Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2024.052931

ARTICLE

A Double-Interactively Recurrent Fuzzy Cerebellar Model Articulation Controller Model Combined with an Improved Particle Swarm Optimization Method for Fall Detection

Jyun-Guo Wang*

The Department of Medical Informatics, Tzu Chi University, Hualien County, 97004, Taiwan

*Corresponding Author: Jyun-Guo Wang. Email: jgwang@gms.tcu.edu.tw

Received: 19 April 2024 Accepted: 02 July 2024 Published: 13 September 2024

ABSTRACT

In many Eastern and Western countries, falling birth rates have led to the gradual aging of society. Older adults are often left alone at home or live in a long-term care center, which results in them being susceptible to unsafe events (such as falls) that can have disastrous consequences. However, automatically detecting falls from video data is challenging, and automatic fall detection methods usually require large volumes of training data, which can be difficult to acquire. To address this problem, video kinematic data can be used as training data, thereby avoiding the requirement of creating a large fall data set. This study integrated an improved particle swarm optimization method into a double interactively recurrent fuzzy cerebellar model articulation controller model to develop a cost-effective and accurate fall detection system. First, it obtained an optical flow (OF) trajectory diagram from image sequences by using the OF method, and it solved problems related to focal length and object offset by employing the discrete Fourier transform (DFT) algorithm. Second, this study developed the D-IRFCMAC model, which combines spatial and temporal (recurrent) information. Third, it designed an IPSO (Improved Particle Swarm Optimization) algorithm that effectively strengthens the exploratory capabilities of the proposed D-IRFCMAC (Double-Interactively Recurrent Fuzzy Cerebellar Model Articulation Controller) model in the global search space. The proposed approach outperforms existing state-of-the-art methods in terms of action recognition accuracy on the UR-Fall, UP-Fall, and PRECIS HAR data sets. The UCF11 dataset had an average accuracy of 93.13%, whereas the UCF101 dataset had an average accuracy of 92.19%. The UR-Fall dataset had an accuracy of 100%, the UP-Fall dataset had an accuracy of 99.25%, and the PRECIS HAR dataset had an accuracy of 99.07%.

KEYWORDS

Double interactively recurrent fuzzy cerebellar model articulation controller (D-IRFCMAC); improved particle swarm optimization (IPSO); fall detection

1 Introduction

Declining birth rates have led to the gradual aging of society in many Eastern and Western countries. According to a World Health Organization report, the proportion of the population that is aged ≥65 and ≥70 years is increasing by 28%–35% and 32%–42% each year, respectively [1]. A survey from the US National Institutes of Health indicates that in the United States, approximately

Copyright © 2024 The Author. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1.6 million older adults experience a fall-related accident each year [2]. According to Carone et al. [3], the European Union's elderly dependency ratio (the number of people aged ≥65 years divided by the number of people aged 15–64 years) is expected to double by 2050 to greater than 54%. China is also confronting a rapidly aging population. In 2020, older adults comprised approximately 17% of the Chinese population, and this percentage is predicted to increase to 35% by 2050 [2]. In Taiwan, which has been considered an aging society since 2018, individuals aged ≥65 years comprise 14% of the population. Families previously lived in multigenerational households, and at least one adult was available to care for older adult relatives. However, most adults now work outside the home; thus, the older adults in their families are often left alone at home. An accident experienced by an older adult when they are alone at home can have disastrous consequences. Falls are the main cause of injury-related death in older adults [4], and 87% of all fractures in older adults are caused by falls. Although many falls do not cause injury, 47% of older adults who are uninjured after a fall are nonetheless unable to move normally without assistance, which indirectly affects their health. In addition, in one study, 50% of older adults who fell and then remained on the floor for a certain period after their fall were found to die within 6 months, even if they had not been directly injured in the fall [5]. Therefore, a system that can automatically and immediately detect fall events and inform a caregiver or family member can substantially benefit older adults. Additionally, increasing economic stress has reduced the time available for young people to care for their older adult relatives. This crucial social problem can be addressed by using information collected through different systems, such as cameras.

Many methods have been proposed for posture recognition and fall detection. These methods can mainly be divided into two categories: non-computer-vision (NCV) and computer-vision (CV) methods. NCV methods involve automatically recognizing human actions by conducting mathematical modeling and using sensors. These methods can be divided into three subcategories: dynamic system methods [6–9], stochastic methods [10–12], sensor methods [13–16], and wearable-sensors [17–19]. Dynamic system methods involve encoding action data as differential equations, whereas stochastic methods involve encoding action data as state transitions and data representing the distribution of states. Finally, sensor methods involve using a set of model parameters, which are regarded as discrete points in a multidimensional parameter space. Thus, the obtained action data are compressed into a mathematical model through symbolic representations called action symbols. Action symbols facilitate not only the synthesis of human-like actions but also the recognition of human behavior from observations. However, such mathematical models are not intuitive representations. A model that classifies observations does not understand action categories such as "walking," "running", or "squatting". In sensor methods, sound, vibration, and other data are captured using various sensors—fall detection can be achieved through the use of accelerometers, acoustic sensors, and floor vibration sensors. In [13], an activity detection method based on dynamic and static measurements made by a single-axis acceleration sensor was developed. Changes were observed by placing acceleration sensors on the chest and feet, and the measured signal is analyzed using a threshold-based algorithm for fall detection. In [14], a three-axis acceleration sensor was attached to different areas of the body, and the severity of the fall is determined by comparing the measured dynamic and static acceleration components to appropriate thresholds. Threshold-based algorithms can be used to detect certain falls, as shown by the experimental results of [14]. In [15], an acoustic fall detection system was designed that can automatically notify a guardian of a fall. The performance of this fall detection system was evaluated using simulated sounds corresponding to fall and nonfall events. A fall detection system based on ground vibration and sound induction was proposed in [16]. Suitable temporal and spectral characteristics were selected from signals generated by ground vibrations and sound induction, and fall and non-fall events were classified using a Bayesian classifier. The system above detects falls with a rate of 97.5% and detects errors with a rate of 1.4%. Although NCV methods can accurately detect falls, these methods have some limitations, such as the need for the individual under monitoring to wear an acceleration sensor. Moreover, environmental noise can affect the systems used in these methods. To overcome these problems, some scholars have proposed fall detection methods based on CV technology. However, the implementation of CV-based fall detection systems raises privacy concerns regarding the use of surveillance equipment. To alleviate these concerns, most such systems send an alarm to a caregiver or family member only when a fall is detected, and video of normal activities is neither stored nor transmitted.

In the past decade, substantial improvements have been achieved in CV systems using real-time object motion and video or image processing technologies, thus enabling the development of various fall detection methods. In [20–22], simple adaptive threshold or shape-based methods were used to distinguish falling from nonfalling actions. However, these methods had high false positive rates (e.g., the action of sitting down quickly was misclassified as a fall). The authors of [23] proposed a threshold-based method that involves using a calibrated camera to reconstruct the three-dimensional (3D) shape of a person. In the aforementioned study, fall events were detected by analyzing the volume distribution of the reconstructed 3D shape along the vertical axis. Experimental results demonstrated that the aforementioned method has excellent performance. Rapid developments have been made in pattern recognition technology, and many researchers have adopted this technology for fall detection [24–25]. In [24], a neurofuzzy network was used to classify postures and detect fall events. Similarly, in [25], the k-nearest neighbors' classifier was used for posture classification, and statistical hypothesis testing was performed to determine the threshold time difference suitable for distinguishing falls from and lying-down events. This method was found to have a correct detection rate of 84.44%. In [26,27], fall and nonfall events were classified by extracting feature information (such as contour, bright, and flow features) to achieve a robust system that can be used in bright environments containing multiple objects. In [28], the fall detection performance of three models was compared. Among them, the neural network model was effective in detecting falls with a 92% fall detection rate and a 5% false positive rate, respectively. Short video clips have been used by researchers to categorize fall and non-fall events [29,30]. In [29], continuous contours are analyzed using bounding boxes and motion information and to extract features. These features were then used to train an HMM (hidden Markov model) for the classification of fall and nonfall events. In [30], the 3D position information of an object was extracted using multiple cameras, and an improved hierarchical hidden Markov model was developed for fall detection. In [31], a method was proposed for automatically extracting motion trajectories and thereby detecting abnormal human activities. In [32], a semantic video summarization system based on fuzzy logic for detecting falls was proposed. This system was tested using 14 and 32 data sets on fall and nonfall behaviors, respectively. The test results demonstrated that all fall behaviors were correctly detected by the system, whereas only two nonfall behaviors were incorrectly classified as fall behaviors. Additionally, some techniques [33–37] are designed to identify ADLs (activities of daily living) and create fall monitoring systems that detect falls and provide feedback to doctors or relatives of patients. In the same way, ADL monitoring can be utilized in rehabilitation-assisted living applications [38–40].

Substantial progress has been achieved in deep learning technology [41], which has resulted in natural language processing [42] and improved CV [43] methods. Convolutional neural networks (CNNs) have been particularly successful in image classification tasks [43,44], employed in a variety of tasks as specialized feature extractors [45–49] and related technologies [50–54]. In [55,56], scholars converted articulated coordinates into two-dimensional (2D) image information and then used a CNN model to classify actions on the basis of the 2D image information. In [57,58], the 3D joint trajectory shape of bones was converted into three types of images representing the front, top, and side views of the joint trajectory shape. Actions were then classified using a CNN model and the three types of images. In [59], a CNN model was demonstrated to be a powerful tool for feature extraction and classification. An overview of the latest progress in CNNs (e.g., in terms of aspects such as regularization, optimization, activation function, loss function, and weight initialization) was provided in [60]. In [61], action classification accuracy was improved using multiple types of information in deep CNNs for action recognition. Sharma et al. [62] developed a spatial attention model based on long short-term memory (LSTM) that combined temporal and spatial attention mechanisms into a single scheme. In their work, they employ the proposed model to caption images and recognize actions. The last convolutional layer of GoogleNet [63] generates a feature map tensor for each frame, which is later sent to an LSTM to forecast action labels and regions of interest. Lu et al. [64] developed a visual attention–guided 3D CNN, which incorporated a soft attention mechanism.

Although the aforementioned methods have acceptable performance, achieving highly accurate fall detection with them remains difficult. Simulating falls in a controlled environment is how training data for fall detection is typically generated; thus, obtaining a large number of training samples is difficult, and the performance of a trained classifier is limited. To create a method for detecting falls that is not dependent on simulated fall data, the present study integrated an improved particle swarm optimization (IPSO) method into a double interactively recurrent fuzzy cerebellar model articulation controller (D-IRFCMAC) model for fall detection. The D-IRFCMAC model is based on a modified fuzzy cerebellar model articulation controller (CMAC) model [65]. The CMAC is a feedforward neural network with a high learning speed. We extensively modified the fuzzy CMAC to enhance its precision, reduce its memory requirements, and enable differential computation. Kinematic data sets were employed instead of smaller fall data sets to train the D-IRFCMAC model efficiently. The proposed method can accurately detect human behavior and fall events in long-term care centers or in homes. This study makes three main contributions to the literature. First, it obtained an optical flow (OF) trajectory diagram from image sequences by using the OF method, and it solved problems related to focal length and object offset by employing the discrete Fourier transform (DFT) algorithm. Second, this study developed the D-IRFCMAC model, which combines spatial and temporal (recurrent) information. Third, it designed an IPSO algorithm that effectively strengthens the exploratory capabilities of the proposed D-IRFCMAC model in the global search space.

This paper is organized as follows. Section 1 introduces the background of the present study and describes related studies, Section 2 explains the proposed method, Section 3 details the experimental procedure and results, and Section 4 provides the conclusions of this study.

2 Feature Selection Using the Proposed Methods

Four algorithms are described in this section: an OF algorithm, the DFT, the proposed D-IRFCMAC, and an IPSO algorithm.

No line breaks between paragraphs belonging to the same section.

2.1 OF Algorithm

The goal of the OF method is to capture movement trajectories. To achieve this goal, the Lucas–Kanade (LK) OF algorithm [66] was employed in the present study for calculating the image flow vector (v_x, v_y), where v_x and v_y are the velocities in the x-direction and y-direction, respectively. The OF vector for each pixel is obtained as follows:

$$OFI\,(x, y) = \begin{cases} 255, & \text{if } TH_{lb} \leq v_x \leq TH_{ub} \\ 255, & \text{if } TH_{lb} \leq v_y \leq TH_{ub} \\ 0, & \text{else} \end{cases} \tag{1}$$

where TH_{lb} and TH_{ub} are the lower- and upper-bound thresholds, respectively.

Once the OF vector has been calculated, the OF method is used to analyze overlapping images of a continuous sequence (the trajectory image). Trajectory images are categorized as OF trajectory images (OFTIs) or foreground trajectory images (FTIs), as illustrated in Fig. 1a,b, respectively. An OFTI presents a representation of an action, which is defined as a motion over time.

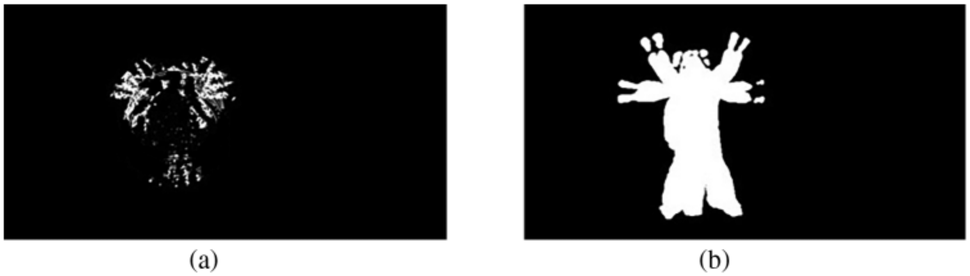

Figure 1: Examples of an (a) OFTI and (b) FTI

The image $OFTI\,(x, y, \pi)$ is defined as follows:

$$OFTI\ (x, y, \pi) = U_{t=t_0}^{\pi} OFI\,(x, y, t), \quad t_0 = 0, 3, 6, \ldots, \pi \tag{2}$$

where $OFI(x,\ y,\ t)$ represents the pixel values at positions x and y in an OF image at time t, defines the duration of a movement within a specific period, and U indicates the range of a set from one period to another. Although the LK OF algorithm can recognize most actions, it has difficulty in recognizing actions in some situations, such as when similar actions have different postures. Therefore, to distinguish different types of actions, a circle mark is added to the centroid of the action calculated using the OFTI. This circle mark facilitates calculation of the direction of motion to distinguish similar actions with different postures. However, the OFTI only approximately represents the movement of an object, and it does not enable the recognition of target types. Therefore, FTI features are also used because these features encode considerable foreground information. An FTI [denoted $FTI\,(x, y, \pi)$] is defined as follows:

$$FTI\,(x, y, \pi) = U_{t=t_0}^{\pi} FI\,(x, y, t), \quad t_0 = 0, 3, 6, \ldots, \pi \tag{3}$$

where $FI\,(x, y, t)$ represents the pixel values at positions x and y in a foreground image at time t. Moreover, π defines the duration of an action within a specific period. In experimental analysis, the foreground sampling interval of FTIs was set as six frames. The foreground sampling interval was found to be too small; thus, distinguishing different FTIs generated for different actions was impossible.

2.2 DFT

First, the horizontal and vertical projections of the OFTI and FTI are calculated (Fig. 2). Subsequently, the DFT [67] is used to convert the projection images obtained from the OFTI and FTI.

The advantages of the DFT are not limited to the focal distance of the camera and the location of the object; it can also reduce the number of feature dimensions by extracting low-frequency components. The conversion formula of the DFT is given as follows:

$$DFT_H[h] = \frac{1}{N}\sum_{y=1}^{N} H(y)\exp\left(\frac{-2j\pi yh}{N}\right), \quad h = 0, \ldots, N-1 \tag{4}$$

$$DFT_V[v] = \frac{1}{M}\sum_{x=1}^{M} V(x)\exp\left(\frac{-2j\pi xv}{M}\right), \quad v = 0, \ldots, M-1 \tag{5}$$

where $H(y)$ and $V(x)$ are the horizontal and vertical projections, respectively. The terms N and M denote the width and height of the trajectory image, respectively. Low-frequency features are selected using the DFT, and the 15 lowest-frequency values are used as features in the horizontal and vertical projections. Thus, a total of 60 features are selected from the horizontal and vertical projection images of the OFTI and FTI.

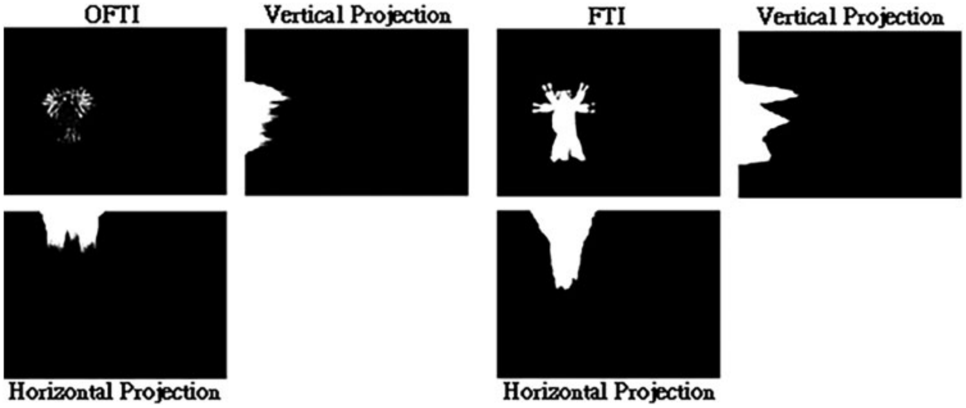

Figure 2: Horizontal and vertical projections of the OFTI and FTI

2.3 D-IRFCMAC Model

In this study, a D-IRFCMAC model that uses an interactive recurrent mechanism and a Takagi–Sugeno–Kang (TSK)-type linear function was developed for action recognition. The TSK-type linear function enables systematic extraction of fuzzy rules from data. From an input–output structure, this function systematically generates the antecedent and consequent of the relevant IF–THEN rule. The function adjusts the antecedent and consequent frameworks and variables in accordance with the adopted data. The structure of the proposed D-IRFCMAC model is displayed in Fig. 3. The model comprises the following six layers in sequence: an input layer, a membership function layer, a spatial firing layer, a temporal firing layer, a consequent layer, and an output layer. The function of each layer is detailed in the following paragraphs. The architecture implements similar fuzzy IF–THEN rules as described by using expression. The expression is computed as follows:

Rule j:

IF x_1 is A_{1j} and x_2 is A_{2j} ... and x_i is A_{ij}

THEN $y_j = \sum_{j=1} O_j^{(4)}\left(\alpha_{0j} + \sum_{i=1}^{N_D}\alpha_{ij}x_i\right)$

where x_i and y_j represent the input variables and local output variables, respectively; A_{ij} is the linguistic term of the antecedent part created using the Gaussian membership function; O_j is the output of the interactive feedback; and $\alpha_{0j} + \sum_{i=1}^{N_D} \alpha_{ij} x_i$ is the TSK-type linear function of the input variables.

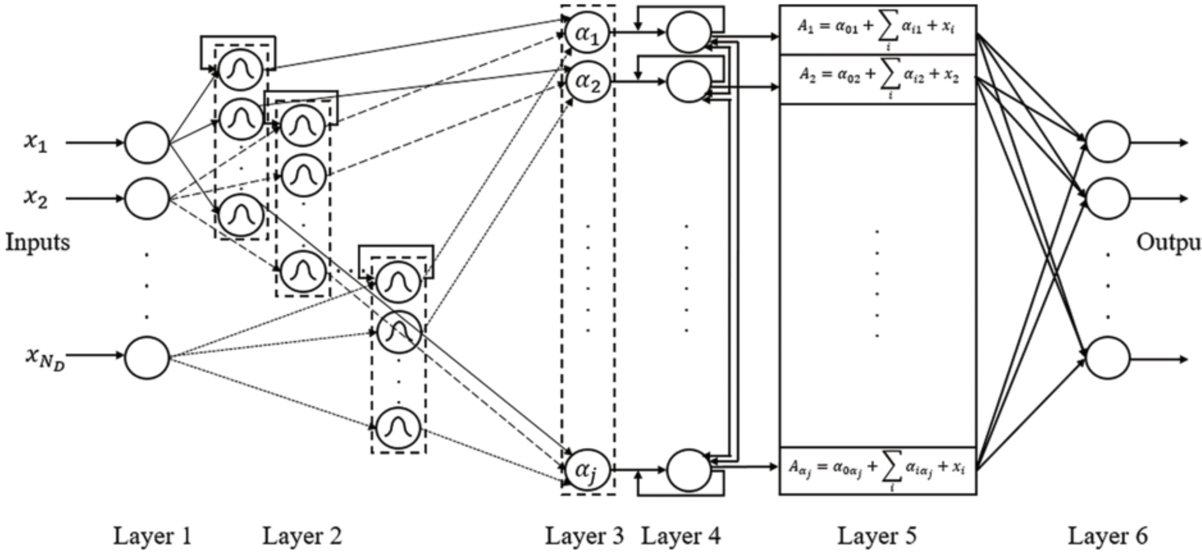
x_1
x_2
Inputs
x_{N_D}
α_1
α_2
α_j
$A_1 = \alpha_{01} + \sum_i \alpha_{i1} + x_i$
$A_2 = \alpha_{02} + \sum_i \alpha_{i2} + x_2$
$A_{\alpha_j} = \alpha_{0\alpha_j} + \sum_i \alpha_{i\alpha_j} + x_i$
Outputs
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Layer 6

Figure 3: Architecture of the D-IRFCMAC model

Input layer (layer 1): This layer is used to input feature vectors into the D-IRFCMAC model; these input vectors consist of crisp values.

$$O_i^{(1)} = u_i^{(1)}, \text{ and } u_i^{(1)} = x_i \tag{6}$$

Membership function layer (layer 2): This layer is a fuzzy layer. The single linguistic label of the input variables in layer 1 corresponds to each node in layer 2. Input values are used to calculate membership values through layer 2 fuzzy sets. The Gaussian membership function implemented in layer 2 is expressed as follows:

$$O_{ij}^{(2)} = \exp\left(-\frac{\left[r_{ij} - m_{ij}\right]^2}{\sigma_{ij}^2}\right) \tag{7}$$

where m_{ij} and σ_{ij} are the mean and variance, respectively, of the Gaussian membership function. The input of layer 2 contains the memory terms $O_{ij}^{(2)}\ (t-1)$ of the previous time point; these terms store past information regarding the network.

$$r_{ij} = O_i^{(1)}\ (t) + O_{ij}^{(2)}\ (t-1) \cdot \theta_{ij} \tag{8}$$

where θ_{ij} is the feedback weight.

Spatial firing layer (layer 3): The nodes in this layer are called rule nodes or hypercube cells. Each node in layer 3 determines the firing strength information for a hypercube cell in accordance with the information acquired from the nodes of a fuzzy set in layer 2; each of these nodes represents one fuzzy rule or hypercube cell rule. Therefore, the hypercube cell rule nodes perform a fuzzy product operation, and the output function is as follows:

$$O_j^{(3)} = \prod\nolimits_i^{N_D} u_{ij}^{(3)}, \text{ and } u_{ij}^{(3)} = O_{ij}^{(2)} \tag{9}$$

Temporal firing layer (layer 4): Each node of this layer is a cyclic hypercube element node and comprises two parts: an internal feedback loop and external feedback loop interactively. Because the internal feedback of a hypercube element node alone is insufficient for representing all necessary information, the external feedback process is used obtain additional information from other nodes. In layer 4, each node has its own (self-loop) and other nodes (external loop) information reference. The temporal firing strength is expressed as follows:

$$O_j^{(4)} = \sum\nolimits_{k=1} \left(\omega_{kj}^{q} \cdot O_k^{(4)}\left(t-1\right)\right) + \left(1 - r_j^q\right) \cdot u_j^{(4)}, \text{ and } u_j^{(4)} = O_j^{(3)} \tag{10}$$

where ω_{kj}^{q} represents the ratio between the weight of the current and previous inputs and the weight of the network outputs. Moreover, r_j^q is an interactive weight between the self-loop and external loop.

$$O_j^{(5)} = O_j^{(4)}\left(\alpha_{0j} + \sum_{i=1}^{N_D} \alpha_{ij} x_i\right) \tag{11}$$

Consequent layer (layer 5): In this layer, each node is a linear combination function associated with the input variables. The relevant equation is as follows:

$$y = \frac{\sum_{j=1}^{N_L} O_j^{(4)}\left(\alpha_{0j} + \sum_{i=1}^{N_D} \alpha_{ij} x_i\right)}{\sum_{j=1}^{N_L} O_j^{(4)}} \tag{12}$$

Output layer (layer 6): This layer defuzzifies the fuzzy output of previous layers into a scalar output by using the center-of-gravity approach.

2.4 Proposed IPSO Algorithm

This section describes the particle swarm optimization (PSO) algorithm and the proposed IPSO algorithm.

2.4.1 PSO Algorithm

The PSO algorithm was introduced by Kennedy and Eberhart [68] and was first used in the field of social and cognitive behavior. This algorithm is based on how birds and fish forage for food. The algorithm is simple to understand and implement and thus has been widely employed as a problem-solving method in engineering and computer science. The position and velocity vectors of the *i*th particle in an N-dimensional search space can be represented as $x_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{id})$ and $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{id})$, respectively. The velocity of each particle is used to find the solution space. An evaluation function is designed on the basis of the relevant problem, and the output of this function is a fitness value indicating the position of the particle. The new velocity and position of each particle in the particle swarm can be obtained by utilizing its local best position (p_i) and global best positions $\left(p_g\right)$. The equations for calculating the fitness values for a new velocity and position are as follows:

$$v_{vid}^{k+1} = \omega \times v_{vid}^{k} + c_1 \times rand\left(\cdot\right) \times \left(P_{id} - x_{id}^{k}\right) + c_2 \times rand\left(\cdot\right) \times \left(P_{gd} - x_{id}^{k}\right) \tag{13}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{14}$$

Among those, ω is designed as coefficient of the inertia term, c_1 is designed as the coefficient of the cognitive term, and c_2 is designed as the coefficient of the social term. The term in Eq. (13) is a generated random number that is uniformly distributed over the range [0, 1]. Eq. (13) comprises three parts: a part representing the particle's previous velocity, a cognitive component, and a social component. The first part (related to the previous velocity) represents the momentum required by particles to move across the search space. The second part (cognitive component) guides each particle to move toward its local best position. Finally, the third part (social component) guides all particles to move toward the global optimal position. However, the conventional PSO algorithm is affected by the dimensionality of the data. If the size of the search space is greater, the efficiency of this algorithm is lower. Therefore, to improve the performance of PSO and enhance its ability to explore the global search space, this study developed an IPSO algorithm.

2.4.2 Proposed IPSO Algorithm

The overall flowchart of the proposed IPSO algorithm is displayed in Fig. 4, and the steps of this algorithm are described in the following sections. The IPSO algorithm can be employed to analyze and optimize the internal parameters $\left(m_{ij}, \sigma_{ij}, \theta_{ij}, \omega_{ij}\right)$ in the D-IRFCMAC model, consequently unveiling superior solutions and enhancing identification performance.

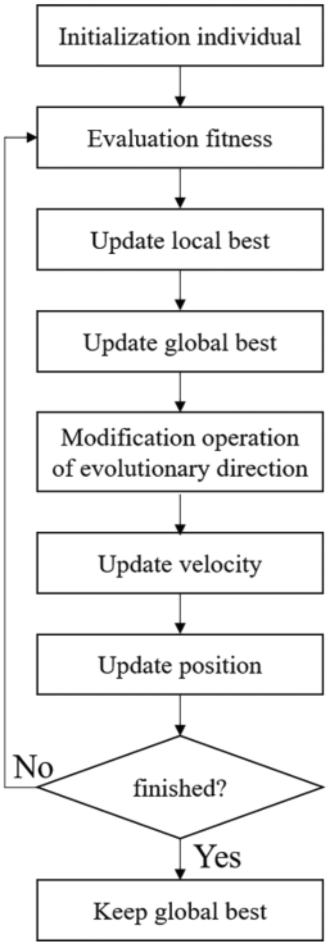

Figure 4: Flowchart of the proposed IPSO algorithm

Step 1 Individual initialization

Random values are first generated in the range [−1, 1] for the individual initialization of each particle. These values represent the average, standard deviation, and weight of each variable in the D-IRFCMAC model. Subsequently, coding is performed. The coding step is related to the membership functions and fuzzy rules for particles suitable for IPSO. Fig. 5 shows how parameters are coded for a particle. The parameters *i* and *j* represent the input variables and rules in the D-IRFCMAC model.

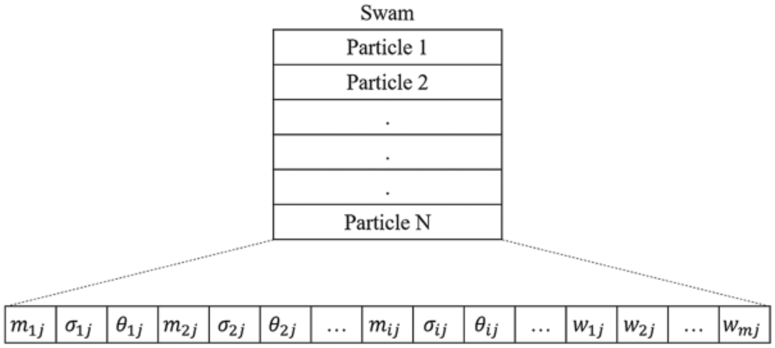

Figure 5: Coding of the fuzzy rule parameters of a particle in the IPSO algorithm

Step 2 Fitness evaluation

The fitness function for evaluating each particle in the swarm is defined as follows:

$$f_i = 1/Y \tag{15}$$

$$Y = \sqrt{\frac{1}{N}\sum_{p=1}^{N}\left(y_p - \overline{y}_p\right)}, \quad for \quad p = 1, 2, \ldots, N \tag{16}$$

where N represents the number of input data points and y_p and $\overline{y}_p$ represent the model output and desired output, respectively.

Step 3 Updating the local best value

In case the fitness value of the evaluated particle surpasses the local optimal value, this fitness value takes the place of the local optimal value.

Step 4 Updating the global best value

In case the fitness value of the evaluated particle surpasses the global optimal value, this fitness value takes the place of the global optimal value.

Step 5 Modification operation for the evolutionary direction

Evolutionary optimization is performed by selecting the three most effective solutions from each generation. The goal of each new solution is to surpass the previous best solution. The three best particles are ordered in accordance with their fitness, and the particles with the highest, second-highest, and third-highest fitness values are called the first, second, and third particles, respectively (denoted P_f, P_s, and P_t, respectively). These particles are expressed as follows:

First particle: $P_f = \left(P_{f1}, P_{f2}, P_{f3}, \ldots, P_{fd}\right)$, with fitness F_f

Second particle: $P_s = (P_{s1}, P_{s2}, P_{s3}, \ldots, P_{sd})$, with fitness F_s

Third particle: $P_t = (P_{t1}, P_{t2}, P_{t3}, \ldots, P_{td})$, with fitness F_t

The initial step is to update the third particle using a migration operation to create a new third particle. Subsequently, the second and first particles are used as references for the new third particle to move in the target direction; that is, the new third particle is updated again in accordance with the direction indicated by the second and first particles. The global search capacity is improved through this process. The modification operation for the evolutionary direction (MOED) is detailed as follows:

Step 5.1 The parameters of the two evolution directions and the initial index of the MOED are set as 1; that is, the initial number of iterations is set as 1. The number of MOED loops is set as N_L. The three particles (P_f, P_s, and P_t) with the highest fitness values in the swarm are selected for evolutionary optimization.

Step 5.2 A migration operation is used to generate a new diverse population and to prevent particles from gradually clustering; thus, the migration operation considerably increases the size of the search space for small-swarm exploration. Migrating particles are generated through nonuniform random selection based on the three best particles. The objective is to prevent the MOED from becoming trapped in a local extremum. The second and third particles P_s and P_t, respectively) are updated using Eq. (17).

$$x_{id} = \begin{cases} x_{id} + r\left(x_{id}^{L} - x_{id}\right), & if\ \ th < \dfrac{x_{id} - x_{id}^{L}}{x_{id}^{U} - x_{id}^{L}} \\ x_{id} + r\left(x_{id}^{U} - x_{id}\right), & otherwise \end{cases} \tag{17}$$

where r and th are random numbers in the interval [0, 1].

Step 5.3 The current evolutionary direction and current evolutionary step size of the parallelogram are used to generate particle P_o, as expressed in Eq. (18). By using the two difference vectors $D_1 \times \left(P_{tj} - P_{sj}\right)$ and $D_2 \times \left(P_{tj} - P_{fj}\right)$, one can determine the parallelogram's next evolutionary direction and step.

$$P_{oj} = P_{tj} + D_1 \times \left(P_{tj} - P_{sj}\right) + D_2 \times \left(P_{tj} - P_{fj}\right) \tag{18}$$

Step 5.4 The fitness F_o is evaluated using the newly generated output particle (P_o).

Step 5.5 The third, second, and first particles (P_t, P_s, and P_f, respectively) are updated in sequence as follows:

(1) If $F_o > F_f$, then $P_t = P_s$, $P_s = P_f$, and $P_f = P_o$.

(2) If $F_o < F_f$ and $F_o > F_s$, then $P_t = P_s$, and $P_s = P_o$.

(3) If $F_o > F_t$ and $F_o < F_s$, then $P_t = P_o$.

(4) If $F_o = F_t = F_s$, then $p_o = p_o + DF$ ($DF \in [0, 1]$).

(5) If $F_o < F_t$, then $D_1 = D_1 \times (-C)$ and $D_2 = D_2 \times (-C)$.

Under the first three aforementioned conditions, the first, second, and third particles are updated on the basis of the fitness value of P_o. The fourth condition is considered if P_o, P_s, and P_t have the same fitness values. In this case, a random disturbance factor is added to the fitness value of P_o to prevent the MOED from becoming trapped at a local optimum. Finally, if the fitness value of P_o is lower than those of P_o, P_s, and P_t to obtain better fitness, the particle velocity is decreased with a constant penalty parameter (fifth condition).

Step 5.6 The global best value is updated. Replace the current global best particle with P_o if the fitness value of P_o is greater than the current global best value.

Step 6 Updating the particle velocity

By using the Eq. (13), the velocity of every particle dimension is updated.

Step 7 Updating the particle position

By using the Eq. (14), the position of every particle dimension is updated.

2.5 Proposed D-IRFCMAC-IPSO Method

A D-IRFCMAC model that combines spatial and temporal (recurrent) information is proposed. An IPSO algorithm that effectively strengthens the exploratory capabilities in the global search solution space is proposed. The IPSO is used to optimize the internal parameters of the D-IRFCMAC model and decrease the subjective influence of chosen parameters. The IPSO algorithm is employed to optimize the internal parameters of the D-IRFCMAC module, as depicted the overall process is shown in Fig. 6. The process area of the D-IRFCMAC model is depicted in the red block, while that of IPSO is depicted in the green block.

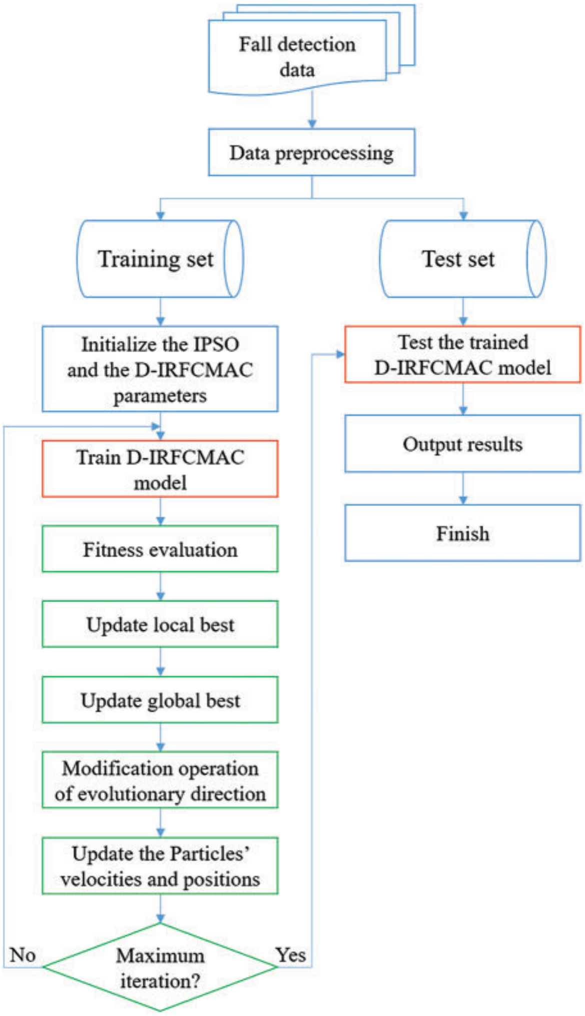
Fall detection data
Data preprocessing
Training set
Test set
Initialize the IPSO and the D-IRFCMAC parameters
Train D-IRFCMAC model
Fitness evaluation
Update local best
Update global best
Modification operation of evolutionary direction
Update the Particles' velocities and positions
Maximum iteration?
No
Yes
Test the trained D-IRFCMAC model
Output results
Finish

Figure 6: Flowchart of the D-IRFCMAC model and IPSO algorithm for fall detection

3 Experiments and Discussion

Two experiments were conducted to examine the fall detection performance of the proposed method, which involves integrating the proposed IPSO algorithm into the designed D-IRFCMAC model, and to also compare this performance with that of several existing state-of-art methods.

3.1 Experimental Data Sets

Three fall event data sets and two activity classification data sets were used to test the proposed model. The first fall event data set was collected using multiple cameras (eight cameras for each scene in each scenario) [69]; thus, this data set is called the multiple-camera fall data set. The aforementioned data set comprised 736 videos of different actions: 120 videos of falls, 192 videos of walking, 88 videos of pseudofalls, 80 videos of sitting, 32 videos of lying down, 16 videos of standing, 72 videos of crouching, 8 videos showing objects being dropped, 88 videos showing objects being carried (picked up or put down), 8 videos of taking off or hanging a coat on a coat rack, and 32 videos of sweeping. Each of the aforementioned videos had resolution of 720 × 480 pixels and a frame rate of 30 fps. A total of 80% and 20% of the aforementioned data set were used for training and testing, respectively. The second fall event data set, namely the fall detection data set (FDD) [70], comprised 192 videos of falls (forward falls, falls caused by an inappropriate sitting-down action, and falls caused by loss of balance) and 58 videos on several normal activities (walking in different directions, sitting down, standing up, crouching, housekeeping, and moving a chair). Each of these videos had resolution of 320 × 240 pixels and a frame rate of 25 fps. Each video was recorded from seven directions and positions in one of four locations: a home, a coffee room, an office, or a lecture room. Regarding the multiple-camera fall data set, 80% and 20% of the FDD were used for training and testing, respectively. The examined methods were also evaluated on the UR fall data set (URFD) [71], with 80% and 20% of the data set being used for training and testing, respectively. The URFD contains 70 videos: 30 videos on falls (e.g., forward falls and falls caused by loss of balance) and 40 videos on activities of daily living (e.g., walking, sitting down, crouching, and lying down) in various locations.

The second type tends to activity classification datasets, The UCF11 dataset [62] is an activity classification dataset that was also used to evaluate the performance of the proposed method and other methods. UCF11 is an action dataset composed of 1600 videos, including 11 action videos. The action videos included those of diving, cycling, basketball shooting and so on. Each action video only displayed one type of action. A similar dataset, UCF101 [72], was also used for evaluation. The UCF101 dataset contains 13,000 video clips from 101 activity categories. Both databases were divided into training, validation, and test sets at a 3:1:1 ratio.

All videos from the aforementioned five data sets were adjusted to resolution of 128 × 128 pixels before being processed. A total of 60 features obtained using the OF method and DFT for each video frame were used as the input in the training, validation, and testing of the D-IRFCMAC model integrated with the proposed IPSO algorithm.

3.2 Ablation Study

Ablation studies were performed to determine the effectiveness of the added IPSO for improving FDD and FURD performance. The results are presented in Table 1. With the IPSO algorithm, the proposed D-IRFCMAC model has better accuracy.

Table 1: Model performance with different mechanisms

Method	Accuracy (%)
D-IRFCMAC (without IPSO)	99.42
D-IRFCMAC (with IPSO)	99.80

3.3 Experimental Results for Fall Detection

For the three fall event data sets, the parameters of the proposed IPSO algorithm were set as follows: the initial coefficient w was set as 0.5, the cognitive and society coefficients were set as 1, the swarm size was set as 100, the number of iterations was set as 5000, and the number of fuzzy rules (hypercube cells) of the D-IRFCMAC model was set as 6 (six cells). The parameters ω, C_1, and C_2 were determined through trial and error. Table 2 presents the fall detection performance of the proposed method and existing state-of-the-art methods on the multiple-camera fall data set. The proposed method as found to outperform the other tested methods. To verify the effectiveness of the proposed method, this method was also tested on the FDD and URFD. It exhibited average classification accuracy of 99.41% and 99.32% on the aforementioned data sets, respectively. Thus, the results indicate the proposed method can accurately detect falls.

Table 2: Fall detection performance of the proposed method and different state-of-the-art methods on the multiple-camera fall data set

Method	Accuracy (%)
Silhouette area variation [30]	94.01
Full procrustes distance [22]	96.20
Mean matching cost [22]	95.40
Bounding box ratio [22]	56.60
2D vertical velocity [22]	89.70
Normalized 2D vertical velocity [22]	87.30
3D CNN [64]	99.73
Proposed method	99.80

3.4 Experimental Results for Activity Recognition

The activity recognition performance of the proposed method was investigated on the UCF11 and UCF101 data sets. The parameters of the IPSO algorithm in the activity recognition experiment were the same as those in the fall detection experiment. Tables 3 and 4 present the average testing classification accuracy of the proposed method on the UCF11 and UCF101 data sets in five cross-validation trials (93.13% and 92.19%, respectively). Various existing state-of-the-art methods were also employed on the aforementioned data sets, and their performance was compared with that of the proposed method. The proposed method was discovered to outperform the state-of-the-art-methods in activity recognition on both the UCF11 and UCF101 data sets (Table 5). The proposed method had 1.12% and 1.73% higher accuracy than the method with the second-best performance, namely visual attention–guided 3D CNN, on the UCF11 and UCF101 data sets, respectively. The superior performance of the proposed method is reflected in the confusion matrix for the UCF11 data set, which is presented in Fig. 7. Most activities in this data set were correctly classified by the proposed method.

Table 3: Average classification accuracy (%) of the proposed method on the UCF11 data set

Proposed method	Cross validation 1	Cross validation 2	Cross validation 3	Cross validation 4	Cross validation 5	Avg
Training	92.11	91.73	91.81	93.05	92.52	92.24 ± 0.55
Validation	93.61	93.03	92.12	92.86	93.03	92.93 ± 0.53
Testing	92.42	93.47	93.03	93.61	93.11	93.13 ± 0.46

Table 4: Average classification accuracy (%) of the proposed method on the UCF101 data set

Proposed method	Cross validation 1	Cross validation 2	Cross validation 3	Cross validation 4	Cross validation 5	Avg
Training	92.24	91.79	91.57	92.57	92.02	92.04 ± 0.39
Validation	92.65	92.03	92.22	92.65	93.08	92.53 ± 0.41
Testing	92.36	91.82	92.03	92.61	92.13	92.19 ± 0.30

Table 5: Average classification accuracy (%) of the proposed method and existing state-of-the-art methods on the UCF11 and UCF101 data sets

Dataset	Visual attention guided 3D CNN [64]	GoogleNet [63] with average pooling	GoogleNet with max pooling	GoogleNet with LSTM soft attention	VGGNet [44] with LSTM soft attention	Proposed method
UCF11	92.01 ± 0.35	82.56 ± 0.98	81.60 ± 1.75	84.96 ± 0.32	84.87 ± 0.53	93.13 ± 0.46
UCF101	90.46 ± 0.38	81.13 ± 0.55	81.00 ± 0.65	83.38 ± 0.39	84.27 ± 0.49	92.19 ± 0.30

To verify the stability and effectiveness of the proposed method, this method and existing state-of-the-art methods [73–77] were tested on the UR-Fall, UP-Fall, and PRECIS HAR [78] data sets, which comprise data on activities of daily living, such as walking, standing, squatting, and sitting. For each data set, 90%, 5%, and 5% of the data were used for training, validation, and testing, respectively. Table 6 presents the results obtained on the aforementioned data sets; the proposed method outperformed the existing state-of-the-art methods.

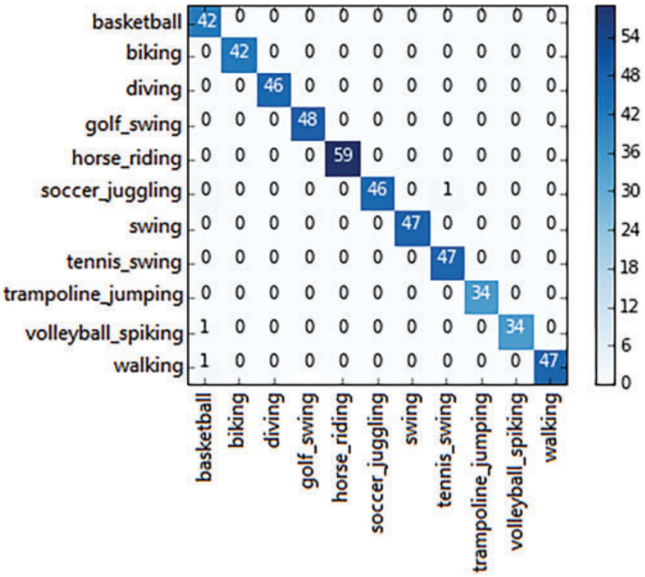

Figure 7: Confusion matrix for the proposed method when applied to the UCF11 data set

Table 6: Average classification accuracy (%) of the proposed method and existing state-of-the-art methods on the UR-Fall, UP-Fall, and PRECIS HAR data sets

UR-Fall				
Model	Accuracy	Sensitivity	F1	Specificity
Model of Harrou et al. [76]	96.66%	1.0	0.96	0.94
Framework (best result) [75]	100%	1.0	1.0	1.0
Proposed method	100%	1.0	1.0	1.0
UP-Fall				
Model of Martinez-Villasenor et al. [73]	95.93%	0.74	0.66	0.69
Framework (best result) [75]	98.62%	0.92	0.93	0.99
ARFDNet [74]	96.7%	0.967	0.966	–
Proposed method	99.25%	0.97	0.97	0.99
PRECIS HAR				
Model of Popescu et al. [77]	94.38%	–	–	–
Framework (best result) [75]	98.59%	0.95	0.97	0.97
Proposed method	99.07%	0.97	0.98	0.98

3.5 Discussion

The aforementioned experimental results indicate that the proposed method outperforms existing state-of-the-art methods in fall detection and activity recognition. The superior performance of the proposed method is attributable to the D-IRFCMAC model using both spatial and temporal (recurrent) information and the IPSO algorithm improving the exploration of the global search space by this model. The proposed method enables highly accurate fall detection to be achieved after training with a small fall event data set. However, this method can only detect a fall and not predict one. The prediction of fall events is a possible topic for future research.

4 Conclusions

This study developed a method that involves combining the proposed D-IRFCMAC model with an IPSO algorithm to achieve highly accurate fall detection and activity recognition. The designed D-IRFCMAC model processes spatial and temporal information in videos by using recurrent mechanisms. In addition, the proposed IPSO algorithm improves this model's exploration of the global search space. The integration of the IPSO algorithm into the D-IRFCMAC model improves the model's overall efficiency and effectiveness. Motion information features are crucial to fall detection and activity recognition. In the proposed method, OFTIs and FTIs are employed for feature selection and for efficiently encoding movement features. The selected features are then input to the D-IRFCMAC model integrated with the proposed IPSO algorithm to obtain accurate and robust fall detection or activity recognition results. The D-IRFCMAC model uses features obtained from OFTIs and FTIs to locate regions of interest in each video frame. The D-IRFCMAC model was discovered to exhibit excellent fall detection performance after being trained on different action-related data sets. Because simulated fall data sets are difficult to obtain, the aforementioned characteristic is a key advantage of this model. Experiments indicated that the proposed method outperforms existing state-of-the-art methods in terms of fall detection accuracy on the FDD and URFD and activity recognition accuracy on the UCF11 and UCF101 data sets. The proposed method also outperforms existing state-of-the-art methods in terms of action recognition accuracy on the UR-Fall, UP-Fall, and PRECIS HAR data sets. The UCF11 dataset had an average accuracy of 93.13%, whereas the UCF101 dataset had an average accuracy of 92.19%. The UR-Fall dataset had an accuracy of 100%, the UP-Fall dataset had an accuracy of 99.25%, and the PRECIS HAR dataset had an accuracy of 99.07%.

Acknowledgement: Not applicable.

Funding Statement: This research was supported by the National Science and Technology Council under grants NSTC 112-2221-E-320-002 and the Buddhist Tzu Chi Medical Foundation in Taiwan under Grant TCMMP 112-02-02.

Availability of Data and Materials: 1. The data that support the findings of this study are openly available in [Multiple Cameras fall dataset-MFD] at [https://www.iro.umontreal.ca/~labimage/Dataset/ (accessed on 07 Oct. 2010)]. 2. The data that support the findings of this study are openly available in [UCF101] at [https://www.crcv.ucf.edu/data/UCF101.php (accessed on 17 Oct. 2013)]. 3. The data that support the findings of this study are openly available in [UCF11] at [https://www.crcv.ucf.edu/data/UCF_YouTube_Action.php (accessed on 31 Oct. 2011)]. 4. The data that support the findings of this study are openly available in [UP-Fall] at [https://sites.google.com/up.edu.mx/har-up/ (accessed on 04 Aug. 2019)]. 5. The data that support the findings of this study are openly available in [UR-Fall] at [http://fenix.ur.edu.pl/~mkepski/ds/uf.html (accessed on 12 Jun. 2014)]. 6. The data that support the findings of this study are openly available in [PRECIS HAR] at [https://ieee-dataport.org/open-access/precis-har (accessed on 01 Feb. 2020)].

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that there are no conflicts of interest. The contents of the manuscript have been reviewed and approved by all co-authors, and there is no financial interest in reporting. We confirm that the submission is original work and is not currently under review at any other publication.

References

[1] G. Sannino, I. De Falco, and G. De Pietro, “A supervised approach to automatically extract a set of rules to support fall detection in an mHealth system,” *Appl. Soft Comput.*, vol. 34, pp. 205–216, 2015. doi: 10.1016/j.asoc.2015.04.060.

[2] L. Yang, Y. Ren, H. Hu, and B. Tian, “New fast fall detection method based on spatio-temporal context tracking of head by using depth images,” *Sensors*, vol. 15, no. 9, pp. 23004–23019, 2015. doi: 10.3390/s150923004.

[3] G. Carone and D. Costello, “Can Europe afford to grow old?” *Int. Monet. Fund Finance Dev. Mag.*, vol. 43, no. 3, 2006.

[4] J. Halter *et al.*, “Hazzard’s Geriatric Medicine and Gerontology,” in *JAMA Network*, 8th ed., McGraw Hill/Medical, 2009, vol. 302. pp. 1813.

[5] S. Lord, S. Smith, and J. Menant, “Vision and falls in older people: Risk factors and intervention strategies,” *Clin. Geriatr. Med.*, vol. 26, no. 4, pp. 569–581, 2010. doi: 10.1016/j.cger.2010.06.002.

[6] M. Okada, K. Tatani, and Y. Nakamura, “Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion,” in *Proc. ICRA*, Washington, DC, USA, 2002, pp. 1410–1415.

[7] A. J. Ijspeert, J. Nakanishi, and S. Shaal, “Learning control policies for movement imitation and movement recognition,” *Neural Inf. Process. Syst.*, vol. 15, pp. 1547–1554, 2003.

[8] J. Tani and M. Ito, “Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment,” *IEEE Trans. Syst., Man Cybern. A: Syst. Humans*, vol. 33, no. 4, pp. 481–488, 2003. doi: 10.1109/TSMCA.2003.809171.

[9] H. Kadone and Y. Nakamura, “Symbolic memory for humanoid robots using hierarchical bifurcations of attractors in nonmonotonic neural networks,” in *Proc. IROS*, Edmonton, AB, Canada, 2005, pp. 2900–2905.

[10] D. Kulic, H. Imagawa, and Y. Nakamura, “Online acquisition and visualization of motion primitives for humanoid robots,” in *Proc. RO-MAN*, Toyama, Japan, 2009, pp. 1210–1215.

[11] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura, “Active learning of confidence measure function in robot language acquisition framework,” in *Proc. IROS*, Taipei, Taiwan, 2010, pp. 1774–1779.

[12] I. Mordatch, K. Lowrey, G. Andrew, Z. Popovic, and E. Todorov, “Interactive control of diverse complex characters with neural networks,” in *Proc. NIPS*, Montreal, QC, Canada, 2015, pp. 3132–3140.

[13] P. Veltink, H. Bussmann, W. Vries, W. Martens, and R. Lummel, “Detection of static and dynamic activities using uniaxial accelerometers,” *IEEE Trans. Rehab. Eng.*, vol. 4, no. 4, pp. 375–385, 1996. doi: 10.1109/86.547939.

[14] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa, “Comparison of low-complexity fall detection algorithms for body attached accelerometers,” *Gait Posture*, vol. 28, no. 2, pp. 285–291, 2008. doi: 10.1016/j.gaitpost.2008.01.003.

[15] Y. Li, K. Ho, and M. Popescu, “A microphone array system for automatic fall detection,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 2, pp. 1291–1301, 2012. doi: 10.1109/TBME.2012.2186449.

[16] 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, 2009. doi: 10.1109/TBME.2009.2030171.

[17] L. Chen, R. Li, H. Zhang, L. Tian, and N. Chen, “Intelligent fall detection method based on accelerometer data from a wrist-worn smart watch,” *Measurement*, vol. 140, no. 2, pp. 215–226, 2019. doi: 10.1016/j.measurement.2019.03.079.

[18] 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, 2019. doi: 10.1109/MNET.001.1900100.

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

[20] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Fall detection from human shape and motion history using video surveillance," in *Proc. AINAW*, Niagara Falls, ON, Canada, 2007, pp. 875–880.

[21] C. Rougier and J. Meunier, "3D head trajectory using a single camera," *Int. J. Future Gener. Commun. Netw., Invited Paper Special Issue Image Signal Process.*, vol. 3, no. 4, pp. 43–54, 2010. doi: 10.1007/978-3-642-13681-8.

[22] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Robust video surveillance for fall detection based on human shape deformation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 611–622, 2011. doi: 10.1109/TCSVT.2011.2129370.

[23] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier, "Fall detection with multiple cameras: An occlusion-resistant method based on 3-D silhouette vertical distribution," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 2, pp. 290–300, 2011. doi: 10.1109/TITB.2010.2087385.

[24] C. Juang and C. Chang, "Human body posture classification by a neural fuzzy network and home care system application," *IEEE Trans. Syst., Man, Cybern. A: Syst. Humans*, vol. 37, no. 6, pp. 984–994, 2007. doi: 10.1109/TSMCA.2007.897609.

[25] C. Liu, C. Lee, and P. Lin, "A fall detection system using k-nearest neighbor classifier," *Expert. Syst. Appl.*, vol. 37, no. 10, pp. 7174–7181, 2010. doi: 10.1016/j.eswa.2010.04.014.

[26] T. Lee and A. Mihailidis, "An intelligent emergency response system: Preliminary development and testing of automated fall detection," *J. Telemed. Telecare*, vol. 11, no. 4, pp. 194–198, 2005. doi: 10.1258/1357633054068946.

[27] M. Belshaw, B. Taati, J. Snoek, and A. Mihailidis, "Towards a single sensor passive solution for automated fall detection," in *Proc. 2011 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2011, pp. 1773–1776.

[28] J. Jefn Gracewell and S. Pavalarajan, "Fall detection based on posture classification for smart home environment," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 3, pp. 3581–3588, 2019. doi: 10.1007/s12652-019-01600-y.

[29] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Proc. IEEE EMBS*, New York, NY, USA, 2006, pp. 6388–6391.

[30] N. Thome, S. Miguet, and S. Ambellouis, "A real-time, multiview fall detection system: A LHMM-based approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1522–1532, 2008. doi: 10.1109/TCSVT.2008.2005606.

[31] S. McKenna and H. Nait-Charif, "Summarising contextual activity and detecting unusual inactivity in a supportive home environment," *J. Pattern Anal. Appl.*, vol. 7, no. 4, pp. 386–401, 2004. doi: 10.1007/s10044-004-0233-2.

[32] D. Anderson, R. Luke, J. Keller, M. Skubic, M. Rantz and M. Aud, "Linguistic summarization of video for fall detection using voxel person and fuzzy logic," *Comput. Vis. Image Underst.*, vol. 113, no. 1, pp. 80–89, 2009. doi: 10.1016/j.cviu.2008.07.006.

[33] R. Espinosa, H. Ponce, S. Gutierrez, L. Martinez-Villasenor, J. Brieva and E. Moya-Albor, "A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the up-fall detection dataset," *Comput. Biol. Med.*, vol. 115, no. 11, pp. 103520, 2019. doi: 10.1016/j.compbiomed.2019.103520.

[34] S. Maldonado-Bascon, C. Iglesias-Iglesias, P. Martin-Martin, and S. Lafuente-Arroyo, "Fallen people detection capabilities using assistive robot," *Electronics*, vol. 8, no. 9, pp. 1–20, 2019. doi: 10.3390/electronics8090915.

[35] A. Biswas and B. Dey, "A literature review of current vision based fall detection methods," *Adv. Commun., Devices Netw. Lecture Notes Electr. Eng.*, pp. 411–421, 2020. doi: 10.1007/978-981-15-4932-8_46.

[36] P. V. C. Souza, A. J. Guimaraes, V. S. Araujo, L. O. Batista, and T. S. Rezende, "An interpretable machine learning model for human fall detection systems using hybrid intelligent models," in *Challenges and Trends in Multimodal Fall Detection for Healthcare*. Springer, 2020, vol. 273, pp. 181–205, doi:10.1007/978-3-030-38748-8_8.

[37] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva and E. Moya-Albor, "Application of convolutional neural networks for fall detection using multiple cameras," in *Challenges and Trends in Multimodal Fall Detection for Healthcare*. Springer, 2020, pp. 97–120, doi:10.1007/978-3-030-38748-8_5.

[38] M. Buzzelli, A. Albe, and G. Ciocca, "A vision-based system for monitoring elderly people at home," *Appl. Sci.*, vol. 10, no. 1, pp. 1–25, 2020. doi: 10.3390/app10010374.

[39] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, no. 6, pp. 501–513, 2020. doi: 10.1016/j.neucom.2018.10.104.

[40] H. Zhu, S. Samtani, H. Chen, and J. F. Nunamaker Jr, "Human identification for activities of daily living: A deep transfer learning approach," *J. Manag. Inf. Syst.*, vol. 37, no. 2, pp. 457–483, 2020. doi: 10.1080/07421222.2020.1759961.

[41] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. doi: 10.1038/nature14539.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. NIPS*, 2014, pp. 3104–3112.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017. doi: 10.1145/3065386.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

[45] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3D human pose estimation: Motion to the rescue," in *Proc. NeurIPS*, Vancouver, BC, Canada, 2019.

[46] Y. M. Galvão, J. Ferreira, V. A. Albuquerque, P. Barros, and B. J. T. Fernandes, "A multimodal approach using deep learning for fall detection," *Expert. Syst. Appl.*, vol. 168, no. 15, pp. 114226, 2021. doi: 10.1016/j.eswa.2020.114226.

[47] T. -H. Tsai and C. -W. Hsu, "Implementation of fall detection system based on 3D skeleton for deep learning technique," *IEEE Access*, vol. 7, pp. 153049–153059, 2019. doi: 10.1109/ACCESS.2019.2947518.

[48] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, no. 1, pp. 3–11, 2019. doi: 10.1016/j.patrec.2018.02.010.

[49] A. Dhillon and G. K. Verma, "Convolutional neural network: A review of models, methodologies and applications to object detection," *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, 2020. doi: 10.1007/s13748-019-00203-0.

[50] H. Sadreazami, M. Bolic, and S. Rajan, "Fall detection using standoff radar-based sensing and deep convolutional neural network," *IEEE Trans. Circuits Syst. II: Express Briefs*, vol. 67, no. 1, pp. 197–201, 2019. doi: 10.1109/TCSII.2019.2904498.

[51] J. He, Z. Zhang, X. Wang, and S. Yang, "A low power fall sensing technology based on FD-CNN," *IEEE Sens. J.*, vol. 19, no. 13, pp. 5110–5118, 2019. doi: 10.1109/JSEN.2019.2903482.

[52] I. Kiprijanovska, H. Gjoreski, and M. Gams, "Detection of gait abnormalities for fall risk assessment using wrist-worn inertial sensors and deep learning," *Sensors*, vol. 20, no. 18, pp. 5373, 2020. doi: 10.3390/s20185373.

[53] M. Musci, D. D. Martini, N. Blago, T. Facchinetti, and M. Piastra, "Online fall detection using recurrent neural networks on smart wearable devices," *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 3, pp. 1276–1289, 2020. doi: 10.1109/TETC.2020.3027454.

[54] L. Yao, W. Yang, and W. Huang, "A fall detection method based on a joint motion map using double convolutional neural networks," *Multimed. Tools Appl.*, vol. 81, no. 4, pp. 1–18, 2020. doi: 10.1007/s11042-020-09181-1.

[55] Y. Du, Y. Fu, and L. Wang, "Skeleton based action recognition with convolutional neural network," in *Proc. ACPR*, 2015, pp. 579–583.

[56] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition," in *Proc. CVPR*, Honolulu, HI, USA, 2017, pp. 4570–4579.

[57] Y. Hou, Z. Li, P. Wang, and W. Li, "Skeleton optical spectra-based action recognition using convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 807–811, 2018. doi: 10.1109/TCSVT.2016.2628339.

[58] P. Wang, W. Li, C. Li, and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks," in *Proc. ACM Multimedia*, 2016.

[59] J. Koushik, "Understanding convolutional neural networks," in *Proc. ICET*, Antalya, Turkey, 2016.

[60] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, no. 11, pp. 354–377, 2018. doi: 10.1016/j.patcog.2017.10.013.

[61] E. Park, X. Han, T. L. Berg, and A. C. Berg, "Combining multiple sources of knowledge in deep CNNs for action recognition," in *Proc. WACV*, 2016, pp. 1–8.

[62] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," in *Proc. ICLR*, Canada, 2016, pp. 1–11.

[63] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, 2015, pp. 1–9.

[64] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, 2019. doi: 10.1109/JBHI.2018.2808281.

[65] K. Mohajeri, M. Zakizadeh, B. Moaveni, and M. Teshnehlab, "Fuzzy CMAC structures," in *Proc. FUZZ-IEEE*, Jeju, Republic of Korea, 2009, pp. 2126–2131.

[66] K. F. Sim and K. Sundaraj, "Human motion tracking of athlete using optical flow & artificial markers," in *Proc. ICIAS*, Kuala Lumpur, Malaysia, 2010, pp. 1–4.

[67] B. Sharma, K. S. Venkatesh, and A. Mukerjee, "Fourier shape-frequency words for actions," in *Proc. ICIIP*, Shimla, India, 2011, pp. 1–6.

[68] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN*, Perth, WA, Australia, 1995, pp. 1942–1948.

[69] E. Auvinet, C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Multiple camera fall dataset," *Techn. Report*, pp. 1–24, 2010.

[70] I. ICharfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust svm based fall detection solution," in *Proc. SITIS*, Sorrento, Italy, 2012, pp. 218–224.

[71] 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, 2014. doi: 10.1016/j.cmpb.2014.09.005.

[72] S. Khurram, Z. Amir Roshan, and S. Mubarak, "UCF101: A dataset of 101 human actions classes from videos in the wild," in *Proc. CVPR*, Orlando, FL, USA, 2012, pp. 1–7.

[73] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez and C. Peñafort-Asturiano, "UP-Fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, pp. 1988, 2019. doi: 10.3390/s19091988.

[74] S. K. Yadav, A. Luthra, K. Tiwari, H. M. Pandey, and S. A. Akbar, "ARFDNet: An efficient activity recognition & fall detection system using latent feature pooling," *Knowl.-Based Syst.*, vol. 239, no. 7, pp. 107948, 2022. doi: 10.1016/j.knosys.2021.107948.

[75] Y. M. Galvão, L. Portela, J. Ferreira, P. Barros, R. A. D. Fagundes and B. J. T. Fernandes, "A framework for anomaly identification applied on fall detection," *IEEE Access*, vol. 9, pp. 77264–77274, 2021. doi: 10.1109/ACCESS.2021.3083064.

[76] 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. doi: 10.1109/ACCESS.2019.2936320.

[77] A. C. Popescu, I. Mocanu, and B. Cramariuc, "Fusion mechanisms for human activity recognition using automated machine learning," *IEEE Access*, vol. 8, pp. 143996–144014, 2020. doi: 10.1109/ACCESS.2020.3013406.

[78] A. Popescu, I. Mocanu, and B. Cramariuc, *Precis har*. Accessed: Feb. 01, 2020. [Online]. Available: https://ieee-dataport.org/open-access/precis-har