

Article

Using Temporal Covariance of Motion and Geometric Features via Boosting for Human Fall Detection

Syed Farooq Ali ¹ , Reamsha Khan ¹, Arif Mahmood ² , Malik Tahir Hassan ¹
and Moongu Jeon ^{3,*}

¹ Department of Software Engineering, University of Management and Technology, UMT Road, C-II Johar Town, Lahore 54000, Pakistan; farooq.ali@umt.edu.pk (S.F.A.); reamshakhan1122@gmail.com (R.K.); tahir.hassan@umt.edu.pk (M.T.H.)

² Department of Computer Science, Information Technology University (ITU), 346-B, Ferozepur Road, Lahore, Punjab 54000, Pakistan; arif.mahmood@itu.edu.pk

³ School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Korea

* Correspondence: mgjeon@gist.ac.kr; Tel.: +82-62-715-2406

Received: 9 April 2018; Accepted: 11 June 2018; Published: 12 June 2018



Abstract: Fall induced damages are serious incidences for aged as well as young persons. A real-time automatic and accurate fall detection system can play a vital role in timely medication care which will ultimately help to decrease the damages and complications. In this paper, we propose a fast and more accurate real-time system which can detect people falling in videos captured by surveillance cameras. Novel temporal and spatial variance-based features are proposed which comprise the discriminatory motion, geometric orientation and location of the person. These features are used along with ensemble learning strategy of boosting with J48 and Adaboost classifiers. Experiments have been conducted on publicly available standard datasets including *Multiple Cameras Fall (with 2 classes and 3 classes)* and *UR Fall Detection* achieving percentage accuracies of 99.2, 99.25 and 99.0, respectively. Comparisons with nine state-of-the-art methods demonstrate the effectiveness of the proposed approach on both datasets.

Keywords: intelligent surveillance systems; human fall detection; health and well-being; safety and security

1. Introduction

The increasing number of aged persons has led to the uncertainty of unaided and unprovoked falls which may cause physical harm, injuries and health deterioration. These problems may become more intense if timely aid and assistance is not available. To mitigate such effects and to control the risks, there must be an accurate fall detection system. For this reason, surveillance added technology for the timely and retrospective detection of falls has become a priority for the health care industry. Therefore, the development of an intelligent surveillance system is essential, specifically, a system which has the capacity to automatically detect fall incidences using surveillance cameras.

To cope with such a crucial need, various fall detection wearable devices have been developed [1]. Some devices contain buttons and sensors that can be pressed if there is an emergency [1]. However, these devices become ineffective or even useless if the subject is unable to press the button due to unconsciousness or being far from the device. Due to the failure of wearable devices, video controlling and monitoring systems have entered the arena [1], but these systems also suffer from inaccuracy and unreliability [1]. Generic action recognition systems such as [2] may not efficiently detect falls. Due to the lack of efficiency of wearable devices and generic video systems [3], it has become necessary for

customized automatic fall detection systems to be developed to cope with the challenges posed by fall detection problems. Such systems could dramatically improve the health care of older people.

In the current paper, we propose a fall detection system based on the spatial and temporal variance of different discriminative features. The proposed system is compared with nine existing algorithms on two publicly available datasets. The proposed system has exhibited excellent performance on both datasets. It may be noted that, a video surveillance system may result in a privacy issues during the continuous monitoring of older adults, and thus, has its own limitations for individual home monitoring, but it could be useful for rehabilitation centers and elderly health care houses, reformation centers [4,5], nursing homes and hospitals, electronic care environments [6], disabled care centers [7], and firearm shot damage spotting centers [8].

The remainder of this paper is organized as follows. Section 2 contains related work, Section 3 contains the proposed system, experimental evaluations are given in Section 4, conclusions and future follow in Section 5.

2. Related Work

In recent years, several vision-based techniques have been proposed for human fall detection. Some of these are based on wearable devices, such as accelerometers, while others use depth sensors, such as Microsoft Kinect. Moreover, RGB video cameras and different types of audio sensors have also been used for fall detection. Depending upon the underlying hardware, algorithms have been proposed to use one or more modalities, including accelerometer output, depth data, audio data and/or video data. Following are the broad categories of fall detection approaches and techniques.

2.1. Classification Based on Input Data Types

2.1.1. Sensor-Based

Luo et al. [9] developed a dynamic movement detection system to detect human falls. Their algorithm is based on the output of digital signals from mounted accelerometers. They filtered noisy segments using a Gaussian filter and set up a 3D body movement display which related different postures of the body to the yields of the accelerometers. Bourke et al. [10] described a procedure under supervised conditions to identify falls using tri-axial accelerometer sensors based on thresholding techniques. These sensors were mounted on the trunks and thighs of subjects. Makhlouf et al. [11] developed a multi-modal system that provided a fall detection service and an emergency service. Their system used photoelectric sensors and accelerometers to get information regarding the state of a person. If a person is in fall state, then the emergency service informs the doctor. The message sent to the specialist incorporates data about the localized area and condition of the individual. Recently, Casilari et al. [12] proposed a public repository of datasets that could be used as a common reference among the researchers to compare their algorithms. They created the UMAFall dataset that contains information about day-to-day activities and human falls. Contrary to other existing datasets that use one or two sensing points, they obtained data using five wearable sensors.

2.1.2. Audio-Based

Zigel et al. [13] presented an automatic fall discovery framework for elderly individuals particularly for when a person is unconscious or tense. The framework depends on indoor vibration and sound detection to classify human falls and other events. The classification uses features that include the shock response spectrum and Mel-frequency cepstrum coefficients. Doukas et al. [14] proposed a fall detection system based on audio and video data. Tracking of a person was done using video, and sound directionality analysis was made from audio data. Various features, including acceleration, sound proximity, average peak frequency, average signal relative amplitude, visual blob size and average movement speed were trained and tested using an SVM classifier for the detection of falls.

The post-fall analysis was conducted as well to further predict if a person recovered his state or remained unconscious.

2.1.3. Image and Video-Based

Foroughi et al. [15] studied the morphological variations of silhouettes acquired from a series of videos. They concluded that the amalgamation of relative ellipse along the human body, the projection of histogram along the x -axis and y -axis and the change in the position of person's head provided beneficial cues for the determination of various behaviors of falls. Miaou et al. [16] suggested that visual detection may cause false readings as sometimes a movement appears as a fall, but, actually it may not be a fall. It could be simply a motion towards the direction of ground.

Lee et al. [17] recognized falls using image-based sensors. The data was generated by asking the subjects to randomly repeat five scenarios including lying down in a 'tucked' position, lying down in a 'stretched' position, stooping, sitting/lying down in an inactive zone and walking/standing.

Rougier et al. [18] emphasized the use of computer vision techniques to provide promising solutions for the detection of human falls. They used human shape deformation in a video sequence for human fall detection. The shape deformation from the person's silhouette was tracked along the video sequence. The fall was detected using a Gaussian mixture model. In [19], a vision-based system was proposed for human fall detection. The system used novel features of motion history for the detection of a fall. The system was run on video sequences of daily activities and simulated falls. Doulamis et al. [20] proposed a human fall detection system using cheap and low resolution cameras. Their system used adaptive background modeling using Gaussian Mixtures, and hierarchical motion estimation was used to distinguish falls from other activities, including lying, walking and sitting.

2.2. Classification Based on Classifier Types

2.2.1. Thresholding-Based

Chariff et al. [21] proposed intelligent surveillance technology to be used for the detection of dangerous events in the home environment. They tracked individuals as ellipses, and the direction of motion was utilized to recognize irregular and abnormal activities. Tao et al. [22] described the use of activity summarization in supportive homes where care is provided to aged people, but the proposed fall detection system was not capable of differentiating between a real fall incident and when the subject was just lying down.

Zaid et al. [23] used mobile robots to provide an efficient solution for fall detection in elderly people. The mobile robot system used Kinect sensor to track a target person and detect when they had fallen. Moreover, in case of a fall, an alarm was generated by sending an SMS message notification or making an emergency call.

Sumiya et al. [24] proposed a versatile robot to recognize human falls and to give details to observers. It comprised of a family portable robot with Microsoft Kinect and a PC. For simplicity, a sensor was placed on the robot which limited the blind zone by moving around with the robot. This technique improved the accuracy of fall detection compared to monitoring techniques based on fixed sensors.

2.2.2. Machine Learning-Based

Various classifiers including the Support Vector Machine (SVM), Adaboost, Multilayer Perceptron (MLP) and J48 have been used for human fall detection. Support Vector Machine, a pattern classification algorithm developed by V. Vapnik and his team at AT&T Bell Labs maps the data into higher dimensional input space and constructs an optimal plane separating the hyper-plane in this space. In [25], Foroughi et al. in 2008 implemented a Support Vector Machine to classify an event either as a fall or not a fall using feature-based approach and achieved a reliable recognition rate of 88.08%.

Various other human fall detection systems use SVM for the classification of fall events (by their features) [26–28].

Debard et al. [26] proposed a feature-based approach for the detection of human falls. The proposed features included the angle of fall, aspect ratio, center velocity and head velocity. These features were trained and tested using SVM method. The drawbacks of this system include its inaccuracy to discriminate gradual fall from a person who is sitting down normally.

Ni et al. [29] developed a fall prevention framework for application in hospital wards. Their system detects if a patient gets up from the bed and generates an alarm for hospital staff to provide help. Their system used various features to detect human falls on a dataset of videos obtained from RGBD sensors of Microsoft Kinect. These included the region of interest (ROI), motion-based features, and shape-based features.

Adaboost [30], an adaptive boosting algorithm, utilizes a small number of weak classifiers that are used to construct cascades of strong classifiers. The combination of the strong classifiers into a cascade results in high accuracy and time efficiency for human fall detection. Multilayer Perceptron (MLP), a feedforward artificial neural network, is used for human fall detection systems, and achieved an accuracy of 90.15% on an ADL dataset [31].

The rule-based algorithm, J48, has also been used for the detection of human falls [32,33]. J48 is a C4.5 decision tree which is used to present different models of classification and also reveals human reasoning [34]. It has many advantages over various learning algorithms, such as its low computational cost of model generation, noise robustness and ability to handle redundant attributes and modules. It is robust even if training data contains errors or have missing attribute values [35]. Shi et al. proposed a human fall detection algorithm for classifying the human motion using the J48 decision tree classifier and achieved a sensitivity of 98.9%, a specificity of 98.5% and an overall accuracy of 98.6% [36]. In 2017, Guvensan et al. developed a system that implements the decision tree learning algorithm of J48, using five features, for detection of fall events [37]. Motivated by these algorithms, we chose the boosted J48 classifier due to its significantly higher F-measure, low computational cost and robustness to outliers and reduction of the feature space.

In the current work, we propose an algorithm under a boosting framework based on RGB video data for human fall detection. We emphasize its fast detection speed with high accuracy. We compare our work with existing state-of-the-art approaches, including Kepski et al. [38] by using KNN and SVM on the Multiple Cameras Fall Dataset [26] and the UR Fall Detection Dataset [38]. The results of our approach under an ensemble learning strategy with J48 and Adaboost outperform existing state-of-the-art approaches on both datasets in terms of percentage accuracy and execution time.

3. Proposed Fall Detection System

The first step in our proposed algorithm was to segment the foreground from the background and to identify the foreground as a person or non-person. In the second step, we computed various features from the foreground and in the third step, we trained a boosted J48 classifier for per frame classification of the foreground as a falling person or a stable moving person. In addition to the spatial information, our features also use temporal information as velocity and acceleration. Each of these steps is explained in the following sections.

3.1. Foreground Detection

A clean background image can be computed using the recently proposed background detection algorithms, such as [39–41]. The computed background image is subtracted from each frame to find the change region. Pixels with a change larger than a fixed percentage of the background image are considered to be , while the rest are considered to be the result of variation due to noise. In our experiments, we fixed this percentage at 15% of each pixel value in the background image. Then, a distance transform was computed over the changed region, followed by morphological operations, including erosion and flood filling to fill the holes in this region. To ensure that the change region

contained only humans, connected components were computed and components with a size less than the minimum human size were deleted from the foreground. If a connected component had a size larger than the minimum person size threshold, it was considered to correspond to a human. Thus, a foreground mask containing only a human object was obtained. The minimum person size threshold helped to discard frames not containing a full person. Components with a size larger than the minimum person size threshold are referred to as foreground blobs in the rest of the paper.

3.2. Temporal and Spatial Variance for Falling Person Detection

In this section we discuss various types of temporal variance that we used in the proposed real-time person falling detection system. These variances include temporal variations of the aspect ratio, fall angle, speed, upper half area of bounding box and geometric center of the connected component. The temporal variance for each parameter was computed over a temporal window of size $k = 30$. This value of k was chosen after analyzing the video data and carefully conducting experiments. The value of $k = 30$ corresponded to a duration of 1 s as the frame rate of the videos was 30 frames per second. Figure 1 shows variation of the aspect ratio and other parameters with a change in person position during the process of fall.

3.2.1. Temporal Variance of the Aspect Ratio

A bounding box was computed containing the foreground blob. The aspect ratio refers to the ratio of the width to the height of this bounding box. The temporal variation of the aspect ratio is unique during the fall of a person, which was used as a feature. For a person in a stable position, the temporal variation in the aspect ratio is small, while during a fall, this variation is large. The temporal variation of the aspect ratio in a current frame was computed by taking the standard deviation of the aspect ratios of the previous k frames. After analyzing the video data and conducting experiments, the value of $k = 30$ was found to be reasonable to capture temporal variations in this parameter.

$$\sigma_{ar}^2 = \frac{1}{k} \sum_{i=1}^k (a_r(i) - \mu_{ar})^2 \quad (1)$$

3.2.2. Temporal Variation of the Person Angle

An ellipse was fitted in the foreground blob and a person angle was computed as the angle between the major axis of ellipse and the x -axis (horizontal axis). The person angle changed when a person fell from a standing state to a fall state. The temporal variation in the fall angle of the current frame was computed by computing the standard deviation of the fall angle of the previous $k = 30$ frames.

$$\sigma_{pa}^2 = \frac{1}{k} \sum_{i=1}^k (p_a(i) - \mu_{pa})^2 \quad (2)$$

3.2.3. Temporal Variation of the Motion Vector

The motion vector is the variation in the foreground blob's position between the current and the next frame. The motion magnitude increases when the body is in motion, and it reaches a high value during fall and then it becomes zero after the fall. The change in the magnitude of the body speed (B.S.) serves as an important parameter for human fall detection. The body speed is calculated by computing the motion vectors of the centroid of the foreground blob. The magnitude of motion vectors is calculated as shown in Equation (3).

$$|m_v| = \sqrt{(m_v(x))^2 + (m_v(y))^2}, \quad (3)$$

where $m_v(x)$ is the magnitude of motion vectors along the x -direction and $m_v(y)$ is the magnitude of motion vectors along the y -direction. The temporal variation in body speed gives the acceleration of the body. Figure 2 shows the body speed variation of a falling person where the number of frames and magnitude of motion vectors are plotted along x -axis and y -axis respectively. The temporal variance of m_v is computed as follows:

$$\sigma_{mv}^2 = \frac{1}{k} \sum_{i=1}^k (m_v(i) - \mu_{mv})^2, \quad (4)$$

where μ_{mv} is the mean motion vector over the current time window.

3.2.4. Temporal Variation of Shape Deformations

The orientation of the body shape changes significantly when a person falls. The upper half area of the foreground blob bounding box (U.H.B.B.) was used as a shape descriptor (s_d) to capture variations in the orientation of the body when fall occurs. The bounding box is divided into two equal halves. The upper half area of the upper half bounding box is high when a person is in fall state as compared to standing state. Hence, the area of the upper half of the bounding box serves as a strong feature as its value changes significantly when a person enters into a fall state from a normal state. Figure 2 shows the temporal variation of the upper half area of the bounding box of a falling person of video 1 (camera position 7) and video 3 (camera position 3), respectively. In these figures, the number of frames and number of pixels in the upper half area of bounding box are plotted along the x -axis and y -axis, respectively. The temporal variation in the upper half area of the bounding box of a current frame is computed by calculating the standard deviation in the upper half area of the bounding box from the previous $k = 30$ frames to detect if an event is a fall or not a fall.

$$\sigma_{sd}^2 = \frac{1}{k} \sum_{i=1}^k (s_d(i) - \mu_{sd})^2, \quad (5)$$

where μ_{sd} is the mean area of the upper-half bounding box over the current time window.

3.2.5. Temporal Variation in the Geometric Center Position

The geometric location of a foreground blob changes significantly when a person falls. This change in geometric location can be captured by taking the temporal variation of geometric center (G.C.) as a feature. The temporal variation in the x -component and y -component of the geometric center of a current frame is computed by taking the standard deviation of the x -component and y -component of the previous $k = 30$ frames. The geometric center, (x_g, y_g) , is the average of the x -coordinates and y -coordinates of all boundary points (edge points) of the object

$$(x_g, y_g) = \frac{1}{n} \left(\sum_{i=1}^n x_i, \sum_{i=1}^n y_i \right), \quad (6)$$

where (x_i, y_i) represents the coordinates of the pixels in the foreground blob. Figure 2 shows the temporal variation of the geometric center of a falling person. In this figure, the number of frames and geometric center are plotted along the x -axis and y -axis, respectively.

$$\sigma_{gp}^2 = \frac{1}{k} \sum_{i=1}^k (g_p(i) - \mu_{gp})^2, \quad (7)$$

where μ_{gp} is the mean position of the geometric center over the current time window.

3.2.6. Temporal Variation of the Ellipse Ratio

The ellipse ratio, e_r , is defined as the ratio between the length of the major and minor axes of the ellipse containing the foreground blob. It was used as a scale invariant feature in our proposed approach for the classification of a person as fall or not a fall. Figure 1A,B shows that the ellipse ratio changes significantly when the person is moving from a standing state (A (a) & B (a)) to intermediate states (A (b), B (b)) and A (c), B (c)) and then to the fall state (A (d) & B (d)). Hence, the ratio of these axes of ellipse serves as an important feature for detecting the human fall. The temporal variation of ellipse ratio is computed over the current time window

$$\sigma_{er}^2 = \frac{1}{k} \sum_{i=1}^k (e_r(i) - \mu_{er})^2, \quad (8)$$

where μ_{er} is the mean ellipse ratio over the current time window.

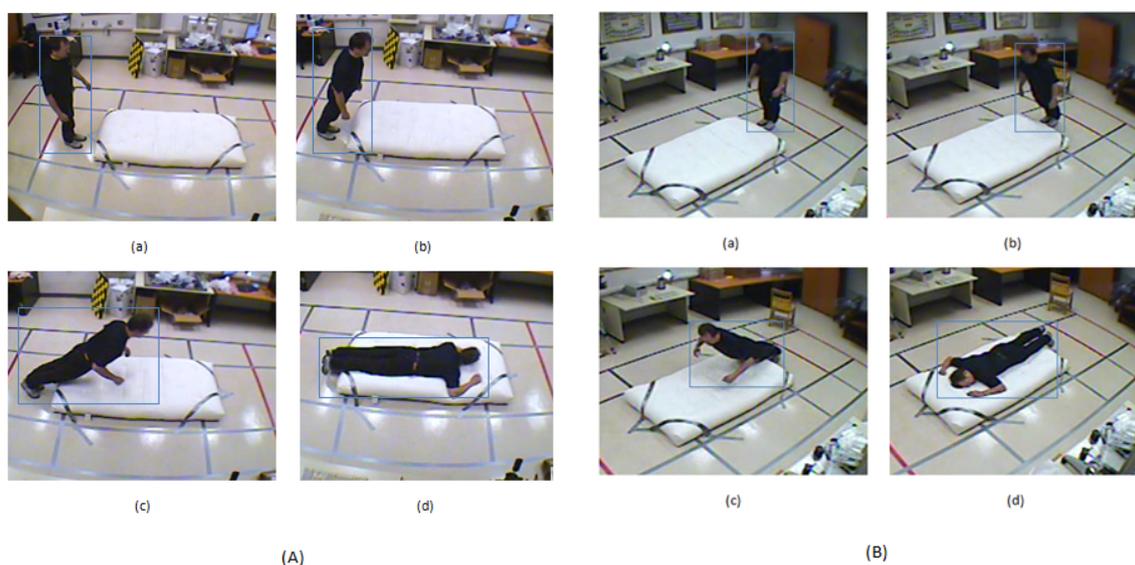


Figure 1. Temporal variation of the aspect ratio (a_r) of a falling person in the Multiple Camera Fall (MCF) dataset where $a_r = \text{width}/\text{height}$ of bounding box of a foreground blob. The person angle (p_a), shape deformation (s_d), and ellipse ratios (e_r) are also shown for each case. **(A)** Camera 7, video 1: **(a)** $a_r = 0.45$, $p_a = 90^\circ$, $s_d = 880.93$, $e_r = 2.02$, **(b)** $a_r = 0.48$, $p_a = 68^\circ$, $s_d = 670.35$, $e_r = 2.17$, **(c)** $a_r = 1.45$, $p_a = 27^\circ$, $s_d = 435.39$, $e_r = 1.85$, **(d)** $a_r = 2.85$, $p_a = 0^\circ$, $s_d = 3252.91$, $e_r = 0.43$; **(B)** Camera 3, video 3, **(a)** $a_r = 0.44$, $p_a = 90^\circ$, $s_d = 580.99$, $e_r = 2.12$, **(b)** $a_r = 0.52$, $p_a = 55^\circ$, $s_d = 356.45$, $e_r = 2.05$, **(c)** $a_r = 1.46$, $p_a = 27^\circ$, $s_d = 193$, $e_r = 1.42$, **(d)** $a_r = 2.8$, $p_a = 0^\circ$, $s_d = 2152.48$, $e_r = 0.38$.

Figure 1 shows a comparison of different proposed feature values for two different camera views. The range of feature values despite significant view changes remained almost the same. Figure 2 shows a comparison of the temporal variation of different features in two different views. Despite significant variations in the camera viewing angle, the shape of the temporal variation remained almost the same. Both figures show that the feature values and temporal variation of values remained almost unchanged regardless of the camera viewing angle. This is the main reason for the consistent performance of the proposed algorithm across multiple camera views.

Similarly, a different camera view is shown in Figure 2. The behaviour of our features is equally good in this new figure with different camera views; this shows that our features exhibit comparable performance even with different views.

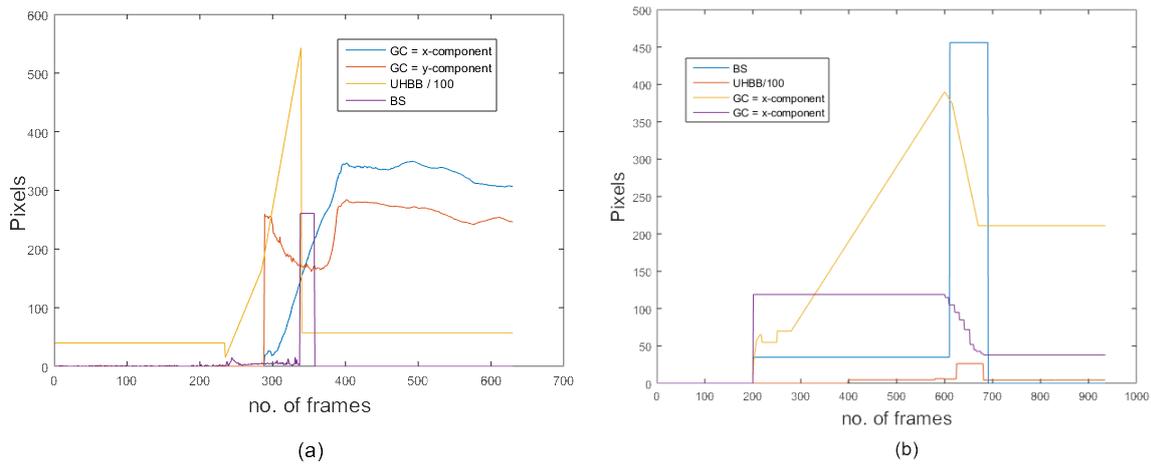


Figure 2. The temporal variation in body speed (B.S.), upper half area of the bounding box (U.H.B.B), and x -component and y -component of the geometric centre (G.C.) of a falling person on the MCF dataset: (a) camera 2, video 2; (b) camera 3, video 3.

3.3. Training Boosted J48 Classifier

Classification is the task of finding a target function that maps an attribute set to a particular class among a predefined set of classes. Various classifiers including SVM, neural networks, rule-based methods, prototype methods and exemplar-based methods exist in the literature. In the current work, we trained an ensemble of J48 classifiers, which we named ‘Boosted J48’. We prefer this classification strategy over the other existing classifiers mainly because of its speed and accuracy.

Boosting is a method for combining multiple classifiers [42,43]. As the name suggests, it is a meta algorithm that is used to improve the results of the base classifier. In our case, the base classifier was J48 which is an extension of the ID3 algorithm that generates rules for the prediction of target variables [44]. The additional features of J48 include finding missing values, derivation of rules, continuous attribute value ranges, and decision tree pruning.

Boosting works sequentially, whereby the first algorithm is trained on the entire dataset, and then the rest of the algorithms are developed by fitting the residuals of the first algorithm. In this process, higher weight is given to the observations that have been poorly predicted by the previous model. Boosting is known to be sensitive to noisy data and outliers. The reason for this is that boosting overfits noisy datasets. Boosting on stable algorithms like J48 improves performance, while boosting on unstable algorithms, like MLP, may reduce performance [45].

In each iteration, the base classifier is used with a different weight over the samples of the training set. At each iteration, the computed distributions assign more weight to the incorrectly classified samples. The final classifier is obtained as a weighted average of the previous hierarchical designed classifier. We focused on the two-class classification task, where the training set is $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, x_i is some feature vector and $y_i \in \{-1, 1\}$, $i = 1, 2, \dots, N$ is the label. The aim was to design an optimal classifier to predict the label of a test feature vector, x_t .

$$\hat{y} = \text{sign}\{F(x_t)\} \quad (9)$$

where \hat{y} is the predicted label and

$$F(x_t) = \sum_{k=1}^K \alpha_k \phi(x_t; \theta_k), \quad (10)$$

where $\phi(x_t; \theta_k)$ denotes the base classifier that returns a binary class label, $\{-1, 1\}$. The corresponding parameter vector, θ_k , describes the base classifier. An important property of boosting is its relative immunity to overfitting with an increasing K . It has been verified that even with a high number of

terms, K , and consequently, a high number of parameters, the error rate on a test set does not increase but keeps decreasing and finally, reaches a low, asymptotic value.

Random Decision Forests (RDF) and J48 are both tree-based classifiers. RDF is a mixture of tree predictors, where each tree is a predicate of the values of a random vector sampled autonomously and all the trees of the forest have the same distribution [46]. As the trees in a forest become large in number, the generalization error for the forest converges to a certain limit. The votes from all trees determine the class assignments. The main limitation of RDF is its increased complexity with an increasing depth of trees when training RDF, compared to J48. That is, RDF requires more computational resources and has higher memory complexity. The learning rate of RDF is slower and its prediction process also has more computational complexity compared to an equivalent J48 classifier.

As discussed, J48 is an extension of the ID3 algorithm that generates rules for the prediction of target variables [44]. The additional features of J48 include finding missing values, the derivation of rules, continuous attribute value ranges, decision tree pruning, etc. J48 is based on the information gain ratio that is evaluated by entropy. The information gain ratio measure is used to choose the test features (attributes) at each node in the tree. The attribute with the highest information gain ratio is selected as the test feature for the current node. If we have a feature, X , and we examine the values for this feature in the training set and they are in increasing order, A_1, A_2, \dots, A_m , then for each value, $A_j, j = 1, 2, \dots, m$, the records are partitioned into 2 sets: the first set includes the X values up to and including A_j and the second set includes the X values greater than A_j [47]. For each of these m partitions, the $GainRatio(X(j), T)$ where $j = 1, 2, \dots, m$ is computed, and the partition that maximizes the gain is chosen. The $GainRatio(X, T)$ is given in Equation (11):

$$GainRatio(X, T) = \frac{Gain(X, T)}{SplitInfo(X, T)}. \quad (11)$$

Considering the information content of a message that indicates not the class to which the case belongs, but the outcome of the test on feature X , the $SplitInfo$ is given by Equation (12):

$$SplitInfo(X, T) = -\sum_i^n \frac{|T_i|}{|T|} \log_2 \frac{|T_i|}{|T|}. \quad (12)$$

The $GainRatio(X, T)$ is thus the proportion of information generated by the split that is useful for the classification.

4. Experiments and Results

4.1. Datasets

We performed experiments on two publicly available datasets: the Multiple Cameras Fall (MCF) dataset (<http://www.iro.umontreal.ca/~labimage/Dataset/>) and the UR Fall Detection (URFD) dataset (<http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html>). Table 1 contains the descriptions of 20 videos from the MCF dataset captured at camera position 2. The details mentioned in Table 1 include the video number, total frames in each video, frames with falls and frames without falls. Figure 1 shows some example frames from the MCF dataset from video 1 (camera 7) and video 3 (camera 3) respectively containing various states of a falling person.

The UR Fall Detection (URFD) dataset contains frontal and overhead video sequences obtained by two Kinect sensors, with one placed at the height of 1 m from the floor and the other mounted on the ceiling with a height of 3 m. The dataset contains two kinds of falls that were performed by five people—from standing position and from sitting on the chair. The dataset was recorded at 30 frames per second. The frontal sequence contains 314 frames, in which 74 frames contain falls and 240 frames have no falls. The key frames of the URFD dataset are shown in Figure 3. The overhead sequence contains a total of 302 frames, in which 75 frames contain falls, while 227 have no falls.

Table 1. Details of the Multiple Cameras Fall (MCF) Dataset: total number of frames, number of frames containing a fall, and number of frames containing no fall in each video.

Videos	Total Frames	Fall	No Fall	Videos	Total Frames	Fall	No Fall
Video 1	1411	314	1097	Video 11	1486	608	878
Video 2	756	331	425	Video 12	1041	420	621
Video 3	883	272	611	Video 13	1240	360	880
Video 4	1033	409	624	Video 14	970	385	585
Video 5	600	266	334	Video 15	1007	367	640
Video 6	1203	513	690	Video 16	1023	320	703
Video 7	912	290	622	Video 17	992	432	560
Video 8	700	240	460	Video 18	1217	627	590
Video 9	905	360	545	Video 19	1155	390	765
Video 10	813	230	583	Video 20	2328	90	2238



Figure 3. Selected frames from the frontal sequence of the URFD dataset.

In all experiments, 10-fold cross-validation was used. For the purpose of temporal variance computation in Equations (1)–(8), different temporal window values, k , were used. All results reported in the paper are for $k = 30$. Since the video frame rate was 30 fps, $k = 30$ corresponds to a duration of 1 s.

4.2. Comparison with Existing Approaches

The proposed algorithm was compared with existing approaches including Kepski et al. [38] using KNN and SVM, Debard (De) [26], Debard Kyrkou (DeKy) [26,30], Debard Foroughi (DeFo) [26,48], Osuna (Osu) [27], Kyrkou (Ky) [27,30] and Foroughi (Fo) [27,48].

The proposed approach, 'PA_B-J48', uses the temporal variance of motion and geometric features with an ensemble learning strategy of 'boosting with J48'. The second proposed approach, 'PA_Ada', uses the same set of features but with the Adaptive Boosting (AdaBoost) classifier. AdaBoost is an ensemble learning approach based on game theory with a main aim of combining many weak classifiers to produce a strong classifier.

Adaboost is an iterative algorithm that is used in conjunction with many other types of learning algorithms (weak learners) to improve their performance [49]. The output of weak learning algorithms is combined into a weighted sum that forms the final output of the boosted classifier. Adaboost, short for adaptive boosting, tweaks the weak learners in favor of misclassified instances by the previous classifiers. The prior knowledge of the lower bound of prediction accuracy of weak learning algorithms is not required; hence, Adaboost is suitable for many practical purposes. The algorithm is sensitive to noisy data and outliers.

The proposed algorithms work better than deep learning due to various reasons. Deep networks require very large training datasets containing millions of images to achieve good performances. Due to the limited training data being used in this study, our proposed algorithms were more suitable than the deep learning approaches.

Deep networks are computationally expensive as they require high-end GPUs to be trained on large datasets. Expensive GPUs, fast CPU, SSD storage and large RAM significantly increase the hardware cost and computational complexity of deep networks. Hence, it is not feasible to train these deep networks on the current systems (64-bit machine with Intel core i3-3110M CPU @2.40GHz, and 4GB RAM) on which our proposed algorithm gives more than 99% percentage accuracy for various data sets.

Classical machine learning algorithms are easier to interpret and understand compared to deep learning methods. Due to thorough understanding of data and algorithms, it is easy to tune hyper-parameters and to change the design of a model. Deep learning networks have often been used like a “black box”.

4.3. Performance Measures

In addition to accuracy, we used sensitivity and specificity for a comparison of our system with existing systems. These measures have also been used by other fall detection systems [50].

$$\text{Sensitivity} = \frac{TP}{(TP + FN)} \quad (13)$$

$$\text{Specificity} = \frac{TN}{(TN + FP)} \quad (14)$$

where true positive (TP) is the number of falls correctly identified by the system, and false negative (FN) is the number of falls missed by the system, and true negative (TN) is the number of ‘no falls’ correctly identified by the system, and false positive (FP) is the number of ‘no falls’ missed by the system.

4.4. Experiments on the MFC Dataset

The performance of the proposed fall detection system was evaluated on the MFC dataset in three different experiments and compared with the existing state-of-the-art approaches. These experiments are discussed below.

4.4.1. Experiment 1

In the first experiment, frames of all 20 videos were combined and 10-fold cross validation was used to train and test the proposed system. All test frames were classified as fall or no fall. The resulting accuracy, sensitivity, and specificity of the existing as well as proposed methods are shown in Figure 4. The proposed PA_B-J48 has previously been shown to obtain larger accuracy than all compared methods. The proposed PA_Ada and the existing algorithms were not able to efficiently handle within class variations. A possible reason for the accuracy degradation of PA_Ada is overfitting [51]. Adaptive boosting uses a training set over and over and hence, is more prone to overfitting.

Figure 4 shows the ROC curves of the existing approaches, including De, DeKy, DeFo, Osu, Ky, Fo and proposed approaches ‘PA_B-J48’ and ‘PA_Ada’ on MCF dataset (Experiment 1). It may be observed that ROC curve of the proposed approach PA_B-J48 is better than the existing approaches.

Table 2 shows the percentage accuracy of existing approaches, including De, DeKy, DeFo, Osu, Ky, Fo, and the proposed approaches ‘PA_B-J48’ and ‘PA_Ada’ from camera position 3 and video 3 of the MCF dataset (Experiment 1). Camera 3 was placed exactly on the opposite wall from camera 2 (<http://www.iro.umontreal.ca/~labimage/Dataset/>). It may be observed that the proposed approach ‘PA_Ada’ exhibited a percentage accuracy of 99.14% with camera position 3 which is comparable with

the percentage accuracy shown by camera position 2 (as can be seen in Table 3). Hence, our proposed algorithm performed robustly across different camera positions.

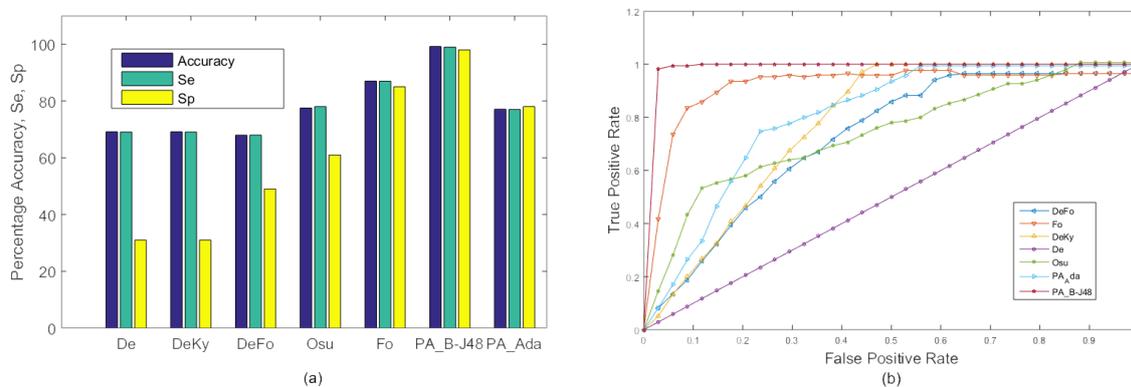


Figure 4. Experiment 1: A comparison of the existing approaches and the proposed approaches on the MCF dataset with 10-fold cross validation using two categories: ‘Fall’ and ‘No Fall’. (a) Graph; (b) ROC curve.

Table 2. Experiment 1: Comparison of the existing approaches, De, DeKy, DeFo, Osu, Ky, and Fo, with the proposed approaches ‘PA_B-J48’ and ‘PA_Ada’ in terms of % accuracy (A) at camera position 3 of video 3 (V) of the MCF dataset with 10-fold cross validation using two categories: ‘Fall’ and ‘No Fall’. Maximum value in each row is shown in bold.

		Existing Approaches					Proposed Approaches	
		De [26]	DeKy [30]	DeFo [48]	Osu [27]	Fo [48]	PA_B-J48	PA_Ada
Cam3 (V3)	A	73.23	77.09	78.37	93.47	98.39	92.4	99.14

Table 3. Experiment 2: Comparison of the existing approaches, De, DeKy, DeFo, Osu, Ky, and Fo, with the proposed approaches, ‘PA_B-J48’ and ‘PA_Ada’, in terms of % accuracy (A), % sensitivity (Se) and % specificity (Sp) for each Video (V) of the MCF dataset with 10-fold cross validation using two categories: ‘Fall’ and ‘No Fall’. Maximum value in each row is shown in bold.

		Existing Approaches					Proposed Approaches	
Video		De [26]	DeKy [30]	DeFo [48]	Osu [27]	Fo [48]	PA_B-J48	PA_Ada
V1	A	88.65	93.76	90.50	99.23	99.86	99.65	98.99
	Se	89.00	94.00	91.00	98.00	99.00	99.00	99.00
	Sp	69.00	95.00	86.00	97.00	99.00	99.00	99.00
V2	A	61.83	91.76	71.24	93.03	99.74	99.08	99.34
	Se	62.00	92.00	71.00	98.00	99.00	99.00	99.00
	Sp	49.00	85.00	61.00	97.00	99.00	99.00	99.00
V3	A	72.79	92.86	74.94	99.88	99.89	99.89	100.00
	Se	73.00	93.00	75.00	99.00	99.00	99.00	100.00
	Sp	42.00	91.00	71.00	99.00	99.00	99.00	100.00
V4	A	68.41	82.95	70.45	87.40	98.35	99.42	88.85
	Se	68.00	83.00	70.00	87.00	98.00	99.00	89.00
	Sp	31.00	55.00	64.00	77.00	98.00	99.00	70.00
V5	A	63.42	89.38	90.12	92.03	95.72	98.82	94.99
	Se	63.00	89.00	90.00	92.00	96.00	99.00	95.00
	Sp	65.00	98.00	98.00	87.00	93.00	99.00	98.00

Table 3. Cont.

Video		Existing Approaches				Proposed Approaches		
		De [26]	DeKy [30]	DeFo [48]	Osu [27]	Fo [48]	PA_B-J48	PA_Ada
V6	A	91.51	89.33	94.01	95.34	99.58	99.67	98.17
	Se	91.00	90.00	94.00	95.00	100.00	99.00	98.00
	Sp	85.00	89.00	87.00	91.00	100.00	99.00	98.00
V7	A	68.09	85.42	79.17	95.39	99.23	99.45	98.90
	Se	68.00	85.00	79.00	95.00	99.00	99.00	98.00
	Sp	32.00	62.00	76.00	89.00	99.00	99.00	98.00
V8	A	65.57	83.86	72.57	89.14	99.00	99.29	98.42
	Se	66.00	84.00	73.00	89.00	99.00	99.00	98.00
	Sp	32.00	61.00	78.00	72.00	98.00	99.00	98.00
V9	A	70.50	85.19	77.46	91.60	97.13	99.45	96.46
	Se	71.00	85.00	78.00	92.00	97.00	99.00	97.00
	Sp	52.00	90.00	74.00	97.00	95.00	99.00	76.00
V10	A	77.74	80.93	80.07	91.39	98.40	99.02	95.69
	Se	78.00	81.00	81.00	91.00	98.00	99.00	96.00
	Sp	65.00	52.00	52.00	84.00	94.00	99.00	93.00
V11	A	59.02	59.02	65.75	84.86	98.92	99.53	87.35
	Se	59.00	59.00	66.00	85.00	99.00	99.00	87.00
	Sp	41.00	70.00	55.00	84.00	99.00	99.00	95.00
V12	A	63.01	83.19	74.54	93.28	98.85	99.33	95.58
	Se	79.00	79.00	79.00	93.00	99.00	99.00	96.00
	Sp	21.00	21.00	21.00	89.00	99.00	99.00	97.00
V13	A	78.93	78.93	78.93	78.93	95.24	99.11	89.26
	Se	79.00	79.00	79.00	93.00	95.00	99.00	89.00
	Sp	21.00	21.00	21.00	6.00	84.00	97.00	74.00
V14	A	64.08	64.29	68.84	79.81	96.58	98.96	89.64
	Se	64.00	64.00	68.00	80.00	97.00	99.00	90.00
	Sp	36.00	39.00	51.00	57.00	95.00	98.00	92.00
V15	A	63.72	90.85	72.56	91.65	96.02	99.50	86.18
	Se	63.00	91.00	73.00	92.00	96.00	99.00	86.00
	Sp	40.00	70.00	72.00	96.00	95.00	99.00	87.00
V16	A	69.60	85.92	69.60	74.19	94.33	99.41	89.34
	Se	79.00	86.00	70.00	74.00	94.00	99.00	75.00
	Sp	30.00	67.00	30.00	42.00	90.00	99.00	89.00
V17	A	72.53	83.13	75.46	78.30	97.46	98.83	93.26
	Se	72.00	81.00	76.00	78.00	98.00	99.00	93.00
	Sp	27.00	70.00	47.00	49.00	93.00	98.00	86.00
V18	A	65.24	79.79	78.14	90.30	99.06	99.18	92.76
	Se	65.00	90.00	78.00	98.00	99.00	99.00	99.00
	Sp	37.00	72.00	88.00	99.00	99.00	99.00	100.00
V19	A	76.19	78.79	77.06	93.42	97.84	99.48	93.42
	Se	76.00	79.00	77.00	93.00	99.00	99.00	93.00
	Sp	48.00	52.00	56.00	88.00	55.00	99.00	92.00
V20	A	96.09	96.09	96.06	98.58	99.87	99.87	99.83
	Se	96.00	96.00	96.00	99.00	100.00	100.00	100.00
	Sp	3	3	3	65.00	96.00	99.00	96.00

In all experiments, the proposed algorithm performed equally well for different types of falls for various videos in the MCF dataset. We experimentally observed that the performance of our proposed algorithm was not greatly effected by the view variations in this dataset. The falls in these videos can

be divided into three categories with respect to the camera angle: falling towards a side or falling towards or away from the camera. In videos 1, 3, 5, 6, 9, 10, 11, 12, 14, 15, 18, 19, and 22, the person is walking from the right to the left of the scene and falls to the side with respect to camera. In all of these videos, the person falls on his chest, while in video 6, the person falls on his back. In videos 9 and 10, the person sits on a sofa before falling. Video 11 is similar to video 9, but the person falls on a sofa rather than sitting on it. In videos 12 and 14, the person is crouching or picking up stuff from the ground before falling. In videos 15, 18, and 19, the sofa is replaced by a chair. In video 22, the person slips to the side.

The second type of fall occurs in videos 2, 4, 7, 8, and 13; the person moves from the right to the left of the scene and falls towards the camera. In video 13, the person falls on a sofa as well. In video 4, the person falls twice. The third type of fall occurs in videos 16, 17 and 20; the fall occurs away from the camera. In video 16, the person moves in a circular motion around the sofa and then falls on the sofa away from the camera. In video 20, the person moves from the left to the right and then from the right to the left multiple times, picking up stuff from the floor, falling on a sofa multiple times and then falling on the floor in a direction away from the camera. The proposed algorithm performed almost the same on all these fall variations.

4.4.2. Experiment 2

The performance of the existing as well as the proposed algorithms on the MFC data set with 10-fold cross validation is shown in Table 3 and Figure 5. Due to relatively less with-in-class variations, performance of all algorithms increased. In this experiment, both proposed algorithms PA_B-J48 and PA_Ada performed quite well and obtained very high accuracy.

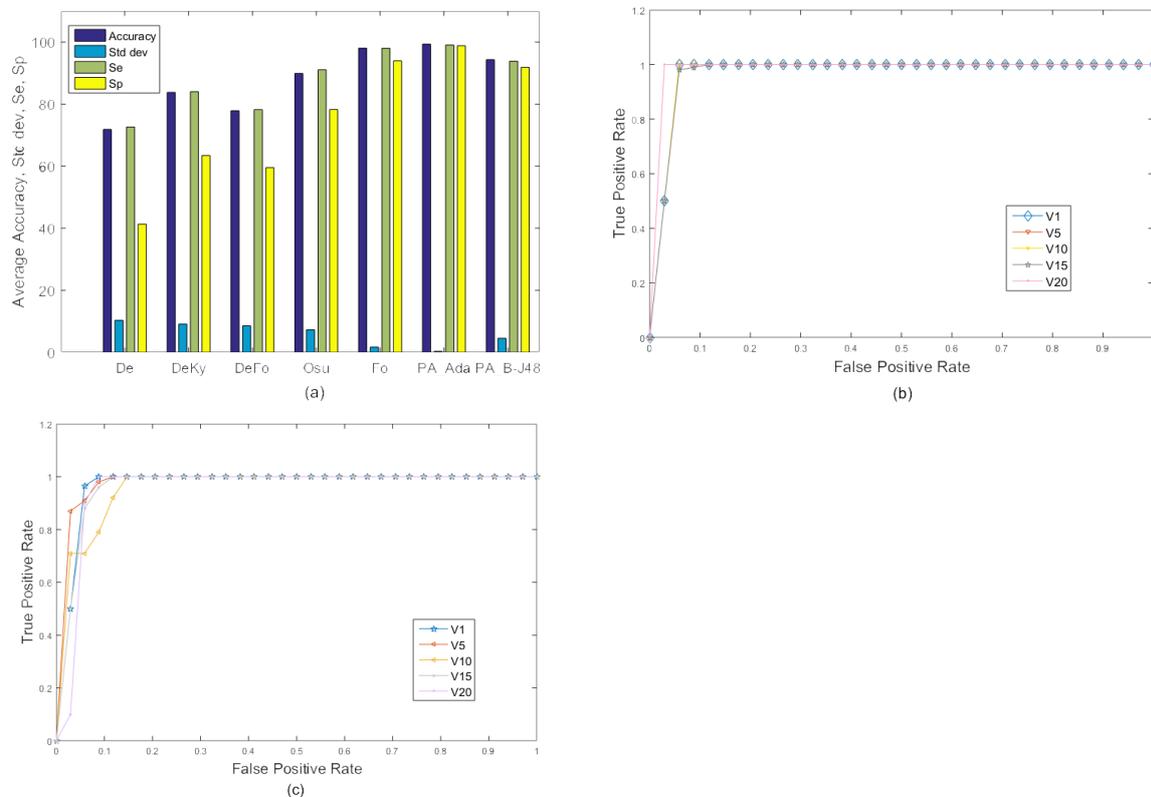


Figure 5. Experiment 2: Average comparison of the existing approaches and the proposed approaches on the MCF dataset with 10-fold cross validation using two categories: 'Fall' and 'No Fall'. (a) Graph; (b,c) ROC curve of selected videos (videos 1, 6, 10, 15, and 20) using camera 2 for the proposed approaches, PA_B-J48 and PA_Ada.

4.4.3. Experiment 3

In the third experiment, a three-class classification was performed. We added a sitting down class to evaluate the performance of the proposed algorithm in differentiating falling down from sitting down. The main difference between these two actions is the speed of performing the action. Similar to the first experiment, all 20 videos were merged together to form a single video and 10-fold cross validation was used for training and testing. Sitting down is slow and relatively gradual while falling down is rapid and relatively random. The results of this experiment are shown in Figure 6. In this experiment, the performance of PA_B-J48 remained excellent while all other methods suffered from significant accuracy degradation. This was mainly due to the similarity between the sitting and falling down classes. Moreover, adaptive boosting is more prone to overfitting large datasets and it could also serve as an important reason for the reduced performance of PA_Ada [51]. The reason for the better performance of PA_B-J48 is due to an improvement in the efficiency of the basic J48 algorithm through boosting, an ensemble learning method and more effective feature set. The J48 is a powerful decision tree method that can handle both discrete and continuous attributes. The algorithm also handles the missing values in the training data.

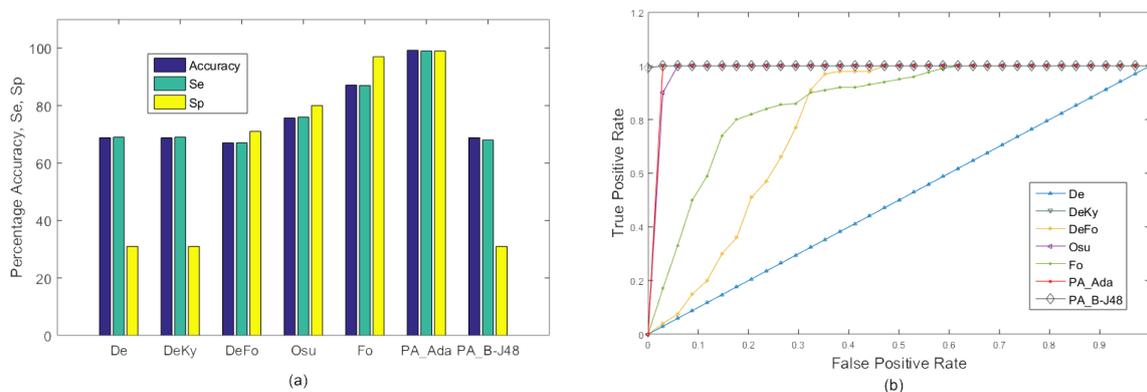


Figure 6. Experiment 3: Accuracy comparison of the existing approaches, De, DeKy, DeFo, Osu, Ky, Fo and the proposed approaches, 'PA_B-J48' and 'PA_Ada', on all videos in the MCF dataset with 10-fold cross validation using three classes i.e., 'fall', 'sitting' and 'no fall'. (a) Graph; (b) ROC curve.

4.5. Experiments on the URFD Dataset

The URFD dataset is challenging as it contains abrupt human falls and short video sequences. Moreover, the data set contains falls not only from the standing position, but also while sitting on a chair. It can be seen in the Figure 7 that our proposed approach 'PA_B-J48' outperformed the existing approaches of De, DeKy, DeFo, Osu, Fo, and Kepski et al. using KNN and SVM [38] on both frontal and overhead data sequences in terms of average accuracy, sensitivity, and specificity. Moreover, our second proposed approach also exhibited better percentage accuracy than the existing approaches of De, DeKy, DeFo, Fo, and Kepski et al. using KNN and SVM [38].

The proposed algorithm exhibited an accuracy of 99.13% on the frontal video sequence of URFD while it showed an accuracy of 99.03% of the overhead video sequence of the same dataset. This shows that accuracy degradation occurs despite significant view variation being quite minor.

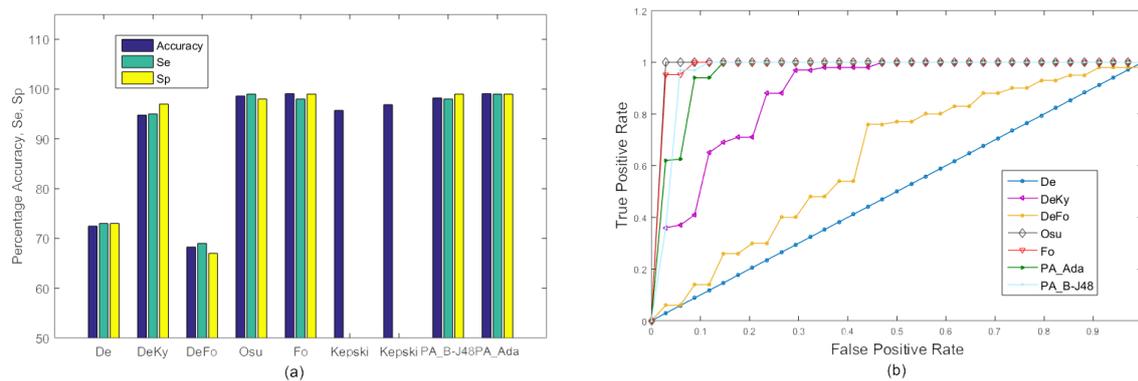


Figure 7. Experiments on the URFD dataset: comparison of accuracy, Se and Sp between the existing approaches under consideration and the proposed approach on the frontal and overhead data sequence from the URFD dataset. (a) Graph; (b) ROC curve.

4.6. Execution Time Comparison

A comparison of the execution times of the proposed approaches, PA_B-J48 and PA_Ada, with the existing approaches was performed on the MCF dataset (Experiments 1 & 3) and the URFD dataset, as shown in Table 4. All experiments were performed on a 64-bit machine with Intel Core i3-3110M CPU @2.40GHz, and 4GB RAM. In Experiment 1, the proposed PA_Ada was 9.06, 74.28, and 27.29 times faster than DeFo, Osu and Fo, respectively, while the PA_B-J48 method was 0.57, 4.72, and 1.73 times faster, respectively. In Experiment 3, the proposed PA_Ada was 4.28, 37.14, and 16.06 times faster than DeFo, Osu and Fo, respectively, while PA_B-J48 was 0.60, 5.24, and 2.27 times faster, respectively, than these methods. In both of these experiments, PA_Ada remained significantly faster than the compared algorithms. PA_B-J48 remained faster than both Osu and Fo. In both experiments, DeFo remained faster than PA_B-J48; however, its accuracy remained low. Similarly De and DeKy were faster but significantly less accurate. This shows a trade-off between accuracy and speed. The proposed algorithm uses a larger feature set than these methods to achieve a higher accuracy. By using a parallel implementation of the proposed PA_B-J48, its execution time can be significantly reduced over larger datasets. On smaller datasets such as URFD, the proposed PA_B-J48 is already among the fastest methods (Table 4).

Table 4. Comparison of execution times (seconds) of the existing approaches under consideration and the proposed approaches on the MCF dataset (Experiments 1 & 3) and on the URFD dataset. The execution time was measured on the same machine under similar operating conditions.

Experiment	Existing Approach			Proposed Approach	
	DeFo	Osu	Fo	PA_B-J48	PA_Ada
Experiment 1	16.5 s	135.19 s	49.65 s	28.63 s	1.82 s
Experiment 3	13.05 s	113.28 s	48.98 s	21.6 s	3.05 s
URFD (average)	0.12 s	0.01 s	0.32 s	0.01 s	0.025 s

5. Conclusions

An accurate and fast human fall detection system is of utmost importance for patients and aged persons for timely intervention if a fall happens, to avoid serious injuries or consequences from a fall. The work presented in this paper used the temporal variance of various discriminatory features including motion, geometric orientation and geometric location to build a fall detection system. The proposed system was trained and tested using the ensemble learning strategy and boosting with the J48 classifier as well as with the AdaBoost classifier. The proposed system was tested on two publicly available fall detection datasets and compared with eight existing algorithms. From the

experiments, it was concluded that the accuracy of the proposed system was better than existing approaches for both the datasets. The proposed system was better able to differentiate between sitting down and falling down compared to existing algorithms. The proposed system also offers a faster execution time than existing methods.

This work can be extended to further improve the performance and time efficiency of human fall detection systems, specifically in more challenging datasets containing multiple people with occlusions, the same colored clothes as that of background and multisource, non-Lambertian lighting. In addition, the proposal of more novel and robust features and the development of large video repositories will play vital roles in improving the accuracy and robustness of fall detection systems. The addition of night vision functionality to a system will be an important feature of outdoor fall detection systems.

Author Contributions: Conceptualization, S.F.A.; Methodology, S.F.A. and A.M.; Software, R.K. and S.F.A.; Validation, R.K., S.F.A. and A.M.; Formal Analysis, A.M.; Investigation, S.F.A.; Resources, M.T.H. and M.J.; Data Curation, S.F.A.; Writing—Original Draft Preparation, S.F.A.; Writing—Review & Editing, A.M.; Visualization, S.F.A.; Supervision, A.M.; Project Administration, M.T.H.; Funding Acquisition, M.J.

Funding: This work was supported by the Institute for Information & Communications Technology Promotion (IITP), grant funded by the Korea government (MSIP) (No. B0101-15-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lmberis, A.; Dittmar, A. Advanced wearable health systems and applications—research and development efforts in the European union. *IEEE Eng. Med. Biol. Mag.* **2007**, *26*, 29–33. [[CrossRef](#)]
2. Rahmani, H.; Mahmood, A.; Huynh, D.; Mian, A. Histogram of oriented principal components for cross-view action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2430–2443. [[CrossRef](#)] [[PubMed](#)]
3. Tazeem, H.; Farid, M.S.; Mahmood, A. Improving security surveillance by hidden cameras. *Multimedia Tools Appl.* **2017**, *76*, 2713–2732. [[CrossRef](#)]
4. Rimmer, J.H. Health promotion for people with disabilities: The emerging paradigm shift from disability prevention to prevention of secondary conditions. *Phys. Ther.* **1999**, *79*, 495–502. [[PubMed](#)]
5. Tinetti, M.E. Preventing falls in elderly persons. *N. Engl. J. Med.* **2003**, *2003*, 42–49. [[CrossRef](#)] [[PubMed](#)]
6. Gutta, S.; Cohen-Solal, E.; Trajkovic, M. Automatic System for Monitoring Person Requiring Care and His/Her Caretaker. U.S. Patent 6,968,294, 22 November 2005.
7. Darwish, A.; Hassanien, A.E. Wearable and implantable wireless sensor network solutions for healthcare monitoring. *Sensors* **2011**, *11*, 5561–5595. [[CrossRef](#)] [[PubMed](#)]
8. Mac Donald, C.L.; Johnson, A.M.; Cooper, D.; Nelson, E.C.; Werner, N.J.; Shimony, J.S.; Snyder, A.Z.; Raichle, M.E.; Witherow, J.R.; Fang, R.; et al. Detection of blast-related traumatic brain injury in US military personnel. *N. Engl. J. Med.* **2011**, *364*, 2091–2100. [[CrossRef](#)] [[PubMed](#)]
9. Luo, S.; Hu, Q. A dynamic motion pattern analysis approach to fall detection. In Proceedings of the 2004 IEEE International Workshop on Biomedical Circuits and Systems, Singapore, 1–3 December 2004; pp. 1–5.
10. Bourke, A.; O’Brien, J.; Lyons, G. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait Posture* **2007**, *26*, 194–199. [[CrossRef](#)] [[PubMed](#)]
11. Makhoulouf, A.; Nedjai, I.; Saadia, N.; Ramdane-Cherif, A. Multimodal System for Fall Detection and Location of person in an Intelligent Habitat. *Procedia Comput. Sci.* **2017**, *109*, 969–974. [[CrossRef](#)]
12. Casilari, E.; Santoyo-Ramón, J.A.; Cano-García, J.M. UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection. *Procedia Comput. Sci.* **2017**, *110*, 32–39. [[CrossRef](#)]
13. Zigel, Y.; Litvak, D.; Gannot, I. 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.* **2009**, *56*, 2858–2867. [[CrossRef](#)] [[PubMed](#)]
14. Doukas, C.N.; Maglogiannis, I. Emergency fall incidents detection in assisted living environments utilizing motion, sound, and visual perceptual components. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 277–289. [[CrossRef](#)] [[PubMed](#)]

15. Foroughi, H.; Aski, B.S.; Pourreza, H. Intelligent video surveillance for monitoring fall detection of elderly in home environments. In Proceedings of the 11th International Conference on Computer and Information Technology (ICCIT 2008), Khulna, Bangladesh, 24–27 December 2008; pp. 219–224.
16. Miaou, S.G.; Sung, P.H.; Huang, C.Y. A customized human fall detection system using omni-camera images and personal information. In Proceedings of the 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2 2006), Arlington, VA, USA, 2–4 April 2006; pp. 39–42.
17. Lee, T.; Mihailidis, A. An intelligent emergency response system: Preliminary development and testing of automated fall detection. *J. Telemed. Telecare* **2005**, *11*, 194–198. [[CrossRef](#)] [[PubMed](#)]
18. Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Robust video surveillance for fall detection based on human shape deformation. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 611–622. [[CrossRef](#)]
19. Rougier, C.; Meunier, J.; St-Arnaud, A.; Rousseau, J. Fall detection from human shape and motion history using video surveillance. In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07), Niagara Falls, ON, Canada, 21–23 May 2007; Volume 2, pp. 875–880.
20. Doulamis, A.; Doulamis, N.; Kalisperakis, I.; Stentoumis, C. A real-time single-camera approach for automatic fall detection. In *ISPRS Commission V, Close Range Image Measurements Techniques*; John Wiley & Sons: Hoboken, NJ, USA, 2010; Volume 38, pp. 207–212.
21. Nait-Charif, H.; McKenna, S.J. Activity summarisation and fall detection in a supportive home environment. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), Cambridge, UK, 23–26 August 2004; Volume 4, pp. 323–326.
22. Tao, J.; Turjo, M.; Wong, M.F.; Wang, M.; Tan, Y.P. Fall incidents detection for intelligent video surveillance. In Proceedings of the 2005 Fifth International Conference on Information, Communications and Signal Processing, Bangkok, Thailand, 6–9 December 2005; pp. 1590–1594.
23. Mundher, Z.A.; Zhong, J. A real-time fall detection system in elderly care using mobile robot and kinect sensor. *Int. J. Mater. Mech. Manuf.* **2014**, *2*, 133–138. [[CrossRef](#)]
24. Sumiya, T.; Matsubara, Y.; Nakano, M.; Sugaya, M. A mobile robot for fall detection for elderly-care. *Procedia Comput. Sci.* **2015**, *60*, 870–880. [[CrossRef](#)]
25. Foroughi, H.; Rezvanian, A.; Pazirae, A. Robust fall detection using human shape and multi-class support vector machine. In Proceedings of the Sixth Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP'08), Bhubaneswar, India, 16–19 December 2008; pp. 413–420.
26. Debar, G.; Karsmakers, P.; Deschodt, M.; Vlaeyen, E.; Van den Bergh, J.; Dejaeger, E.; Milisen, K.; Goedemé, T.; Tuytelaars, T.; Vanrumste, B. Camera based fall detection using multiple features validated with real life video. In *Workshop Proceedings of the 7th International Conference on Intelligent Environments*; IOS Press: Amsterdam, The Netherlands, 2011; Volume 10, pp. 441–450.
27. Yu, M.; Yu, Y.; Rhuma, A.; Naqvi, S.M.R.; Wang, L.; Chambers, J.A. An online one class support vector machine-based person-specific fall detection system for monitoring an elderly individual in a room environment. *IEEE J. Biomed. Health Inform.* **2013**, *17*, 1002–1014. [[PubMed](#)]
28. Bosch-Jorge, M.; Sánchez-Salmerón, A.J.; Valera, Á.; Ricolfe-Viala, C. Fall detection based on the gravity vector using a wide-angle camera. *Expert Syst. Appl.* **2014**, *41*, 7980–7986. [[CrossRef](#)]
29. Ni, B.; Nguyen, C.D.; Moulin, P. RGBD-camera based get-up event detection for hospital fall prevention. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; pp. 1405–1408.
30. Kyrkou, C.; Theocharides, T. A flexible parallel hardware architecture for AdaBoost-based real-time object detection. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2011**, *19*, 1034–1047. [[CrossRef](#)]
31. Kerdegari, H.; Samsudin, K.; Ramli, A.R.; Mokaram, S. Evaluation of fall detection classification approaches. In Proceedings of the 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lumpur, Malaysia, 12–14 June 2012; Volume 1, pp. 131–136.
32. Gjoreski, H.; Lustrek, M.; Gams, M. Accelerometer placement for posture recognition and fall detection. In Proceedings of the 2011 7th International Conference on Intelligent environments (IE), Nottingham, UK, 25–28 July 2011; pp. 47–54.
33. Kansiz, A.O.; Guvensan, M.A.; Turkmen, H.I. Selection of time-domain features for fall detection based on supervised learning. In Proceedings of the World Congress on Engineering and Computer Science, San Francisco, CA, USA, 23–25 October 2013; pp. 23–25.

34. Kaur, R.; Gangwar, R. A Review on Naive Bayes's (NB), J48 and K-Means Based Mining Algorithms for Medical Data Mining. *Int. Res. J. Eng. Technol.* **2017**, *4*, 1664–1668.
35. Kapoor, P.; Rani, R.; JMIT, R. Efficient Decision Tree Algorithm Using J48 and Reduced Error Pruning. *Int. J. Eng. Res. Gen. Sci.* **2015**, *3*, 1613–1621.
36. Shi, G.; Zhang, J.; Dong, C.; Han, P.; Jin, Y.; Wang, J. Fall detection system based on inertial mems sensors: Analysis design and realization. In Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Shenyang, China, 8–12 June 2015; pp. 1834–1839.
37. Guvensan, M.A.; Kansiz, A.O.; Camgoz, N.C.; Turkmen, H.; Yavuz, A.G.; Karsligil, M.E. An Energy-Efficient Multi-Tier Architecture for Fall Detection on Smartphones. *Sensors* **2017**, *17*, 1487. [[CrossRef](#)] [[PubMed](#)]
38. Kepski, M.; Kwolek, B. Embedded system for fall detection using body-worn accelerometer and depth sensor. In Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, Poland, 24–26 September 2015; Volume 2, pp. 755–759.
39. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Background-Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE Trans. Image Process.* **2017**, *26*, 5840–5854. [[CrossRef](#)] [[PubMed](#)]
40. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Spatiotemporal Low-rank Modeling for Complex Scene Background Initialization. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *28*, 1315–1329. [[CrossRef](#)]
41. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Superpixels-based Manifold Structured Sparse RPCA for Moving Object Detection. In Proceedings of the British Machine Vision Conference (BMVC 2017), London, UK, 4–7 September 2017.
42. Mahmood, A. Structure-less object detection using adaboost algorithm. In Proceedings of the International Conference on Machine Vision (ICMV 2007), Islamabad, Pakistan, 28–29 December 2007; pp. 85–90.
43. Mahmood, A.; Khan, S. Early terminating algorithms for Adaboost based detectors. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 1209–1212.
44. Kaur, G.; Chhabra, A. Improved J48 classification algorithm for the prediction of diabetes. *Int. J. Comput. Appl.* **2014**, *98*, 13–17. [[CrossRef](#)]
45. Tanwani, A.K.; Afridi, J.; Shafiq, M.Z.; Farooq, M. Guidelines to select machine learning scheme for classification of biomedical datasets. In Proceedings of the European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, Tübingen, Germany, 15–17 April 2009; pp. 128–139.
46. Devasena, C.L. Comparative Analysis of Random Forest REP Tree and J48 Classifiers for Credit Risk Prediction. In Proceedings of the International Conference on Communication, Computing and Information Technology (ICCCMIT-2014), Chennai, India, 12–13 December 2015.
47. Freund, Y.; Schapire, R.; Abe, N. A short introduction to boosting. *J.-Jpn. Soc. Artif. Intell.* **1999**, *14*, 771–780.
48. Foroughi, H.; Naseri, A.; Saberi, A.; Yazdi, H.S. An eigenspace-based approach for human fall detection using integrated time motion image and neural network. In Proceedings of the 9th International Conference on Signal Processing (ICSP 2008), Beijing, China, 26–29 October 2008; pp. 1499–1503.
49. Lv, T.; Yan, J.; Xu, H. An EEG emotion recognition method based on AdaBoost classifier. In Proceedings of the Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 6050–6054.
50. Wang, S.; Chen, L.; Zhou, Z.; Sun, X.; Dong, J. Human fall detection in surveillance video based on PCANet. *Multimedia Tools Appl.* **2016**, *75*, 11603–11613. [[CrossRef](#)]
51. Dietterich, T.; Bishop, C.; Heckerman, D.; Jordan, M.; Kearns, M. *Adaptive Computation and Machine Learning*; The MIT Press: Cambridge, MA, USA; London, UK.

