algorithms. Classification-based algorithms are R-CNN, Fast-RCNN, R-FCN, and other algorithms. Through these algorithms, continuous optimisation has achieved a high target detection accuracy, but their real-time performance is not high. The regression-based algorithm optimises this problem and improves the real-time performance of the algorithm. The characteristics of the you only look once (Yolo) algorithm [14] are simple network, fast detection speed, but low positioning accuracy. The single shot multibox detection (SSD) algorithm [15] model is simple, and the positioning accuracy is high, but the network is difficult to converge. The Yolov3 [16] algorithm improved the Yolo algorithm, and its detection accuracy is 57.9 AP50, which took 51 ms, which is the best performance target detection method. Deep learning methods can extract higher-level features feature extraction, target selection, and classification in the model. In

this paper, the target detection part of this paper adopts the Yolov3 algorithm.

2.2 | Action feature extraction

The key step of human action recognition is the selection and extraction of human behaviour features. Avgerinakis et al. [17] detected interest points in the motion boundary active area (MBAA) and described them through histogram of gradient (HOG) and histogram of optical flow (HOF) descriptors, and finally, supported vector data description (SVDD) as a classifier for action classification. Iqbal et al. [18] used scale invariant feature descriptor (SIFT) to extract motion edge and boundary features as feature vectors for action description and performed abnormal action matching classification through the hidden Markov model (HMM). Although this method does not rely on positioning and tracking of people. It needs to extract a large number, strong correlation, and stable interest points and requires a complicated preprocessed process.

Global features generally take the human body as a whole for action feature extraction. For example, Adhikari et al. [7] used background subtraction to extract contour features to describe human actions. Espinosa et al. [10] used the optical flow method to extract optical flow information of continuous pictures as human body motion features for motion recognition. Although the method based on contour features and optical flow features has achieved good results, it still has limitations for background clutter, appearance changes, and lighting changes. Johansson et al. [19] proposed that key points of the human body can provide effective information on human motion. Yao et al. [20] used the three-dimensional skeleton data collected by the Kinect camera and used joint vectors to calculate pose features. Since pose features do not contain colour information, there are no restrictions for RGB images. Depth features refer to information that can describe the three-dimensional position of the human body and are generally obtained by depth cameras. Alzahrani et al. [21] used a depth camera to collect the depth images by calculating the human bone points to classify the actions. However, the effective distance of the Kinect camera [22] has limitations. With the research of many gesture recognition algorithms [23], it is possible to obtain human bone points through ordinary cameras. Xu et al. [11] used the OpenPose algorithm to extract bone points for classification and recognition and achieved good recognition results. This article intends to use posture feature bone data as the human motion feature in the feature extraction link.

2.3 | Human movement classification

The most critical step in human action recognition is the choice of action classifier. Nguyen et al. [12, 24] used template matching to classify actions. The disadvantage of this classification method is that it requires a large amount of stable data. The K-neighbour classifier uses the distance classification method, and its principle is to classify the sample as the largest of the K neighbouring samples. This classification method is commonly used in sensor-based fall detection. HMM, as a commonly used state-space model, analyses human behaviour through posture transformation within a period of time. For example, Farzad et al. [25] extracted pose features based on the constructed framework model, trained HMM using pose sequences, and finally used HMM as a classifier for behaviour recognition. The support vector machine (SVM) algorithm is the traditional state statistics method. For example, Kong et al. [26] used HOG to extract human motion features and then used the SVM algorithm to classify images to detect falls.

State statistics methods include the K-means clustering method and the bag of words (BOW) model. For example, Jalal et al. [27] generated skeleton models from contours, calculated motion feature vectors from node motion parameters, used K-means clustering, and HMM model training. Traditional action classifiers require a large amount of feature data for classifier training. The classification method of deep learning generally adopts the deep neural network (DNN) model, a neural network, including an input layer and multiple hidden layers. Yoo et al. [28] used DNN to learn and classify the data collected by the acceleration sensors. For multi-classification problems, the DNN classifier can simulate complex linear problems well. Therefore, the action classification step in this paper will use the DNN network for classification.

3 | FALL DETECTION BASED ON BONE KEY POINTS

We propose a human motion recognition system based on bone key points. The system includes a target detection module, a motion feature extraction module, and a motion classification module. The system uses the Yolov3 algorithm for human target detection, the OpenPose algorithm for human bone coordinate calculation, and the deep learning algorithm for indoor motion classification.

3.1 | Human target detection

The Yolo algorithm uses a deep number of neural networks for target detection classification. It directly divides the original picture into not coincident small squares, then predicts the target whose centre point is in the small square. The Yolov3 algorithm [16] proposed the Darknet-53 network structure and added a residual module to solve the deep network gradient explosion problem. The "conv" layer in this network includes two 1×1 and 3×3 convolutional layers. The Yolov3 network structure is shown (see Figure 1). In the figure, the "DBL" represents the operation of the model added by the "conv", "BN", and "Leaky Relu". The "res nuit" represents the residual operation between two "DBL" operations. The "zero padding" means zero padding operation module. The "resn" module represents the zero padding operation plus "DBL" operation followed by n times 'res unit' operation. The "BN" module represents the data batch normalisation operation. The "up sample" module

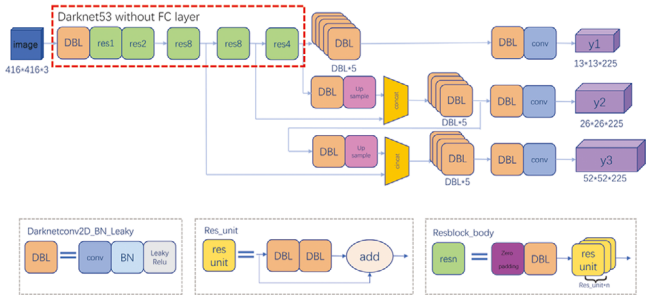

FIGURE 1 Yolov3 network structure

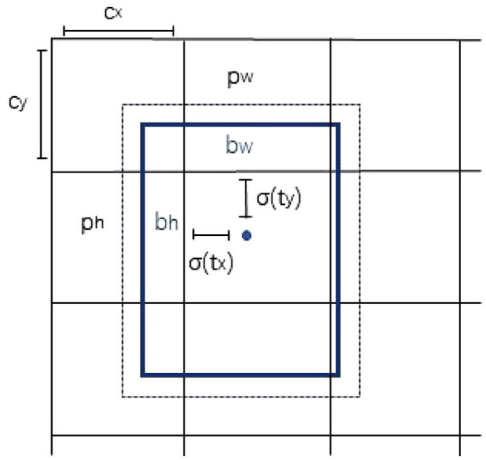

FIGURE 2 The process of determining the prediction bounding box, where the solid box predicts the bounding box and the dashed box is the preset bounding box

represents the up-sample operation on the image. The "Leaky Relu" module represents the activation function using Leaky Relu. The "concat" represents the tensor splicing operation module.

Yolov3 used three different scale feature maps to detect the target and the size of the a priori frame obtained by K-means clustering. Meanwhile, the algorithm used the sigmoid activation function, which differs from the softmax activation function sigmoid activation function, which is more suitable for multi-object classification. The final convolutional layer predicts a three-dimensional tensor code: bounding box, box target, and classification prediction. The predicted target bounding box is calculated by the preset bounding box and centre offset of the predicted bounding box (see Figure 2).

The calculation process from the preset bounding box to the final predicted bounding box as flow:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= P_w e^{t_w}, \\ b_h &= P_h e^{t_h} \end{aligned} \tag{1}$$

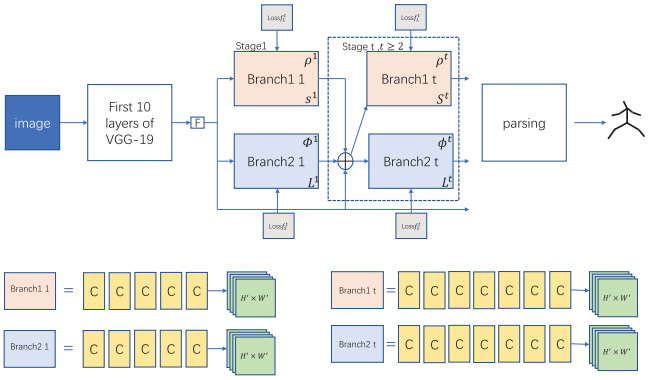

FIGURE 3 OpenPose algorithm framework

The function $\sigma(x)$ is the sigmoid activation function. (c_x, c_y) is the centre coordinate of the preset bounding box on the feature map. P_w and P_h are the width and height of the preset border on the feature map. (t_x, t_y, t_w, t_h) is the prediction bounding box. The final prediction bounding box is (b_x, b_y, b_w, b_h), calculated by the centre offset and the height and width zoom ratio between the preset bounding box and the prediction bounding box.

3.2 | Bone and joint extraction

OpenPose algorithm [8] is a human pose analysis algorithm with high real-time and accuracy proposed by Carnegie Mellon University. The OpenPose algorithm is a bottom-up pose analysis algorithm. The human body key point prediction process is as follows: In the first step, a set of feature maps is obtained after the features are extracted from the first ten layers of the convolution network of Visual Geometry Group19 (VGG19) on the input detection frame image of the previous step. In the second step, input the feature map into the two branches to predict the joint point confidence map (PCM) and the part affinity fields (PAFs) between the two joint points. In the third step, the bipartite graph matching is performed by the Hungarian algorithm to connect the bone joints belonging to the same person into a whole human skeleton. OpenPose provides 18 or 25 human body joints. The algorithm framework is shown (see Figure 3).

In the figure (see Figure 3). The "F" represents the feature map of the image after passing through the convolution network. The "Branch1" and "Branch2" represent the multi-layer convolution operation. The "C" represents the convolution operation. The 'parsing' represents the joint operation of the human body.

The first ten layers of the VGG19 convolution network extract a series of feature maps for the input image of the network. Those feature maps are divided into two branches by the "F" module. The first branch predicts the PCM S^t. The second branch predicts PAFs L^t. After calculating the loss for each stage, connect L, S, and the output of "F", and then send it to the next stage for further extract features. The following is the

calculation formula of the loss:

$$f_S^t = \sum_{j=1}^{J} \sum_{p} W(p) \cdot \left\| S_j^t(p) - S_j^*(p) \right\|_2^2,$$
$$f_L^t = \sum_{i=1}^{c} \sum_{p} W(P) \cdot \left\| L_c^t(p) - L_j^*(p) \right\|_2^2, \quad (2)$$
$$f = \sum_{t=1}^{T} \left(f_s^t + f_l^t \right)$$

Among them:

- $w(p)$ is the weight. When the P position is not marked, $w(p) = 0$ to avoid prediction errors.
- $s_j^t(p)$ refers to the confidence value at point p of the jth part output by Branch1 in the tth network.
- $s_j(p)$ refers to the confidence value of the ground truth jth joint at point p.
- $L_c^t(p)$ is the vector at point p of the jth part output by Branch2 in the tth network.
- $L_c(p)$ is the vector of the ground truth output at the jth part at point t in the tth network.

Confidence maps for part detection:

$$S_{j,k}^*(p) = \exp\left(-\frac{\left\| p - x_{j,k} \right\|_2^2}{\sigma^2} \right),$$
$$S_j^*(p) = \max_k S_{j,k}^*(p) \quad (3)$$

Among them:

- $S_j^*(p)$, which means that the jth joint point of the kth person exists at point o as the confidence level.

Part affinity fields for part association:

$$L_{c,k}^*(p) = \begin{cases} v & \text{if } p \text{ on limb } c, k \\ o & \text{otherwise} \end{cases},$$
$$v = \left(x_{j2,k} - x_{j1,k} \right) / \left\| x_{j2,k} - x_{j1,k} \right\|_2 \quad (4)$$

If v satisfies the following conditions, it is judged that the pixel p is above the limb c.

$$0 \le v \cdot \left(p - x_{j1,k} \right) \le l_{c,k} \text{ and } \left| v_\perp \cdot \left(p - x_{j1,k} \right) \right| \le \sigma_l,$$
$$L_c^*(p) = \frac{1}{n_c(p)} \sum_k L_{c,k}^*(p) \quad (5)$$

Among them:

- $L_{c,k}^*(p)$ represents the vector of the cth limb of the kth person at the point $x_{j1,k}$ and $x_{j2,k}$ are the actual coordinates of the joints j1 and j2.
- $l_{c,k}$ is the length of limb $\| x_{j2,k} - x_{j1,k} \|_2$.
- σ_l is the width of the limb, and $v_\perp$ is the unit vector of vertical v.

Joint splicing, by calculating the linear integral of PAFs to express the correlation of bone points:

$$E = \int_{u=0}^{u=1} L_c(p(u)) \cdot \frac{d_{j2} - d_{j1}}{\left\| d_{j2} - d_{j1} \right\|_2} \mathrm{d}u,$$
$$p(u) = (1-u) d_{j1} + u d_{j2} \quad (6)$$

In the above formula:

- E refers to the PCM calculated by L_c. It is the line segment weight between the predicted points d_{j1} with d_{j2}. The value range of $p(u)$ is d_{j1} to d_{j2}.

The final step is to use PAFs for multi-body analysis. Using non-maximum suppression (NMS) on the confidence map, the candidate bodies joint set is obtained. The relationship between candidate points is obtained by connecting the confidence E for each limb, and finally, each person posture is obtained.

$$D_J = \left\{ d_j^m : \text{ for } j \in \{1 \cdots J\}, m \varepsilon \left\{ 1 \cdots N_j \right\} \right\} \quad (7)$$

$$Z = \left\{ z_{j1j2}^{mn} : \text{ for } j1, j2 \varepsilon \{1 \cdots J\}, m \varepsilon \left\{ 1 \cdots N_{j1} \right\}, n \in \left\{ 1 \cdots N_{j2} \right\} \right\} \quad (8)$$

In the formula (7) and (8):

- d_j^m represents the position of the mth candidate key point of the j-th body part and N_j represents the number of candidate points for the jth key point.
- Z refers to the relationship set between all prediction points of body parts j1 and j2. The size of z_{j1j2}^{mn} is between 0 and 1.

Calculate the highest affinity value:

$$\max_{z_c} E_c = \max_{z_c} \sum_{m \in D_{j1}} \sum_{n \in D_{j2}} E_{mn} \cdot z_{j1j2}^{mn},$$

Among them：

$$\forall m \in D_{j1}, \sum_{n \in D_{j2}} z_{j1j2}^{mn} \le 1,$$
$$\forall n \in D_{j2}, \sum_{m \in D_{j1}} z_{j1j2}^{mn} \le 1 \quad (9)$$

In the formula:

- $\max_{z_c} E_c$ is the total affinity value, E_{mn} represents the affinity between d_{j1}^m and d_{j1}^m.

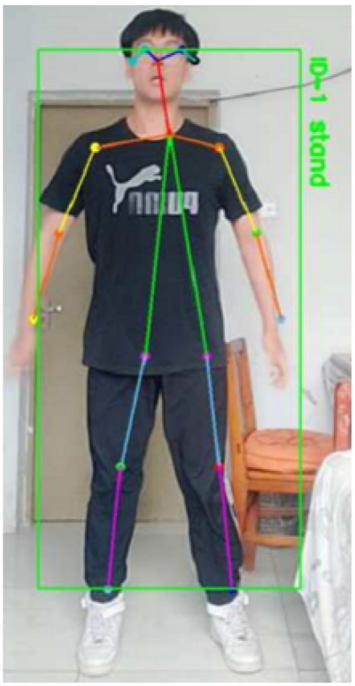

FIGURE 4 Bone diagram extracted

The purpose of this article is full-body motion recognition, so eighteen human key point coordinates are selected as input to the motion classification algorithm (see Figure 4). Including 18 joints are the nose, left shoulder, right shoulder, right hand, left shoulder, left elbow, left hand, right hip, right knee, right foot, left hip, left knee, left foot, right eye, left eye, right ear, left ear.

3.3 | Deep learning classifier

In selecting the action feature classifier, we chose the DNN network to extract further and classify the bone data we extracted. The DNN network structure is generally composed of an input layer, multiple hidden layers, and an output layer. These layers are connected by adjacent neuron nodes. Each layer can be regarded as a logistical regression model. In the training process of the neural network model, first set the initial parameters randomly, calculate the network output parameters to obtain the deviation of the predicted value and the actual value. Then readjust the network parameters according to these deviations. Finally, optimise the model parameters until the model deviation between the predicted value and the actual value gradually decreases to stabilisation.

Our DNN network (see Figure 5) is composed of four fully connected layers. The input layer includes 128 neurons. The second and third layers are hidden layers, including 64 and 16 neurons, respectively. In deep neural network networks, the Relu linear activation function is generally used to prevent the gradient disappearance problem. Therefore, the activation functions of our first three layers all choose the Relu function. The last layer has four neurons in the output layer and uses the suitable softmax activation function for classification. For

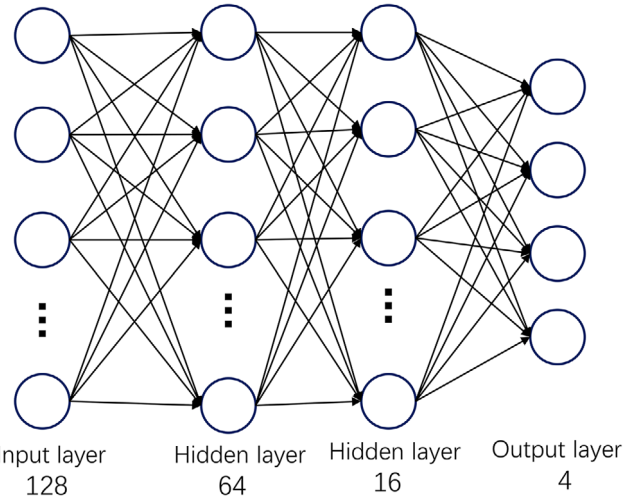

FIGURE 5 DNN network structure

the gradient optimiser, we choose the Adam method with an adaptive learning rate.

3.4 | Our fall detection system

In our fall detection system, we pre-trained the Yolov3 algorithm on the coco dataset. The purpose of this is that the algorithms trained on the large dataset are universal, and the use of pre-trained models also greatly reduces our computational complexity.

The input of the fall detection algorithm is an RGB image, and the output is the action type or falls action warning. First, load the pre-trained model for file configuration and initialization. For each image, we take the following actions:

Step 1: Adjust the input image size and change it to 416 × 416 as the input of the Yolov3 algorithm.
Step 2: Perform image extraction on the detection frame of the human body category detected by Yolov3.
Step 3: Adjust the size of the human body image we extracted to 368 × 368, which is used as the human skeleton point extraction steps input.
Step 4: Classify the bone point data extracted by the OpenPose algorithm.

4 | EXPERIMENT AND DISCUSSION

Our model training and testing process were conducted on a computer with an NVIDIA GeForce GTX960m graphics card. For software, we design our algorithm based on python3.6 language and use Tensorflow and Keras as deep learning toolkit.

4.1 | Dataset

We have built a small database of 3289 posture movements containing four different movements. These actions include

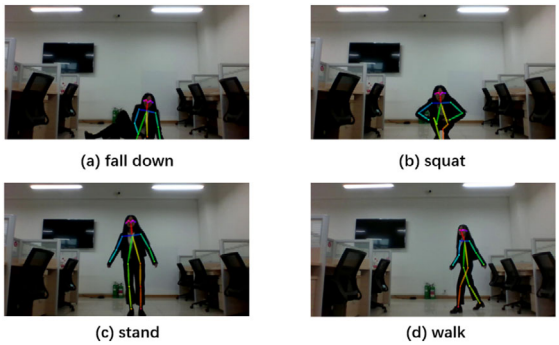

FIGURE 6 A sample of our dataset

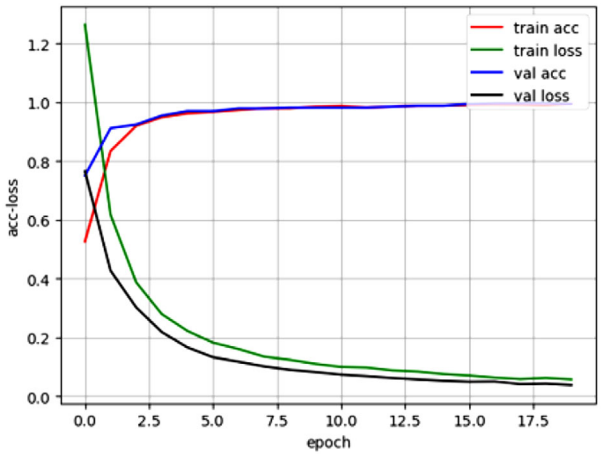

FIGURE 7 DNN network training learning curve

walking, squatting, falling, and standing (see Figure 6). Each activity collected 1078 standing pictures, 387 walking action pictures, 815 squat pictures, and 1008 falling action pictures for training and testing our classification algorithm.

Before training the deep learning classifier, we preprocessed the dataset images. At the same time, we pre-trained Yolov3 on the coco dataset. We use the OpenPose algorithm with high accuracy to extract the skeleton data in the figure and label these skeleton data with the same label as the original picture. In the end, we get a set of digital sequences, including joint coordinates and action classes. Afterward, we use these data containing human posture information in the process of classifier training and optimisation. In the experiment, we use the cross-validation method. We randomly select 10% of the overall sample as the test set and the remaining 90% as the training set. In optimising the classifier parameters with gradient descent, we used a learning rate of 0.0001.

4.2 | Action classification results

We record the learning situation of our action recognition classifier training and testing (see Figure 7).

In the process of model training and learning, the "train acc" curve and "val acc" curve rise smoothly and tend to stabilize

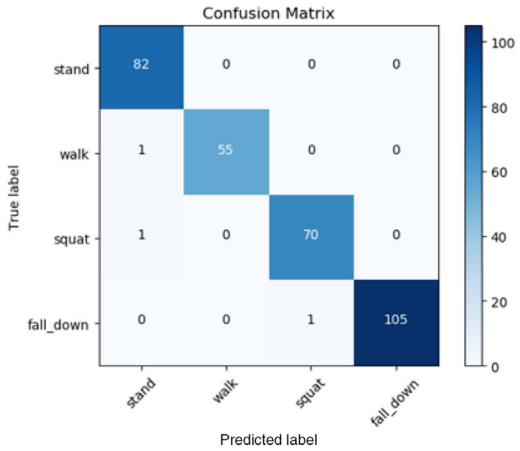

FIGURE 8 Test confusion matrix

TABLE 1 Compare with other methods on FDD dataset [7]

	Accuracy (%)	Sensitivity of fall (%)	Specificity of fall (%)
Adhikar [7]	74	99	91
Wafa [29]	81	-	-
ours	86	86	95

after 7.5 batches. It can be seen that our model has reached high training accuracy. From the comparison of the "train loss" and "val loss" curve changes, the "train loss" curve and the "val loss" curve decrease smoothly during the training process and tended to converge after 15 batches, indicating that our network does not have overfitting or underfitting (see Figure 7).

We analyse the classification effect of our model through the confusion matrix collected during the training (see Figure 8). The blue elements on the diagonal indicate the number of correctly classified gestures. Other elements represent confusion due to model classification. Sensitivity is the ratio of identified negative examples to all negative examples and measures the classifiers ability to recognise positive examples. The accuracy rate is the proportion of the accurately predicted samples in all samples and is an essential indicator for evaluating the quality of the classifier. We can know that the "fall" class is correctly classified 105 times by analysing the matrix results. Only one time is mistakenly classified as "squat". The sensitivity of the "walk" category is 98.2%. The sensitivity of the "stand" category is 99%, and the sensitivity of "squat" is 98.5%. The sensitivity of the "fall" action classification of this model is as high as 99.05%, and the average accuracy rate of our method in the dataset is 99.04%, indicating that our method can effectively and accurately detect and recognise falling actions.

In order to further evaluate our proposed algorithm, on the standard dataset FDD, we compare our fall recognition algorithm with methods [7] and [29].

The experimental results are shown in Table 1. The overall accuracy of our method is significantly higher than [7] and [29], which shows that our method can effectively achieve falls and other daily motion recognition. The method [29] improved

the accuracy of action classification by using the CNN network to extract action features and pre-training the action classifier, then feature sharing. Compared with the traditional CNN classification method, our method of further classification training of bone point features has achieved better classification results. Nevertheless, the performance of the fall action sensitivity and specificity is poor because our method only classifies and recognises the RGB image data. For the problems of occlusion or light change, [7] extracts the image depth information to constructing the human body contour. The method of extracting the key points of the pose that we adopt needs to be further optimised. In the future, our work will focus on how to overcome the effect of occlusion on the efficiency of human bone point extraction.

5 | CONCLUSION AND OUTLOOK

This paper proposes a fall detection algorithm based on deep learning. Unlike the traditional dependence on the depth camera, we extract human skeleton information through a pose estimation algorithm to classify and learn it as human motion features. This article defines four human movements: squatting, walking, falling, and standing. These actions are natural and simple. Human bone information is data that can directly represent human actions. We conduct deep learning classification and training. We conduct experiments on the self-collected indoor basic motion skeleton dataset. The experimental results show that the average accuracy of our method is 99.04%, verifying that the deep learning method has achieved an excellent recognition effect on our dataset. At present, more and more people are beginning to install cameras in the family public environment to view the activities of the elderly at home or as home security. The fall detection framework proposed in this paper can be applied to a home monitoring system to monitor and warn the elderly health activities in real-time by uploading video data.

The continuous development of computer vision technology has made the smart home medical security monitoring system less restrictive to hardware. People obtain more and more human body motion information through cameras. The generation of human body posture analysis algorithm has brought new inspiration for human body motion recognition research. In the past, people needed depth cameras or sensors to obtain human bone information, but now we can easily obtain human bone information by relying on simple cameras and pose estimation algorithms. The human motion recognition analysis method based on 3D bone information and video information will bring new research ideas for the subsequent research on recognising dangerous motions such as falls based on ordinary cameras.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61871021 and the Pyramid Talent Training Project of BUCEA.

ORCID

Guangzhe Zhao https://orcid.org/0000-0002-6850-9335

REFERENCES

1. World Health Organization.: WHO global report on falls prevention in older age. 1st edn. Geneva: World Health Organization (2008)
2. Mao, A., et al.: Highly portable, sensor-based system for human fall monitoring. Sensors 17, 2096–2111 (2017)
3. Sucerquia, A., López, J.D., Vargas-Bonilla, J.F.: Real-life/real-time elderly fall detection with a triaxial accelerometer. Sensors 18, 1101–1119 (2018)
4. Kumar, V.S., et al.: Wearable sensor-based human fall detection wireless system. Wireless Communication Networks and Internet of Things. pp. 217–234. Springer, Berlin (2019)
5. Feng, G., et al.: Floor pressure imaging for fall detection with fiber-optic sensors. IEEE Pervasive Comput. 15, 40–47 (2016)
6. Taramasco, C., et al.: A novel monitoring system for fall detection in older people. IEEE Access 6, 43563–43574 (2018)
7. Adhikari, K., Bouchachia, H., Nait-Charif, H.: Activity recognition for indoor fall detection using convolutional neural network. In: 2017 Fifteenth IAPR International Conference on Machine Vision Applications, pp. 81–84. IEEE, Piscataway, NJ (2017)
8. Xu, Y., et al.: Human posture recognition and fall detection using kinect v2 camera. In: 2019 Chinese Control Conference, pp. 8488–8493. IEEE, Piscataway, NJ (2019)
9. Manekar, R., et al.: Activity recognition for indoor fall detection in 360-degree videos using deep learning techniques. In: Proceedings of 3rd International Conference on Computer Vision and Image Processing, pp. 417–429. Springer, Berlin (2020)
10. Espinosa, R., et al.: Application of convolutional neural networks for fall detection using multiple cameras. Challenges and Trends in Multimodal Fall Detection for Healthcare, pp. 97–120. Springer, Berlin (2020)
11. Xu, Q., et al.: Fall prediction based on key points of human bones. Physica A 540, 123205–123218 (2020)
12. Nguyen, V.D., et al.: An efficient camera-based surveillance for fall detection of elderly people. In: 2014 9th IEEE Conference on Industrial Electronics and Applications, pp. 994–997. IEEE, Piscataway, NJ (2014)
13. Xiang, Y., et al.: Remote safety monitoring for elderly persons based on omni-vision analysis. PloS One 10, 1–16 (2015)
14. Redmon, J., et al.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788. IEEE, Piscataway, NJ (2016)
15. Liu, W., et al.: Ssd: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer, Berlin (2016)
16. Redmon, J., Farhadi, A.: YOLOV3: An incremental improvement. arXiv:180402767 (2018)
17. Avgerinakis, K., Briassouli, A., Kompatsiaris, Y.: Activity detection using sequential statistical boundary detection (SSBD). Comput. Vision Image Understanding 144, 46–61 (2016)
18. Iqbal, J.M., Lavanya, J., Arun, S.: Abnormal human activity recognition using scale invariant feature transform. International Journal of Current Engineering and Technology 5, 3748–3751 (2015)
19. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception & Psychophysics 14, 201–211 (1973)
20. Yao, B., et al.: A big bang–big crunch type2 fuzzy logic system for machine-vision-based event detection and summarization in real-world ambient-assisted living. IEEE Trans. Fuzzy Syst. 24, 1307–1319 (2016)
21. Alzahrani, M.S., et al.: Comprehensive evaluation of skeleton features-based fall detection from microsoft kinect v2. Signal, Image and Video Processing 13, 1431–1439 (2019)
22. Zhang, Z.: Microsoft kinect sensor and its effect. IEEE Multimedia 19, 4–10 (2012)
23. Cao, Z., et al.: Openpose: realtime multi-person 2d pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. 43, 172–186 (2019)
24. Vishwakarma, D., et al.: Recognition of abnormal human activity using the changes in orientation of silhouette in key frames. In: 2015 2nd International Conference on Computing for Sustainable Global Development, pp. 336–341. IEEE, Piscataway, NJ (2015)
25. Farzad, A., Asli, R.N.: Recognition and classification of human behavior in intelligent surveillance systems using hidden markov model. IJ Image, Graphics and Signal Processing 12, 31–38 (2015)

26. Kong, X., et al.: A hog-svm based fall detection IOT system for elderly persons using deep sensor. Procedia Comput. Sci. 147, 276–282 (2019)
27. Jalal, A., Kamal, S.: Real-time life logging via a depth silhouette-based human activity recognition system for smart home services. In: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 74–80. IEEE, Piscataway, NJ (2014)
28. Yoo, S., Oh, D.: An artificial neural network–based fall detection. Int. J. Eng. Bus. Manage. 10, 1847979018787905–1847979018787912 (2018)
29. Abedi, W.M.S., Ibraheem-Nadher, D., Sadiq, A.T.: Modified deep learning method for body postures recognition. International Journal of Advanced Science and Technology 29, 3830–3841 (2020)

How to cite this article: Zhu, N., Zhao, G., Zhang, X., Jin, Z.: Falling motion detection algorithm based on deep learning. IET Image Process. 16, 2845–2853 (2022). https://doi.org/10.1049/ipr2.12208