Fall detection using supervised machine learning algorithms: A comparative study

Fall Detection Using Supervised Machine Learning Algorithms: A Comparative Study

Nabil Zerrouki[a], Fouzi Harrou[b]
[a] University of Sciences and Technology Houari Boumédienne, LCPTS, Faculty of Electronics and Computer Science, Algiers, Algeria
Email: nzerrouki@usthb.dz, fouzi.harrou@kaust.edu.sa

Amrane Houacine[a], and Ying Sun[b]
[b] King Abdullah University of Science and Technology, CEMSE Division, Thuwal, 23955-6900, Saudi Arabia
ahouacine@usthb.dz,

***Abstract*— Fall incidents are considered as the leading cause of disability and even mortality among older adults. To address this problem, fall detection and prevention fields receive a lot of intention over the past years and attracted many researcher efforts. We present in the current study an overall performance comparison between fall detection systems using the most popular machine learning approaches which are: Naïve bayes, K nearest neighbor, neural network, and support vector machine. The analysis of the classification power associated to these most widely utilized algorithms is conducted on two fall detection databases namely FDD and URFD. Since the performance of the classification algorithm is inherently dependent on the features, we extracted and used the same features for all classifiers. The classification evaluation is conducted using different state of the art statistical measures such as the overall accuracy, the F-measure coefficient, and the area under ROC curve (AUC) value.**

Keywords: Human fall detection, feature extraction, classification, Support vector machine.

I. INTRODUCTION

In a recent study from the Health Evidence Network for the World Health Organization [1], it is established that 30% of elderly over 65 years old falls at least once each year. In addition, falls are considered as a major cause of hospitalizations for injuries for people with special needs like elderly people, disabled, overweight and obese, etc [1]. Generally, falls are the major reason for injuries, but it also degenerate to serious wound and even to deaths (especially when elderly remain lying for longtime for hours or even days after the fall accident) [1]. The statistic studies indicated that only in United States, fall-related medical costs caused more than 8 billion dollars each year. Furthermore, more than $43 billion dollars are estimated to be allocated to fall-medical cares by 2020.

In this paper, the choice of fall detection problematic was motivated by the imminent need, and the availability of different data repositories in the literature. In other word, the recent rapid development of machine learning for pattern recognition, many researches have been exploited for posture identification-based fall detection. Actually in fall detection techniques, two approaches emerge as the most prominent: the first one is sensors-based and the second one is vision-based fall detection [2]. However, the majority cases of sensors-based methods define threshold values for triggering alarms. These threshold values are manually fixed, and no learning step is performed. This shortcoming generally caused the considerable number of false alarms.

-

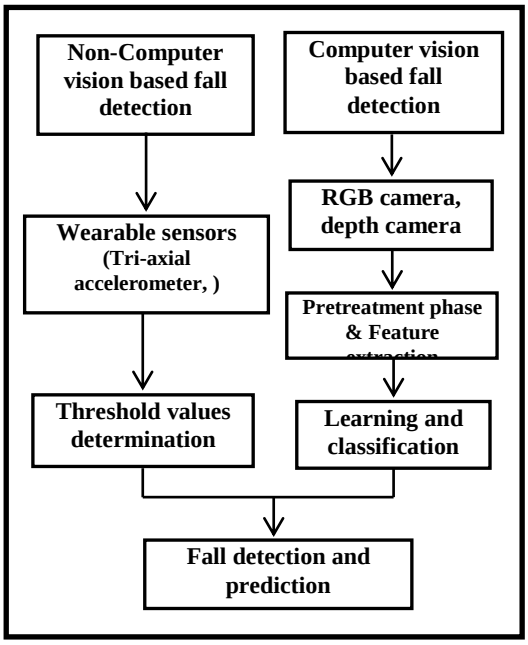

Figure 1: Fall detection approaches.

In addition, some sensors are sensitive to the presence of noise in the living environment, and then generate false alarms. In order to overcome these shortcomings, many researchers adopted the computer vision-based fall detection methods. In this section a briefly review of some existing fall detection systems is presented. In [3], Juang and Chang used a neural fuzzy network for posture classification, and when the detected posture changed from stand" to "lie" in a short time (defined by a fixed threshold), a fall activity is detected. A similar idea was proposed in [4] by Liu, except that the classifier was replaced with a more common k-nearest neighbor (KNN); moreover, statistical hypothesis testing was applied to obtain the critical time difference to separate a fall incident event from a lying down event, and a correct detection rate of 84.44% was obtained according to their experimental results. Thome et al. proposed a Layered Hidden Markov Model (LHMM) for a real-time fall detection system,

however only two kinds of activities walking and falling were concerned, and multiple cameras were needed [5]. Yu et al. introduced an improved version of SVM: a Directed Acyclic Graph SVM (DAGSVM) for posture classification and fall detection [6].

This paper presents a comparative study among several supervised classification techniques while using the video sequences during different daily and fall activities. The classification algorithms tested in this paper are: The K-Nearest Neighbors method (K-NN), the Naïve Bayes (NB), Neural Network (NN) and the Support Vector Machine (SVM). Finally, several statistical measures are conducted to evaluate the performance of each classification method on the human activity recognition.

The remainder of this paper is organized as follows: section 2, presents the image segmentation and preprocessing phases. Section 3 briefly reviews feature extraction phase. Next, section 4 is dedicated to the description of classification algorithms. Finally, comparative tests and the results of the proposed scheme are illustrated in Section 5.

II. SEGMENTATION AND PREPROCESSING

The segmentation consists of extracting body's silhouette from the input image sequence. In this work, human body is typically discriminated using background subtraction technique [7].

Figure 2: Results of the background subtraction algorithm.

The background image is defined as reference to eliminate the unchanged pixels in the frame sequences. This method is suitable here because it can manage multiple component models [7]. However, some noise regions can be observed in the segmented image. To eliminate this noise, the morphological operator, which performs the erosion and dilation operator with3 $\times$3 structuring elements, is applied.

Figure 2 illustrates an example of the background subtraction algorithm, where background and input frames are represented by the two images on the first line, respectively, and the two images on the second line, illustrate the respective results from background subtraction before and after applying morphological operators.

III. FEATURE EXTRACTION

Feature extraction is considered as crucial step in video-based fall classification, where extracted features have a direct impact on the classification performances. The extracted features have to be invariant to image (i) translation when the position of human body changes in the image, and (ii) to scaling when the silhouette dimensions change as the distance between the monitored person and camera varies.

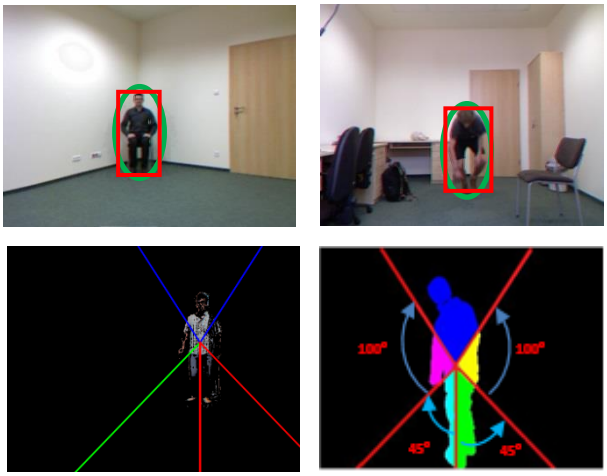

Figure 3: Features used in human posture description.

Recently, several works have focused on shape information to detect and classify falls. One can cite: the body's center of gravity [8], the bounding box and approximated ellipse of the silhouette [9]. However, most of the proposed features cannot always distinguish among various body postures, especially when there is a high degree of similarity between activities (e.g., dimensions and orientations of the bounding boxes and approximated ellipses are nearly the same for both bending and sitting postures, as shown in Figures 3 (a) and (b)). For this reason, we discard the idea to consider the body as a geometric shape. In this work, we base the extracted features on pixels constituting human body.

More specifically, we use five partial occupancy areas of the body to detect and classify falls. These areas typically correspond to the action of body parts when in a standing posture, as shown in Figures 3 (c) and (d). These areas are determined using a portioning centered on the body's gravity center. These ratios are discriminative enough to describe human body, and not too complex in order to permit a fast processing [10-12]. Figure 4 represents a time evolution of the five ratios while performing daily and fall activities respectively. Whereas, Figure 5 characterizes area's occupancy variation while performing an intentional lying activity

Finally, since frames of a video sequence are assimilated to an observation sequence, the set of ratios that are computed for each frame are then concatenated to form the whole feature vector corresponding to the video sequence.

IV. FALL CLASSIFICATION

The principle of supervised classification can be stated as follows, given a set of training data points along with associated training labels, determine the class label for an unlabeled testing data [13].

First, in training phase a model is constructed from the training data (e.g., assign the system certain fall sequences as training samples). In the testing phase, the constructed model is used to classify the rest of sequences (testing samples) according to their features. In this section, the machine learning algorithms (*k*-NN, NB, NN, and SVMs used to perform the classification task) are briefly described.

A. Support vector machine algorithm

The key idea behind the SVM algorithm is first to map the input space into a high dimensional feature space (transformed space) via a kernel function, which makes the data approximately linear, and then to build an optimal separating hyperplane in the new space to distinguish between classes [14]. The support vectors are the training data points that are closest to the separating hyperplane; these points represent the maximum-margin hyperplane for the training data.

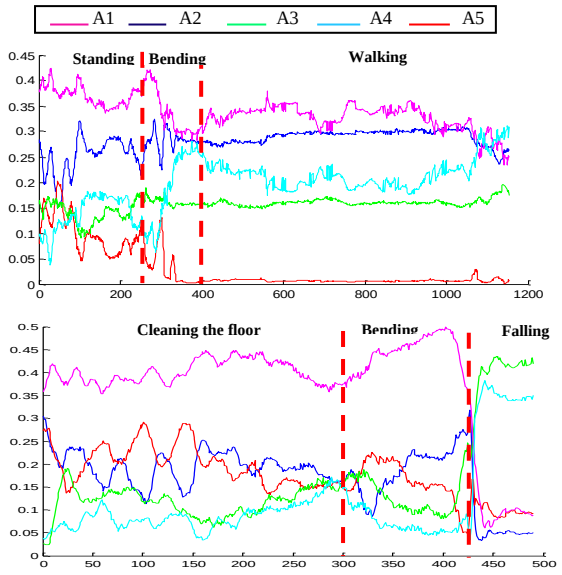

Figure 4: A sample of five ratios time evolution of daily (a) and fall (b) activities

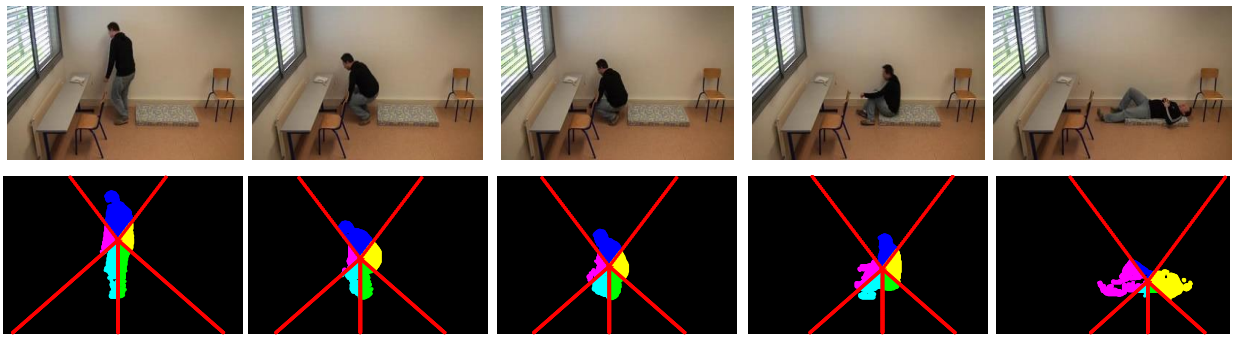

Figure 5: Samples of human body areas partitioning.

In this way, SVMs look for a maximum margin hyperplane to separate data. Further details on SVM can be found in [14].
In this study, the SVM was employed to find a function (i.e., an optimal separating hyperplane) to map each feature vector into it corresponding label space $y_k \epsilon \{+1, -1\}$, where +1 and -1 represent true fall and false fall, respectively. The optimization separation function in SVM can be defined as follow:

$$\min_{w\xi j} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \qquad (1)$$

Where C is the regularization parameter, ξ correspond to slack variables introduced which acts as a penalty term, and *w* represents the normal vector to the hyperplane. The optimal *w* is obtained as:

$$w = \sum_{i=1}^{n} y_i \alpha_i \varphi_i(x_i) \qquad (2)$$

and the decision function is given by:

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x_j) + b\right) \qquad (3)$$

where $\varphi_i(x_i)$ is the transformation applied to x_i; α_i corresponds to Lagrange multipliers, b is the bias term, and x_j is a test observation. Whereas $K(x_i, x_j)$ represents the kernel function. Various kernel functions can be used for mapping process. In this study, Radial basis function kernel has been selected.

B. Neural network classification

Neural network classifier is a supervised algorithm, which can use multiple input, output and hidden layers with arbitrary number of neurons. The most widely used neural classifier today is the Multi-layer Perceptron (MLP) network with back propagation (BP) learning algorithm. BP is used for the optimization of MLP [15].

The classic architecture of an ANN classifier includes three layer types, namely: input layers, hidden layers, and output layers. The input layers correspond generally to features to classify. The hidden layers are generally determined empirically and relatively to the expected classification accuracy. The output layer corresponds to the defined classes. Each class corresponds to a node in output layer. The output node value should provide the corresponding class for the input data, i.e. a high output value is expected on the correct class node and a low output value on all the rest.

During learning phase, a set of training input vectors is presented at the input layers by feature vectors and their corresponding desired output vectors. Initially random weights are assigned to the set of nodes. The neural network adjusts the weights attached to the connections according the difference between the network's output and the desired output for that input vector. More this difference is reduced more is better for classification. A single neuron in the network can be represented as follow:

$$y_j = f_j \sum W_{ij} \times x_i \quad (4)$$

where, x_i is data input to neural network, w_{ij} represents weights between i^{th} neuron of previous layer and j^{th} neuron of the current layer and f_j represents the activation function. Various activation functions exist in the literature, one can cite: linear, sigmoid, hyperbolic tangent.

C. K-Nearest Neighbor

The K-Nearest Neighbor classifier is a classical supervised machine learning technique. To make a classification for a test sample, a training phase is requiring. During learning stage, distance to every training example is computed. Then, keep the k closest training examples, where k represents an integer [15]. Finally, a new sample is classified by a majority vote of its neighbors. This basic method is called the KNN algorithm. The most used voting strategies in the literature are: Standard voting, Weighted voting, and Distance-based voting.

Given a test sequence *x*, its k closest neighbors $y_i \ldots y_k$ are found and vote are conducted to assign the dominant class to *x*. This class of *x* is denoted by c(x), and determined by the following equation:

$$C(x) = \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c(y_i), c(x_i)), \quad (5)$$

where $c(y_i)$ is the class of y_i. Based on a simple vote of nearest neighbors, the estimation of the belonging probability into class c is given by the following equation:

$$\hat{p}(c / x) = \frac{\sum_{i=1}^{k} \delta(c, c(y_i))}{k} \quad (6)$$

D. Naïve Bayes classifier

Naïve Bayes classifier is a probabilistic supervised machine learning technique, based on Bayes' theorem [15]. It assumes that every input data related to a class is independent of each other. So, the probability of occurrence of a class y_i, provided the input data x_i is given by:

$$P(y_i / x_i) = \arg\max_{y_i \in \{0,1\}} P(y_i) \prod_{i=1}^{m} P(x_i / y_i) \quad (7)$$

Since Naïve Bayes algorithm belongs to supervised classifiers, a training phase is required before classification. During this learning stage, the algorithm learns the conditional probability of samples from the training data. During the test phase, the classification is accomplished by computing the posterior probability $P(y_i / x_i)$ for all classes, and then predicts the class with the highest probability value. Given that an input observation x_i containing m attributes or features, the Naïve Bayes classifier assumes that the features $x_{i\ldots m}$ are conditionally independent of one another, the probability $P(x_i / y_i)$ is then calculated as follows:

$$P(y_i / x_i) = \prod_{i=1}^{m} P(x_{i\ldots m} / y_i) \quad (8)$$

Though the independence assumption is far reaching and often inappropriate in real world data, Naïve Bayes algorithm performs surprisingly well reel data for most recognition fields, and widely used for classification problems

V. Experimental Results

A. Data and experiments

In the present study, to assess the detection performance of the proposed methods experiments are conducted on two fall detection databases publicly available: UR Fall Detection Dataset (URFD) [2] and fall detection dataset (FDD) datasets [16]. URFD comprises 70 sequences of several actions performed in different ways. Fall events and activities of daily living (ADL) events used in this work are recorded with RGB camera. The URFD consists around for 30 images value per sequence, for both classes: falls and typical ADLs like sitting, down, crouching down, picking-up an object from the floor.

The second database is extracted from fall detection dataset (FDD). This dataset contains 191 Video sequences. The frame rate is 25 frames / second with the resolution of 320 × 240 pixels. For both databases, video sequences are recorded from different environments and contain variable illumination as well as shadows and reflections that can be detected as moving objects.

B. Parameters tuning

It is worth noting that classification performance depends on the classifier parameters, therefore, for each classification method, the optimal parameters have been carefully selected during the training phase (parameter tuning phase) which correspond to the maximum classification accuracy. In the case of neural network, we varied classifier parameters to select the optimal Multi-Layer Perceptron MLP architecture (one hidden

layer with fifteen (15) neurons) which corresponds to the best classification rate. For the k-NN classification, the choice of k value depends strongly on the type of the data. In this work, we varied the parameter k from 1 to 20 to select the optimal value. Based on the highest accuracy, the corresponding value of k is 3. Concerning Naïve Bayes classifier, a Gaussian model is adapted as predictor distribution model. The values of the corresponding mean and standard deviation are 3.43 and 0.38, respectively.

Finally, we have iteratively tested different SVM-kernel parameters namely sigma σ and cost C; where σ is the width of Gaussian kernel and C the parameter for the soft margin cost function, (which controls the influence of each support vector; this process involves trading error penalty for stability). The pair presenting to the highest accuracy has been selected (σ 2^{-3}=0.125 and C=2^{7}=128).

C. Result and interpretation

Table.1 depicts a result of comparison between different machine learning techniques. For each classification algorithm, the overall accuracy is computed, along with sensitivity (Se), specificity (Sp), precession, recall, F-measure, and area under ROC curve coefficients (AUC).

Table.1 Evaluation of the fall detection systems

Classifier	Accuracy	Se	Sp	Precession	Recall	F-measure	AUC
KNN	90.48	0.955	0.808	0.714	0.73	0.833	0.887
NN (MPL)	92.24	0.968	0.848	0.75	0.778	0.857	0.91
Naïve Bays	91.29	0.961	0.819	0.7317	0.749	0.845	0.896
SVM	**93.96**	**0.98**	**0.894**	**0.83**	**0.808**	**0.90**	**0.937**

The results shown in Table 1 and Figure.4 demonstrate the outperformance of the SVM classification over detecting falls by the rest of classifiers. One can also note that an acceptable accuracy is achieved. In addition, SVM provides the highest sensitivity which reveals its ability to detect real falls. This detail is very important, which means that a number of false negatives (FNs) or missed falls is reduced using SVM classification. The highest specificity value is also obtained by SVM algorithm which reflects the capacity to avoid false positives (FPs), only a reduced number of like-fall activities was confused as real falls.

The reason that SVM algorithm has outperformed Neural Network is that SVMs have a simple geometric interpretation and give a sparse solution using structural risk minimization, unlike Neural Networks, which use empirical risk minimization.

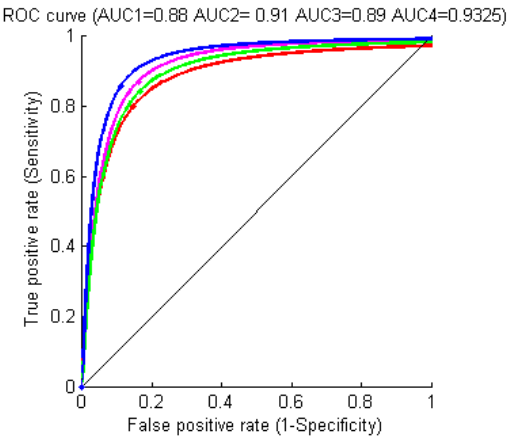

Fig. 5. ROC curves for different fall classifications: KNN (AUC1=0.88), NN (AUC2=0.91), NB (AUC3=0.89) and SVM (AUC4=0.93).

In addition SVMs can deal with the major problem than ANNs, as SVMs are less prone to over fitting. One can also note, that Naïve Bayes algorithm presented several misclassification cases compared to SVM classification; this misclassification could be related to the conditional independence assumption, which can affect the estimation of the posterior probability. SVM classification has also outperformed K-nearest neighbor, duo to the complexity of k-NN algorithm in searching the nearest neighbors for each sample.

VI. CONCLUSION

We have presented a comparative study between most popular machine learning approaches for human fall classification using computer vision data. This comparative examination is founded on various and complementary statistical performance measures. The experiments conducted via these metrics have revealed the superiority of the SVM approach, where it provided the greatest sensitivity, specificity, and accuracy in distinguishing falls from non-falls. On the other hand, it has shown that acceptable accuracies are obtained even using other classifiers. This is duo to the extracted features that offered a key description of human posture in most activities. We believe that this evaluation study may hold clues about a possible combination of non-computer and computer vision based approaches. As perspective, we will seek to collaborate with clinics or nursing institutes, where the performance of the proposed approach will be tested and validated on real data coming from elderly people.

Finally, we consider that this work can represent a helpful tool for practitioners that perform the assistance of patients and elderly people.

REFERENCES

[1] W. H. O. Ageing and L. C. Unit, WHO global report on falls prevention in older age. World Health Organization, 2008.

[2] B. KWOLEK and M. KEPSKI, "Improving fall detection by the use of depth sensor and accelerometer", Neurocomputing, vol. 168, pp. 637-645, 2015

[3] Juang, Chia-Feng, and Chia-Ming Chang. "Human body posture classification by a neural fuzzy network and home care system

application." IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 37.6 (2007): 984-994.

[4] C. Liu, C. Lee, and P. Lin, “A fall detection system using k-nearest neighbor classifier,” Expert Syst. Appl., vol. 37, no. 10, pp. 7174–7181, 2010.

[5] N. Thome, S. Miguet, and S. Ambellouis, “A real-time, multiview fall detection system: A LHMM-based approach,” IEEE Trans. Circuits Syst.Video Technol., vol. 18, no. 11, pp. 1522–1532, Nov. 2008.

[6] Yu, M., Rhuma, A., Naqvi, S. M., Wang, L., & Chambers, J. Yu, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment." IEEE Transactions Information Technology in Biomedicine, on 16.6, 1274-1286, 2012.

[7] B. Shoushtarian and H. E. Bez, “A practical adaptive approach for dynamic background subtraction using an invariant colour model and objecttracking,” Pattern Recognit. Lett., vol. 26, no. 1, pp. 5–26, Jan. 2005.

[8] Chua, J. L., Chang, Y. C., & Lim, W. K. (2013, November). Visual based fall detection through human shape variation and head detection. In Multimedia, Signal Processing and Communication Technologies (IMPACT), 2013 International Conference on (pp. 61-65). IEEE.

[9] H. Foroughi, B.S. Aski, H. Pourreza, Intelligent Video Surveillance for Mon-itoring Fall Detection of Elderly in Home Environments, 11th IEEE Interna- tional Conference on Computer and Information Technology ICCIT), pp. 219–224, 2008

[10] Zerrouki, N., and A. Houacine. "Automatic Classification of Human Body Postures Based on Curvelet Transform." Image Analysis and Recognition. Springer International Publishing, 2014. 329-337.

[11] Zerrouki, N., Harrou, F., Sun, Y., & Houacine, A. (2016). A Data-Driven Monitoring Technique for Enhanced Fall Events Detection. *IFAC-PapersOnLine*, *49*(5), 333-338.

[12] Fouzi Harrou, Nabil Zerrouki, Ying Sun, Amrane Houacine A Simple Strategy for Fall Events Detection INDIN 2016 IEEE International Conference on Industrial Informatics 18-21 July 2016, Futuroscope-Poitiers, France

[13] Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

[14] Chang, C.C., Lin, C.J, "LIBSVM: a library for support vector machines." ACM Transactions on Intelligent Systems and Technology 2(3), 1–27, 2011.

[15] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

[16] Charfi, I., Mitéran, M., Dubois, J., Atri, M., Tourki, R. "Definition and Performance Evaluation of a Robust SVM Based Fall Detection Solution", 8th International Conference on,Signal Image Technology and Internet Based Systems (SITIS), 2012.