IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 5, 2019, accepted September 16, 2019, date of publication October 9, 2019, date of current version November 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946522

Fall Detection for Elderly People Using the Variation of Key Points of Human Skeleton

ABDESSAMAD YOUSSFI ALAOUI, SANAA EL FKIHI, AND RACHID OULAD HAJ THAMI
IRDA Team, ADMIR Laboratory, Rabat IT Center, ENSIAS, Mohammed V University, Rabat 10000, Morocco

Corresponding author: Abdessamad Youssfi Alaoui (abdessamad.youssfialaoui@gmail.com)

This work was supported by the ANGEL PROJECT: Video Surveillance of Elderly People through the CNRST and MESRSFC.

ABSTRACT In the area of health care, fall is a dangerous problem for aged persons. Sometimes, they are a serious cause of death. In addition to that, the number of aged persons will increase in the future. Therefore, it is necessary to develop an accurate system to detect fall. In this paper, we present spatiotemporal method to detect fall form videos filmed by surveillance cameras. Firstly, we computed key points of human skeleton. We calculated distances and angles between key points of each two pair sequences frames. After that, we applied Principal Component Analysis (PCA) to unify the dimension of features. Finally, we utilized Support Vector Machine (SVM), Decision Tree, Random Forest and K Nearest Neighbors (KNN) to classify features. We found that SVM is the best classifier to our method. The results of our algorithm are as follow: accuracy is 98.5%, sensitivity is 97% and the specificity is 100%.

INDEX TERMS Fall detection, health care, human pose estimation.

I. INTRODUCTION

UNion Nation published a report in 2005 [1] which presents statistics about the population of aged people in the world. They show the percentage of older adults (60 years or over) in the past, and in the future. They cited that the proportion in 1950 is 8%, they also mentioned that this percentage grew to 11% in 2009 and they estimated that the rate would also increase to 22% in 2050. They also show that these statistics include developed, developing countries and even countries of the third world. Since, all the nations will have a growth in the population of aged persons.

Fall is one of the most dangerous and vital problems in health-care. It is one of the leading causes of unintentional injury and accidental deaths in the world. In addition to that, the World Health Organization [2] show statics about fall causes. They cited that each year, an estimated 646000 persons die from fall. However, elderly persons (65 years of age or over) represent the highest percentage of fatal falls. More than that, WHO cited that each year, 37.3 million falls, which are sever, dangerous, and require medical attention to occur. Since, we notice that falls are a big problem for the elderly, and this problem will be more severe if these persons live alone. For these reasons, we have to look for solutions to save them. And we also note that the percentage of aged persons will grow. These factors guarantee that investing in this topic will be successful. We can invest by developing surveillance systems to detect falls and send messages or signals to help fallen persons quickly.

In the last years, a lot of surveillance systems have been developed for fall detection [3]–[9]. There are a lot of proposed works to decrease injuries for aged persons. They used different support to build their algorithms. All works done in this area can be classified into three categories [3]. There are works based on the use of wearable sensors. These works used the tri-axial accelerometer. They detect posture and also generated inactivity. On the other hand, a group of researchers used ambient/ fusion to detect fall. They utilized vibrations or/and audio and video. Finally, a lot of works used vision computing. We find in this type a lot of categories. For example, there are works based on the use of body shape change, posture, head change analysis, inactivity or/and spatiotemporal approach.

In the current paper, we will present an algorithm to detect fall, by using the spatial and temporal features extracted from videos. Our work is compared to others works done in the state of the art. We evaluated our work by using two datasets. The proposed method has achieved an excellent performance. Since, we can note that the vision computing approach may be uesd in systems with high performance. Consequently, our proposed pipeline can be a good support to detect fall. We can also rescue the fallen person quickly.

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

Our paper will be organized as follows. Sections 2 contains some of works cited in state of art; section 3 shows our proposed approach. Performance of our algorithm and discussion will be in section 4. Finally, we conclude our paper and we present our perspectives in section 5.

II. RELATED WORK

In the current years, a lot of works were done in state of the art to detect fall [3]–[9]. We can find that different materials were used to detect fall. By the way, existing approaches can be presented and classified into three categories [3]. Firstly, we find a fall detection system which used wearable sensors [10]–[15], they utilized tri-axial accelerometer to generate data. The goal of these methods is detecting the posture and the inactivity of the person. The second category of works uses ambient/fusion. They detect fall by using vibrations, audios, and videos. Finally, we find computing vision category, it used videos filmed by surveillance cameras to detect fall, it calculated the change of body shape, posture, 3D head change analysis, inactivity and/or spatiotemporal features. Since, we can conclude that there are three types of support that we can use to detect fall, we can use wearable sensors, ambiance sensors, and/or cameras (vision).

In state of the art, a lot of works were done to detect fall by using the sensor array. In this paper, we have cited some of them [10]–[14]. In fact, they developed their algorithm by computing vectors features and applying classifiers such as KNN, MLP neural network, and recurrent neural network (LSTM, bi-LSTM, and GRU) to classify fall and non-fall. For example, Sixmith and Johnson [10] used low-resolution infrared sensor arrays ($16 * 16$ pixels). He trained his model with 108 scenarios and 10000 vectors. He utilized MLP neural networks to classify fall. He also generated an alert to send via GSM. Although, his results were not encouraging because of the Dataset used. Mashiyama *et al.* [13] also utilized a low-resolution infrared array sensor. They have used the sensor on the ceiling and K-nearest neighbor to classify fall or non-fall. They have computed four features: the number of consecutive frames where motion is detected, the maximum number of pixels which were changing during the number of consecutive frames, maximum of the variance of temperature during the number of consecutive frames, and distance of maximum temperature before and after an activity. Their proposed method achieved a detection rate higher than 94%. In addition to that, Fan *et al.* [12] used Grid-Eye infrared array sensor to detect fall. They have followed two steps approach: the first one is pre-processing data filtering. They have applied Wavelet, Gaussian, and median filter. The second step is the classification. They have used deep learning models, including multi-layer perceptron's, LSTM, GRU, and GRUATT. They have also created their own dataset containing over 300 falls in multiple configurations. They found that Median filter is more accurate to the others filter and LSTM classifier is also more classifier to others classifier.

On the other hand, Taniguchi *et al.* [11] utilized two thermal array sensors. They utilized the first sensor on the ceiling and the second on the wall to acquire $16 * 16$ temperature distributions. They computed the sum of temperature and the diagram of the time series posture transition. They used the room as a model of a nursing home. The exactitude of their system were around 72.7% in the scenarios that they had used. They successfully estimated posture and detected fall. Furthermore, Taramasco *et al.* [14] used a thermal sensor array for older people who live alone. They classified fall or non-fall by applying three recurrent classifiers (bi-LSTM, LSTM, and GRU). Each classifier produced an accurate performance. But, bi-LSTM is the most performant to others classifiers. Although, their proposed method achieved drawbacks, such as uncertainty on ambient temperature and the presence of objects in the area of coverage.

In the other side, there are works done using accelerometer and sensor to detect fall [15], [16]. For example, Tolkiehn *et al.* [15] computed basic amplitude and angular features from the accelerometer sensor. They also used a pressure threshold. Recently, Shahzad *et al.* [16] developed a system to detect fall using smartphones (SPs), namely FallDroid. They exploited two-step to monitor and detect fall by using accelerometer signals. In fact, they combined between comprising of the threshold based method (TBM) and multiple kernels learning support vector machine (MKL-SVM). They also utilized others techniques to identify fall events (such as lying on a bed or sudden stop after running) and to reduce false alarms. Their system achieved the lowest false alarm rate of 1 alarm per 59 hours of usage.

In 2018, Jokanovic *et al.* [17] applied deep learning for Range-Doppler radar-bases fall detecting. They stacked between auto-encoders and regression classifier. They also used spectrograms and range maps. They verified walking, falling, bending/straightening, and sitting. Their method demonstrated the superiority of deep learning based on the use of the convolution approach and PCA-based methods to detect fall.

The sound also has used as a signal to detect fall [18], [19]. They utilized a Mel-frequency spectral coefficient. For example, Li *et al.* [18] used eight-microphone circular to track the person. Zhuang *et al.* [19] applied the Gaussian Mixture Model super vector using the fall segment. They combined their model with the supervisor model to classify audio segments into falls.

Others works done to detect fall by using computer vision techniques. All these work-based their works by using the deformation of shape into the video. More than that, they used ellipse and bounding box to surround the moving object into the video, silhouette of the person, the change of person posture into the videos and the change of motion vectors of the moving object into the videos.

For example, [20]–[22] used a silhouette of the person to detect fall. Nasution and Emmanuel [20] used the projection of histograms of human body silhouette as a feature vector. They utilized histogram to distinguish between postures in standing, sitting, bending/squatting, lying on the side and lying toward the camera. They classified their features by

using a K-nearest neighbor (KNN). Lee *et al.* [21] based their work on the use of computer vision approach. They used state and geometrical orientation of the silhouette at each epoch of time t. By the way, they utilized spatial orientation and speed of the center of the silhouette as features to detect fall. In addition to that, they calculated a threshold by using the height of the silhouette. Cucchiara *et al.* [22] presented a fall detection system by using a fixed and calibrated camera. They tucked away from the background of the current frame of the video. They also removed shadows and ghost pixels to result in a good segmentation and to detect the silhouette of the person efficiently. They also computed the posture of the person by using the histogram of the projection.

On the other hand, a lot of works utilized the motion history image as a factor to detect fall. For example, Lu *et al.* [23] developed visual attention guided 30 CNN. They applied LSTM into 3DCNN for video analysis. They used 3DCNN to encode the motion features. They have utilized temporal and spatial attention to detect fall and recognize activities. Their experiments results have shown the effectivity of their proposed method to detect fall and activities recognition. Especially when they used multiple cameras fall detection dataset. Alaoui *et al.* [24], [25] used motion vectors computed of the person's silhouette using optical flow to detect fall. They also applied directional distribution called von Mises distribution.

There are also a lot of works used box bounding an ellipse to detect fall. They surrounded the moving object into the video by using box [26]–[32] or ellipse [33]–[37]. They computed the height and width of the box [26], [27], [31], [32]. More than that, Khan *et al.* [32] combined between human's height, width, and the vertical velocity. He also applied a neural network to detect fall. On the other hand, [33], [37] used the deformation of the ellipse shape as a feature to detect fall. In addition to that, [35], [37] added to the strain the angle of ellipse. But, Mirmahboub *et al.* [36] modeled the motion of the silhouette by integrating motion energy computed over a short video.

Using ellipse is insufficient to measure the posture deformation more presciently, and it is also hard to make a difference between two postures using just global information [39]. Therefore, more information is necessary to differentiate postures. For example, the projection histogram features achieved excellent performance for posture classification [40].

There are also works used the human pose estimation to detect fall. For example, Solbach and John [41] used stereo camera data and CNN to estimate the 3D human pose. They reconstructed 3D human pose. They also estimated the 3D ground plane. After that, they computed the Center of gravity (CoG) for all key points detected, and the Upper body critical (UbC) by computing the center of gravity for a subset of points. They detected fall by using CoG and UbC.

Overall, accelerometer and cameras were the most used to detect fall after 2014. But, accelerometer, Wifi or radar, and Kinect are the most used from 2014 [4].

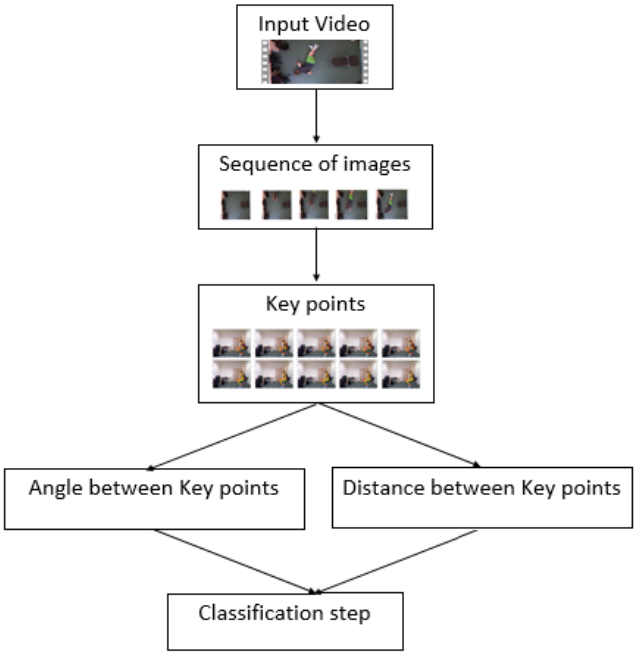

FIGURE 1. The steps which we use to implement our proposed method.

III. PROPOSED METHOD

In this section, we will present our proposed method to detect fall. Firstly, we extracted features from the videos. Indeed, extracting features is devised to two steps: we detect key points of the human body's skeleton. We used a 2D skeleton model to detect key points. Our challenge is using just simple RGB camera to detect fall. After that, we use these key points to compute the change of distance and angle between the same key points into each two pair sequential frames. After that, we apply Principal Component Analysis (PCA) to unify the size of the videos. Finally, we classify the features that we have computed to detect fall in the video.

The first step of our algorithm is detecting key point of the person into an image [42]. The result of this step is an image contains the skeleton and key points of human body. Firstly, a set of $2D$ confidence map S and $2D$ vectors fields L of part affinity are predicted simultaneously. Where S contains J confidence maps $(S_1, S_2, ..., S_J)$ and L contains C vector fields $(L_1, L_2, ..., L_C)$. Finally, the output is a $2D$ key points of people into the image. The result is computed by parsing confidence maps and affinity fields. Although, this model is utilized to predict simultaneously confidence maps and parts affinity fields. Which encodes association between parts. By the way, the network contains two branches. The first is used to detect the confidence maps, and the second is utilized to detect the part affinity fields. Each branch represents an architecture to predict with a number of iterations.

A convolution neural network is used to analyze the image (10 layers of VGG-19 [43] initialize network and fine-tuned). On the other hand, the feature map F of the input image is the input of each branch. The first branch produces confidence

maps $S' = \varrho'(F)$ and the second result affinity fields $L' = \phi'(F)$. Parameters ϱ' and ϕ' are a CNNS inference at epoch 1. The prediction from both branches, in the previous stage and the original feature map F, are concatenated and used to produce predictions.

$$S^t = \varrho^t(F, S^{t-1}, L^{t-1}), \quad \forall t \geq 2 \tag{1}$$
$$L^t = \phi^t(F, S^{t-1}, L^{t-1}), \quad \forall t \geq 2 \tag{2}$$

where ϕ^t and ϱ^t are the CNNs for inference at epoch t.

On the other hand, to predict confidence maps and affinity fields of human body, a loss function L_2 is applied at the end of each branch across epochs. Indeed, L_2 is computed between the estimated predictions and the ground-truth map and fields. The loss function at both branches can be written as follow:

$$f_S^t = \sum_{j=1}^{J} \sum_{p} w(p).\|S_j^t(p) - S_j^*(p)\|_2^2 \tag{3}$$

$$f_L^t = \sum_{c=1}^{C} \sum_{p} w(p).\|L_c^t(p) - L_c^*(p)\|_2^2 \tag{4}$$

where S_j^* and L_c^* are ground-truth confidence maps and affinity fields respectively. w is used to regularize the true positive predictions during training. At each stage, The intermediate supervision addresse,s vanishing gradient problem by replenishing the gradient periodically.

Overall, the objective loss function is computed by the following equation which used the sum between f_L^t and f_S^t at all stages.

$$f = \sum_{t=1}^{T} (f_S^t + f_L^t) \tag{5}$$

On the other hand, the confidence map detect the position of the person in the image. Indeed, a peak exists if the image contains a person. In addition to that, body parts are also extracted. And affinity fields are used to associate each pair body part. Therefor, each pair affinity formed two key points. And if we associate all the parts affinity fields of the body, the skeleton of the person will be made. By the way, key points of the person will be detected, and these key points will be used to represent the person. We just utilize the key points of the person, if we want to detect the position of the body in the video. We can also use the key points to apply our algorithm. In fact, we use videos to detect fall of the elderly person. Thus, we have to detect key points for each image in the video, e.g. we have to detect the position of the person in each image in the video. Then, for each image containing the person we will produce a set of key points. We get a set of a set of key points for each video, e.g. the video for fall or non fall will be represented by a set of key points. We use spatiotemporal computing. We compute the position of each part of the person body in each image into the videos, e.g. at each moment in the video. Finally, we generate all position of each part of the person body and at each moment

FIGURE 2. This figure represents the input and the result of each step in the key point detection step: the first image represents the input image, the second represents the confidence maps, the third represents the part affinity fields and the final image represents the key point and the skeleton of the human body.

in the video. Now, we can just use the key points to detect if the person is falling or not. But, which criteria we use to detect the fall of the person? We need more information from video to detect fall of the person. For example, we have to compute the cost of body changing, e.g. how the length of body parts change when the person is falling. We can also use the orientation of the body parts during the fall of the person. Furthermore, we can use the combination between these both methods to improve a high performance.

We detect 15 key points for each image containing a person. Since, we have a matrix with $15*2$ dimension, where we stock the position of the key points in the image. On the other hand, to compute the change of the person skeleton in the video, we compute the distance between the same key points into two sequential images, e.g. we compute the distance between two matrix ($15 * 2$ dimension) which represent two sequential images. By the way, if we want to compute the variation of body position. We first detect the key points of the body in each image in the video. After that, we compute the distances between each two sequential matrices of key points. Thus, if we use a video with m images, the result will be a set of set of distance with $m - 1$ set of distances. Each distance between two sequences matrix of key points result in a vector of 15 values. Then, each video (m image) will be represented by a set of vectors ($m - 1$ vectors). And each vector contains 15 values.

$$\forall i \in \{0, .., m-1\}, I_i, I_{i+1} \in \mathbb{R}^{15*2},$$
$$Distance(I_i, I_{i+1}) = \|I_i - I_{i+1}\|_2 \tag{6}$$

where I_i and I_{i+1} are a two sequential matrix of key points.

On the other side, we also use the matrix of key points to compute the orientations of the human body during fall and during normal activities. After that, we can find the difference between the orientations in these two different cases. For this reason, we calculate the angle between the same key points

into successive frames. Thus, we compute the orientation for each key points, e.g. each part of the human body. We have in the first step a video, we compute the key points of the skeleton for each frame. If we have a video with m frame containing a person, we produce m matrix of key points. Where each key points matrix contains $15 * 2$ values. We will have a matrix of $15 * 2$ dimensions. After that, we compute the angle between each key points in two successive frames. We will get in a $m - 1$ vectors where each vector contains 15 values, and each value into the vector represents the angle between two positions of the same key point.

$$\forall i \in \{0, .., m-1\}, I_i, I_{i+1} \in \mathbb{R}^{15*2},$$
$$Angle(I_i, I_{i+1}) = atan(I_i, I_{i+1}) \quad (7)$$

where I_i and I_{i+1} are two sequential matrices of key points.

After computing distances and angles between each key points in two successive frames, we get two matrices M_1 and M_2. The first represents distances and the second represents angles. If we have a video with m frames, we compute two sets of vectors with dimension $m - 1$ where $M_1 \in \mathbb{R}_+^{m-1}$ and $M_2 \in \mathbb{R}^{m-1}$. On the other hand, we don't have the same dimension for all videos, e.g. we have the same height and width of frames but we don't have the same number of frames for all videos. Thus, the value of m is not the same for all videos. For this reason, we apply Principal Component Analysis (PCA) to unify the number of distances and angles vectors for all videos. We just take the more representative vectors for all videos, e.g. we take just the principal component in the video. We reduce the number of distances and angles vectors, and we take the same number for all videos. Then, each video will be represented by two matrices of p vectors (the first represents distances and the second represents angles). And p have the same value for all videos. We apply these computing for fall videos and non-fall videos. And we will get in two matrix with dimension $15 * p$.

To classify vectors of distances and angles, we used the following algorithms:

A. SUPPORT VECTOR MACHINE (SVM)

SVM is a supervised learning algorithm, used for classification and regression [44]. The general concept of this algorithm is defining an hyper-plan to distinguish between fall and non-fall videos. We used a D data $D = \{(x_d, y_d)|x_d \in R^m, y_d \in \{1, -1\}\}$. Where x_d is a vector of m values, 1 represents fall class and -1 represents non-fall classes. We used also Radial Basis Function as a kernel of SVM. $K(x_i, x) = exp(\frac{-\|x_i - x\|^2}{2\sigma^2})$. Where x_i is a vector support and σ is a positive float representing the high of kernel band. The hyper-plan used to classify our class is written as follows: $H(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b$. Where b is bias, α_i is weight (Lagrangian factor), x_i is a vector support and S is a set of vectors support.

B. DECISION TREE

Decision tree is a classification algorithm.It is used to set up a tree from the root to the leaf node. Where each node represent an attribute, and each branch represents a test (i.e relates one possible value for the attribute related to this branch). Classification in decision tree starts from the root to the leaf node by testing values specified in branches. There are a lot of algorithms that can be used to set up a decision tree. For our method, we use ID3 [45] algorithm. In this algorithm, they used Entropy and Gain to chose nodes in each level of tree (the chosen node is the node which has the most important gain). Entropy is written as follows: $Entropy(S) = \sum_{i=1}^{C} -p_i log_2 p_i$. Where p_i is the proportion of S related to the class i (fall or no-fall class). And Gain is written as follows: $Gain(S, A) = Entropy(S) - \sum_{v \in value(A)} \frac{|S_v|}{|S|} Entropy(S_v)$. Where $value(A)$ is all possible values for attribute A, S_v is a subset of S(collection of examples used) which the attribute A has as value.

C. RANDOM FOREST

Random Forest classifier [46] is set up by using combination between tree classifiers, where classification is realized by using the most popular class generated by tree casts. Decision tree classifier was set up by using Gain Ratio [47] or Gini Index [48]. Random Forest classifier uses Gini Index to select attributes. Equation of Gini Index can be written as follow: $Gini(p) = 1 - \sum_{k=1}^{C} p(k|p)^2$.Where $p(k|p)$ is the probability that the selected cases related to the class k.

D. K NEAREST NEIGHBORS (KNN)

KNN is one of the simplest algorithms in machine learning. It can be used for supervised and unsupervised learning. It is based on using K nearest neighbors of example to classify. we classify by taking the class which has the majority vote of K nearest neighbors. However, to compute the nearest neighbors, we can use a lot of distances (For example: Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$, Manhattan $\sum_{i=1}^{k} |x_i - y_i|$, etc...)

IV. PERFORMANCES

We have calculated distances and angles between each two sequential key points of the human skeleton. After that, we used these features to classify non-fall and fall videos.Now, we will show the performance of our proposed method. We will also present the result of applying our algorithm by using two datasets.

A. DATASETS

1) UR FALL DETECTION DATASET

URFD (University of Rzeszow Fall Detection) dataset is realized by Interdisciplinary Centre for Computational Modeling University of Rzeszow. It contains 70 (30 falls +40 activities of daily living) sequences. In addition to that, activities of daily life are filmed like sitting down, crouching down, picking-up an object from the floor, lying on the floor and

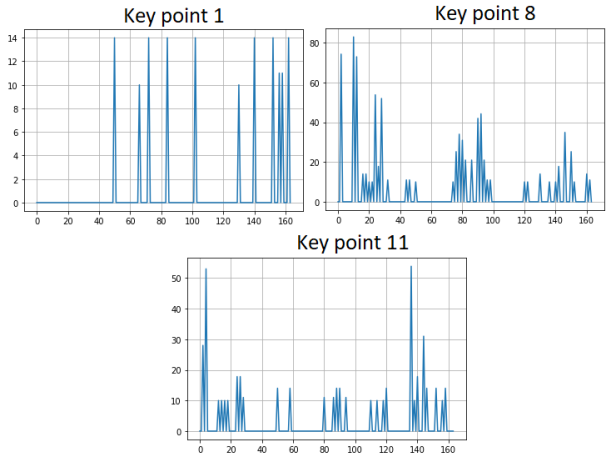

FIGURE 3. The variations of distances between the same key points in two sequential frames for a video where the person is not falling.

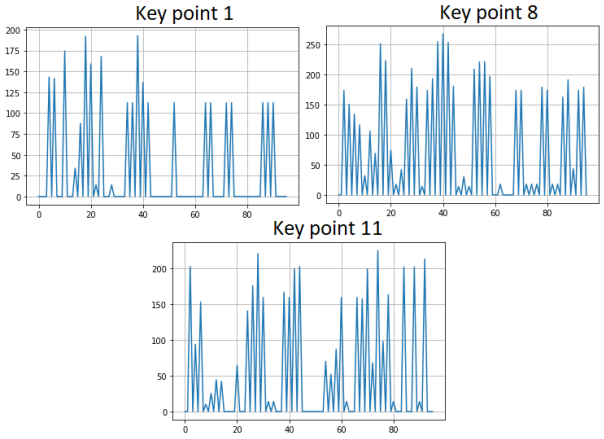

FIGURE 4. The Variations in distances between the same key points in two pairs sequential of frames for a video where the person is falling.

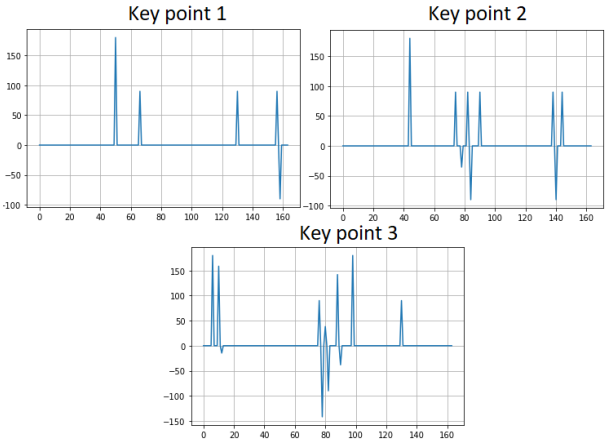

FIGURE 5. The variations of angles between the same key points in two pairs of sequential frames for a video where the person is not falling.

lying on the bed/couch. On the other hand, the number of images used in this dataset is 13000. Where, 3000 images are used for fall sequences, and 10000 images are used for ADLs sequences. More than that, this dataset contains three classes: the first one contains sequences for a person not falling, the second contains sequences for person after falling (this class not used in classification), and the third contains sequences for person during fall [49].

2) CHARFI DATASET

Charfi dataset is realized by ''Laboratoire Electronique, Informatique et Image'' (LE2I). They built dataset by using a single camera. They also filmed in different locations in an elderly environment (''home'', ''Coffee room'', ''Office'' and ''Lecture room''). They used 25 frames/s as a frame rate and $320 * 240$ as resolution. On the other hand, this dataset contains 191 videos annotated (fall and not fall videos). Each frame is annotated by the label and the location of human body(defined by using box bounding the person) [50].

B. TRAINING PROCESS

During the training process, we used the variation of distances and angles between sequential key points of the skeleton. The following figures show the variation of distances and angles during fall and non fall videos.

Figure 3 presents the variation of distances between the same key points into sequential frames. We used a video which contains a person not falling. And we computed key points of the person in each frame. After that, we computed distances between the same key points in two sequential frames. Consequently, we produced a vector that contains distances for each key point, e.g we calculated 15 vectors. After that, we presented these vectors as graphs to show the variations in distances. On the other hand, we found that, the variations in distances during time are not very large. This figure (figure 3) presents the variation of three key points (neck, right hip and left ankle). Generally, the variation of distances is not important, if we compared these variations with the variations of distances for the person who is falling (figures 4).

Figure 4 presents the variations of distances for a person during fall. We computed distances between the same key points of two sequential frames. After that, we presented these variations as a graphs. We found that, these variations are very important, if we compare these variations with the variations shown in figure 3. More than that, the variation of all key points are very important. By the way, we can conclude that, these features can produce a very performed model.

Figure 5 and figure 6 show the variations of angles between the same key points in two sequential frames. The variation of each key point is presented by a graph. These graphs illustrate the variation of non fall (figure 5) and fall (figure 6) videos. We found that, the variations are negatives and positives (e.g the change of person's direction changes the direction of the variation). But, there are a lot of peaks in a fall video and the time between these peaks are very short, if we compare the variation shown in figure 5 and figure 6. Figure 5 and

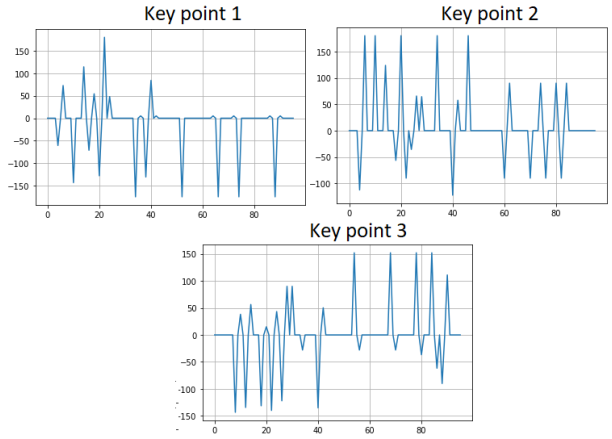

FIGURE 6. The variations of angles between the same key points in two pairs of sequential frames for a video where the person is falling.

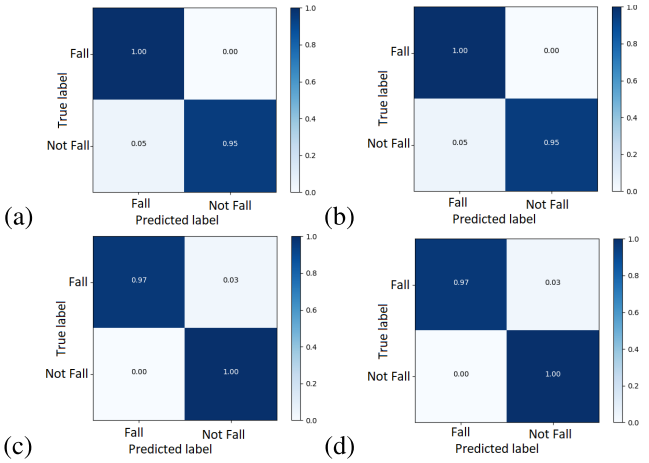

FIGURE 7. Confusion matrix for angles (a) and distances (b) using URFD dataset, angles (c) and distances (d) using L2i Charfi dataset with SVM classifier.

Figure 6 show just the variation of three key points (neck, right hip and left ankle).

C. RESULT AND DISCUSSION

After computing distances and angles, and applying classifiers to classify falls and non-fall videos. We have used three criteria to evaluate our proposed method [7]:

1) Sensitivity: to evaluate detecting falls. And compute the ratio of trues positives to the number of falls.

$$Sensitivity = \frac{TP}{TP + FN} * 100$$

2) Specificity: Compute how much our algorithm detects just falls.

$$Specificity = \frac{TN}{TN + FP} * 100$$

3) Accuracy: Compute how much our algorithm can differ between falls and non-fall videos.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} * 100$$

4) We compute also ROC (Receiver Operating Characteristics) curve. We use ROC curve to measure the probability curve of separability, how much our algorithm is capable to distinguish between falls and non-fall videos.

Where TP means that the videos present fall and our algorithm detect fall in those videos, TN refers to the videos don't contain fall and our algorithm doesn't detect fall in those videos, FN designate the videos contain falls and our algorithm doesn't detect fall in those videos and FP indicate the videos don't contain fall and our algorithm detect fall.

We have applied four classifiers using distances and angles as features. We used SVM, Decision Tree, Random Forest and KNN classifiers. We used also two datasets to test the performance of our algorithm. We computed the confusion matrices, ROC curve, Sensitivity, specificity and accuracy to evaluate our algorithm.

Figure 7 shows confusion matrices of our algorithm. The plots (b) and (d) present confusion matrices using angles with SVM classifier for Charfi and URFD datasets. For URFD dataset, our algorithm detected fall with 100% as a rate, and did not detect fall for non fall videos with 95%. For Charfi dataset, our algorithm detected fall with 97% and did not detect fall for non fall videos with 100%. On the other hand, we used distances and SVM classifier (plots (a) and (c)). We found that, our algorithm detected fall with 97% for Charfi dataset and 100% for URFD dataset. Our algorithm did not detect fall for non fall videos with 100% for charfi dataset and 95% for URFD dataset. The following figure illustrates these results.

Table 1 shows the performance (Sensitivity, specificity and accuracy) of our algorithm with four classifiers (SVM, KNN, Decision Tree and Random Forest) using URFD and Charfi datasets.We found that, the ratio of detecting all true positives fall is 100% for URFD using distances or angles, and 97% using angles and 95% using distances from Charfi dataset. Furthermore, our algorithm differs between fall and non fall videos with 97.5% for distances or angles, and 97.5% using distances and 98.5% using angles form Charfi dataset. The rate of detecting just fall videos is 95% using distances or angles. Also, we extracted features from Charfi dataset, and 97.5% using distances and 98.5% using angles from Charfi dataset.

Figure 8 contains four plots. Each plots presents the ROC curve using distances or angles form URFD or Charfi dataset. More than that, each plot contains the ROC curve of fall (red line), non fall (green line), micro average (pink line), and macro average (blue line). We utilized micro average to measure the sum of the individual true positive, false positives, and false negatives classification of our algorithm. On the other hand, we computed macro average to measure the average of the sensitivity and specificity of our algorithm. Overall, all allures are in the upper left of plots. Since, our algorithm achieved a high performance. We can also use it

TABLE 1. The performance of classifiers (Sensitivity, specificity and accuracy) Decision Tree, Random Forest, SVM and KNN using angles and distances by using URFD and Charfi datasets.

Features	Classifier	Sensitivity	Specificity	Accuracy
Angles	SVM+URFD	100	95	97.5
	Decision Tree+URFD	98	98	96
	Random Forest+URFD	97.6	97.7	98.4
	KNN+URFD	97.8	97.8	97.6
	SVM+Charfi	97	100	98.5
	Decision Tree+Charfi	98.88	98.7	97.77
	Random Forest+Charfi	81.88	73.89	73.77
	KNN+Charfi	81	73	73.3
Distances	SVM+URFD	100	95	97.5
	Decision Tree+URFD	98	98	96
	Random Forest+URFD	97.6	97.7	98.4
	KNN+URFD	97.8	97.8	97.6
	SVM+Charfi	95	100	97.5
	Decision Tree+Charfi	98	98	97
	Random Forest+Charfi	91	98	88
	KNN+Charfi	90	89	88

TABLE 2. The performance of others' work realized in the state of art and our method, using Distances or angles from URFD dataset.

Classifier	Sensitivity	Specificity	Accuracy
Ali, Syed Farooq et al. [51]	99.03-99.13	99.03	-
Kepski et al. [52]	100	96.67	95.71
Bourke et al. [53]	100	90	-
Kepski et al.(KNN) [49]	100	92.5	95
Yixiao Yun et al. [54]	96.77	89.74	-
Our method	100	95	97.5
Our method	100	95	97.5

TABLE 3. The performance of other works realized in the state of art and our method, using Distances and angles from L2i Charfi dataset.

Classifier	Sensitivity	Specificity	Accuracy
Georgios Goudelis et al. [35]	-	-	100 - 96.6
Charfi et al. [26]	73	97.7	-
M. Chamle [29]	83.47	73.07	79.31
Arisa Poonsri et al. [55]	93	64.29	86.21
Alaoui et al. [25]	94.55	90.84	90.9
Our method+Angles	97	100	98.5
Our method+Distances	95	100	97.5

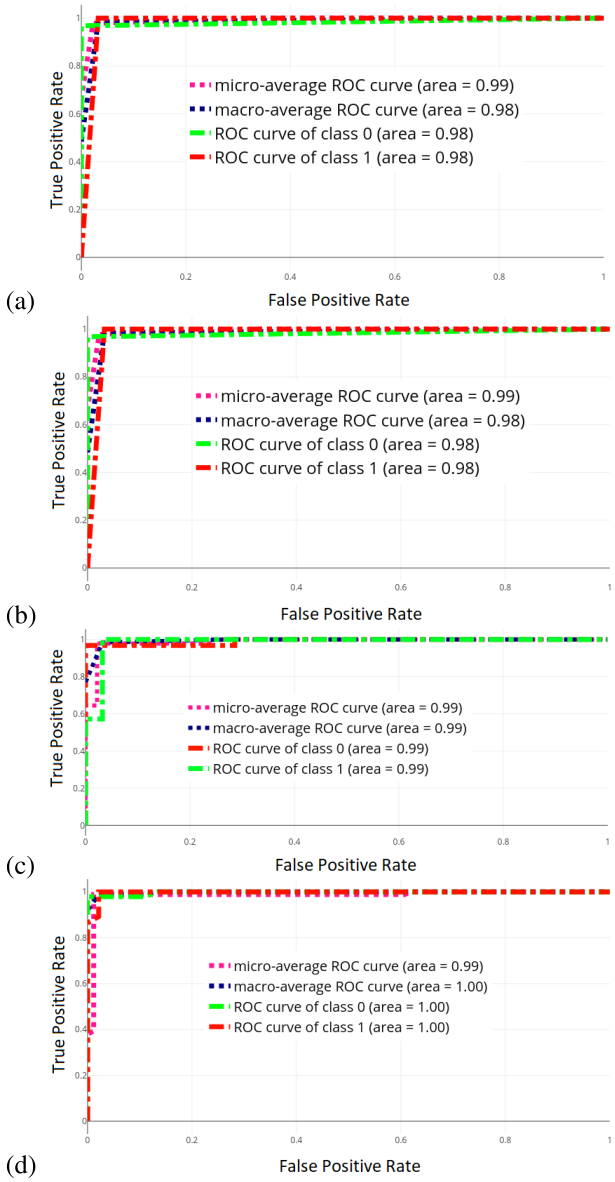

FIGURE 8. The representation of ROC curve for Angles from Charfi dataset (a), distances from Charfi dataset (b), Angles from URFD dataset (c) anddistances from URFD dataset (d) applying SVM classifier.

to develop a surveillance system, if we improve the time of processing. We have to reduce the time of computing the key points of the skeletons' person from videos.

To evaluate our algorithm, we compared our work with others' works done on the state of art. We cited some work realized using Charfi and URFD datasets. For example, table 1 shows the performance of our work and others' work utilized URFD dataset. We found that, our algorithm is performed. We can use it to realize a surveillance's system. We found also the same thing using Charfi dataset shown in table 3. We conclude that, our algorithm is performed with two different datasets.

Firstly, we annotated images used in these dataset. And we tested the performance of detecting key points. We found 99.9% as accuracy. After that, we computed accuracy, sensitivity and specificity of classifying fall and non-fall videos. And, we found results shown in the previous tables. Since, the test of our algorithm show that our proposed method is a performed work. More than that, we found also that the performance of our algorithm can be compared with works realized in the state of art. We found that, our algorithm had achieved the best accuracy using URFD dataset and one of the best accuracy using Charfi dataset. Since, our algorithm achieved an accurate method to differ between fall and non fall videos. On the other hand, in this paper,we used a machine learning classifiers. And we produced preferment results. In the future, we will use sequential classifiers.We will also look for a more performed model.

V. CONCLUSION

In this work, we have developed an algorithm to detect fall for elderly people. We have based our algorithm on using the computer-vision approach. However, we used videos to

detect fall. These videos used are set up by RGB images. In the first step, we detected key points and skeletons of the human body. After that, we calculated distances and angles between each two pairs of sequential points. We have applied also PCA to unify dimension of features. Finally, we used SVM, KNN, Decision Tree and Random Forest classifiers to classify fall and non fall videos by using generated features (distances and angles). We got a performed results using two datasets. In the future, we will use sequential classifiers to take all features in consideration. On the other hand, our algorithm can't be used in dark room. We have also to fix the camera in the same place for all time. For this reason, We will update our method using IR emitters, we will remove IR filter of the camera to detect fall in dark room. We will also improve our method to generate a better performance and reduce time processing. Because, our algorithm consomme time during execution. We will also integrate our algorithm in a real time system.

REFERENCES

[1] C. Griffiths, C. Rooney, and A. Brock, "Leading causes of death in England and Wales–how should we group causes?" *Health Statist. Quart.*, vol. 28, no. 9, pp. 1–12, 2005.
[2] S. Yoshida-Intern, "A global report on falls prevention epidemiology of falls," WHO, Geneva, Switzerland, Tech. Rep., 2007.
[3] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, Jan. 2013.
[4] T. Xu, Y. Zhou, and J. Zhu, "New advances and challenges of fall detection systems: A survey," *Appl. Sci.*, vol. 8, no. 3, p. 418, 2018.
[5] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: Providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiolog. Meas.*, vol. 25, no. 2, pp. R1–R20, 2004.
[6] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. Online*, vol. 12, p. 66, Dec. 2013.
[7] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection—Principles and methods," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 1663–1666.
[8] J. T. Perry, S. Kellog, S. M. Vaidya, J.-H. Youn, H. Ali, and H. Sharif, "Survey and evaluation of real-time fall detection approaches," in *Proc. 6th Int. Symp. High Capacity Opt. Netw. Enabling Technol. (HONET)*, Dec. 2009, pp. 158–164.
[9] Y. S. Delahoz and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19806–19842, 2014.
[10] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *IEEE Pervasive Comput.*, vol. 3, no. 2, pp. 42–47, Apr./Jun. 2004.
[11] Y. Taniguchi, H. Nakajima, N. Tsuchiya, J. Tanaka, F. Aita, and Y. Hata, "A falling detection system with plural thermal array sensors," in *Proc. Joint 7th Int. Conf. Soft Comput. Intell. Syst. (SCIS), 15th Int. Symp. Adv. Intell. Syst. (ISIS)*, Dec. 2014, pp. 673–678.
[12] X. Fan, H. Zhang, C. Leung, and Z. Shen, "Robust unobtrusive fall detection using infrared array sensors," in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 194–199.
[13] S. Mashiyama, J. Hong, and T. Ohtsuki, "A fall detection system using low resolution infrared array sensor," in *Proc. IEEE 25th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2014, pp. 2109–2113.
[14] C. Taramasco, T. Rodenas, F. Martinez, P. Fuentes, R. Munoz, R. Olivares, V. H. C. De Albuquerque, and J. Demongeot, "A novel monitoring system for fall detection in older people," *IEEE Access*, vol. 6, pp. 43563–43574, 2018.
[15] M. Tolkiehn, L. Atallah, B. Lo, and G.-Z. Yang, "Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug./Sep. 2011, pp. 369–372.
[16] A. Shahzad and K. Kim, "FallDroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 35–44, Jan. 2019.
[17] B. Jokanovic and M. Amin, "Fall detection using deep learning in range-Doppler radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 1, pp. 180–189, Feb. 2018.
[18] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate fast fall detection using gyroscopes and accelerometer-derived posture information," in *Proc. 6th Int. Workshop Wearable Implant. Body Sensor Netw.*, Jun. 2009, pp. 138–143.
[19] X. Zhuang, J. Huang, G. Potamianos, and M. Hasegawa-Johnson, "Acoustic fall detection using Gaussian mixture models and GMM supervectors," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2009, pp. 69–72.
[20] A. H. Nasution and S. Emmanuel, "Intelligent video surveillance for monitoring elderly in home environments," in *Proc. Int. Workshop Multimedia Signal Process. (MMSP)*, Crete, Greece, Oct. 2007, pp. 203–206.
[21] S.-W. Lee, Y.-J. Kim, G.-S. Lee, B.-O. Cho, and N.-H. Lee, "A remote behavioral monitoring system for elders living alone," in *Proc. Int. Conf. Control, Automat. Syst.*, Seoul, South Korea, Oct. 2007, pp. 2725–2730.
[22] R. Cucchiara, A. Prati, and R. Vezzani, "An intelligent surveillance system for dangerous situation detection in home environments," *Intell. Artif.*, vol. 1, no. 1, pp. 11–15, 2004.
[23] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 1, pp. 314–323, Jan. 2019.
[24] A. Y. Alaoui, A. El Hassouny, R. O. H. Thami, and H. Tairi, "Video based human fall detection using von Mises distribution of motion vectors," in *Proc. Intell. Syst. Comput. Vis. (ISCV)*, Apr. 2017, pp. 1–5.
[25] A. Y. Alaoui, A. El Hassouny, R. O. H. Thami, and H. Tairi, "Human fall detection using von Mises distribution and motion vectors of interest points," in *Proc. BDCA*, 2017, Art. no. 82.
[26] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Definition and performance evaluation of a robust SVM based fall detection solution," in *Proc. 8th Int. Conf. Signal Image Technol. Internet Based Syst.*, Nov. 2012, pp. 218–224.
[27] D. Anderson, J. M. Keller, M. Skubic, X. Chen, and Z. He, "Recognizing falls from silhouettes," in *Proc. 28th IEEE Eng. Med. Biol. Soc. (EMBS)*, Aug./Sep. 2006, pp. 6388–6391.
[28] C.-W. Lin and Z.-H. Ling, "Automatic fall incident detection in compressed video for intelligent homecare," in *Proc. 16th Int. Conf. Comput. Commun. Netw.*, Aug. 2007, pp. 1172–1177.
[29] M. Chamle, K. G. Gunale, and K. K. Warhade, "Automated unusual event detection in video surveillance," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Coimbatore, India, Aug. 2016, pp. 1–4.
[30] V. Vishwakarma, C. Mandal, and S. Sural, "Automatic detection of human fall in video," in *Pattern Recognition and Machine Intelligence*. Berlin, Germany: Springer, 2007, pp. 616–623.
[31] B. U. Töreyin, Y. Dedeoğlu, and A. E. Çetin, "HMM based falling person detection using both audio and video," in *Proc. IEEE Int. Workshop Hum.-Comput. Interact.*, Beijing, China, 2005, pp. 211–220.
[32] M. S. Khan, M. Yu, P. Feng, L. Wang, and J. Chambers, "An unsupervised acoustic fall detection system using source separation for sound interference suppression," *Signal Process*, vol. 110, pp. 199–210, May 2015.
[33] H. Rajabi and M. Nahvi, "An intelligent video surveillance system for fall and anesthesia detection for elderly and patients," in *Proc. 2nd Int. Conf. Pattern Recognit. Image Anal. (IPRIA)*, Mar. 2015, pp. 1–6.
[34] G. Goudelis, G. Tsatiris, K. Karpouzis, and S. Kollias, "Fall detection using history triple features," in *Proc. 8th ACM Int. Conf. Pervasive Technol. Rel. Assistive Environ.*, Corfu, Greece, 2015, Art. no. 81.
[35] B. Mirmahboub, S. Samavi, N. Karimi, and S. Shirani, "View-invariant fall detection system based on silhouette area and orientation," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 176–181.
[36] K. G. Gunale and P. Mukherji, "Fall detection using k-nearest neighbor classification for patient monitoring," in *Proc. Int. Conf. Inf. Process. (ICIP)*, Dec. 2015, pp. 520–524.
[37] Y. T. Liao, C.-L. Huang, and S.-C. Hsu, "Slip and fall event detection using Bayesian belief network," *Pattern Recognit.*, vol. 45, pp. 24–32, Jan. 2012.
[38] M. Yu, Y. Yu, A. Rhuma, S. Naqvi, L. Wang, and J. A. Chambers, "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.*, vol. 17, no. 6, pp. 1002–1014, Nov. 2013.

[39] C.-L. Liu, C.-H. Lee, and P.-M. Lin, "A fall detection system using k-nearest neighbor classifier," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 7174–7181, 2010.
[40] M. D. Solbach and J. K. Tsotsos, "Vision-based fallen person detection for the elderly," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1433–1442.
[41] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 1302–1310.
[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.
[43] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.
[44] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[45] L. Breiman, "Random forests-random features," Dept. Statist. Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. 567, 1999. [Online]. Available: ftp://ftp.stat.berkeley.edu/pub/users/breiman
[46] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA, USA: Wadsworth, 1984.
[47] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Ann. Math. Artif. Intell.*, vol. 41, no. 1, pp. 77–93, 2004.
[48] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, Dec. 2014.
[49] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki, "Optimised spatio-temporal descriptors for real-time fall detection: Comparison of SVM and Adaboost based classification," *J. Electron. Imag.*, vol. 22, no. 4, p. 17, Oct. 2013.
[50] S. F. Ali, R. Khan, A. Mahmood, M. T. Hassan, and M. Jeon, "Using temporal covariance of motion and geometric features via boosting for human fall detection," *Sensors*, vol. 18, no. 6, p. 1918, 2018.
[51] B. Kwolek and M. Kepski, "Improving fall detection by the use of depth sensor and accelerometer," *Neurocomputing*, vol. 168, pp. 637–645, Nov. 2015.
[52] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, 2007.
[53] Y. Yun and I. Y.-H. Gu, "Human fall detection via shape analysis on Riemannian manifolds with applications to elderly care," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 3280–3284.
[54] A. Poonsri and W. Chiracharit, "Fall detection using Gaussian mixture model and principle component analysis," in *Proc. 9th Int. Conf. Inf. Technol. Elect. Eng. (ICITEE)*, Oct. 2017, pp. 1–4.

ABDESSAMAD YOUSSFI ALAOUI was born in Errachidia, Drâa-Tafilalet, Morocco, in 1992. He received the License in computer science from the Sciences and Techniques Faculty, University of Moulay Ismail, Errachidia, Morocco, in 2014, and the M.Sc. degree in image processing and business intelligent from Sidi Mohamed Ben Abdellah University, Fez, Morocco, in 2016. He is currently pursuing the Ph.D. degree with the Higher National School of Computer Science and Systems Analysis (ENSIAS), Mohammed V University, Rabat, Morocco. His research interests include machine learning, deep learning, pattern recognition, and computer vision applied to the intelligent surveillance home systems and other applications.

SANAA EL FKIHI received the Ph.D. degree in computer science from Mohammed V Rabat University and the University of Science and Technology of Lille, in 2008. She is currently a Full Professor of computer engineering with the Higher National School of Computer Science and Systems Analysis (ENSIAS), Rabat IT Center, Mohammed V University, Rabat, Morocco. Her research interests include multimedia and information retrieval, image and video analysis, intelligent video surveillance, and health applications.

RACHID OULAD HAJ THAMI received the Ph.D. degree in computer science from the Faculty of Sciences Ben M'Sik Sidi Otthman, Casablanca, Morocco, in 2002. He is currently a Full Professor of computer engineering with the Higher National School of Computer Science and Systems Analysis (ENSIAS), Rabat IT Center, Mohammed V University, Rabat, Morocco. His research interests include multimedia and information retrieval, image and video analysis, intelligent video surveillance, and health applications.