Research Article

Fall Detection and Direction Judgment Based on Posture Estimation

Chunmiao Yuan, Pengju Zhang, Qingyong Yang, and Jianming Wang

1School of Software, Tiangong University, Tianjin 300387, China
2School of Computer Science and Technology, Tiangong University, Tianjin 300387, China
3School of Software and Communication, Tianjin Sino-German University of Applied Sciences, Tianjin 300350, China

Correspondence should be addressed to Jianming Wang; wangjianming@tiangong.edu.cn

Received 22 April 2022; Accepted 18 May 2022; Published 15 June 2022

Academic Editor: Chun Wei

Copyright © 2022 Chunmiao Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the problem of elderly people falling easily, it is very necessary to correctly detect the occurrence of falls and provide early warning, which can greatly reduce the injury caused by falls. Most of the existing fall detection algorithms require the monitored persons to carry wearable devices, which will bring inconvenience to their lives and few algorithms pay attention to the direction of the fall. Therefore, we propose a video-based fall detection and direction judgment method based on human posture estimation for the first time. We predict the joint point coordinates of each human body through the posture estimation network, and then use the SVM classifier to detect falls. Next, we will use the three-dimensional human posture information to judge the direction of the fall. Compared to the existing methods, our method has a great improvement in sensitivity, specificity, and accuracy which reaches 95.86, 99.5, and 97.52 on the Le2i fall dataset, respectively, whereas on the UR fall dataset, they are 95.45, 100, and 97.43, respectively.

1. Introduction

According to statistical information, accidental fall is a phenomenon with a high frequency among the elderly population [1, 2]. Even for people living independently, falls are common occurrences. At the same time, the injury of falling to the elderly is also extremely serious, which is the main factor causing the death of the elderly [3]. It will take a lot of time and energy if unexpected fall activities are monitored manually. Therefore, it is very important to realize a platform that can monitor people’s activity status, detect the occurrence of falls and other unexpected behaviors in real time, and give a timely early warning of falls.

In recent years, the researches on fall detection have achieved remarkable results and have a wide application prospect [4, 5]. However, many existing studies are based on wearing wearable devices [6–8]; these approaches have led to the appearance of smart environments for elderly assistance, which had been traditionally limited to home settings. And most strategies often only focus on detecting whether the falling behavior occurs, but pay little attention to the direction of the fall and the degree of injury caused by the fall. Actually, the extent of injury caused by a fall is largely affected by the direction of the human body fall, and different fall directions will cause different intensities of injury to the human body. The injury caused by a human body falling forward is generally smaller than that caused by falling back. Therefore, a perfect and practical fall prediction system should not bring a burden to the monitored person and it can not only accurately detect falls but also classify the risk of falls.

As a result of the research carried out, this study presents the following main contributions:

1) A vision-based fall detection algorithm is proposed which extracts human features directly from video through posture analysis and does not need the observed person to carry any wearable devices.
2. Related Works

With the in-depth study of fall detection, a large number of schemes have been proposed and great achievements have been made. According to different research methods and experimental equipment, fall detection can be divided into two classifications: wearable sensor-based and vision-based.

The wearable sensor-based collects motion and other parameter signals through relevant sensor devices and then uses the calculation formula to convert the collected signals into information that can represent the motion state, such as acceleration information [9]. According to this information, the real state of the current target can be judged. Sensors are usually placed on the waist, legs, or near the neck. The commonly used equipment is an accelerometer, three-axis gyroscope, magnetometer, and barometer. Zerrouki et al. [10] used an accelerometer to get the motion state to detect falls. Chen et al. [11] proposed a fall detection system that integrated a three-axis gyroscope, a three-axis accelerometer, and a Bluetooth module for wireless communication to design a waist-mounted, miniature fall detection device. The device collected information from the gyroscope and accelerometer for analysis to deduce continuous signals representing the human body postures. According to the posture and signal relationship, artificial intelligence was used to construct a highly accurate model. Alarifi and Alwadain [12] used the wearable sensor device composed of a magnetometer, gyroscope, and accelerometer which was placed in six different positions on the subject’s body, then they used the intelligent AlexNet convolution network for fall detection. A waist-mounted device was presented to detect possible falls in elderly people [13] through data coming from a three-axis accelerometer, a three-axis gyroscope, a three-axis magnetometer, and a barometer sensor integrated into the device. Built-in smartphone sensors can be utilized to detect falls [14]. Built-in smart helmets are also wearable sensors [15] to detect falls for ease of use. After getting the angular motion or trunk inclination by sensors, machine learning or deep learning methods are often used to judge the fall. The wearable sensor-based method has high recognition accuracy, but is less convenient because sensor devices need to be carried by the monitored person.

The vision-based method records the activities of people in the scene by using different types of cameras, such as ordinary cameras and depth cameras. According to the changes in human characteristics in each frame, the target people are analyzed by image processing or neural network. The human body silhouettes or bounding boxes can be extracted by traditional computer vision methods, such as the frame difference method, background subtraction, or foreground segmentation, then the methods use those features as input for a classifier (e.g., Gaussian mixture model (GMM), SVM, and MLP) to automatically detect if a fall has occurred. For example, Sehairi et al. [16] obtained human contour from a series of video frames through background subtraction, extracted the change of aspect ratio according to the difference of contour, calculated the vertical velocity of the head using a finite state machine, and inferred the state of target by synthesizing these features. Zerrouki et al. [17] computed occupancy areas around the body’s gravity center, extracted their angles, and fed them into various classifiers; the SVM being the one that obtained the best results. The same authors extended their previous work by adding curvelet coefficients as extra features and applying a hidden Markov model (HMM) to model the different body poses [18]. A less frequent technique was used by Harrou et al. [19], who applied multivariate exponentially weighted moving average (MEWMA) charts. Rougier et al. [20] detected whether fall behavior occurred by quantifying the change of human body shape in the video sequence. For these solutions, the shape of the human body will produce different results due to different camera positions, at the same time, carrying backpacks, crutches, and other objects will also affect the detection results.

Another vision-based method judges according to the joint information of pose estimation. The pose estimation method can accurately predict the position of each joint of the body and represent the abstract human body information with a set of 2D or 3D joint points. Asif et al. [21] obtained the feature information of the human body represented by human joint point information through a stacked hourglass network, inputted the features into a CNN model with modal-specific layer and multimodal embedding layer, and learned high-level feature embedding to distinguish fall posture and non-fall posture. Chen et al. [22] used OpenPose [23] to get the data of human joint points and then identified whether the falling behavior occurred by calculating the falling speed of the hip joint center, the angle between the human body centerline and the ground, and the width-height ratio of the rectangle outside the human body. At the same time, whether the person stood up independently after falling was considered, and the state of the target was judged by combining the above conditions. Compared with contour information, using the human bone joint feature is easier for action recognition and is less affected by external factors. Therefore, this study selects a pose estimation method to extract human features. The injury caused by falling is largely affected by the direction of the human body falling when the falling action occurs; however, most of these studies pay no attention to the direction of the fall, which is a major issue that is taken into account in our solution.

3. Proposed Method

3.1. Overview of Our Method. Our method is divided into two modules: fall detection and direction judgment. In the fall detection module, the videos captured by an ordinary camera are analyzed frame by frame and then sent to the fall detection network to detect whether there are accidents such as falls in the current scene. After the fall action is detected, the 2D joint feature will be sent to the direction judgment module: first the dimension transformation network will
transform the 2D feature into 3D joint points with depth information, and then the direction judgment network will estimate the specific fall direction by calculating 3D joint set, so as to give appropriate early warning for the fall event. The overview of this method is shown in Figure 1.

3.2. Fall Detection Module

3.2.1. Human Posture Extraction. In order to realize fall detection, we used the joint point information as the feature. We first extracted the relevant joint point information from the video frame through the posture estimation method, screened the bone points with high correlation as the feature input into the classification network, and then trained and learned a group of fall and non-fall data by support vector machine, so as to judge whether the human body is in a fall state.

Through the comprehensive analysis and comparison of current posture estimation algorithms, the OpenPose algorithm with better real-time performance was finally selected to realize posture estimation to obtain joint point parameters. When using the OpenPose algorithm to test our dataset, it was found that there was misjudgment in some data, such as identifying background or objects similar to the human body as a part of the body, which greatly affected the final recognition effect. After analysis, the reason for this is that OpenPose is a bottom-up method that first predicts all possible joint points from the picture, and then clusters these points according to the dependency relationship to form an independent skeleton. In this process, it is impossible to ensure whether the predicted points belong to a part of the human body which results in some points offset from the human body area being predicted.

In order to solve the above problems and accurately obtain the joint point coordinates within the human body area, we proposed a top-down posture estimation network, which took the OpenPose network as the baseline, and added a layer of human body detection network to detect the human body and determine the regional range of each person. Through the combination of human body boundary features and joint features, the final joint information was more accurate.

The human detection network uses Fast R-CNN to detect boundary features of the human body and these features will be used to screen the false human joint points. The posture estimation network puts the human picture into a 10-layer VGG19 convolution network and extracts a set of feature map $F$ after a series of convolution pooling operations. Next, $F$ is sent to a two-branch structure to predict the confidence $S$ of body joint points and affinity vector $L$ between the inter joints, respectively. $S$ represents the possibility of a joint at the pixel position, and $L$ is used to determine the position and direction of the body part. In order to improve the
accuracy of prediction, an iterative architecture is used to establish a multilevel two-branch structure. Each time, the prediction results and image features of the previous stage are inputted to the next two-branch structure for prediction. After continuous iteration, a more accurate $S$ and $L$ are finally generated. A candidate body joint point set $D_j$ of multiple persons is obtained.

$$D_j = \{d_{mj}^n : j \in \{1 \ldots J \}, m \in \{1 \ldots N_j \}\}, \quad (1)$$

where $d_{mj}^n$ represents the coordinate of the $n$th joint point of body part $j$.

In the process of posture estimation, some structures are often judged as human joint points because they are very similar to human parts. At the same time, they have high confidence, which will produce errors in the final joint set and affect the accuracy of recognition results. In order to ensure that the obtained joint point set is a real human joint and is located within the contour of the human body, we screened and judged the boundary features and joint features extracted earlier to obtain a more accurate joint set. By means of traversal screening, for each set of joint points in the joint features, we compared each joint coordinate with all the bounding box features to judge whether the current joint coordinate was within one of the bounding boxes and selected all the qualified joint points as the final joint feature output.

3.2.2. Fall Detection by SVM. After optimizing the joint points combined with the human body boundary box, the misidentified and out-of-range joints are screened out, and the joint points are input into the SVM classifier for fall and non-fall classification training. Then, according to the trained model, the input human joint points coordinate parameters are directly used to predict the fall of the target person. But it is found that because the accuracy of each pose estimation is different, for the pictures with unclear human structure, many joint points will not be recognized, resulting in the loss of features, which will affect the final fall classification results. Furthermore, different camera shooting angles will affect the height and body shape of the human body and even lead to deformation, which will also affect the results of fall classification.

Therefore, in order to eliminate these factors that may cause deviation results, this study uses also a direction vector instead of a direct coordinate as the input of SVM. The process of fall detection using SVM is shown in Figure 2. By comparing and observing the changes in joint points before and after falling and the division relationship of body parts, three joint points with significant changes in the shoulder, hip, and ankle are selected, and a directed vector is formed according to their coordinates; and finally, the calculated vector set is inputted into the fall classification network as features for training and learning, so as to judge whether the current person is in a fall state.

3.2.3. Pseudocode of Fall Detection. The pseudocode of fall detection is described as Algorithm 1.

3.3. Fall Direction Judgment Module

3.3.1. Feature Dimension Transformation. After the human joint features are obtained through the fall detection module, the next work is to deeply analyze and calculate these features, so as to obtain the specific fall direction information. Because the characteristics of human joints are 2D information, in 2D plane space, we can only simply distinguish the left and right, and there is no way to judge other directions. Therefore, we transformed the 2D parameters into 3D parameters and then used the 3D coordinates with depth information as the parameters to construct the 3D coordinate system, selected specific joint points to form the 3D vector, and calculated the spatial transformation angle according to the change of the vector, so as to obtain more detailed direction change information. At the same time, we calculated the change range and direction of specific joint points in space, so as to accurately calculate the direction of the fall. The process of the fall direction module is shown in Figure 3.
Dimension transformation network adopts the idea of Wandt and Rosenhahn [24] to convert 2D parameters into 3D parameters which uses the idea of generative adversarial networks (GANs) consisting of three modules, namely generator module, discriminator module, and re-projection module.

The input 2D human pose is first processed by the 3D generator, which is composed of two branches: 3D pose estimation and camera parameter estimation, which are used to generate a preliminary estimation of 3D human coordinates and internal parameters of the camera. The 3D pose estimation branch continuously learns the mapping from 2D pose to 3D coordinates to obtain a better representation. It is composed of two continuous residual blocks, and each block contains a fully connected network of two 1000 neurons. After full connection, leaky ReLU is used as the activation function; and finally, a set of vectors are outputted as 3D estimated coordinates.

In order to match the 3D pose estimation branch generation with the real pose, in the re-projection module, the camera parameters obtained from the camera parameter estimation branch are used to convert the predicted 3D coordinate results into 2D pose representation, so as to compare the deviation between the predicted results and the originally inputted 2D pose, so as to realize weak supervision. The loss function of the camera parameter estimation branch is as follows:

$$L_{\text{rep}}(X, K) = \|W - KX\|_F,$$

where $W$ is the original input 2D posture matrix, $K$ represents the camera parameter matrix, $X$ represents the estimated 3D posture matrix, and $KX$ is the result of projecting the estimated 3D posture into 2D space by re-projection.

The posture discrimination module is the same as the discriminator in GAN to judge whether the picture generated by the generator is correct. For the 3D posture predicted above, it is necessary to determine whether the prediction is accurate by comparing the discrimination model with the real 3D coordinates. The discrimination network model is also composed of two branches. The first branch introduces the kinematic chain space layer, transforms the generated 3D posture into the kinematic chain space transformation angle, and finally outputs a fall direction judgment.

**Algorithm 1: Fall detection.**

**Figure 3: Flowchart of fall direction judgment module.**
space matrix, and then feeds them into the full connection layer containing 100 neurons. The kinematic space matrix is a match matrix, which introduces the symmetry of human body structure. The elements on the diagonal of the kinematic chain space matrix represent the length of each bone, which ensures the size of the generated posture. Elements outside the diagonal actually represent the angle information of bone motion, because they can actually be regarded as the cosine of the included angle. The calculation formula of the motion chain space matrix is as follows:

\[
B = X C;
\]

\[
\Psi = B^T B = \begin{pmatrix}
\tilde{l}_1^2 & \ldots & \ldots & \ldots \\
\ldots & \ddots & \ddots & \ddots \\
\ldots & \ldots & \ddots & \ddots \\
\ldots & \ldots & \ldots & \tilde{l}_b^2
\end{pmatrix},
\]

where \(X\) represents the estimated 3D posture matrix, \(C\) is an adjacency matrix, and the non-zero elements of each column are composed of 1 and −1.

The other branch is the pure fully connected layer, which is directly represented by the coordinate of 3D posture. After two-branch processing, the features of the two parts are spliced, and finally inputted into a full connection layer. Finally, combined with the real 3D coordinates, Wasserstein is used as the loss function to calculate the loss.

In the training process, the discrimination module and generator are trained in turn, so that the discriminator can effectively judge the authenticity of the current generator input. After continuous learning, the generator gradually improves the reliability of the generated samples to pass the authentication of the discrimination module. In this way, through mutual training and learning, the loss between the two parts converges to the lowest. Finally, the input 2D human posture can be used to obtain the corresponding precise joint coordinates in 3D space to achieve the transformation of coordinate feature dimension. The converted 3D space coordinates are shown in Figure 4.

3.3.2. Calculating the Angle of the Fall. After obtaining the 3D joint coordinates of the human body at the timing of falling, and before/after falling, the specific fall direction can be calculated from the perspective of 3D space.

Fall direction is a broad concept, which includes both angle size and direction. In a 3D space, for the human body represented by different coordinate points, calculating the angle of a human fall is to find the included angle between the plane composed of human joint points when falling and the plane composed of human joint points before/after falling. Because the human body in this article is composed of 18 joint points, if all points are used to form a complex irregular graph rather than a plane, it is impossible to calculate the angle of human falling by calculating the included angle of the two planes. We used the vector as input to solve this problem. According to the relationship between joints, we selected specific joint points to form a vector. By calculating the included angle between the vector composed of the same group of joints when falling and before/after falling, we could get the fall angle of the human body. Therefore, the problem of calculating the fall angle of the human body in 3D space can be transformed into the problem of calculating the included angle between vectors composed of specific joint points.

As for the selection of vectors, when the body is blocked or blurred, some joints will not be recognized, resulting in the problem that vectors cannot be formed. In order to prevent the nonexistence of joint points when calculating the included angle with only one vector, we selected four easily identifiable joint points from the joint set, combined them in pairs to form four vectors, and calculated the included angle by transformation, respectively, and finally synthesized the results to reduce the probability of errors. According to the observation, the occurrence probability and recognition accuracy of joint points of the left shoulder, left hip, right shoulder, and right hip are higher than other joint points. Therefore, these four points are combined in pairs to form feature vectors.

Vectors \(\vec{a}\) and \(\vec{b}\) are, respectively, composed of the same group of joints before and after the fall in 3D space. The angle \(\theta\) between two vectors can be calculated as follows:

\[
\theta = \arccos\frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \cdot \frac{180}{\pi}.
\]

If the coordinates of the vectors \(\vec{a}, \vec{b}\) are \((x_1, y_1, z_1)\) and \((x_2, y_2, z_2)\), then angle \(\theta\) is as follows:

\[
\theta = \arccos\frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{\sqrt{x_1^2 + y_1^2 + z_1^2} \cdot \sqrt{x_2^2 + y_2^2 + z_2^2}} \cdot \frac{180}{\pi}.
\]

3.3.3. Calculating the Direction of the Fall. Through the above method, the angle of the fall is calculated, and the specific direction of the fall also needs to be calculated. In this study, the direction of the human fall is divided into four kinds: forward, backward, left, and right. Next, the method of calculating the direction of falling will be introduced.

When we analyze the human body in the video frame, the fall direction is compared with the original position of the human body, not from the perspective of the observer. Therefore, when estimating the direction of a human falling, it is first necessary to determine whether the human body is facing the surveillance camera or back to the camera before
falling. To solve this problem, we determine the orientation of the human body in the frame by comparing the position relationship of some joints in the 3D joint points.

According to experience, when the front of the human body faces the camera, the left and right of the body are just opposite to the left and right of the camera. In the previously established 3D coordinate system of the human body, we selected two joint points of the left shoulder and right shoulder to compare their positional relationship. For the human body in 3D space, if the coordinate of the left shoulder joint point in the $x$-axis direction is greater than that of the right shoulder joint point in the $x$-axis direction, it indicates that the human body faces the camera, otherwise it indicates that the human body faces away from the camera. After determining the orientation of the human body, these two situations are discussed, respectively:

1. When the human body faces the camera, we supervise the changes in the human head joint points in 3D space. For the same joint point, if the coordinate of the joint point in the depth $z$-axis direction is greater than the original coordinate and exceeds a certain range after falling, it can be judged as falling backward. If the coordinate of the joint point in the $z$-axis direction is lesser than the original coordinate and exceeds a certain range after falling, it can be judged as falling forward. If the coordinate of the joint point in the horizontal $x$-axis direction is greater than the original coordinate and exceeds a certain range after falling, it can be judged as falling to the left. If the coordinate of the joint point in the $x$-axis direction is lesser than the original coordinate and exceeds a certain range after falling, it can be judged as falling to the right. The calculation formula of the fall direction is as follows:

\[
\text{Direction} = \begin{cases} 
\text{left: } C \leq \text{Position 1 (x)} - \text{Position (x)}, \\
\text{right: } C < \text{Position (x)} - \text{Position 1 (x)}, \\
\text{backward: } C \leq \text{Position 1 (z)} - \text{Position (z)}, \\
\text{forward: } C < \text{Position (z)} - \text{Position 1 (z)} 
\end{cases}
\]

$Position (x)$ and $Position (z)$ refer to the original coordinates of the joint point on the $x$-axis and $z$-axis; $Position1 (x)$ and $Position1 (z)$ refer to the coordinates of the same point on the $x$-axis and $z$-axis after falling; and $C$ is the threshold.

2. If the human body is facing away from the camera, it shall be calculated according to the reverse rule.

According to the above discussion, the direction judgment of the fall is divided into two scenarios and eight possibilities. Finally, it is necessary to comprehensively match all conditions to estimate the specific direction of falling. According to the observation, different fall directions of the human body will cause different degrees of injury. When falling backward, the protection ability of the human body is the worst and the probability of head injury is the largest, so the injury of falling backward is much greater than that of falling in other directions. Therefore, we classify the risk level of the previously estimated human fall direction: when the human body falls backward, the corresponding risk level is very dangerous, and the risk level of falls in other directions is general.
3.3.4. Pseudocode of Fall Direction Judgment. The pseudocode of fall direction judgment is described as Algorithm 2.

4. Main Results

4.1. Dataset. At present, the datasets mainly used for fall detection include the Multiple Cameras Fall Dataset (Multicam) [25], Le2i fall dataset (Le2i) [26], and the UR Fall Dataset (URFD) [27], which contain a large number of scenes of normal human movement and fall action. For our study, in addition to detecting whether the fall event occurs, we will estimate the direction of the fall. Each dataset above only contains the fall action in partial directions, which cannot completely cover the fall events in each direction. Therefore, we screened the above datasets to form a new fall dataset, in which we selected some videos containing fall events, and some videos of normal daily activities as a comparison. In our fall dataset, the fall direction of the human body includes forward, backward, left, and right.

4.2. Evaluation Metrics. In fall detection, all samples are classified by the fall detection algorithm, and the classification results can be divided into the following four categories: TP (true positive) indicates that there is a fall event in the sample, and the fall detection algorithm correctly identifies it as a fall; TN (true negative) indicates that there is no fall event in the samples, and the fall detection algorithm correctly identifies it as non-fall; FP (false positive) indicates that there is no fall event in the samples, but the fall detection algorithm incorrectly identifies it as a fall; and FN (false negative) indicates that there is a fall event in the samples, but the fall detection algorithm incorrectly identifies it as a non-fall.

In order to measure the performance of the fall detection algorithm, the existing evaluation standards such as precision, accuracy, specificity, sensitivity, and F1 score are used generally [28]. The values of the above standards are calculated from the classification results of the algorithm. The higher the value, the better the performance of the method. The evaluation metrics we used are defined as follows:

\[
\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{7}
\]

The precision represents the ratio of the samples accurately judged as fall to all the samples judged as fall:

\[
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}
\]

Sensitivity refers to the ratio of the samples correctly identified as falls to all fall samples. The higher its value in fall detection, the better the recognition performance of fall events. Here, the sensitivity is the same as the recall rate.

\[
\text{Sensitivity} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}
\]

Specificity refers to the ratio of correctly identified non-fall samples to all non-fall samples. The higher its value in fall detection, the better the recognition performance of non-fall events.

\[
\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{10}
\]

F1 score is the harmonic average of precision and recall to punish extreme cases.

\[
\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}
\]

For the performance of the fall direction estimation algorithm, we also use the above evaluation criteria to calculate.

4.3. Results of Fall Detection. Our proposed algorithms are compared with other fall detection algorithms in different literatures. According to the classification of fall detection results above, we calculated different fall detection evaluation standard values, and the results are shown in Table 1.

In the above table, our method 1 and our method 2 represent the fall detection results obtained by using human body coordinates directly and using direction vectors, respectively.

It can be seen from Table 1 that the experimental results of all methods are relatively good in URFD. The effect of using coordinates in our method 1 is lower than that of using direction vectors in our method 2. For the standard of accuracy, although our methods are not the highest, our method 2 has only a little difference from the first place and is higher than other methods. For the sensitivity standard, our methods are lower than that of other methods, because ours depend on the joint points identified by posture estimation. In some cases, the positions of joint points estimated are not accurate enough, which cause the failure of fall detection.

As shown in Figure 5, the first row of pictures is the result of behavior detection as a fall, and the second row of pictures is the result of non-fall caused by inaccurate joint point recognition due to occlusion, incomplete human body, or other reasons.

For the standard of specificity, our methods are higher than other methods, which shows that our methods are very accurate in judging non-fall behaviors.

<table>
<thead>
<tr>
<th>Detection methods</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kwolek and Kepski [30]</td>
<td>100</td>
<td>96.67</td>
<td>95.71</td>
</tr>
<tr>
<td>Bourke et al. [31]</td>
<td>100</td>
<td>90</td>
<td>—</td>
</tr>
<tr>
<td>Kwolek and Kepski [27]</td>
<td>100</td>
<td>92.5</td>
<td>95</td>
</tr>
<tr>
<td>Yun and Gu [32]</td>
<td>96.77</td>
<td>89.74</td>
<td>—</td>
</tr>
<tr>
<td>Alaoui et al. [33]</td>
<td>100</td>
<td>95</td>
<td>97.5</td>
</tr>
<tr>
<td>Our method 1</td>
<td>95</td>
<td>96.47</td>
<td>95.64</td>
</tr>
<tr>
<td>Our method 2</td>
<td>95.45</td>
<td>100</td>
<td>97.43</td>
</tr>
</tbody>
</table>
It can be seen from Table 2 that compared with other methods, our methods have better results on the Le2i dataset. Our method 2 of using direction vector has higher sensitivity, accuracy, and precision than other methods, but for the specificity standard, it is slightly lower than the method of Alaoui et al. [33]. Overall, our methods have a good recognition effect for fall detection.

It can be seen from Table 3 that in the MultiCam dataset, our method 1 of using coordinate features for fall detection is higher than the other two methods in terms of accuracy and F1 score, and slightly lower than the method of Asif et al. [21] in terms of recall rate. Overall, the performance of our methods is slightly better than other methods.

### Table 2: Fall detection results on Le2i.

<table>
<thead>
<tr>
<th>Detection methods</th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charfi et al. [34]</td>
<td>73</td>
<td>97.7</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Chamle et al. [35]</td>
<td>83.47</td>
<td>73.07</td>
<td>79.31</td>
<td>—</td>
</tr>
<tr>
<td>Poonsri and Chiracharit [36]</td>
<td>93</td>
<td>64.29</td>
<td>86.21</td>
<td>—</td>
</tr>
<tr>
<td>Alaoui et al. [37]</td>
<td>94.55</td>
<td>90.84</td>
<td>90.9</td>
<td>—</td>
</tr>
<tr>
<td>Alaoui et al. [33]</td>
<td>95</td>
<td>100</td>
<td>97.5</td>
<td>—</td>
</tr>
<tr>
<td>Asif et al. [21]</td>
<td>92.44</td>
<td>—</td>
<td>—</td>
<td>92.45</td>
</tr>
<tr>
<td>Asif et al. [38]</td>
<td>89.92</td>
<td>—</td>
<td>—</td>
<td>90.08</td>
</tr>
<tr>
<td>Our method 1</td>
<td>91.73</td>
<td>98.5</td>
<td>94.79</td>
<td>98.66</td>
</tr>
<tr>
<td>Our method 2</td>
<td>95.86</td>
<td>99.5</td>
<td>97.52</td>
<td>99.57</td>
</tr>
</tbody>
</table>

### Table 3: Fall detection results on MultiCam.

<table>
<thead>
<tr>
<th>Detection methods</th>
<th>F1 score (%)</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asif et al. [21]</td>
<td>87.08</td>
<td>87.03</td>
<td>87.05</td>
</tr>
<tr>
<td>Asif et al. [38]</td>
<td>86.38</td>
<td>86.28</td>
<td>86.58</td>
</tr>
<tr>
<td>Our method 1</td>
<td>87.38</td>
<td>88.13</td>
<td>86.66</td>
</tr>
</tbody>
</table>

### 4.4. Results of Fall Direction Judgment.

The direction judgment module will be executed only after the human body is detected as in a fall state. In order to verify the performance of fall direction judgment, it needs to be tested in the dataset containing only fall events. Therefore, the direction judgment experiments are carried out on our fall dataset. Results of converting to a 3D posture by our method are shown in Figure 6.

It can be seen from Table 4 that the method in this article has the highest recognition accuracy for humans falling forward and low recognition accuracy for falling backward. The main reason for this phenomenon is that the direction judgment module in this article depends on the 3D joint points obtained by the dimension transformation network. For the complex 2D human posture, the 3D joint coordinates obtained by the dimension transformation network are not accurate enough, which affects the final direction judgment.

When the human body falls, the fall direction is usually either forward or backward, either left or right. Therefore, this article divides the fall dataset into forward-backward and left-right groups to verify our fall direction method. For the parameters in the evaluation criteria, TP in the forward-backward fall dataset represents the samples that are correctly judged as falling forward; TN represents the samples
Figure 6: Dimension conversion results.

Table 4: Accuracy of different fall directions.

<table>
<thead>
<tr>
<th>Fall direction</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward</td>
<td>82.5</td>
</tr>
<tr>
<td>Backward</td>
<td>70.37</td>
</tr>
<tr>
<td>Left</td>
<td>72.22</td>
</tr>
<tr>
<td>Right</td>
<td>75</td>
</tr>
</tbody>
</table>

Table 5: Fall direction judgment results on our forward-backward and left-right datasets.

<table>
<thead>
<tr>
<th></th>
<th>Sensitivity (%)</th>
<th>Specificity (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward-backward</td>
<td>80.48</td>
<td>73.07</td>
<td>77.61</td>
</tr>
<tr>
<td>Left-right</td>
<td>78.26</td>
<td>68.42</td>
<td>73.81</td>
</tr>
</tbody>
</table>
that are correctly judged as falling backward; FP refers to the samples whose direction of falling are wrongly judged to be forward; and FN refers to the samples whose fall direction are wrongly judged to be backward. For the left-right fall dataset, the parameters are also selected according to the corresponding rules. The experimental results are shown in Table 5.

5. Conclusion

In this article, a fall detection algorithm based on human posture analysis is proposed. Different situations of falling are studied from the perspective of posture analysis, the timing of fall behavior is detected, and the direction of fall is judged, so as to make an accurate early warning. Human body detection and joint point estimation are combined to screen and calculate the 2D joint information according to the human body boundary box and confidence, and the wrong joint points and redundant information are eliminated to obtain a more accurate set of human joints. Next, the limited 2D joint point information is promoted to 3D spatial coordinates containing depth information; and finally, the 3D coordinates are used to judge the fall direction of the human body.

Although the method proposed in this article has certain advantages in the human fall detection tasks, there are still defects to be improved in the following aspects: (1) this method is only applicable to some simple fall movements, and the judgment of fall direction is not ideal for complex fall movements; (2) the method consists of multiple modules. Each module needs less time to run alone, but it needs more time to realize the whole process. Therefore, in the future, it is necessary to improve the adaptability of the dimension transformation network to the complex fall movements and generate a more accurate 3D human posture, so as to realize more accurate fall direction judgment; at the same time, it is necessary to improve the real-time performance of the method.

Data Availability

The data that are used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Tianjin Science and Technology Program (19PTZWHZ00020).

References


