IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 19, 2020, accepted May 28, 2020, date of publication June 2, 2020, date of current version June 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999503

Fall Detection Based on Dual-Channel Feature Integration

BO-HUA WANG, JIE YU, KUO WANG, XUAN-YU BAO, AND KE-MING MAO
Faculty of Software, Northeastern University, Shenyang 110819, China

Corresponding author: Ke-Ming Mao (maokm@swc.neu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant N2017007, in part by the National Training Program of Innovation and Entrepreneurship for Undergraduates under Grant 201910145221, and in part by the Fundamental Research Funds for the Central Universities under Grant N182410001.

ABSTRACT Falls have caught great harm to the elderly living alone at home. This paper presents a novel visual-based fall detection approach by Dual-Channel Feature Integration. The proposed approach divides the fall event into two parts: falling-state and fallen-state, which describes the fall events from dynamic and static perspectives. Firstly, the object detection model (Yolo) and the human posture detection model (OpenPose) are used for preprocessing to obtain key points and the position information of a human body. Then, a dual-channel sliding window model is designed to extract the dynamic features of the human body (centroid speed, upper limb velocity) and static features (human external ellipse). After that, MLP (Multilayer Perceptron) and Random Forest are applied to classify the dynamic and static feature data separately. Finally, the classification results are combined for fall detection. Experimental results show that the proposed approach achieves an accuracy of 97.33% and 96.91% when tested with UR Fall Detection Dataset and Le2i Fall Detection Dataset.

INDEX TERMS Computer vision, fall detection, dual channel, machine learning.

I. INTRODUCTION

With the development of society and the improvement of medical standards, the aging of the population has become a global trend. The World Health Organization released a report in 2015, stating that the number of people over the age of 60 is expected to double by 2050 [1]. As age increases, the probability of accidental fall events will increase accordingly [2]. It has been pointed out that the incidence of falls in the elderly over 65 is 28% to 35%, and the possibility of falling over the age of 70 is increased to 32% to 42% [3]. These falls mainly caused 90% of hip and wrist fractures and 60% of head injuries[4]. In addition to these injuries, another consequence of the fall is long-term lying (i.e., staying on the ground for a long time), with the consequences of dehydration, hypothermia, and even death. Therefore, how to quickly and effectively detect falls has excellent social benefits. It can not only enable the elderly to get attention and treatment promptly but also reduce the potential harm caused by falls considerably.

Many studies have been undertaken in the field of fall detection. The existing technologies can be classified into mainly three types: ambient device-based method [5], [6], wearable sensor-based method [7]–[12] and computer vision-based method [13]–[24].

An **Ambient device-based** method detects the fall event through the sound, vibration, and other signals collected by sensors installed in the room (walls, floors, etc.). Alwan *et al.* [5] proposed the working principle and designed a fall detector based on floor vibration. Similarly, Zigel *et al.* [6] proposed a detection approach based on floor vibration and sound sense. These approaches employed signal processing and pattern recognition algorithms to distinguish between fall events and other events. A **Wearable sensor-based** method mainly uses various sensors to detect human movement speed and sudden changes in gait to evaluate the balanced state of the human body. Lee and Tseng [7] proposed an Enhanced Threshold-Based Fall Detection System Using Smartphones With Built-In Accelerometers. Montanini *et al.* [8] proposed a detection approach based on footwear. They installed a force sensor and a three-axis acceleration sensor in shoes to analyze the fall event. Hassan *et al.* [9] proposed a MEFD (mobile-enabled

The associate editor coordinating the review of this manuscript and approving it for publication was Hossein Rahmani.

fall detection) framework. In this framework, real-time data retrieved from an accelerometer sensor on a smartphone are processed and analyzed by an online system running on the smartphone itself. Xi *et al.* [10] proposed a detection approach based on Surface Electromyography and Plantar Pressure. Gumaei *et al.* [11] proposed an effective multi-sensors-based framework for human activity recognition using a hybrid deep learning model that combines simple and gated recurrent neural network units. Kerdjidj *et al.* [12] used wearable sensors and compressed sensing to detect fall events. A **Computer vision-based** method generally detects fall events by analyzing the video or image. According to the number and type of camera being used, it can be further divided into ways of single camera, multiple cameras [13], Omni-camera [14], and depth camera [15]. Merrouche and Baha [16] analyzed the shape of the human body by using a depth camera. So the head can be tracked and body centroid can be detected. Lotfi *et al.* [23] used elliptical models and projection histograms for feature extraction. Then, a MLP network is used to classify the features.

It can be found that the ambient device-based method may sense the pressure of other objects around the target and generate false alarms so that the detection accuracy will be affected [24]. In terms of cost, ambient devices are also expensive. For the wearable sensor-based method, although it has high accuracy, these devices need to be carried around, which adds uncertainty to the fall detection. When people forget to take the equipment, the method will be invalid. In contrast, the video-based method reduces this risk and does not interfere with the lives of the elderly. With the development of computing power of edge devices, some current application scenarios involving video information analysis will preprocess video data locally. After removing the data related to privacy, the data will be uploaded to the server, which can solve the problem of privacy leakage caused by video data. In terms of cost, although video-based fall detection requires additional equipment (cameras, embedded devices, etc.), this is tolerable compared to its advantages. With the development of technology, the price of equipment will also be gradually decreased. Therefore, the video-based fall detection method has attracted more and more attention due to its advantages of non-invasiveness and high accuracy.

Most existing video-based fall detection methods treat the input video as a series of frames, and fall event is defined as a sequence of behavior. Then features like the outline of the person, the relative position of the specific part, speed, etc. are extracted. These features are combined to train the classifier. However, the disadvantage of these methods is the low specificity. In this paper, we propose a new model for fall detection in video. We found that different features express different aspects of fall. For example, velocity and acceleration are more suitable for detecting the process of falling, while the body's contours and limbs' positions are more suitable for expressing the state after fallen. In this research, falling and fallen states are defined as two channels to model the fall event. The objection detection model Yolo and posture detection model OpenPose are adopted for video preprocessing. Then, features related to two channels are extracted within a sliding window and two states are recognized. Finally, the fall event is detected with the combined result. The main novelties of our method are as follows:

- Falling-state and fallen-state are defined to describe the fall events from dynamic and static perspectives
- Dual-channel sliding window model (DSW) is proposed to extract dynamic and static features.
- Extensive experiments are evaluated for testing and analyzing the model performance.

The rest of the paper is organized as follows: a brief review of related works is presented in Section II. Section III gives an overview of the proposed method. Detailed implementation is described in Section IV, including Preprocessing, Fall event modeling, State classification, and Fall event detection. Experimental evaluations are discussed in Section V. Section VI concludes the paper and suggests opportunities for future work.

II. RELATED WORK

Existing fall detection methods based on computer vision can be divided into two categories: traditional features based and depth features based.

A. TRADITIONAL FEATURES BASED

Debard *et al.* [17] extracted four features from the bounding box around the human silhouette to describe a fall. These features were aspect ratio, fall angle, centre speed, and head speed. Then, SVM (Support Vector Machine) was used as the classifier. However, the major drawback is the inadequate description of human motion by merely using a bounding box. To overcome this problem, Gunale and Mukherji [18] utilized an ellipse to fit the physical characteristics of a person and used KNN (K-NearestNeighbor) for classification. The accuracy of this approach was 95%. The main problem of this approach is that the ellipse can only describe the external features of the human body while some local features are neglected. Yu *et al.* [19] combined the elliptical model with the projection histogram along the elliptical axis to complete the extraction of the target specific pose. They achieved a high fall detection rate of 97.08%. Similarly, Yun *et al.* [20] detected fall events based on measuring a temporal variation of pose change and body motion. Features of centroid velocity, head-to-centroid distance, the histogram of oriented gradients, and optical flow were computed. It got an accuracy of 90.6%. Lotfi *et al.* [23] used the background subtraction to extract the moving human target and then extracted the external contour, ellipse, centroid, and other characteristics of the human body. Finally, these features were fed into a MLP for fall detection. The test results show a high sensitivity of 99.52% and a high specificity of 97.38% on the UR Fall Detection Dataset[30].

B. DEPTH FEATURES BASED

With the development of deep learning in recent years, many human posture detection models based on Convolutional

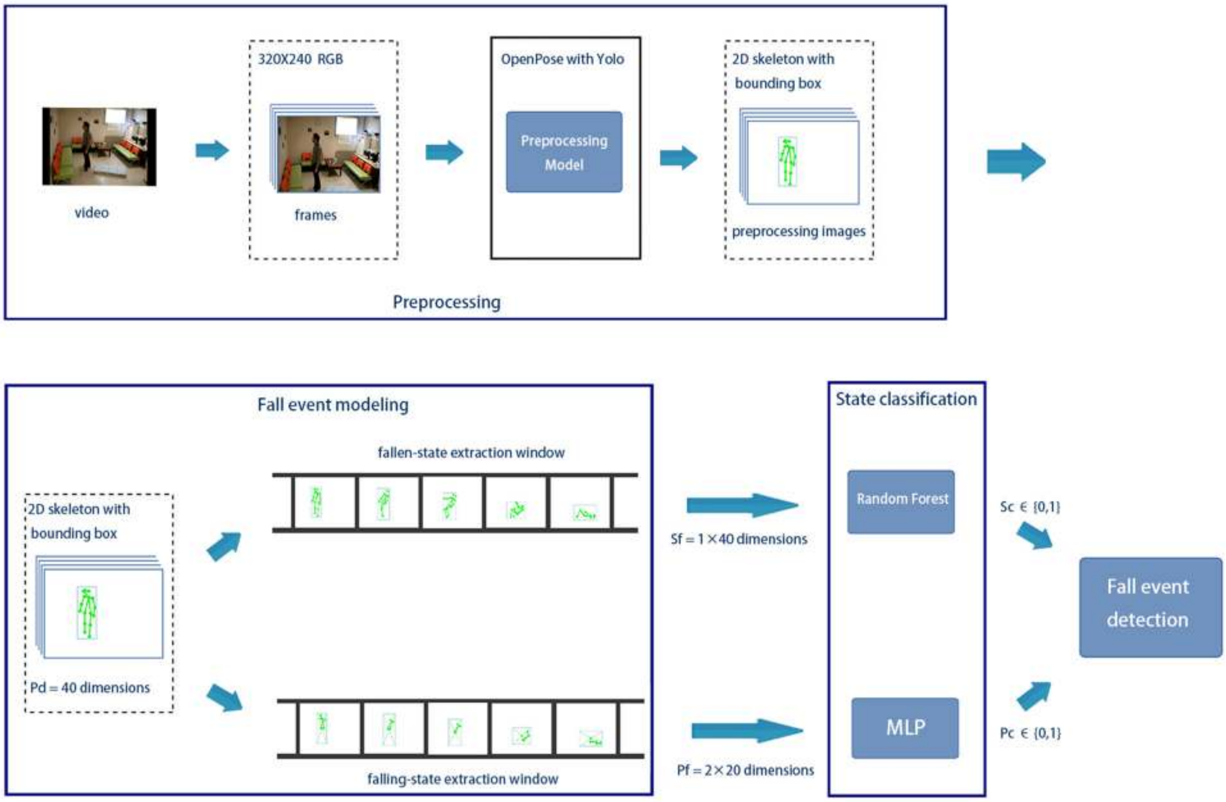


FIGURE 1. The flow diagram of proposed fall detection method.

neural networks (CNN) have been proposed. These models can extract the key part positions of the human body, and it provides a new approach for fall detection. Lie *et al.* [21] adopted DeeperCut (a 152-layer Residual neural network) to extract the skeleton coordinate from the 2D image. The skeleton information composed of 14 key points, which can depict the outline of the central part of the human body. Then a recurrent neural network (RNN) with long and short term memory (LSTM) state units were used to process the keypoint sequence. This method was suitable for the classification of video sequences and finally achieved an accuracy of 88.9%. Huang *et al.* [22] used the OpenPose model to get 18 key points of the human body skeleton. Then a VGG-16 network was used for feature extraction and representation. There was no specific accuracy of this method on public datasets.

The proposed approach in this paper combines the feature extraction schemes of the above two methods, which attempts to overcome the existing shortcomings.

III. SYSTEM OVERVIEW

This paper proposes a method for fall event detection in video. The process of the framework is illustrated in Figure 1. Our

fall detection method includes four main steps: **Preprocessing**, **Fall event modeling**, **State classification**, and **Fall event detection**. First, OpenPose and Yolo are used for preprocessing. The human body is subtracted from the background and then tracked in successive frames. Meanwhile, the position of the human body is also extracted. Second, the rate of centroid drop, the speed of the upper limbs, the location of key points, and the ellipse parameters of the human body are computed. These features are further divided into types of dynamic and static, which are used for describing the human body during the fall event. Two types of features are recognized in a dual-channel sliding window of successive frames of video. Third, two classifiers are used to classify the two types of features independently. At last, the final fall event detection result is judged by merging the above two classifiers.

Fall event in the video is complex, and it can be confused with normal activities for their high similarity in some cases. Experimental results show that features of different frame sequences do not have equal importance. For solving this problem, a novel model is introduced in our method. Features are divided into types of dynamic and static, which are used for describing the falling-state and fallen-state of the human body. The human body keypoint position extracted with OpenPose is used to construct static features. At the same time, the rate of central drop and the speed of the upper limbs are calculated to construct dynamic features. Two types of features are extracted independently within a dual-channel sliding window set in the video. Then MLP and RandomForest are used to classify falling-state and fallen-state. Finally, the final result is judged by the combination of two classifiers.

IV. SYSTEM IMPLEMENTATION

Step 1. Preprocessing: For representing the fall event effectively, some raw features should be computed. The position, outline, and key points of the human body are calculated from input video in the preprocessing step. DeepSort_Yolo and OpenPose models are adopted in this work. DeepSort_Yolo is a method that combines multi-target tracking (DeepSort[27]) with target detection (Yolo[28]). It can detect and track the human body target in consecutive frames of video. OpenPose is a keypoint detection model built by CMU's Perceptual Computing Lab[25], whose purpose is to detect 2D and 3D keypoint coordinates of the human body. In our preprocessing model, OpenPose and DeepSort_Yolo are used to achieve detection, tracking, key points extraction of the human body target in continuous frames. For each frame of an input video, the rectangular area of human target is forecasted by DeepSort_Yolo, and the coordinates of human body key points are calculated by OpenPose (as shown in ***Figure 2***, including ten parts of the human body: nose, eye, ear, neck, shoulder, elbow, wrist, hip, knee, ankle). The raw features of each frame are finally expressed as vector $Pd = [(x_0, y_0), \ldots, (x_{17}, y_{17}), (x_{ul}, y_{ul}), (x_{lr}, y_{lr})]$.

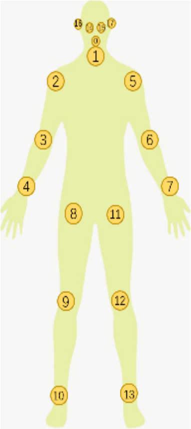


FIGURE 2. Position of human body key points.

The coordinate (x_i, y_i) corresponds to the ith human body keypoint, (x_{ul}, y_{ul}) and (x_{lr}, y_{lr}) represent the coordinates of the upper left corner and the lower right corner of human body target area respectively.

Step 2. Fall Event Modeling: In our model, the fall event in the video is defined as two states: **falling-state** and **fallen-state.** The **falling-state** describes the falling process of the human body, and the **fallen-state** describes the appearance of the human body after he lied down.

Then, feature selection is an essential technique[26], there are two sets of features selected to represent falling-state and fallen-state respectively.

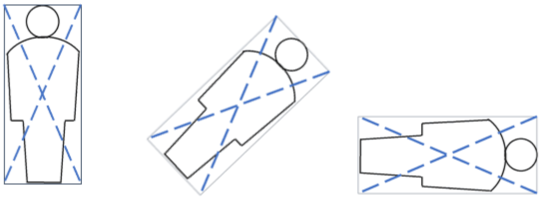


FIGURE 3. The position change of the centre of mass in the process of falling.

A. FALLING-STATE REPRESENTATION

1) CENTROID DROP RATE

As shown in **Figure 3**, human body centroid is formatted as the diagonal intersection of the minimum enclosing rectangle of a human body. The upper left corner coordinate of the rectangle is defined as (x_{ul}, y_{ul}), and the lower right corner coordinate is defined as (x_{lr}, y_{lr}). The coordinates of the centroid point $(Cent\ (x)\ , Cent\ (y))$ are calculated as follows:

$$Cent\ (x) = \frac{x_{ul} + x_{lr}}{2} \tag{1}$$

$$Cent\ (y) = \frac{y_{ul} + y_{lr}}{2} \tag{2}$$

Let $Cent\ (y_i)$ represent the vertical coordinate of the human centroid point in frame i, and Δh represents the height of the rectangle region outside the human body. The centroid drop rate (V_c) describes the average moving speed of $Cent\ (y_i)$ between consecutive frames. Then in Δf frames, the formula of V_c is as follows:

$$V_c = \frac{\left[Cent\ \left(y_{i+\Delta f}\right) - Cent\ (y_i)\right]}{\Delta f * \Delta h} \tag{3}$$

According to our fundamental knowledge of the falling-state, there is a downward process when people are falling. Since people are not controlled by themselves during the fall, the average centroid drop rate will be faster than the controlled squat, sitting down, bending, and other normal movements. As shown in *Figure 4*, the average centroid drop rate of five common behaviours (walking, sitting, squatting, falling, lying) are calculated. According to *Figure 4,* when people are falling, the curve is different from sitting, lying, and walking. However, it also can be found that the curve of the squat is partially similar to the falling-state. Therefore, relying solely on this feature cannot accurately classify the fall event and other behaviours. In the next section, the upper limb velocity will be proposed to solve this problem.

2) UPPER LIMB VELOCITY

For representing the fall event more accurately, new features need to be defined. We find that the body generally moves laterally at a faster speed during the falling process. In the process of falling, the human body is moving like accelerated circular motion with the feet as the centre. Meanwhile, the movement speed of the upper limb in the x-axis direction is faster than that in other directions, and the movement range is also more massive, which leads to drastic variation in

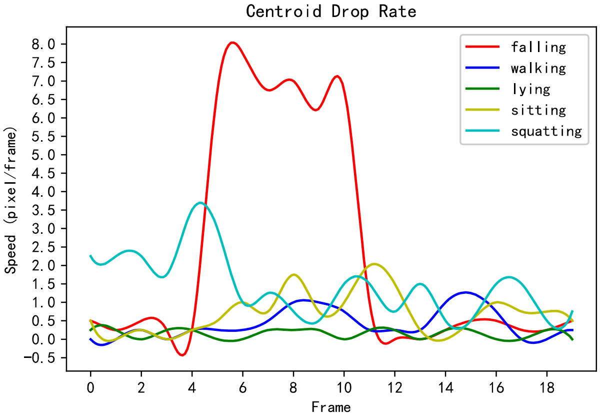


FIGURE 4. Centroid drop rate of five behaviors. The x axis represents video frame and the y axis represents the drop speed of centroid.

feature space. Accordingly, the upper limb velocity can be adopted as a new feature. To ensure stability, six key points, L_Eye, R_Eye, Nose, Nack, L_Shoulder, and R_Shoulder, are extracted as references. The average value of the movement velocity of six points in the x-axis direction between two sampling frames is used to compute upper limb velocity. Let the horizontal coordinates of the six points of Nack, L_ Shoulder, R_Shoulder, L_Eye, R_Eye, and Nose on the j-th frame image be x(0,j)~x(5,j), respectively, and the upper limb velocity (V_l) is calculated as follows:

$$V_l = \sum_{i=0}^{5} \frac{\left(\frac{|x(i,j+\Delta f)-x(i,j)|}{\Delta f*\Delta w}\right)}{6} \tag{4}$$

where Δw is the width of the rectangular region outside the human body, and Δf is the number of frames between two sampling frames.

Figure 5 (a), **(b),** and **(c)** respectively show the upper limb velocity between three different fall postures and four common behaviours. According to the figure, when a fall occurs, the upper limb of a person generally has a relatively apparent lateral movement, which does not exist when squatting down. So this feature can distinguish the fall action and the squat action to a certain extent.

B. FALLEN-STATE REPRESENTATION

1) HUMAN BODY KEYPOINT

OpenPose is a realtime multi-person 2D pose estimation model, which aims to extract the key points and posture of multiple people from the image. In this method, a dual-branch neural network is used for forecasting the confidence maps of body part locations and the part affinities fields concurrently[25], then, a greedy inference is used to match the key points of the human body.

Based on the definition of OpenPose, **Figure 6** shows that 18 2D coordinates are taken advantage of representing the position of 18 key points of the human body, which include ten parts of the human body: nose, eye, ear, neck, shoulder, elbow, wrist, hip, knee, ankle. The key points will be expressed as follows: Kp =[(x_i, y_i)], where i = 0,1, ...,17.

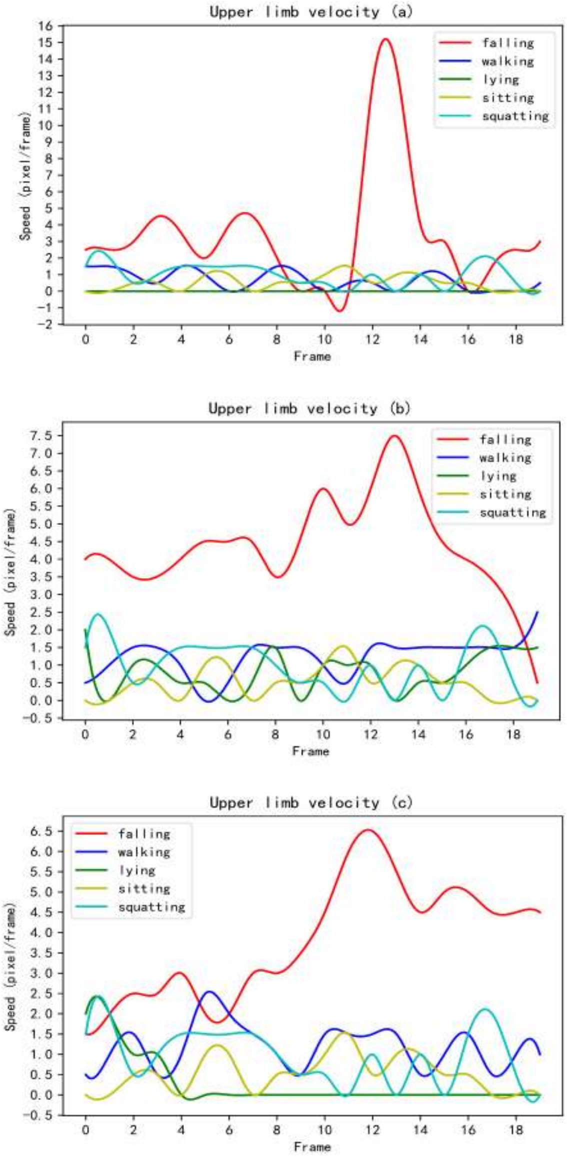


FIGURE 5. The velocity of upper limb in five *behaviours*. The x-axis represents the video frame, and the y-axis represents the speed of upper limb.

2) HUMAN BODY EXTERNAL ELLIPSE

In **part A**, the external rectangle is used to complete the delimitation of the human body, which has achieved excellent results in the area of centroid extraction. However, there are still some differences between the real shape characteristics of the rectangle and the human body, which can not accurately describe the body shape characteristics.

Therefore, an external ellipse, which is closer to the human body shape, is used to represent the human body's posture characteristics. The external ellipse of humans while standing is displayed in **Figure 7(a)**, and the external ellipses during

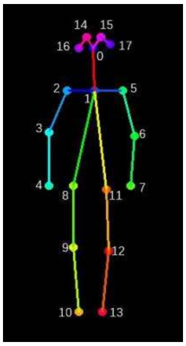


FIGURE 6. Human structure extracted by OpenPose.

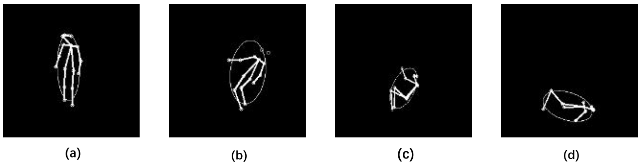

(a) (b) (c) (d)

FIGURE 7. Human body external ellipse.

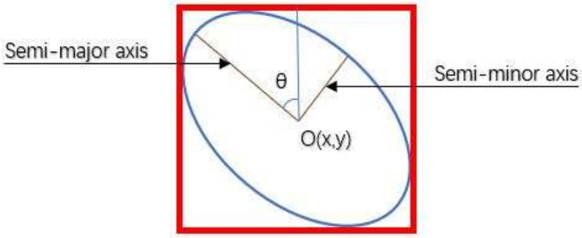


FIGURE 8. Parameters of ellipse.

falling are illustrated in **Figure 7(b)**, **(c),** and **(d)**. The image centre moment method [23] is used for fitting the ellipse. The fitting process and formula are as follows:

As **Figure 8** shows, the ellipse can be represented by four parameters. (1) Ellipse centre position coordinate O(x_c, y_c),

(2) The angle between the long axis of the ellipse and the vertical direction θ, (3) The length of the elliptical semi-major axis e_l, (4) The length e_s of the semi-minor axis of the ellipse.

First, the definition of image moment is as follows:

$$p_{ab} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^a y^b dxdy \tag{5}$$

where p_{ab} is the a + b moment of the image.

The centre moment of the ellipse as follows:

$$\mu_{ab} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-x_c)^a (y-y_c)^b \, d(x-x_c)d(y-y_c) \tag{6}$$

Next, the centre point of the ellipse is defined as the centroid of the human body, and the pixel coordinates within the range of the human body target area are (x_i, y_i), the μ_{11}, μ_{02}, and μ_{20} represent the second-order central moment of the image:

$$\mu_{11} = \sum (x_i - x_c)(y_i - y_c) \tag{7}$$

$$\mu_{02} = \sum (y_i - y_c)(y_i - y_c) \tag{8}$$

$$\mu_{20} = \sum (x_i - x_c)(x_i - x_c) \tag{9}$$

The angle (θ) between the major axis and the horizontal axis gives the ellipse orientation and can be computed with the second-order central moments. The formula is as follows:

$$\theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{10}$$

The eigenvalues I_{max} and I_{min} are given by:

$$I_{max} = \frac{\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \tag{11}$$

$$I_{min} = \frac{\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2}}{2} \tag{12}$$

Then the semi-magor axis e_l and the semi-minor axis e_s of the ellipse can be expressed as:

$$e_l = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[\frac{I_{max}^3}{I_{min}}\right]^{\frac{1}{8}} \tag{13}$$

$$e_s = \left(\frac{4}{\pi}\right)^{\frac{1}{4}} \left[\frac{I_{min}^3}{I_{max}}\right]^{\frac{1}{8}} \tag{14}$$

C. FEATURE EXTRACTION BY SLIDING WINDOW

For realizing the feature extraction based on the sequence, the new features need to be extracted from the video continuously, and the old features also need to be removed from the sequence in real-time. The sliding window model can meet this demand. In the sliding window model, the system uses a fixed size storage space to store data with a certain time series (the storage space is called window). Over time, the window moves directionally, new data is added to the head of the window, and the data at the tail is pushed out. This process will continue until the window traverses all data. A sliding window model is composed of two parameters W_l and V_w. Where W_l is the length of the window, V_w is the moving speed of the window on the data. In our model, the features of the falling-state and fallen-state need to be extracted and determined independently. To meet this demand, we define a dual-channel parallel sliding window, which consists of two branches called **falling-state detection window** and **fallen-state detection window**.

The **falling-state detection window** is used to extract the falling-state (P_f), which represented by the Centroid drop rate (V_c) and the Upper limb velocity (V_l) defined in **part A**. The window is defined as $Pd_w(W_l, V_w = 1)$, which means the initial window contains the feature information of frames 1 to W_l. After sliding, the window will contain the feature information of frames 2 to $(W_l + 1)$, and so on until the end of the video segment. In this process, V_c and V_l between two adjacent frames in the window will be calculated in real-time. The results of falling-state extraction can be expressed as a vector $P_f = [(V_{c1}, V_{l1}), (V_{c2}, V_{l2}), \ldots, (V_{cn}, V_{ln})]$, where n = W_l, V_{ci} and V_{li} corresponds to the V_c and V_l of the i-th window.

For detecting the fallen-state (S_f) feature which defined in part B, the **fallen-state detection window** is defined as $Sd_w(W_l, V_w = 1)$, whose input is the human key points of each frame, and output is the 18 standardized key points and 4 ellipse parameters (formula 5-14) of the last frame in the window. The final output of this step will be expressed as $S_f = [(x_0, y_0), (x_1, y_1), \ldots, (x_{17}, y_{17}), \mathrm{O}, \theta, e_l/e_s]$, where coordinate (x_i, y_i) corresponds to the ith human body keypoint.

Step 3. State Classification: In **Step 2**, the falling-state and fallen-state are finally expressed as two vectors P_f and S_f, which will be fed into two classifiers to detect whether a falling-state or fallen-state happens. The vector P_f is composed of the relevant velocity features in each frame in the sliding window Pd_w, which describes the velocity features in the range of W_l. The vector S_f represents the human state features in the last frame of the sliding window Sd_w, which describes the current state of the human body after the end of a window. In order to detect the falling-state and fallen-state, P_f and S_f are fed as input to a 5-layer MLP Neural Network and a RandomForest, respectively. The classification results can be expressed as $P_c \in \{0, 1\}$ and $S_c \in \{0, 1\}$, where 0 indicates that the condition has not occurred, and 1 suggests that the condition has occurred.

Step 4. Fall Event Detection: In the final fall detection, the classification results Pc and Sc are combined to detect whether the fall event happened. A fall detection window $Fd_w(W_l, V_w = 1)$ is used to realize the temporary storage of S_c and P_c sequences. In a range of W_l, when a falling-state is detected, the state of the human body in the last frame of the current window will be judged at the same time. Therefore, the fall detection result $F_c \in \{0, 1\}$ will be expressed as:

$$F_c = \begin{cases} 1 & if\ S_{cW_l} = 1\ and \sum_{i=1}^{W_l-1} P_{ci} \geq 1 \\ 0 & Others \end{cases} \tag{15}$$

S_{cW_l} indicates the value of the W_l-th S_c in the current Fd_w, and P_{ci} indicates the value of the i-th P_c in the current Fd_w.

V. EXPERIMENTAL PROTOCOL AND RESULTS

A. DATA DESCRIPTION

In order to prove the validity of our method, experiments were conducted on three public databases: Fall Detection Dataset, Le2i Fall Detection Dataset, and UR Fall Detection Dataset.

The Fall Detection Dataset[31] consists of 21499 320∗240 RGB and depth images that show the activate of humans at different angles and different lighting conditions. According to the state of the human, these images are marked in six categories: Standing, Sitting, Lying, Bending, Crawling, and Empty.

The Le2i Fall Detection Dataset [29] consists of 191 human activity videos in four scenes: Office, Home, Coffee Room, and Lecture Room. The format of the video is 320∗240, 25frame/s.

The UR Fall Detection Dataset [30] consists of 70 depth video and corresponding RGB video (the image format is 1920∗1080), which contains 30 fall videos and 40 daily activity videos from different angles and different lighting conditions.

During the training, we process the original dataset to fit our model. 23160 frames are selected from the dataset and preprocessed. According to the length of the window, the generated data vector are grouped. The dataset was relabeled, with the group containing the fall event marked as fall and the other groups marked as non-fall. The generated dataset is taken as our training dataset.

B. PERFORMANCE METRICS

When detecting a video sequence, four possible cases are corresponding to four valid parameters:

True positive (TP): a video segment contains a fall, and is correctly detected as a fall.

False positive (FP): a video segment does not contain falls, but is incorrectly detects as a fall.

True negative(TN): a video segment does not contain falls, and is correctly detects as non-fall.

False negative(FN): a video segment contains a fall, but is incorrectly detected as not a fall.

Based on these four parameters, five indicators, including sensitivity, specificity, accuracy, precision and F-score, are defined to measure the reliability of fall detection method, which are calculated as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{17}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

$$Precision = \frac{TP}{TP\text{+FP}} \tag{19}$$

$$F\text{-}Score = \frac{2TP}{2TP\text{+FP} + \text{FN}} \tag{20}$$

In the following classifier selection and model evaluation, these five indicators will be used as performance indicators. Among them, Sensitivity describes the sensitivity of the model to detect falls. Specificity describes the ability of the model to prevent misjudgment. The higher this index is, the lower the probability that the model misjudges other behaviors as falls. These two parameters can show the characteristics of the classification model more intuitively, so they are used as a relatively more important reference standard in the selection process of the classifier.

C. WINDOW LENGTH COMPARISON

In the sliding window model, the setting of window length (W_l) may affect the classification results. We designed a set of experiments to verify the classification effect of window

TABLE 1. The falling-state accuracy of different window lengths.

W_l(frame)	Accuracy(%)
10	96.12
15	94.19
20	98.28
25	95.56
30	96.67

length from 5 to 30 frames. In the experiment, the training data set is divided into 10 pieces, each of which is 2316 pictures with the format of 320 ∗ 240. The data are grouped according to the window size and tested on the MLP classifier with the method of ten-fold cross-validation. The experimental results are shown in **Table 1**.

According to the experimental results, when the window length is 20, the best classification accuracy is achieved.

D. FALLING-STATE CLASSIFIER COMPARISON

Artificial neural networks adopt more than one neuron to handle the received information, and are effective methods for solving many optimization and classification problems [32]. **Table 2** illustrates the comparison between the MLP Neural Network with some frequently used classifiers on the falling-state classification. In the MLP, the total layer is set to 5; the input layer includes 40 nodes because the dimension of input is 40, the nodes of the hidden layer are set to 240, 60, and 8. During training, ReLU is used as the activation function, Cross-Entropy is selected as the loss function, and Stochastic Gradient Descent is used as the optimizer. During the experiment, the window length was set to 20 frames that achieved the best results in the previous step. The training set contains 1042 ∗ 20 samples, and the test set contains 116 ∗ 20 samples, a ten-fold cross-validation method is used for estimating.

The results in **Table 2** confirm that in the falling-state classification, the MLP classifier achieves the best performance among all classifiers with 98.28% accuracy, 98.26% sensitivity, and 98.07% specificity. Compared with KNN, which gets the same sensitivity, MLP has a great advantage in specificity, which means that the probability of MLP misjudging normal human activities as falls is very low.

E. FALLEN-STATE CLASSIFIER COMPARISON

Table 3 shows the comparison between the RandomForest with other machine learning methods on the fallen-state classification. In the RandomForest, the number of features is set to the max to select all the features, and the estimator number is set to 120. During the experiment, the window length is set to 20 frames, and the ten-fold cross-validation method is used to estimate. The training set contains 1042 ∗ 20 samples, and the test set contains 116 ∗ 20samples.

As shown in **Table 3**, RandomForest achieved the best score (96.98% accuracy, 95.98% sensitivity, 97.88 specificity, 96.73% precision, and 96.8% F-Score) in fallen-state classification.

F. FALL EVENT DETECTION RESULTS

To evaluate the reliability of the proposed dual-channel feature integration model, we tested the model on the Le2i Fall Detection Dataset and UR Fall Detection Dataset which are commonly used in the fall detection field. The detailed parameters of the two datasets are shown in Section V, Part A. During the test, the window length (W_l) is set as 20, the MLP is used as the falling-state classifier, and the RandomForest is used as the fallen-state classifier.

Table 4 shows the experimental results of the proposed method and three other computer vision-based methods on

TABLE 2. The accuracy of different classifiers in falling-state detection.

Approach	Accuracy(%)	Sensitivity(%)	Specificity(%)	Precision(%)	F-Score(%)
GDBTree	95.86	96.15	95.28	95.23	95.69
DTree	90.25	95.19	80.76	89.36	84.85
RandomForest	96.12	96.63	94.17	93.80	95.92
SVM	96.80	96.52	97.59	95.68	96.10
K-NN	94.04	98.26	93.75	89.68	93.78
MLP	98.28	98.26	98.07	94.95	97.41

TABLE 3. The accuracy of different classifiers in fallen-state detection.

Approach	Accuracy(%)	Sensitivity(%)	Specificity(%)	Precision(%)	F-Score(%)
GDBTree	95.61	93.01	97.97	97.65	95.27
DTree	95.70	94.03	97.21	96.83	95.45
RandomForest	96.98	95.98	97.88	97.63	96.80
SVM	95.51	93.94	96.95	96.55	95.22
K-NN	94.45	94.96	93.99	93.48	94.22
MLP	95.64	92.17	97.29	96.86	94.46

TABLE 4. Fall Event Detection Results between our method and the current commonly used methods on the Le2i Fall Detection Dataset.

Method	Accuracy(%)	Precision(%)	Sensitivity(%)	Specificity(%)	F-Score
Our Method	96.91	97.65	96.51	97.37	97.08
Gradient Boosting Classifier [33]	79.31	79.41	83.47	73.07	81.39
Gaussian Mixture Model [34]	86.21	89.13	93.18	64.29	91.11
Consecutive-frame Voting [35]	91.38	88.68	97.92	60	93.07

TABLE 5. Final test results between our method and the current commonly used methods on the UR Fall Detection Dataset.

Method	Accuracy(%)	Precision(%)	Sensitivity(%)	Specificity(%)	F-Score
Our Method	97.33	97.78	97.78	96.67	97.78
Dai et al.[36]	94	——	95	96.70	——
Harrou et al (2017).[37]	96.66	94	100	94.93	96.91
Marcos (2017)	95	——	100	92	——
Kwolek and Kepski(2014)	90	83.30	100	80	90.89

the Le2i Fall Detection Dataset. The data show that the proposed method has advantages in some parameters. The only drawback is that the Sensitivity value of the method is slightly inferior to the Consecutive-frame voting method, which means that the detection ability of our method still needs to be improved. From another perspective, our method has a more significant advantage in Specificity, which means that our method is less likely to misjudge other daily behaviors as falls.

Table 5 shows the experimental results of the proposed method and four other computer vision-based methods on the UR Fall Detection Dataset. The data show that compared with other methods, our method has certain advantages in overall Accuracy and makes a better balance between the two metrics of Sensitivity and Specificity. At the same time, it also gets a higher Precision.

In conclusion, on the basis of ensuring Sensitivity, our method improves Specificity to a certain extent and generally maintains high Accuracy and Precision. This means that in the process of fall detection, our method can sensitively identify the occurrence of fall behavior and reduce the possibility of misidentifying other daily behaviors as falls.

G. MODEL EFFICIENCY

Fall detection requires a certain real-time, so the execution speed is also a necessary standard to measure the effectiveness of a model. For this reason, we tested the execution speed of the model on our device. **Table 6** shows the average processing time of a frame in different stages.

As shown in **Table 6**, the preprocessing stage takes the longest time, and the feature extraction and classification stage takes 0.48ms and 2.78ms each frame. Finally, we get the result of 35.80ms/frame, which means that our model can process 27.93 frame images in each second on average. This execution speed can basically meet the real-time processing of videos which taken by 25-30 fps conventional cameras.

In the preprocessing, two deep convolution neural network models, OpenPose and Yolo, are used to process the image. Their large scale model also brings a large amount of computation, which takes a relatively long time. In the process of feature extraction, two channels are used to complete the calculation of the two types of features defined in Section IV, which only involves a few basic operations, so the efficiency is high. In the classification part, we use MLP and RandomForest to classify the extracted feature vectors P_f and S_f. Among them, the MLP neural network designed by us has only three hidden layers, 240, 60, and 8 nodes respectively, which means that the scale of the trained model is not large. The random forest model contains 20840 samples, each sample is a 40 dimensional vector (S_f), which is not a large amount of computation. Therefore, the time spent in the classification part is relatively short.

TABLE 6. The time spent for processing a frame in different stages.

Stage	Processing Time (ms/frame)
Preprocessing	32.54
Feature Extraction	0.48
Classification	2.78
Total	**35.80**

Then, we compared our model with the execution efficiency of several typical computer vision-based fall detection methods. As shown in **Table 7**, the execution efficiency of our method is seemingly better than two different depth feature-based methods (Lie *et al.* and Huang *et al.*), a background subtraction-based method (Wang *et al.*), and a Multi-camera shape matching method (Rougier *et al.*).

H. TESTING ENVIRONMENT

In terms of hardware, the experiment runs on a server with Intel Core i5 CPU, Nvidia GTX1080 GPU, and 32g RAM. In terms of software, the operating system is Ubuntu 18.04, and the development language is Python 3.6.

TABLE 7. Comparison of our method with several common methods in execution speed.

Method	Speed (fps)
Our Method	27.93
DeeperCut with LSTM	16.26
Openpose with VGG-16,SVM	26.17
Background Subtraction and SVM	18.00
Multi-camera shape matching	16.68

VI. CONCLUSIONS AND FUTURE WORK

The work presented in this paper is mainly focused on investigating an accurate and efficient fall detection for older adults based on computer vision techniques. The approach presented here employs the preprocessing model combined with Yolo and OpenPose to obtain key points and the position information of the human body and then extract two types of features which are falling-state and fallen-state from the preprocessed data. In the detection of the falling-state, the speed change of the human body part is the main feature. Then a MLP is used to classify the feature sequence. In the detection of the fallen-state, the features are the human body keypoint that depict human limbs' position and human external ellipse that describe the external contour features of the human body. These features are then fed into a RandomForest for classification. Finally, the two classification results were combined to obtain the fall detection results.

Experimental results show that the proposed approach is universal. The approach achieves 96.91% accuracy on the Le2i Fall Detection Dataset and achieves 97.33% accuracy on the UR Fall Detection Dataset. In addition, our method achieves a good balance between sensitivity and specificity. This result illustrates that the combination of falling-state and fallen-state features is significant. The falling-state features make contribution to identify sudden changes in the shape of the human body. The fallen-state features can identify the state of the human body lying on the ground, and reduce the misjudgment of similar actions, such as bending, down. The fusion of multiple features makes the judgment of falls more reasonable and accurate.

In the future, we will focus on fall detection in more complex environments, such as outdoor fall detection, crowd stampede event detection. At the same time, our model will be extended to identify other dangerous behaviours that may occur in the life of the elderly. Furthermore, we hope to develop a practical application of the fall detection system in combination with the Internet of things.

REFERENCES

[1] (2015). *WHO Number of People Over 60 Years Set to Double by 2050; Major Societal Changes Required*. [Online]. Available: https://www.who.int/mediacentre/news/releases/2015/older-persons-day/en/

[2] R. Anita and K. Anupama, "A survey on recent advances in wearable fall detection systems," *BioMed. Res. Int.*, vol. 2020, Jan. 2020, Art. no. 2167160.

[3] C. Todd and D. Skelton, "What are the main risk factors for falls among older people and what are the most effective interventions to prevent these falls?" WHO Regional Office Eur., Health Evidence Netw. Rep., Copenhagen, Denmark, 2004. Accessed: Apr. 5, 2004. [Online]. Available: http://www.euro.who.int/document/E82552.pdf

[4] S. R. Lord, C. Sherrington, H. B. Menz, and J. C. Close, *Falls in Older People: Risk factors and Strategies for Prevention*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[5] M. Alwan, P. J. Rajendran, S. Kell, D. Mack, S. Dalal, M. Wolfe, and R. Felder, "A smart and passive floor-vibration based fall detector for elderly," in *Proc. 2nd Int. Conf. Inf. Commun. Technol.*, 2006, pp. 1003–1007.

[6] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls," *IEEE Trans. Biomed. Eng.*, vol. 56, no. 12, pp. 2858–2867, Dec. 2009.

[7] J.-S. Lee and H.-H. Tseng, "Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8293–8302, Sep. 2019, doi: 10.1109/JSEN.2019.2918690.

[8] L. Montanini, A. Del Campo, D. Perla, S. Spinsante, and E. Gambi, "A footwear-based methodology for fall detection," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1233–1242, Feb. 2018.

[9] M. M. Hassan, A. Gumaei, G. Aloi, G. Fortino, and M. Zhou, "A smartphone-enabled fall detection framework for elderly people in connected home healthcare," *IEEE Netw.*, vol. 33, no. 6, pp. 58–63, Nov. 2019.

[10] X. Xi, W. Jiang, Z. Lü, S. M. Miran, and Z.-Z. Luo, "Daily activity monitoring and fall detection based on surface electromyography and plantar pressure," *Complexity*, vol. 2020, pp. 1–12, Jan. 2020.

[11] A. Gumaei, M. M. Hassan, A. Alelaiwi, and H. Alsalman, "A hybrid deep learning model for human activity recognition using multimodal body sensing data," *IEEE Access*, vol. 7, pp. 99152–99160, 2019.

[12] O. Kerdjidj, N. Ramzan, K. Ghanem, A. Amira, and F. Chouireb, "Fall detection and human activity classification using wearable sensors and compressed sensing," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 1, pp. 349–361, Jan. 2020.

[13] W.-Y. Shieh and J.-C. Huang, "Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system," *Med. Eng. Phys.*, vol. 34, no. 7, pp. 954–963, Sep. 2012.

[14] S.-G. Miaou, P.-H. Sung, and C.-Y. Huang, "A customized human fall detection system using omni-camera images and personal information," in *Proc. 1st Transdisciplinary Conf. Distrib. Diagnosis Home Healthcare (D2H2)*, 2006, pp. 39–42.

[15] X. Yuan, L. Kong, D. Feng, and Z. Wei, "Automatic feature point detection and tracking of human actions in time-of-flight videos," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 677–685, 2017, doi: 10.1109/JAS.2017.7510625.

[16] F. Merrouche and N. Baha, "Depth camera based fall detection using human shape and movement," in *Proc. IEEE Int. Conf. Signal Image Process. (ICSIP)*, Aug. 2016, pp. 586–590.

[17] G. Debard, P. Karsmakers, M. Deschodt, E. Vlaeyen, E. Dejaeger, K. Milisen, T. Goedemé, B. Vanrumste, and T. Tuytelaars, "Camera-based fall detection on real world data," in *Outdoor and Large-Scale Real-World Scene Analysis*. Cham, Switzerland: Springer, 2012, pp. 356–375.

[18] K. G. Gunale and P. Mukherji, "Fall detection using k-nearest neighbor classification for patient monitoring," in *Proc. Int. Conf. Inf. Process. (ICIP)*, Dec. 2015, pp. 520–524.

[19] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 6, pp. 1274–1286, Nov. 2012.

[20] Y. Yun, C. Innocenti, G. Nero, H. Linden, and I. Y.-H. Gu, "Fall detection in RGB-D videos for elderly care," in *Proc. 17th Int. Conf. E-Health Netw., Appl. Services (HealthCom)*, Oct. 2015, pp. 422–427.

[21] W.-N. Lie, A. T. Le, and G.-H. Lin, "Human fall-down event detection based on 2D skeletons and deep learning approach," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Chiang Mai, Thailand, Jan. 2018, pp. 1–4.

[22] Z. Huang, Y. Liu, Y. Fang, and B. K. P. Horn, "Video-based fall detection for seniors with human pose estimation," in *Proc. 4th Int. Conf. Universal Village (UV)*, Boston, MA, USA, Oct. 2018, pp. 1–4.

[23] A. Lotfi, S. Albawendi, H. Powell, K. Appiah, and C. Langensiepen, "Supporting independent living for older adults; employing a visual based fall detection through analysing the motion and shape of the human body," *IEEE Access*, vol. 6, pp. 70272–70282, 2018.

[24] R.-D. Wang, Y.-L. Zhang, L.-P. Dong, J.-W. Lu, Z.-Q. Zhang, and X. He, "Fall detection algorithm for the elderly based on human characteristic matrix and SVM," in *Proc. 15th Int. Conf. Control, Autom. Syst. (ICCAS)*, Busan, South Korea, Oct. 2015, pp. 1190–1195.

[25] Z. Cao, G. H. Martinez, T. Simon, S.-E. Wei, and Y. A. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jul. 17, 2019, doi: 10.1109/TPAMI.2019.2929257.

[26] H. Liu, M. Zhou, and Q. Liu, "An embedded feature selection method for imbalanced data classification," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 3, pp. 703–715, May 2019, doi: 10.1109/JAS.2019.1911447.

[27] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 3645–3649.

[28] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[29] *The Fall Detection Dataset*. Accessed: Jun. 30, 2019. [Online]. Available: http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr

[30] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014.

[31] *Fall Detection Dataset*. Accessed: Jun. 30, 2019. [Online]. Available: http://falldataset.com

[32] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019, doi: 10.1109/TNNLS.2018.2846646.

[33] M. Chamle, K. G. Gunale, and K. K. Warhade, "Automated unusual event detection in video surveillance," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Aug. 2016, pp. 1–4.

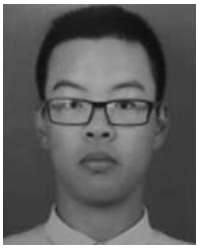
[34] A. Poonsri and W. Chiracharit, "Fall detection using Gaussian mixture model and principle component analysis," in *Proc. 9th Int. Conf. Inf. Technol. Electr. Eng. (ICITEE)*, Phuket, Thailand, Oct. 2017, pp. 1–4.

[35] A. Poonsri and W. Chiracharit, "Improvement of fall detection using consecutive-frame voting," in *Proc. Int. Workshop Adv. Image Technol. (IWAIT)*, Chiang Mai, Thailand, Jan. 2018, pp. 1–4.

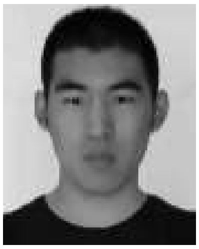
[36] B. Dai, D. Yang, L. Ai, and P. Zhang, "A novel Video-Surveillance-Based algorithm of fall detection," in *Proc. 11th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Beijing, China, Oct. 2018, pp. 1–6.

[37] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "An integrated vision-based approach for efficient human fall detection in a home environment," *IEEE Access*, vol. 7, pp. 114966–114974, 2019.

BO-HUA WANG was born in Shaanxi, China, in 1999. He is currently pursuing the bachelor's degree in engineering with the Software College, Northeastern University. His current research interests include image processing and video content analysis. His research focuses on image feature extraction and human behavior modeling.



JIE YU is currently pursuing the B.Eng. degree with the Software College, Northeastern University. His current research interests include computer vision, pattern recognition, and smart home design. He is trying to use the Internet of Things (IoT) device to detect the fall of the elderly at home alone. His research mainly focuses on the analysis of the action characteristics of the fall event and the computer vision on embedded architectures.



KUO WANG was born in Liaoning, China, in 1999. He is currently pursuing the bachelor's degree in engineering with the Software College, Northeastern University. He has some experience of working on research of computer vision and development of related software.

XUAN-YU BAO was born in Beijing, China, in 1998. He is currently pursuing the bachelor's degree in engineering with the Software College, Northeastern University. His current research interests include pattern recognition and image processing. His research focuses on analyzing, visualizing, charting test results, and proposing the direction of use.

KE-MING MAO received the B.S. and Ph.D. degrees from Northeastern University, Shengyang, China, in 2003 and 2009, respectively. He is currently an Associate Professor with Northeastern University. His research interests include computer vision, deep learning, and intelligence data analysis.