




Article

Synergistic Integration of Skeletal Kinematic Features for Vision-Based Fall Detection

Anitha Rani Inturi¹, Vazhora Malayil Manikandan^{1,*} , Mahamkali Naveen Kumar¹, Shuihua Wang² 
and Yudong Zhang^{2,*} 

¹ Department of Computer Science and Engineering, SRM University—AP, Mangalagiri 522240, India; anitha_rani@srmmap.edu.in (A.R.I.); naveenkumar.m@srmmap.edu.in (M.N.K.)

² School of Computing and Mathematical Sciences, University of Leicester, Leicester LE1 7RH, UK; sw546@le.ac.uk

* Correspondence: manikandan.v@srmmap.edu.in (V.M.M.); yudong.zhang@le.ac.uk (Y.Z.)

Abstract: According to the World Health Organisation, falling is a major health problem with potentially fatal implications. Each year, thousands of people die as a result of falls, with seniors making up 80% of these fatalities. The automatic detection of falls may reduce the severity of the consequences. Our study focuses on developing a vision-based fall detection system. Our work proposes a new feature descriptor that results in a new fall detection framework. The body geometry of the subject is analyzed and patterns that help to distinguish falls from non-fall activities are identified in our proposed method. An AlphaPose network is employed to identify 17 keypoints on the human skeleton. Thirteen keypoints are used in our study, and we compute two additional keypoints. These 15 keypoints are divided into five segments, each of which consists of a group of three non-collinear points. These five segments represent the left hand, right hand, left leg, right leg and craniocaudal section. A novel feature descriptor is generated by extracting the distances from the segmented parts, angles within the segmented parts and the angle of inclination for every segmented part. As a result, we may extract three features from each segment, giving us 15 features per frame that preserve spatial information. To capture temporal dynamics, the extracted spatial features are arranged in the temporal sequence. As a result, the feature descriptor in the proposed approach preserves the spatio-temporal dynamics. Thus, a feature descriptor of size $[m \times 15]$ is formed where m is the number of frames. To recognize fall patterns, machine learning approaches such as decision trees, random forests, and gradient boost are applied to the feature descriptor. Our system was evaluated on the UPfall dataset, which is a benchmark dataset. It has shown very good performance compared to the state-of-the-art approaches.

Keywords: fall detection; video analysis; vision-based human activity recognition; fall prevention; ambient intelligence; assistive technology; signal processing; real-time monitoring; risk assessment



Citation: Inturi, A.R.; Manikandan, V.M.; Kumar, M.N.; Wang, S.; Zhang, Y. Synergistic Integration of Skeletal Kinematic Features for Vision-Based Fall Detection. *Sensors* **2023**, *23*, 6283. <https://doi.org/10.3390/s23146283>

Academic Editor: Mario Munoz-Organero

Received: 26 May 2023

Revised: 29 June 2023

Accepted: 4 July 2023

Published: 10 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Machine learning algorithms, in combination with video processing, analyze the videos and identify the human activity. This may entail activities like scene segmentation, object detection, object tracking, human activity recognition and others. The information contained in the video is automatically extracted with the power of machine learning algorithms. The extracted video data are used for various applications such as surveillance, medicine, criminal activity recognition, assisted living, military and many more. Vision-based fall detection is one such application of video processing that alerts people in case of a fall and has a significant demand in assisted living environments. Falls are a serious risk, especially for adults over 60 owing to their cognitive decline and cell degeneration [1]. Age-related physiological, psychological, neurological and biological changes in elders are the inherent factors for falling. In general, falls occur due to multiple reasons such as

dizziness, chronic illness, vision impairment, gait or other environmental conditions such as slippery floors, highland terrain, outfit etc.

According to centers for disease control and prevention, falling is a major cause of fatality, and the fatality rate is increasing rapidly worldwide, especially among senior citizens. Every year, there are 36 million recorded falls, 3 million of which require emergency room treatment, 95% of which result in hip fractures, and 32,000 deaths are reported [2]. The consequences of falls, which may be social, physical, or psychological, are depicted in Figure 1.

This leads to an increased focus on researching fall detection, highlighting the significance of identifying falls. Consequently, the research community predominantly uses multivariate data in their efforts to develop fall detection methods.

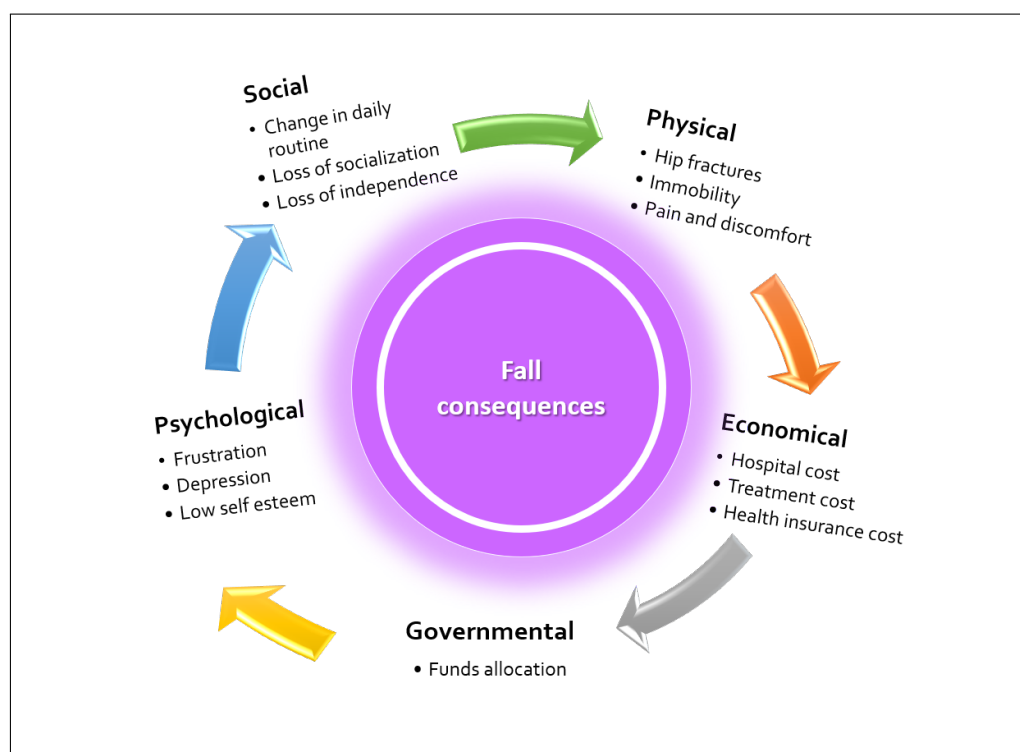


Figure 1. Major consequences of falls.

Falls can be detected using both sensor-based [3–5] and computer vision-based [6–8] technologies. Sensor-based technologies rely on input from wearable sensors worn by the subject. Computer vision-based technologies analyze the data obtained from a monitoring camera. While each method has particular drawbacks, they are both technically capable of identifying falls. Figure 2 depicts the general workflow of a fall detection system. In the model distribution phase, A1, A2, A3, A4 represent the four quadrants where data may be distributed. The feature vector is also a collection of different features from *frame 1* to *frame n*.

Input sources for sensor-based systems typically include accelerometers, magnetometers, and gyroscopes, which are usually embedded in smartwatches, smartphones, and other wearable gadgets. These systems extract features such as speed, velocity, field orientation, variation of the magnetic field, and angular momentum. Contrarily, computer vision-based systems often employ Kinect or depth cameras to record input data in the form of RGB or depth images. The characteristics that can be utilized to detect falls are extracted by these systems after they have analyzed the images or frames. These features are as follows:

- i Local features such as the colour, texture, and intensity of the image
- ii Global features such as the image silhouette, edges, spatial points

iii Depth features, that extract the depth information of the image.

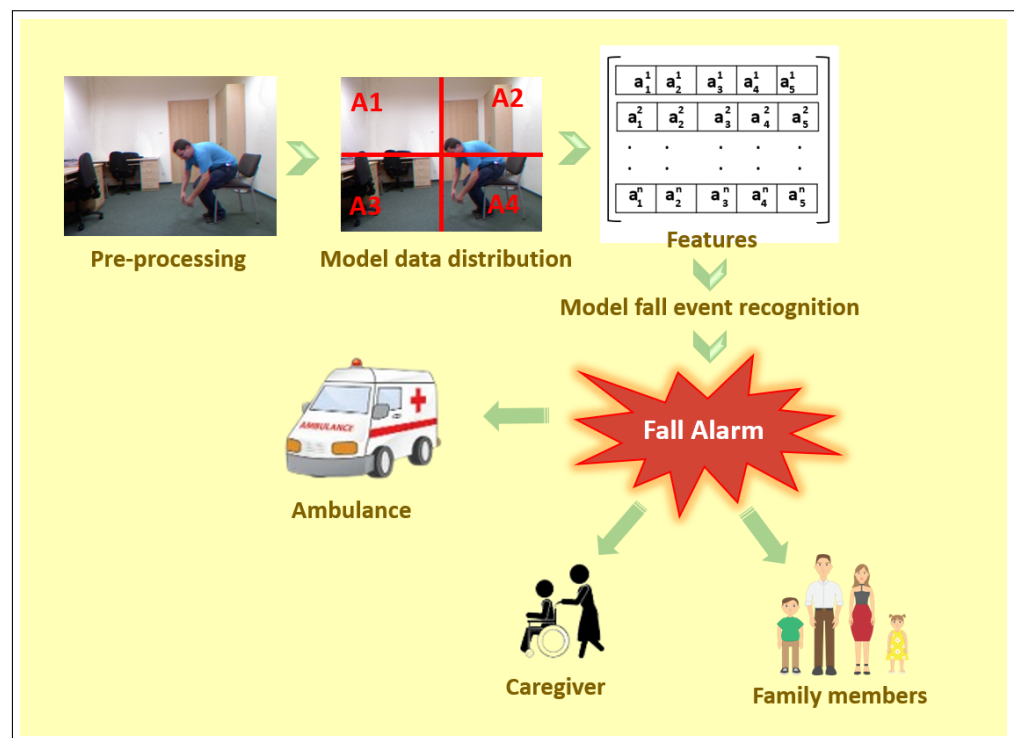


Figure 2. A general fall detection system.

Our main contribution lies in the design of a new feature vector that results in a new fall detection framework. In the proposed approach, we defined a set of new informative features through the following steps:

1. Segmenting the human skeleton into five sections (left hand, right hand, left leg, right leg, and a craniocaudal section).
2. Extracting the distances from the segmented parts (Spatial domain).
3. Calculating the angles within the segments (Spatial domain).
4. Calculating the angle of inclination for every segment (Spatial domain).
5. To capture temporal dynamics, the extracted spatial features are arranged in the temporal sequence. As a result, the feature descriptor in the proposed approach preserves the spatiotemporal dynamics.
6. We have achieved very good performance compared to the state-of-the-art approaches.
7. The performance of our method is evaluated on the UPfall dataset.

In this paper, we proposed a new fall detection system. A comprehensive description of the entire process is given below.

- By extracting m frames from the UPfall dataset using Equation (1).
- The AlphaPose pre-trained network was applied to retrieve 17 keypoints of the subject from every frame.
- The missing keypoints were computed using Equations (2) and (3).
- From the 17 keypoints, 13 keypoints were retrieved and two additional keypoints were computed. In total, 15 keypoints were used for the process.
- These 15 keypoints were used to segment the human skeleton into five sections: left hand, right hand, left leg, right leg, and a craniocaudal section.
- Three features were extracted from each section. Specifically, the length of the section (distances), the angle made by the points that depict a section and the angle made by every section with the x-axis.

- As a result, 15 features were retrieved from each frame. Each feature was represented by a column.
- Thus, we extracted characteristics from m frames and aligned them column-wise so that each row represents one video frame in a temporal sequence.
- Hence a feature descriptor was formed. This descriptor was the input to the machine learning algorithms.
- Also, to preserve the ground truth, every video was labelled as a fall or not a fall.
- The accuracy of the machine learning algorithms was computed using the ground truth data.

The structure of the paper is as follows: Related work is discussed in Section 2, the proposed methodology is presented in Section 3, experimental results are outlined in Section 4, and finally, Section 5 concludes the paper and proposes areas for future research.

2. Literature Review

In this section, we are discussing a few existing fall detection approaches. In the field of computer science, computer vision [9–11] and machine learning [12,13] are widely used for solving various problems, such as sentiment analysis [14], speech recognition [15,16], and image processing [17–19]. Combining computer vision, machine learning, and deep learning has proven to be very effective in resolving a multitude of problems, including human activity recognition (HAR). HAR involves recognizing a person's actions, such as jumping, running, laying, bending, walking, and sitting [20–22]. Detecting falls is a crucial aspect of activity recognition, but it can be a difficult task due to the subtle differences between falls and other activities such as bending or lying down. There are several approaches to fall detection, and we will discuss some of the key ones below.

2.1. Sensor-Based Technology

Fall detection systems process the data that are generated by sensor devices. Sensor devices are more useful in outdoor environments. The authors in their work [23] designed a wearable sensor that detects falls and sends the individual's aid request and position information to carers using a quaternion algorithm. This system applies the algorithm to acceleration data. Their approach, however, might cause false alarms because they might mistake sleeping or lying down for a fall. A bi-axial gyroscope sensor array-based system with three thresholds applied to changes in trunk angle, angular acceleration, and velocity was suggested by the authors of [24]. In controlled studies on young people, their method produced 100% specificity.

Another work proposed in [25] created a straightforward smart carpet-based system that detects falls using piezoresistive pressure sensors. Their approach acquired a sensitivity and specificity of 88.8% and 94.9%, respectively. A smart floor design that retrieves pressure images for fall detection using a sophisticated back-projection method was proposed in [26]. However, implementing this technology in practical applications is expensive. A new long-short-term memory architecture (LSTM) called cerebral LSTM was introduced by [27] on wearable devices to detect falls. A millimetre wave signal-based real-time fall detection system called mmFall was proposed by [28]. It achieved high accuracy and low computational complexity by extracting signal fluctuation related to human activity using spatial-temporal processing and developing a light convolutional neural network.

2.2. Vision-Based Technology

Computer vision systems analyze video feeds from cameras placed in a room to track the subject's movements to identify falls. Every time there is a change in the subject's movement, machine learning or computer vision algorithms are used to analyze the patterns and detect a fall. To detect falls, multiple cameras are utilized to record various types of images.

RGB Images: Red, green, and blue (RGB) images include three color channels and can be used to detect falls by examining alterations in the subject's position and orientation

inside the image. A fall can be categorized as a divergence from usual patterns by tracking the movement of keypoints or additional factors.

In [29], the importance of simultaneously tracking the head and torso regions for fall detection is discussed. Geometrical features are extracted from elliptical contours applied to these regions, and a CNN is used to analyze the correlation of these features and detect falls. However, since falls can be similar to a lying position, tracking only the head and torso may result in false positives.

To preserve the privacy of the subject, Reference [30] uses background subtraction to retrieve the human silhouette, which is then stacked and analyzed for binary motion. Similarly, in [31], human silhouettes are extracted instead of raw images. A pixel-wise multi-scaling skip connection network is used to extract the silhouette, which is then analyzed using convLSTM for fall detection. The authors report an excellent f1-score of 97.68%. However, the human silhouettes may include shadows, which could potentially lead to misclassification of falls.

The work proposed in [32] extracted the skeleton keypoints using the AlphaPose network and applied random forest, support vector machines, multi-layer perceptron, and k-nearest neighbours algorithms. They achieved an accuracy of 97.5%. The authors in [33] proposed a fall detection method by tracking the keypoints in the successive frames. In their approach, they computed the distance and angle between the same keypoints in successive frames. Their system was evaluated on the URfall dataset and they achieved an accuracy of 97%. An overview of existing approaches, the methods adopted, and the machine learning algorithms used are given in Table 1.

Table 1. An overview of existing vision-based approaches.

Approach	Method	Algorithm
[29]	Head and Torso tracking	Convolutional neural network
[30]	Stacked human silhouette	Binary motion is observed
[31]	Pixel-wise multi scaling skip connection network	Conv LSTM
[32]	Skeleton keypoints using AlphaPose	Random forest, Support vector machine, k-Nearest neighbors, Multi layer perceptron
[33]	Distance and angle between same keypoints in successive frames.	Random forest, Support vector machine, k-Nearest neighbors, Multi layer perceptron

Depth Images: Depth images, which measure the distance between objects in the frame and allow for the monitoring of changes in an object's depth, are used to detect falls. In [34], skeletal information is tracked in depth images to enhance privacy preservation. A fall is presumed to have occurred if the head's motion history images show greater variance in head position over time. But relying only on the head position can result in false positives, for instance when the subject is simply lying down.

A fall detection method that utilizes both RGB and depth images has been proposed in [35]. The RGB images are used to identify feature points as either static or dynamic, while the depth images are clustered using k-means. After that, the RGB features are projected onto the depth images and categorized as static or dynamic. This classification is then used for object tracking. This object tracking can then be utilized for fall detection.

3. Proposed Approach

Fall detection systems must respond quickly to prevent significant consequences. Our goal is to improve system performance and minimize response time. To achieve this, we propose a fall detection architecture that reduces computational complexity. We found that computer vision-based approaches are more accurate than sensor-based approaches,

as it is not feasible for individuals to wear sensing devices at all times. Therefore, we adopted a computer vision-based approach and proposed features that can accurately distinguish falling from non-fall activities by analyzing the 2D representation of the human skeleton. The use of 2D representation also reduces computational complexity, improving system efficiency.

3.1. Dataset

The performance of our fall detection system was evaluated using the UP Fall dataset [36]. This dataset is a collection of multimodal data. It contains data obtained from sensors as well as data obtained from two cameras. The setup consists of frontal and lateral cameras. A total of 17 young and healthy subjects performed eleven activities that are a combination of five types of falls and six daily living activities. Three trials were conducted on every subject.

3.2. Pre-Processing

The dataset has been fine-tuned to collect m frames from every video using the following Equation (1)

$$\text{skip_value} = \frac{n}{m} \quad (1)$$

where n = total number of frames; m = number of frames required.

The regional multi-person pose estimation network (RMPE) [37] or the pre-trained AlphaPose network, which extracts the keypoints of the human skeleton, is utilized to process the fine-tuned dataset. These keypoints indicate the locations of the joints in the human body. The COCO dataset [38] was used to train the AlphaPose network. Figure 3 illustrates the architecture of AlphaPose, which consists of three components.

- i The SSTN + SPPE method, which combines a symmetric spatial transformer network (SSTN) with a parallel single-person pose estimation (SPPE) method, is used to create pose recommendations from human bounding boxes. SPPE is used in parallel with SSTN to control the output when it is unable to provide the desired pose.
- ii A technique known as parametric pose non-max-suppression (NMS) is used to find similar poses in the dataset and choose the one with the highest score to prevent redundant poses. The dataset is streamlined in this way, and only the poses that are most accurate and pertinent are kept.
- iii The system's accuracy and robustness are increased using a technique called the pose-guided proposals generator. It operates by locating the human object in the scene and suggesting several bounding boxes that correspond to the various stances the person might strike. These bounding boxes are created using computer vision techniques that estimate the positions of the human joints. This method enables the system to record a large range of potential poses and motions.

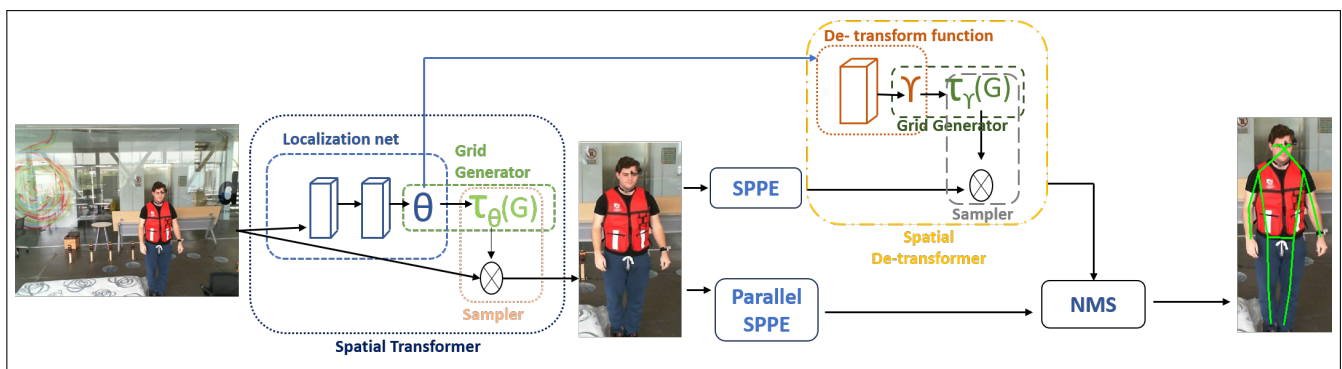


Figure 3. AlphaPose network.

We used the AlphaPose architecture to locate spatial coordinates of human joint positions on the human skeleton in the video data from the UP-FALL detection dataset. To determine the locations of the joints in the human body, this architecture uses a model that generates 17 key points. As they represent the critical body positions that can indicate a fall, these key points serve as crucial information for fall detection. We can examine the human body movements and assess if those movements are related to falls or not. However, due to the position of the subject in certain frames, some key points were found to be missing. To address this issue, we employed an imputation method that calculates the average to estimate the missing key points, as shown in Equations (2) and (3).

$$F_i K_n(x) = \frac{F_{i-1} K_n(x) + F_{i+1} K_n(x)}{2} \quad (2)$$

$$F_i K_n(y) = \frac{F_{i-1} K_n(y) + F_{i+1} K_n(y)}{2} \quad (3)$$

where, F_i is the i -th frame and K_n is the n -th key point. Hence, $F_i K_n(x)$ represents the x coordinate of the n -th key point in i -th frame and $F_i K_n(y)$ represents the y coordinate of the n -th key point in i -th frame. F_{i-1} , F_{i+1} are the earlier and later frames of F_i , respectively.

3.3. Assumptions

Generally, human motion can be observed through the movements of the limbs. Therefore, it is reasonable to assume that tracking a person's limbs can provide additional information about the action being carried out. Hence, we extracted and analyzed the spatial positions of the four limbs. It was also observed that during daily living activities such as walking, sitting, and standing, the craniocaudal axis, which is the segment joining the head and toe, is usually perpendicular to the ground or the x -axis. However, when a person falls, the angle between the craniocaudal axis and the ground (x -axis) is close to zero. To create more precise features using these presumptions, we considered the four limbs and the craniocaudal axis as five individual components and crafted features such as the distance between the points on a component, the angle made by the component with the x -axis, and the angle formed at the center point of each component. From the 17 keypoints obtained through the AlphaPose network, we selected 13 keypoints and 2 other keypoints were computed to generate a 15-keypoint model. The 15-keypoint models shown in Figure 4 were considered for feature extraction.

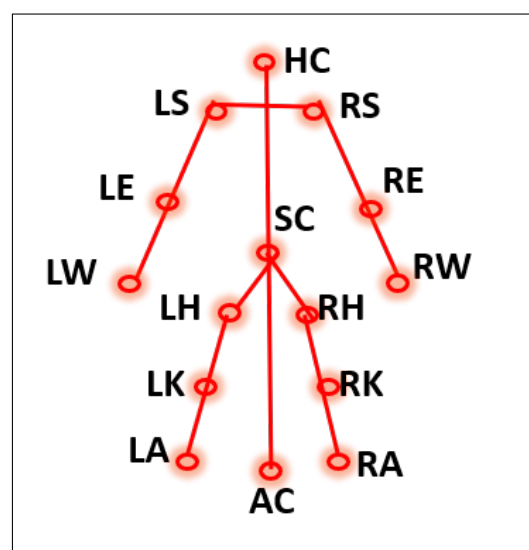


Figure 4. Keypoints considered.

An approximation of the position of the hip/spine center (SC) is calculated from the spatial positions of LH and RH using Equation (4).

$$x_{SC} \approx \frac{x_{LH} + x_{RH}}{2}, \quad y_{SC} \approx \frac{y_{LH} + y_{RH}}{2} \quad (4)$$

Similarly, the ankle center is calculated from the spatial positions of LA and RA using Equation (5).

$$x_{AC} \approx \frac{x_{LA} + x_{RA}}{2}, \quad y_{AC} \approx \frac{y_{LA} + y_{RA}}{2} \quad (5)$$

The five components used in the analysis are the left hand, left leg, right hand, right leg, and the craniocaudal segment. The left-hand component is defined by the non-collinear points of the left shoulder (LS), left elbow (LE), and left wrist (LW) represented by their Cartesian coordinates. Similarly, the non-collinear points of the right shoulder (RS), right elbow (RE), and right wrist (RW) represent the right-hand component. The left leg component is defined by the non-collinear points of the left hip (LH), left knee (LK), and left ankle (LA), while the right leg component is defined by the non-collinear points of the right ankle (RA), right knee (RK), and right hip (RH). The craniocaudal axis component is represented by the center of the head (HC), hip/spine center (SC), and the midpoint of the ankles (AC). The Equation (6) entails these five segments.

$$\{LS, LE, LW\} \{RS, RE, RW\} \{LH, LK, LA\} \{RH, RK, RA\} \{HC, SC, AC\} \quad (6)$$

3.4. Kinetic Vector Calculation

The final set of features that characterizes the movement patterns of the individual in the video is determined by measuring the distance between the initial and final points in each of the five components, as well as the angle formed between each component and the x-axis. Additionally, the angle formed at the center point of each component is also computed to enhance the precision of the features. The representation of features is shown in Figures 5–7.

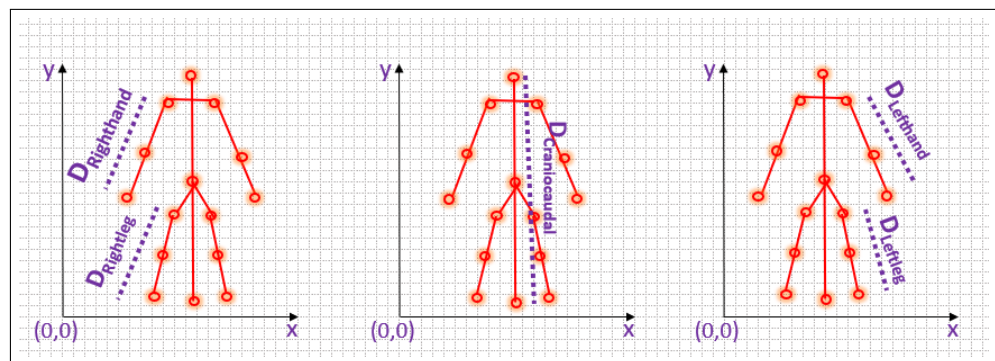


Figure 5. Distance calculation on all segments.

Distance between the non-collinear points is calculated using the Equations (7)–(11).

For left hand

$$D_{Lefthand} = ||LS - LW|| \quad (7)$$

For right hand

$$D_{Righthand} = ||RS - RW|| \quad (8)$$

For left leg

$$D_{Leftleg} = ||LH - LA|| \quad (9)$$

For right leg

$$D_{Rightleg} = ||RH - RA|| \quad (10)$$

For craniocaudal line

$$D_{craniocaudalline} = ||HC - AC|| \tag{11}$$

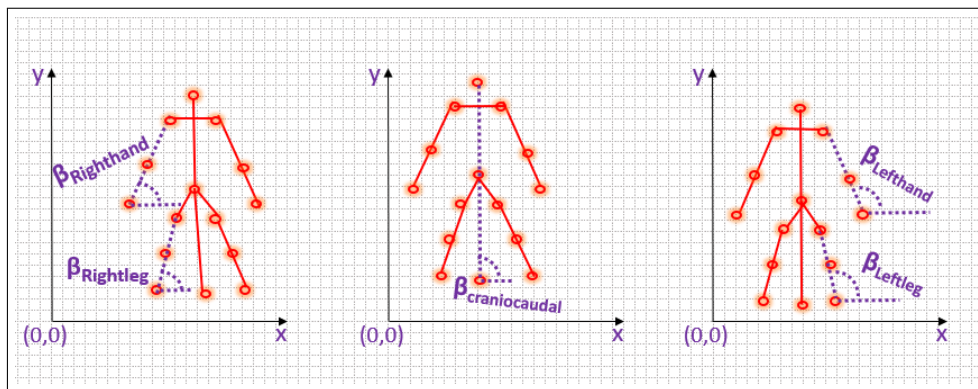


Figure 6. Angle of inclination for all segments.

The angle of inclination β between the non-collinear points is calculated using the Equations (12)–(16). The reason to use atan while calculating the beta angle is that values in the second quadrant result in negative numbers and values in the third quadrant result in positive numbers when atan function is applied. This variation from positive to negative and again to positive offers a chance to identify some deviation in a regular pattern which is consistently either positive or negative. Identifying this deviation is more crucial for an angle of inclination rather than the angle between the non-collinear points of a segment.

For left hand

$$\beta_{Lefthand} = \tan^{-1} \left(\frac{y_{LS} - y_{LW}}{x_{LS} - x_{LW}} \right) * \frac{180}{\pi} \tag{12}$$

For right hand

$$\beta_{Righthand} = \tan^{-1} \left(\frac{y_{RS} - y_{RW}}{x_{RS} - x_{RW}} \right) * \frac{180}{\pi} \tag{13}$$

For left leg

$$\beta_{Leftleg} = \tan^{-1} \left(\frac{y_{LH} - y_{LA}}{x_{LH} - x_{LA}} \right) * \frac{180}{\pi} \tag{14}$$

For right leg

$$\beta_{Rightleg} = \tan^{-1} \left(\frac{y_{RH} - y_{RA}}{x_{RH} - x_{RA}} \right) * \frac{180}{\pi} \tag{15}$$

For craniocaudal line

$$\beta_{craniocaudalline} = \tan^{-1} \left(\frac{y_{HC} - y_{AC}}{x_{HC} - x_{AC}} \right) * \frac{180}{\pi} \tag{16}$$

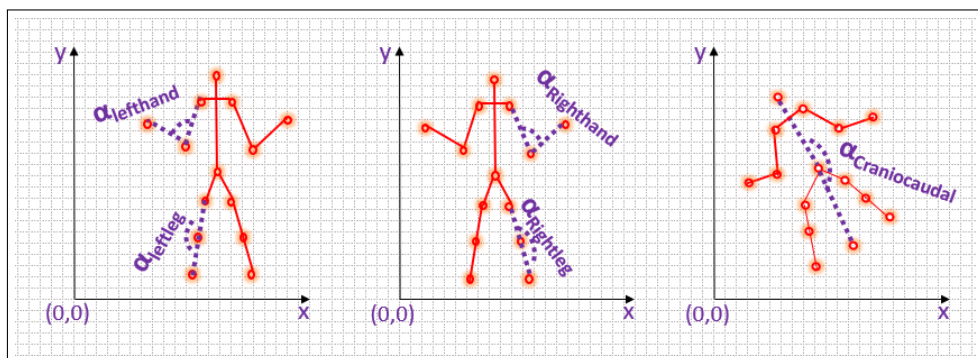


Figure 7. Angle between the non-collinear points of every segment.

Base angle α between the non-collinear points is calculated using the Equations (17)–(21).
For left hand

$$\alpha_{Lefthand} = (\text{atan2}((x_{LS} - x_{LE}), (y_{LS} - y_{LE})) - \text{atan2}((x_{LE} - x_{LW}), (y_{LE} - y_{LW}))) * \frac{180}{\pi} \quad (17)$$

For right hand

$$\alpha_{Righthand} = (\text{atan2}((x_{RS} - x_{RE}), (y_{RS} - y_{RE})) - \text{atan2}((x_{RE} - x_{RW}), (y_{RE} - y_{RW}))) * \frac{180}{\pi} \quad (18)$$

For left leg

$$\alpha_{Leftleg} = (\text{atan2}((x_{LH} - x_{LK}), (y_{LH} - y_{LK})) - \text{atan2}((x_{LK} - x_{LA}), (y_{LK} - y_{LA}))) * \frac{180}{\pi} \quad (19)$$

For right leg

$$\alpha_{Rightleg} = (\text{atan2}((x_{RH} - x_{RK}), (y_{RH} - y_{RK})) - \text{atan2}((x_{RK} - x_{RA}), (y_{RK} - y_{RA}))) * \frac{180}{\pi} \quad (20)$$

For craniocaudal line

$$\alpha_{craniocaudal\ line} = (\text{atan2}((x_{HC} - x_{SC}), (y_{HC} - y_{SC})) - \text{atan2}((x_{SC} - x_{AC}), (y_{SC} - y_{AC}))) * \frac{180}{\pi} \quad (21)$$

A feature vector is constructed from the computed features on the video data, which consists of m frames per video and 15 features per frame. This results in a total of $15m$ features per video, as given in Equation (22).

$$\{D_{Lefthand1}, \beta_{Lefthand1}, \alpha_{Lefthand1}, \dots, D_{craniocaudalm}, \beta_{craniocaudalm}, \alpha_{craniocaudalm}\} \quad (22)$$

4. Experimental Study and Result Analysis

The performance of the fall detection system will be assessed in the following section by comparing the output of various classifiers. Also, a comparison of the performance of the proposed fall detection system using the approaches in the literature is made. The feature set, consisting of 15 features per frame and $m = 100$ frames per video, is generated for the UPfall dataset, which is further utilized for training and testing machine learning models such as gradient boosting, decision tree, and random forest algorithms to classify videos into fall and non-fall categories. The workflow of the system is shown in Figure 8.

4.1. Decision Tree Algorithm

One of the classifiers employed for fall detection is the decision tree algorithm. This algorithm is generally used to address both classification and regression problems. The decision tree method generates predictions by using a succession of feature-based splits organized in a tree-like structure. Every node acts as a decision node starting at the root node and leads to a leaf node that represents the ultimate choice. The input space is recursively divided into subsets up until a halting requirement is satisfied, and this is how decision trees are fundamentally constructed.

Entropy, a statistical metric that measures the degree of unpredictability or impurity in a particular dataset node, is used to determine which node should be the decision tree's root node and is calculated as in Equation (23)

$$e(s) = -P_f \log P_f - P_{nf} \log P_{nf} \quad (23)$$

s denotes the number of samples; $e(s)$ is the entropy; P_f denotes the probability of a fall; P_{nf} denotes the probability of not a fall.

The measure of information gain represents the quantity of information obtained from a specific feature and is utilized to determine both the root and decision nodes in a decision tree algorithm. Information gain (IG) is calculated as in Equation (24)

$$IG = E_{parent} - E_{children} \quad (24)$$

where E_{parent} is entropy of parent node and $E_{children}$ is the average entropy of child nodes.

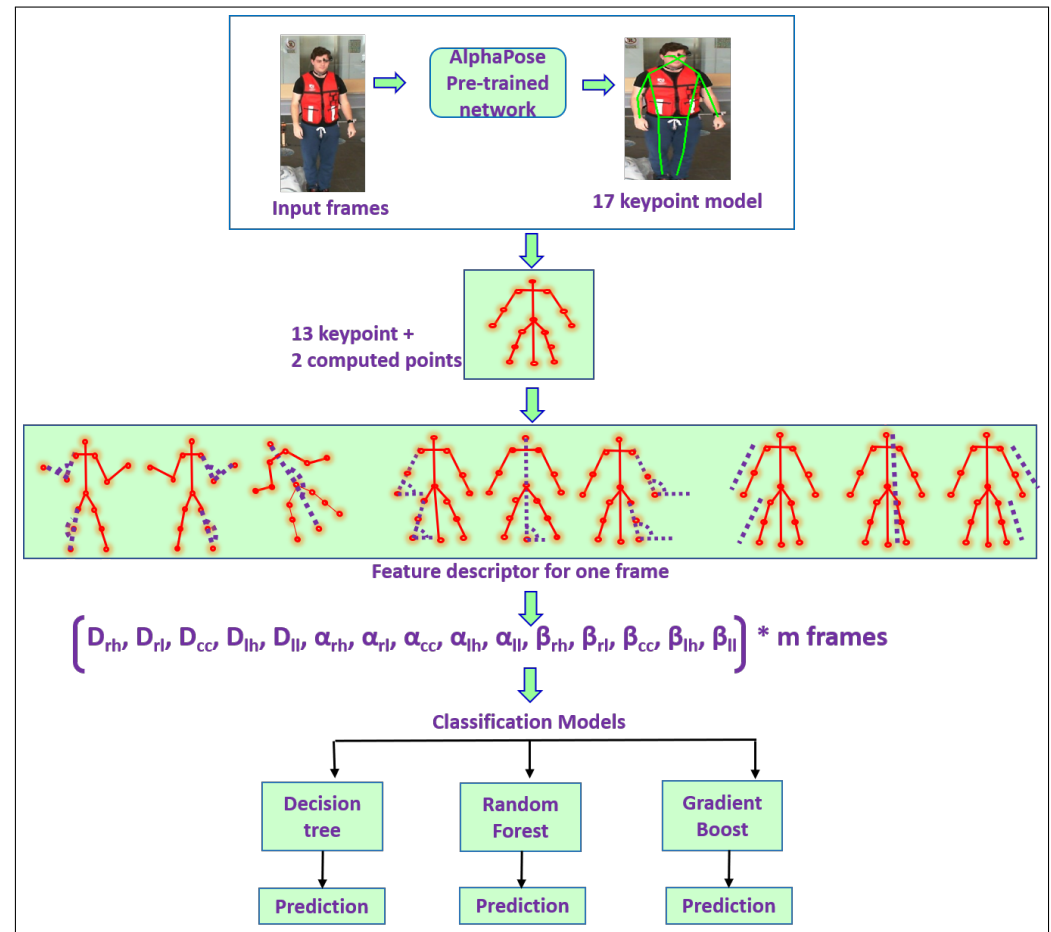


Figure 8. Workflow of the proposed fall detection system.

4.2. Random Forest Algorithm

A random forest is a collection of multiple decision trees, where each tree predicts a class, and the final output of the model is determined by the class that receives the most votes from the individual trees.

4.3. Gradient Boost Algorithm

The goal of the gradient boosting algorithm is to improve weak learners' performance and transform them into strong learners. It also makes use of an ensemble of decision trees, similar to the random forest algorithm. However, it uses a modified version of the AdaBoost algorithm technique. All observations in the AdaBoost algorithm are trained with equal weights. The weights of the hard-to-classify observations are increased with each iteration, while those of the straightforward observations are dropped.

4.4. Evaluation Metrics

Performance measurement is crucial for any classification algorithm in machine learning. The performance metrics used to validate our system are as follows:

1. *Accuracy*: Accuracy is an important measure of the classifier's performance. It is the ratio that relates to the proportion of precise predictions made compared to all other predictions. Accuracy is defined as given in Equation (25).

$$Acc = \frac{ncp}{tnp} \quad (25)$$

Acc: accuracy

ncp: number of correct predictions

tnp: total number of predictions

2. *Confusion matrix*: The confusion matrix is a statistic used to evaluate the performance of a predictive model that can be provided in either tabular or matrix style. It serves as a visual representation of a classification model's true positive, false positive, true negative, and false negative predictions, as shown in Figure 9.

The number of falls that were reported as falling is known as true positives (T_p). The amount of non-falls that were anticipated to be non-falls is known as true negatives (T_n). The term "false positives" (F_p) refers to the number of non-falls that were mistakenly identified as falling. False negative (F_n) refers to the number of falls that were assumed not to occur.

3. *Precision*: Precision is another measure that calculates how accurately the model predicts positive outcomes. From the total number of samples classified as positive, the number of true positive predictions is identified as the model's precision. A model with high precision has a lower number of false positives. The precision (Pre) is calculated in the Equation (26).

$$Pre = \frac{T_p}{T_p + F_p} \quad (26)$$

4. *Sensitivity*: In machine learning, sensitivity refers to the true positive rate. From the total number of positive samples of ground truth, the number of samples predicted as positive defines the model's sensitivity. A model with high sensitivity has predicted most of the positive samples correctly, resulting in low false negatives. Sensitivity is calculated as in the Equation (27).

$$Recall = \frac{T_p}{T_p + F_n} \quad (27)$$

5. *Specificity*: In machine learning, specificity refers to the true negative rate. From the total number of negative samples in the ground truth, the number of samples classified as negative defines the specificity of the model. A model with high specificity has predicted most of the negative instances correctly. The specificity (Spe) is calculated as given in Equation (28).

$$Spe = \frac{T_n}{T_n + F_p} \quad (28)$$

6. *F1-score*: The F1-score metric represents the balance between the true positive rate and the precision. It combines the precision and recall scores to get a single score that assesses a predictive model's overall effectiveness. F1-score ranges from 0 to 1, where 1 indicates the best case. The F1-score is calculated using both precision and recall as given in Equation (29).

$$F1 - score = 2 * \frac{Pre * recall}{Pre + recall} \quad (29)$$

7. *AUC – ROC curve*: The *AUC – ROC* curve represents the model performance of a binary classifier. *AUC* refers to the area under the curve and *ROC* refers to the receiver operating characteristic. The *ROC* is a probability curve that plots the true positive rate (sensitivity) against the false positive rate (1–specificity) at various classification thresholds. A higher *AUC* means that the model is better in its classification.

PV \ AV	Positive	Negative
Positive	T_p	F_p
Negative	F_n	T_n

AV – Actual Values; PV – Predicted Values

Figure 9. General form of a confusion matrix.

Performance of the Algorithms

Performance results of the algorithms used to evaluate the system, i.e., decision tree algorithm, random forest and gradient boost are given in Table 2, the confusion matrix is shown in Figure 10 and the AUC-ROC curve is plotted in Figure 11.

Table 2. Performance measurement of all three algorithms.

	Accuracy (%)	Precision (%)	Sensitivity (%)	Specificity (%)	F1-Score (%)
Decision Tree	88.39	84.48	92.45	84.74	88.28
Random Forest	96.43	98.03	94.33	98.30	96.15
Gradient Boost	98.32	98.11	98.11	98.30	98.11

The accuracy of the decision tree algorithm at different depths is shown in Figure 12. The accuracy of the gradient boost algorithm at different learning rates is shown in Figure 13. The number of estimators was increased from 100 to 500 and the accuracy was measured at different learning rates. It can be observed that the accuracy is the same at *number of estimators* = 400 for all three learning rates. Later, there is an improvement in the accuracy only for learning rate 0.1 at *number of estimators* = 500.

A comparison of the performance of the existing approaches that were evaluated on the UPfall dataset is given in Table 3. It can be observed that our proposed methodology has achieved the highest scores in terms of accuracy, precision, sensitivity, and f1-score. The results obtained from the system are shown in Figure 14 and Figure 15. These results represent the sequence of frames of videos that were classified as falling and daily living activities respectively.

Table 3. Comparison of the performance of the existing approaches that were evaluated on UPfall dataset.

Author	Classifier	Sensitivity (%)	Specificity (%)	Precision (%)	F1-Score (%)	Accuracy (%)
[39]	Convolutional neural network (CNN) Lateral camera (cam1)	97.72	81.58	95.24	97.20	95.24
	Convolutional neural network (CNN) Frontal camera (cam2)	95.57	79.67	96.30	96.93	94.78
[36]	K-Nearest Neighbors (KNN)	15.54	93.09	15.32	15.19	34.03
	Support Vector Machine (SVM)	14.30	92.97	13.81	13.83	34.40
	Random Forest (RF)	14.48	92.9	14.45	14.38	32.33
	Multilayer Perceptron (MLP)	10.59	92.21	8.59	7.31	34.03
[40]	Convolutional neural network (CNN)	71.3	99.5	71.8	71.2	95.1
	Convolutional neural network (CNN)	97.95	83.08	96.91	97.43	95.64
[32]	Average(RF, SVM, MLP, KNN)	96.80	99.11	96.94	96.87	97.59
[8]	Convolutional neural network (CNN) + Long-short term memory (LSTM)	94.37	98.96	91.08	92.47	96.72
[31]	ConvLSTM	97.68	-	97.71	97.68	97.68
Our Proposed Work	Gradient Boost (GB)	98.11	98.30	98.11	98.11	98.32

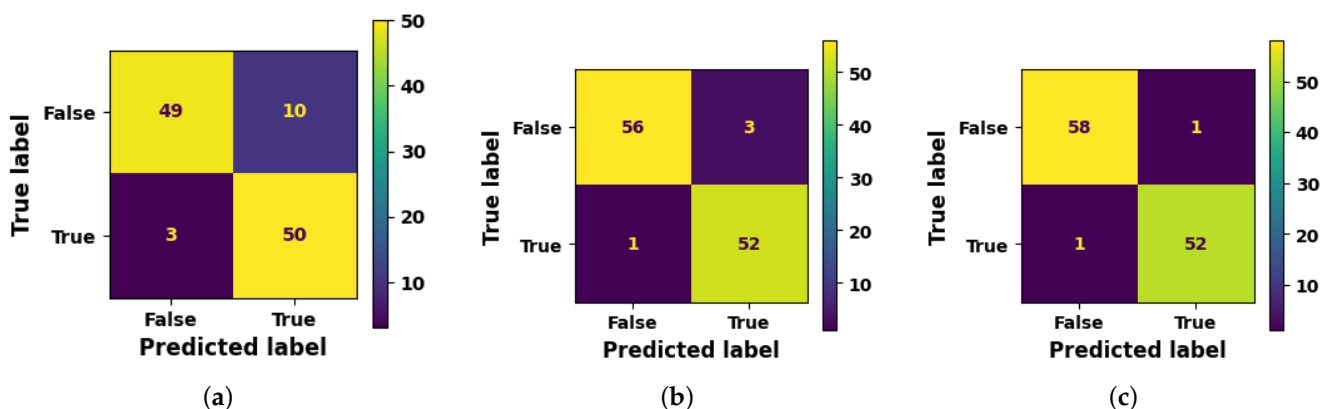
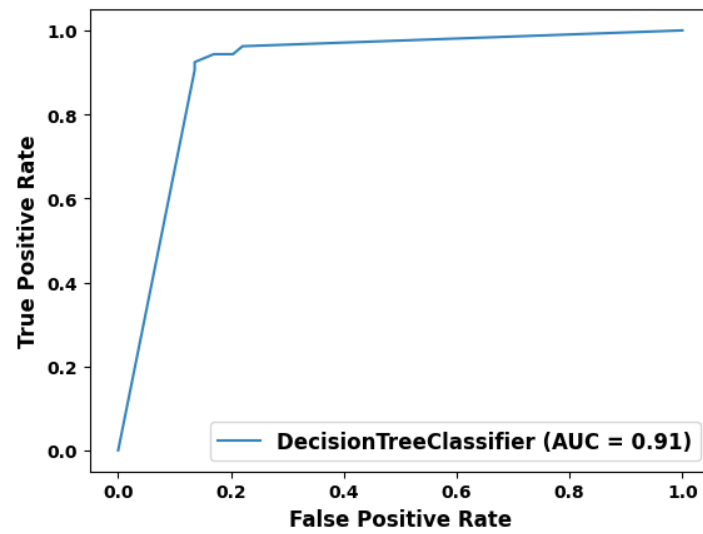
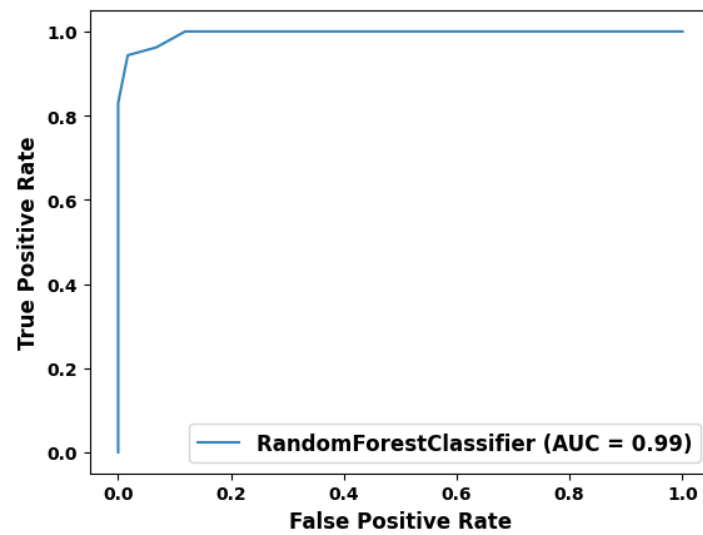


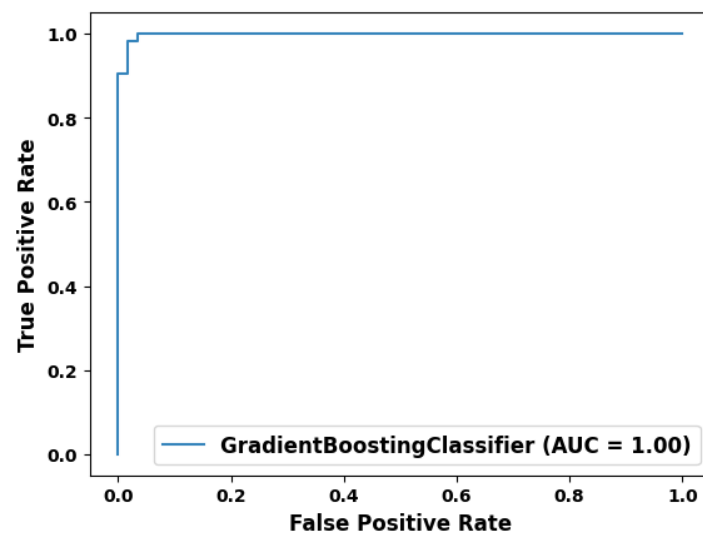
Figure 10. Confusion matrices of the algorithms in the proposed work. (a) depicts the confusion matrix of the Decision tree algorithm, (b) depicts the confusion matrix of the Random Forest algorithm and (c) depicts the confusion matrix of the Gradient boost algorithm.



(a)



(b)



(c)

Figure 11. AUC-ROC Curves of the algorithms in the proposed work. (a) Decision tree algorithm. (b) Random forest algorithm. (c) Gradient boost algorithm.

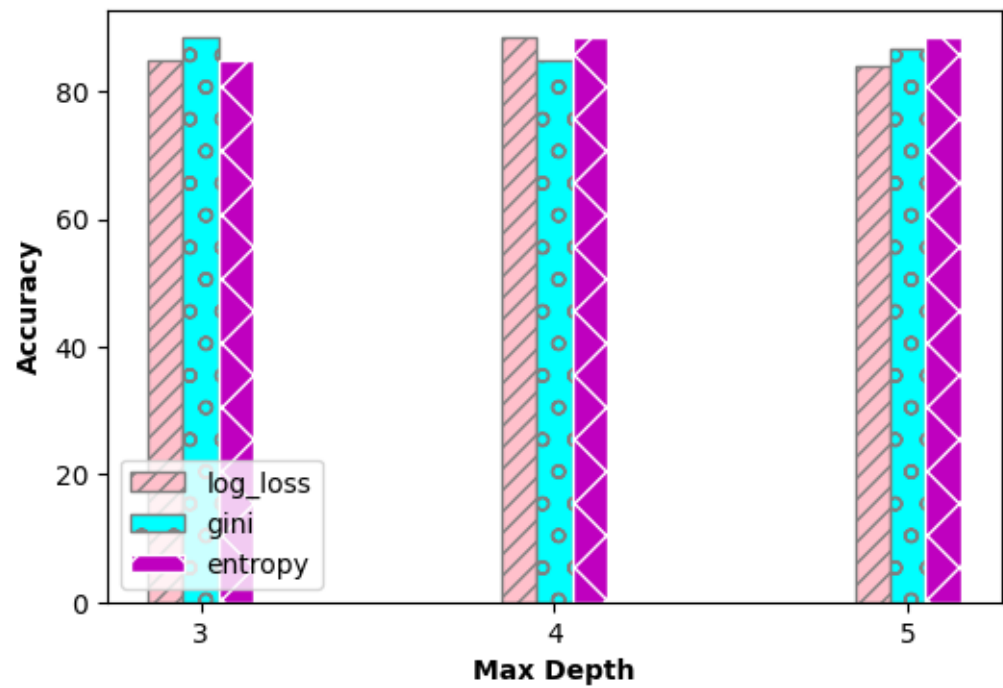


Figure 12. Accuracy of decision tree at different depth.

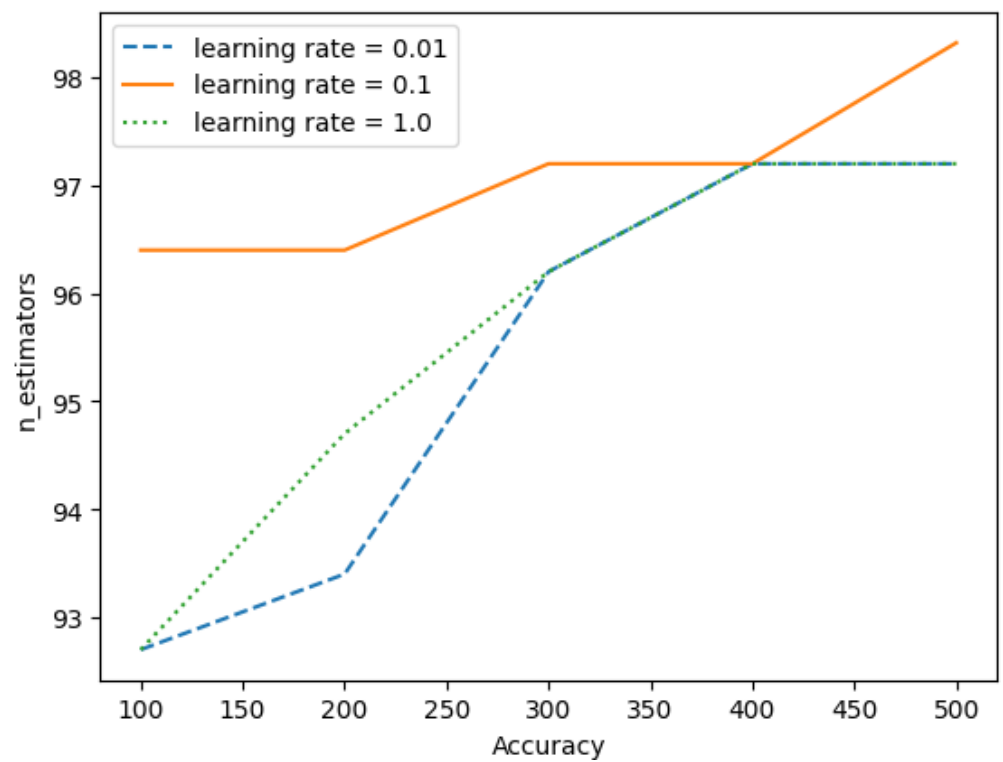


Figure 13. Performance of Gradient Boost algorithm at different learning rates.

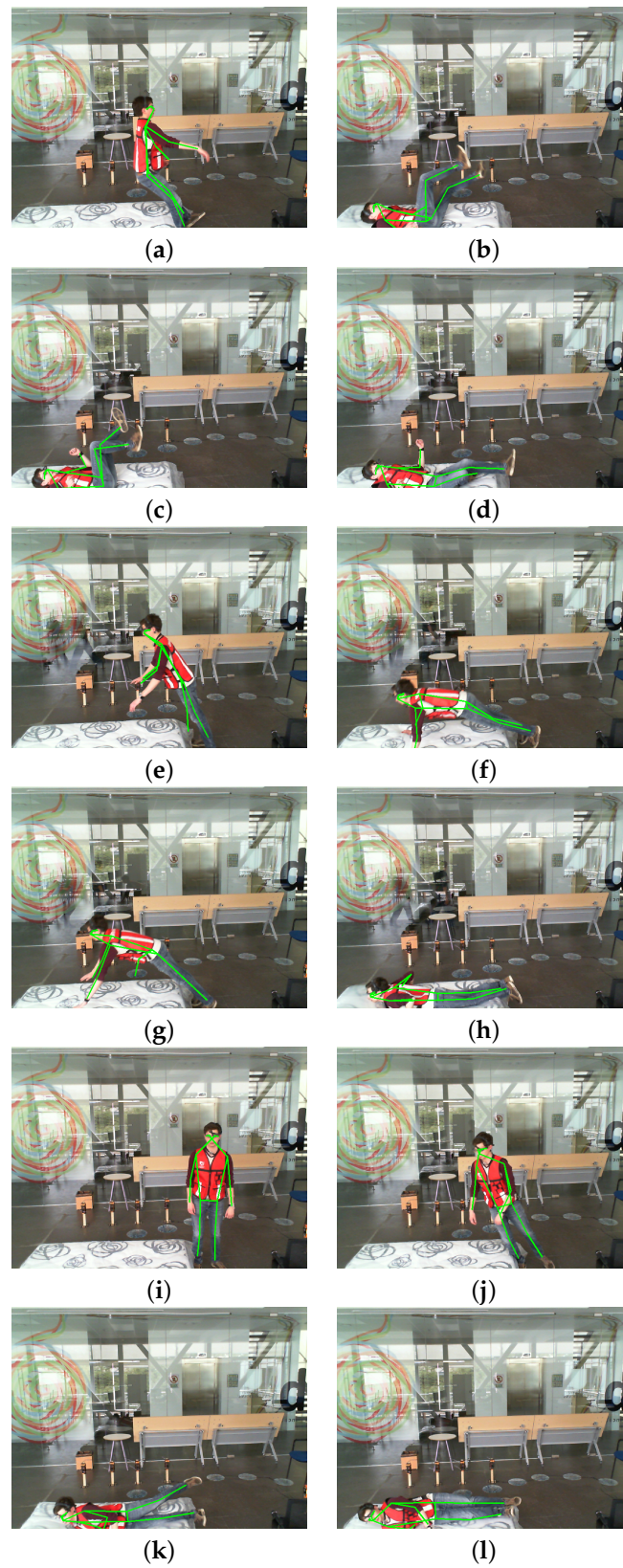


Figure 14. Results of the falling activity of the proposed system. (a–d) show the sequence of falling backwards, (e–h) show the sequence of falling forwards and (i–l) show the sequence of a falling sideways.

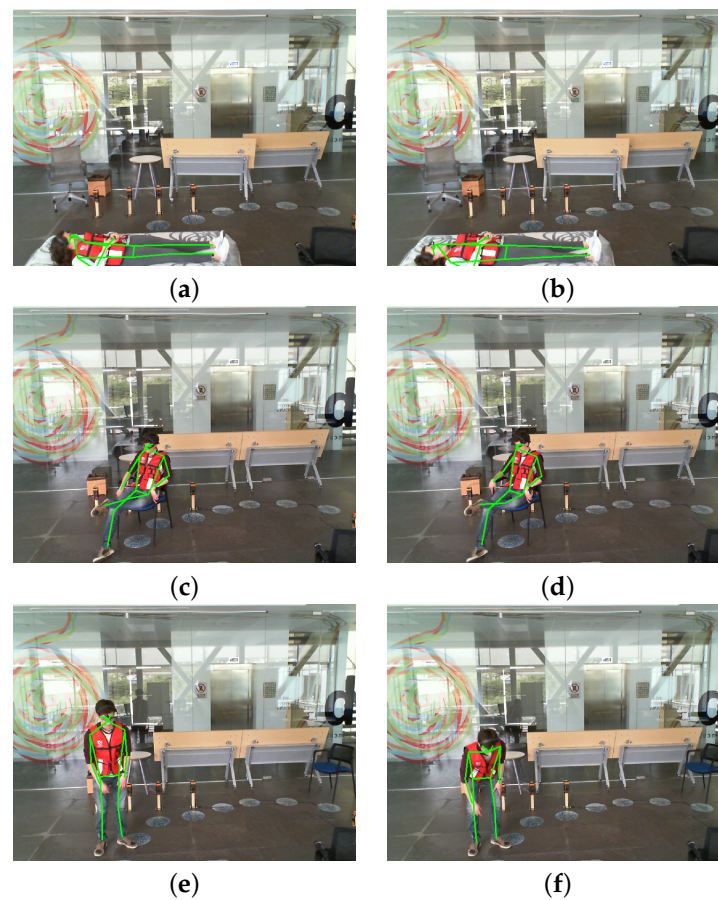


Figure 15. Results of the daily living activities of the proposed system. (a,b) show images of laying activity, (c,d) show images of sitting activity and (e,f) show images of picking up an object.

5. Conclusions and Future Work

This work presents a precise solution for detecting falls and classifying poses by extracting human skeleton features and analyzing their body geometry. Our model which was evaluated on the UPfall dataset achieves an accuracy of 98.32%, which demonstrates its effectiveness in classifying falls and an F1-score of 98.11%. Also, our proposed feature descriptor is invariant to the gender and age of a subject during fall detection. Unlike previous approaches, we do not rely on static techniques such as human silhouette extraction or event-based detection for fall detection, as these methods may not be robust enough to handle changes in the operating environment. Our proposed solution for fall classification has the potential to lower the computational time owing to the features extracted. However, two key constraints of any fall detection system that employs RGB data are preserving the user's privacy, which can be overcome by incorporating depth cameras, and dealing with low lighting conditions, which can be improved by using night vision cameras.

Author Contributions: Conceptualization: A.R.I., V.M.M. and M.N.K.; Formal Analysis: V.M.M. and S.W.; Methodology: A.R.I., V.M.M. and M.N.K.; Investigation and Writing original draft: A.R.I.; Resources: S.W. and Y.Z.; Writing review & editing: V.M.M. and Y.Z.; Visualization: A.R.I., V.M.M. and M.N.K.; Supervision: V.M.M., M.N.K. and Y.Z.; Funding: S.W. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is partially supported by SRM University-AP, Andhra Pradesh, India; MRC, UK (MC_PC_17171), Royal Society, UK (RP202G0230), Hope Foundation for Cancer Research, UK (RM60G0680), GCRF, UK (P202PF11), Sino-UK Industrial Fund, UK (RP202G0289), BHF, UK (AA/18/3/34220), LIAS, UK (P202ED10, P202RE969), Data Science Enhancement Fund, UK (P202RE237), Fight for Sight, UK (24NN201); Sino-UK Education Fund, UK (OP202006); BBSRC, UK (RM32G0178B8).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Liu, Y.; Chan, J.S.; Yan, J.H. Neuropsychological mechanisms of falls in older adults. *Front. Aging Neurosci.* **2014**, *6*, 64. [CrossRef] [PubMed]
2. CDC. Fact Sheet. 2021. Available online: <https://www.cdc.gov/visionhealth/resources/features/vision-loss-falls.html> (accessed on 1 January 2023).
3. Alarifi, A.; Alwadain, A. Killer heuristic optimized convolution neural network-based fall detection with wearable IoT sensor devices. *Measurement* **2021**, *167*, 108258. [CrossRef]
4. Şengül, G.; Karakaya, M.; Misra, S.; Abayomi-Alli, O.O.; Damaševičius, R. Deep learning based fall detection using smartwatches for healthcare applications. *Biomed. Signal Process. Control* **2022**, *71*, 103242. [CrossRef]
5. Wu, X.; Zheng, Y.; Chu, C.H.; Cheng, L.; Kim, J. Applying deep learning technology for automatic fall detection using mobile sensors. *Biomed. Signal Process. Control* **2022**, *72*, 103355. [CrossRef]
6. De, A.; Saha, A.; Kumar, P.; Pal, G. Fall detection method based on spatio-temporal feature fusion using combined two-channel classification. *Multimed. Tools Appl.* **2022**, *81*, 26081–26100. [CrossRef]
7. Galvão, Y.M.; Ferreira, J.; Albuquerque, V.A.; Barros, P.; Fernandes, B.J. A multimodal approach using deep learning for fall detection. *Expert Syst. Appl.* **2021**, *168*, 114226. [CrossRef]
8. Inturi, A.R.; Manikandan, V.; Garrapally, V. A novel vision-based fall detection scheme using keypoints of human skeleton with long short-term memory network. *Arab. J. Sci. Eng.* **2023**, *48*, 1143–1155. [CrossRef]
9. Forsyth, D.; Ponce, J. *Computer Vision: A Modern Approach*; Prentice Hall: Hoboken, NJ, USA, 2011.
10. Baumgart, B.G. A polyhedron representation for computer vision. In Proceedings of the National Computer Conference and Exposition, Anaheim, CA, USA, 19–22 May 1975; pp. 589–596.
11. Shirai, Y. *Three-Dimensional Computer Vision*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
12. Jordan, M.I.; Mitchell, T.M. Machine learning: Trends, perspectives, and prospects. *Science* **2015**, *349*, 255–260. [CrossRef]
13. Sammut, C.; Webb, G.I. *Encyclopedia of Machine Learning*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011.
14. Liu, B. Sentiment analysis and subjectivity. In *Handbook of Natural Language Processing*; Routledge: Abingdon, UK, 2010; Volume 2, pp. 627–666.
15. Jelinek, F. *Statistical Methods for Speech Recognition*; MIT Press: Cambridge, MA, USA, 1997.
16. Yu, D.; Deng, L. *Automatic Speech Recognition*; Springer: Berlin, Germany, 2016.
17. Pavlidis, T. *Algorithms for Graphics and Image Processing*; Springer Science & Business Media: Berlin, Germany, 2012.
18. Russ, J.C. *The Image Processing Handbook*; CRC Press: Boca Raton, FL, USA, 2006.
19. Huang, T.S.; Schreiber, W.F.; Tretiak, O.J. Image processing. In *Advances in Image Processing and Understanding: A Festschrift for Thomas S Huang*; World Scientific: Singapore, 2002; pp. 367–390.
20. Messing, R.; Pal, C.; Kautz, H. Activity recognition using the velocity histories of tracked keypoints. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 104–111.
21. Zhang, C.; Tian, Y. RGB-D camera-based daily living activity recognition. *J. Comput. Vis. Image Process.* **2012**, *2*, 12.
22. Hong, Y.J.; Kim, I.J.; Ahn, S.C.; Kim, H.G. Activity recognition using wearable sensors for elder care. In Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking, Hainan, China, 13–15 December 2008; IEEE: Piscataway, NJ, USA, 2008; Volume 2, pp. 302–305.
23. Wu, F.; Zhao, H.; Zhao, Y.; Zhong, H. Development of a wearable-sensor-based fall detection system. *Int. J. Telemed. Appl.* **2015**, *2015*, 576364. [CrossRef]
24. Bourke, A.K.; Lyons, G.M. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Med. Eng. Phys.* **2008**, *30*, 84–90. [CrossRef] [PubMed]
25. Chaccour, K.; Darazi, R.; el Hassans, A.H.; Andres, E. Smart carpet using differential piezoresistive pressure sensors for elderly fall detection. In Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Abu Dhabi, United Arab Emirates, 19–21 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 225–229.
26. Feng, G.; Mai, J.; Ban, Z.; Guo, X.; Wang, G. Floor pressure imaging for fall detection with fiber-optic sensors. *IEEE Pervasive Comput.* **2016**, *15*, 40–47. [CrossRef]
27. Jagedish, S.A.; Ramachandran, M.; Kumar, A.; Sheikh, T.H. Wearable Devices with Recurrent Neural Networks for Real-Time Fall Detection. In Proceedings of the International Conference on Innovative Computing and Communications—ICICC 2022, Delhi, India, 19–20 February 2022; Springer: Berlin, Germany, 2022; Volume 2, pp. 357–366.

28. Li, W.; Zhang, D.; Li, Y.; Wu, Z.; Chen, J.; Zhang, D.; Hu, Y.; Sun, Q.; Chen, Y. Real-time fall detection using mmWave radar. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 16–20.
29. Yao, C.; Hu, J.; Min, W.; Deng, Z.; Zou, S.; Min, W. A novel real-time fall detection method based on head segmentation and convolutional neural network. *J. Real-Time Image Process.* **2020**, *17*, 1939–1949. [[CrossRef](#)]
30. Khraief, C.; Benzarti, F.; Amiri, H. Elderly fall detection based on multi-stream deep convolutional networks. *Multimed. Tools Appl.* **2020**, *79*, 19537–19560. [[CrossRef](#)]
31. Mobsite, S.; Alaoui, N.; Boulmalf, M.; Ghogho, M. Semantic segmentation-based system for fall detection and post-fall posture classification. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105616. [[CrossRef](#)]
32. Ramirez, H.; Velastin, S.A.; Meza, I.; Fabregas, E.; Makris, D.; Farias, G. Fall detection and activity recognition using human skeleton features. *IEEE Access* **2021**, *9*, 33532–33542. [[CrossRef](#)]
33. Alaoui, A.Y.; El Fkihi, S.; Thami, R.O.H. Fall detection for elderly people using the variation of key points of human skeleton. *IEEE Access* **2019**, *7*, 154786–154795. [[CrossRef](#)]
34. Mansoor, M.; Amin, R.; Mustafa, Z.; Sengan, S.; Aldabbas, H.; Alharbi, M.T. A machine learning approach for non-invasive fall detection using Kinect. *Multimed. Tools Appl.* **2022**, *81*, 15491–15519. [[CrossRef](#)]
35. Zhang, X.; Yu, H.; Zhuang, Y. A robust RGB-D visual odometry with moving object detection in dynamic indoor scenes. *IET Cyber-Syst. Robot.* **2023**, *5*, e12079. [[CrossRef](#)]
36. Martínez-Villaseñor, L.; Ponce, H.; Brieva, J.; Moya-Albor, E.; Núñez-Martínez, J.; Peñafort-Asturiano, C. UP-fall detection dataset: A multimodal approach. *Sensors* **2019**, *19*, 1988. [[CrossRef](#)]
37. Fang, H.S.; Xie, S.; Tai, Y.W.; Lu, C. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2334–2343.
38. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin, Germany, 2014; pp. 740–755.
39. Espinosa, R.; Ponce, H.; Gutiérrez, S.; Martínez-Villaseñor, L.; Brieva, J.; Moya-Albor, E. 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.* **2019**, *115*, 103520. [[CrossRef](#)] [[PubMed](#)]
40. Espinosa, R.; Ponce, H.; Gutiérrez, S.; Martínez-Villaseñor, L.; Brieva, J.; Moya-Albor, E. Application of convolutional neural networks for fall detection using multiple cameras. In *Challenges and Trends in Multimodal Fall Detection for Healthcare*; Springer: Berlin, Germany, 2020; pp. 97–120.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.