Applying Neural Network Architecture in a Multi-Sensor Monitoring System for the Elderly

Shadi Khawandi, Pierre Chauvet
University of Angers
Angers, France
chadi.khawandi@etud.univ-angers.fr,
Pierre.Chauvet@uco.fr

Bassam Daya
Lebanese University
Saida, Lebanon
b_daya@hotmail.com

Abstract— **One of three adults 65 years or older falls every year. As medical science advances, people can live with better health and alone up to a very advanced age. Therefore, to let elderly people live in their own homes leading their normal life and at the same time taking care of them requires new kinds of systems. In this paper, we propose a multi-sensor monitoring system for the fall detection in home environments. The system, which consists of a webcam and heart rate sensor, processes the data extracted from the two different sub-systems by applying neural network n order to classify the fall event in two classes: fall and not fall. Reliable recognition rate of experimental results underlines satisfactory performance of our system.**

Keywords-Neural Network; fall detection; heart rate; webcam

I. INTRODUCTION

Falling and its resulting injuries are an important public-health problem for older adults. The National Safety Council estimates that persons over the age of 65 have the highest mortality rate (death rate) from injuries. Among older adults, injuries cause more deaths than either pneumonia or diabetes. The risk of falling increases with age. Demographic predictions of population aged 65 and over suggest the need for telemedicine applications in the eldercare domain. Many devices have been developed in the last few years for fall detection [1][2], such as a social alarm, which is a wrist watch with a button that is activated by the person in case he/she suffers a fall, and wearable fall detectors, which are based on combinations of accelerometers and tilt sensors. The main problem with social alarms is that the button is often unreachable after a fall, especially when the person is panicked, confused, or unconscious. For the wearable sensors, these autonomous sensors are usually attached under the armpit, around the wrist, behind the ear’s lobe, or at the waist. However, the problem of such detectors is that older people often forget to wear them [3][4]; indeed, their efficiency relies on the person’s ability and willingness to wear them.

The proposed system is composed of two different devices: webcam and heart rate sensor. The extracted data will be processed by a neural network for classifying the events in two classes: fall and not fall. Reliable recognition rate of experimental results underlines satisfactory performance of our system.

In this paper, we review some existing vision-based fall detection systems (Section II), and then we introduce our proposed system (Section III) with additional technical details. The experimental results are presented in Section IV, and finally, the conclusion is presented in Section V.

II. RELATED APPROACHES

Information Technology combined with recent advances in networking, mobile communications, and wireless medical sensor technologies offers great potential to support healthcare professionals and to deliver remote healthcare services, hence, providing the opportunities to improve efficiency and quality and better access to care at the point of need. Existing fall detection approaches can be categorized into three different classes to build a hierarchy of fall detection methods. Fall detection methods can be divided roughly into three categories:

- Wearable Sensors (such as accelerometers or help buttons): These autonomous sensors are usually attached under the armpit, around the wrist, behind the ear’s lobe, at the waist or even on the chest. Merryn [5] used an integrated approach of waist-mounted accelerometer. A fall is detected when the negative acceleration is suddenly increased due to the change in orientation from upright to lying position. A barometric pressure sensor was introduced by Bianchi [6], as a surrogate measure for altitude to improve upon existing accelerometer-based fall event detection techniques. The acceleration and air pressure data are recorded using a wearable device attached to the subject's waist and analyzed offline. A heuristically trained decision tree classifier is used to label suspected falls. Estudillo-Valderrama [7] analyzed results related to a fall detection system through data acquisition from multiple biomedical sensors then processed the data with a personal server. A wearable airbag was incorporated by Tamura [8] for fall detection by triggering airbag inflation when acceleration and angular velocity thresholds are exceeded. Chen [9] created a wireless, low-power sensor network by utilizing small, noninvasive, low power motes (sensor nodes). Wang [10] applied reference velocities and developed a system that uses an accelerometer placed on the head. However the problem of such detectors is that older people often forget to wear them, indeed their efficiency relies on the person’s ability and willingness to wear them, moreover in the case of a help button, it can be useless if the person is unconscious or immobilized.

- Environmental Sensors: Environmental sensors based devices attempt to fuse audio data and event sensing through vibration data. Zhuang [11] proposed an approach the audio signal from a single far-field microphone. A Gaussian mixture model (GMM) super vector is created to model each fall as a noise segment. The pair wise difference between audio segments is measured using the Euclidean distance. A completely passive and unobtrusive system was introduced by Alwan [12] that developed the working principle and the design of a floor vibration-based fall detector. Detection of human falls is estimated by monitoring the floor vibration patterns. The principle is based on the vibration signature of the floor. The concept of floor vibrations with sound sensing is unique in its own way [13]. Pattern recognition is applied to differentiate between falls and other events. Toreyet [14] fused the multitude of sound, vibration and passive infrared (PIR) sensors inside an intelligent environment equipped with the above fusion elements. Wavelet based feature extraction is performed on data received from raw sensor outputs.
  Most ambient device based approaches use pressure sensors for subject detection and tracking. The pressure sensor is based on the principle of sensing high pressure of the subject due to the subject's weight for detection and tracking. It is a cost effective and less intrusive for the implementation of surveillance systems. However, it has a big disadvantage of sensing pressure of everything in and around the subject and generating false alarms in the case of fall detection, which leads to a low detection accuracy.
- Computer Vision Systems: Cameras are increasingly included, these days, in in-home assistive/care systems as they convey multiple advantages over other sensor based systems. Cameras can be used to detect multiple events simultaneously with less intrusion. Cucchiara [15] applied a multi-camera system for image stream processing. The processing includes recognition of hazardous events and behaviors, such as falls, through tracking and detection. The cameras are partially overlapped and exchange visual data during the camera handover through a novel idea of warping "people's silhouettes.
  From tracking data, McKenna [16] automatically obtained spatial context models by using the combination of Bayesian Gaussian mixture estimation and minimum description length model for the selection of Gaussian mixture components through semantic regions (zones) of interest. Tao [17] developed a detection system using background subtraction with an addition of foreground extraction, extracting the aspect ratio (height over width) as one of the features for analysis, and an event-inference module which uses data parsing on image sequences. Foroughi [18] applied an approximated ellipse around the human body for shape change. Projection histograms after segmentation are evaluated and any temporal changes of the head position are noted. Miaou [19] captured images using an Omni-camera called MapCam for fall detection. The personal information of each individual, such as weight, height and electronic health history, is also considered in the image processing task. Rougier [20] proposed a classification method for fall detection by analyzing human shape deformation. Segmentation is performed to extract the silhouette and additionally edge points inside the silhouette are extracted using a canny edge detector for matching two consecutive human shapes using shape context. With Visual fall detection, what appears to be a fall might not be a fall. Most of existing systems are unable to distinguish between a real fall incident and an event when the person is lying or sitting down abruptly.

III. PROPOSED SYSTEM

This paper proposes a multi-sensor fall-detector system (Fig. 1) as a combination between two different commercial devices: a webcam and a heart rate sensor. Data extracted from the two sub-systems will be processed by the neural network [Multi-Layer Perceptron (MLP)] in order to detect the fall. Once the fall is detected, an emergency alert will be activated automatically and sent to care holders through an internet-based home gateway

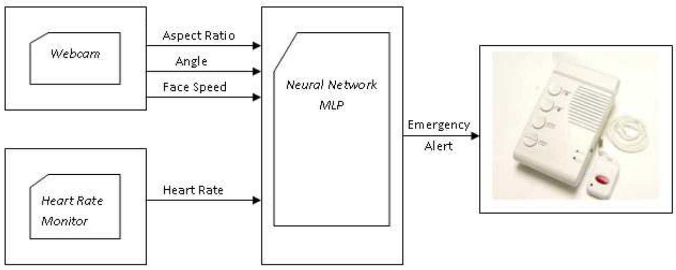

Figure 1. Overview of the proposed system.

A. Webcam System

It is obvious that we need several webcams to cover the entire monitored zone and the switch between the webcams will be based on the face presence. In this paper, we present the webcam system as limited to one webcam, as it will be similar when having multiple webcams.

The webcam system is based on image processing in real time; this system detects the body and face of a person in a given area, collects data such as the aspect ratio, angle, and speed of movement of the person, then sends the extracted data to be processed by the MLP. The system starts by removing the background. After the silhouette is acquired, the next step is the skin color detection, which is an effective way often used to define a set of areas likely to contain a face or hands; then, the system detects the face. Then, features extraction is involved (speed of a person's movement, aspect ratio, and fall angle).

1) Background Subtraction

Background subtraction (Fig. 2) is a particularly popular method to detect moving regions in an image by differentiating between the current image and a reference background image in a pixel-by pixel way

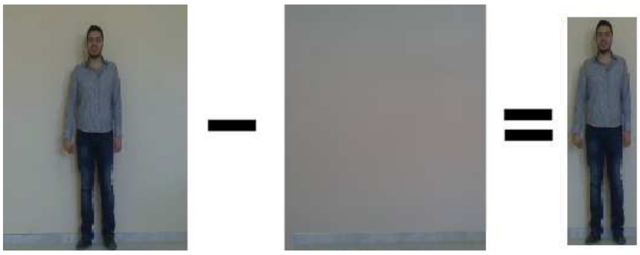

Figure 2. Background Subtraction.

2) Skin-color and HSV detection

The images captured by the webcams are then processed by the system to detect skin color. This is an effective technique for determining whether an image contains a face or hands. In this technique, the appropriate threshold values are defined for all the pixels in a color space (Fig. 3). Different color spaces are used to represent skin color pixels: RGB, RGB standard, HSV (or HSI), YCrCb, and HSV. After the detection of skin-color pixels, image filtering (erosion and dilation) is carried out

Figure 3. Image after skin color detection.

3) Face Detection- Approximated Ellipse

After identifying the skin areas, it is necessary to distinguish the face. For this, the shape of the detected object is compared with an ellipse. This correlation technique is very effective and efficient.

Based on the comparison with an ellipse, we may have more than one image, such as the hand. In order to solve this issue, each image will be converted into a binary image (black and white); then, the white contour will be replaced by black. In this state, the object representing the hand goes black but the object representing the face becomes black except the eyes and mouth. After this transformation, we compute the white surface in each picture, and the object having the greater white surface is the one of the face, and in this case, it is detected.

After calculating the white surface in each image, we found that the white surface in the face is greater than that in the hand; that is why this intelligent system detects the face (Fig. 4).

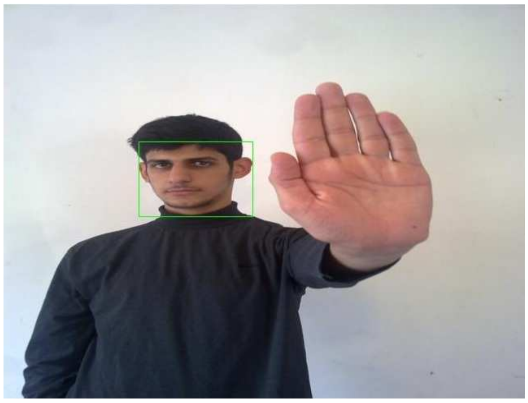

Figure 4. Image after skin color detection.

4) Speed Extraction

One major point in the recognition system is the feature extraction, i.e., the transition from the initial data space to a feature space that will make the recognition problem more tractable. So, we analyze the shape changes of the detected face in the video sequence. The planar speed of movement is calculated using the following formula:

Planar speed = distance/time (pixel/s);

- *Distance*: between the same face in consecutive frames (pixel);
- *Time*: processing time between two consecutive frames.

The range is from 90 to 700 pixels (it can vary depending on the quality of the pictures).

5) Aspect Ratio and Angle Extraction

- Aspect Ratio
  The aspect ratio of a person is a simple yet effective feature for differentiating a normal standing pose from other abnormal poses (Fig. 5). The aspect ratio of the human body changes during a fall. When a person falls, the height and width of his bounding box change drastically (height/width). The range is from 0.15 to 6 (it can vary depending on the dimensions of the subject or on the scaling camera to image coefficients).
- Angle
  Fall angle (θ) is the angle of a vertical line through the centroid of the object with respect to the horizontal axis of the bounding box (Fig. 5). The centroid (Cx, Cy) is the center of mass coordinates of an object. When a person is standing, we assume that he is in an upright position and the angle of a vertical line through the centroid with respect to the horizontal axis of the bounding box should be approximately 90 degrees. When a person is walking, the θ value varies from 45 degrees to 90 degrees. When a person is falling, the angle is always less than 45 degrees. For every frame, we calculate the fall angle (θ), and if θ value is

less than 45 degrees, we confirm that the person is falling. The range is from 0 to 90 degrees.

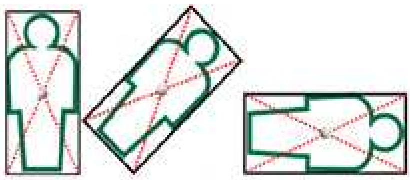

Figure 5. Bounding box and poses of human object.

B. *Heart Rate System*

A heart rate monitor is a personal monitoring device, which allows a subject to measure his or her heart rate in real time or record his or her heart rate for later study. Early models consisted of a monitoring box with a set of electrode leads, which attached to the chest. This paper does not include the design of a heart rate monitor, but we will use an existing heart rate monitor. A Wi-Fi heart rate belt (HRM-2823) (Fig. 6) could be connected to a computer or Wi-Fi operator mobile; this monitor has professional software providing online data exchange model that allows the heart rate belt keeps connecting with the PC and transmits data to PC in real time. The idea is to have a “non-image”-related parameter involved in the fall detection in order to minimize the false alarms.

Figure 6. Heart rate monitor.

C. *Neural Network System*

Throughout the years, the computational changes have brought growth to new technologies. Such is the case of artificial neural networks, that over the years, they have given various solutions to the industry. Designing and implementing intelligent systems has become a crucial factor for the innovation and development of better products for society. In our paper, we decided to design a neural network (Fig. 7) [21] that processes generated input data for classifying the events in two classes: fall and not fall. For the input data, the following sets of parameters are used for falling recognition:

- Speed: The planar speed is calculated using the formula: Planar speed = distance/time (pixel/s);
- Aspect ratio: The aspect ratio of a person is a simple yet effective feature for differentiating normal standing poses from other abnormal poses. The aspect ratio of the human body changes during a fall. When a person falls, the height and width of his bounding box changes drastically (height/width).
- Angle (degree): Fall angle is the angle of a vertical line through the centroid of an object with respect to the horizontal axis of the bounding box. The centroid (Cx, Cy) is the center of mass coordinates of an object. When a person is falling, the angle is always less than 45 degrees. For every frame, we calculate the fall angle (θ), and if θ value is less than 45 degrees, we confirm that the person is falling.
- Heart rate: Measures the number of heart beats per second (bpm).

We generated 5000 such sets of values, each having correspondence with real-life situations that can occur. We chose to have 2500 situations corresponding to non-fall situations and 2500 corresponding to fall situations. We decided that a fall situation occurs when we have measures of high speed, low aspect ratio, the angle under 45 degrees, and a heart rate close to normal. Having four types of data transmitted from the cameras and the sensor made our network have 4 inputs. With each frame, we receive a new set of data, which represent a new pattern that needs to be trained or tested by the network. For our network, we decided to implement a (MLP). The network is a feed-forward network with Back Propagation. The output consists of 1 element that can be 1 or 0. The value one has been assigned to the fall situation class and the value 0 correspond to the situation of non-fall. The training process allowed the neural network to automatically identify the regions in the input pattern space that contained the fall data points. For all of the simulations, we chose a sigmoid transfer function for the hidden layer and a linear transfer function for the output layer.

The network settings are presented below:

- Learning rule: BackPropagation
- Layers = 2
- Inputs = 4
- Hidden Neurons = 8
- Output Neurons = 1
- Transfer function hl = “logsig”
- Transfer function ol = “linear”
- Error function: MSE
- Goal = 0.01
- Max epochs = 10.000
- Momentum Coefficient = 0.01
- Learning method: gradient descent or Levenberg-Marquardt
- Learning rate = 0.01

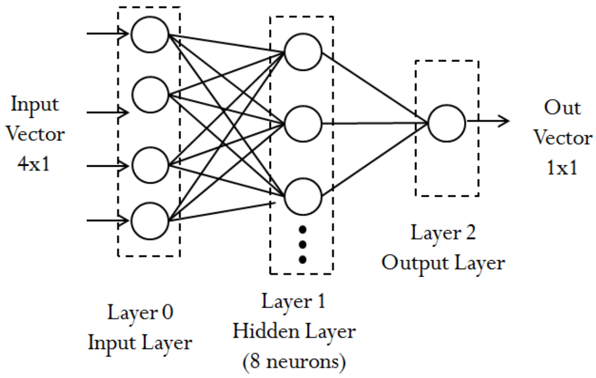

Figure 7. Neural network architecture.

IV. RESUTLS AND DISCUSSION

A. *Preprocessing the training data*

In principle, we can just use any raw input-output data to train our networks. However, in practice, it often helps the network to learn appropriately if we carry out some preprocessing of the training data before feeding it to the network. The ranges of our input are:

- Speed (pixel/s): from 90 to 700;
- Aspect ratio (height/width): from 0.15 to 6 ;
- Angle (degree): from 0 to 90;
- Heartbeat (bpm): from 70 to 200.

All inputs were normalized to [-3, 3]. The range of the output is [0, 1] (sigmoid function), so there is no need for scaling. The number of patterns that we used was 2500 for each class. Because, in our approach, we used batch training, there was no need for shuffling the order of the input patterns. We can observe that the Levenberg-Marquardt method [22] shows better results.

TABLE I. RESULTS FOR DIFFERENT LEARNING METHODS

Learning method	Num of Epochs	Goal	Performance			
			Training (lastepoch)	Validation		
				Sensitivity	Specificity	Accuracy
traingdx	56	0.1	0.0979	87.71	75.43	81.57
trainlm	3	0.1	0.0358	99.53	93.17	96.59
traingdx	10000	0.01	0.0156	92.37	96.12	98.07
trainlm	**7**	**0.01**	**0.00307**	**100**	**98.29**	**99.15**

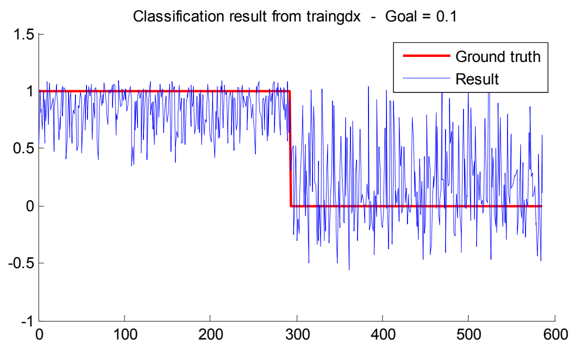

Figure 8. Classification result from traingdx (Goal = 0.1).

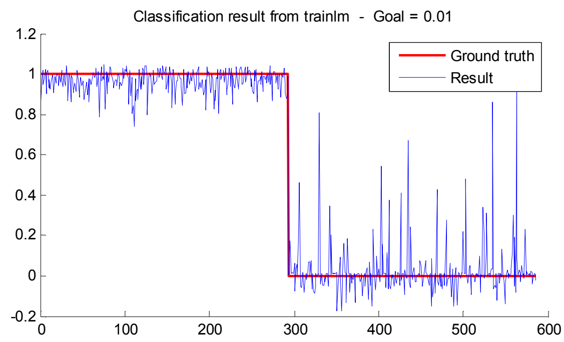

Figure 9. Classification result from trainlm (Goal = 0.01).

B. *Choosing the Initial Weights*

The gradient descent learning algorithm treats all the weights in the same way, so if we start them all off with the same values, all the hidden units will end up doing the same thing, and the network will never learn properly. For that reason, we generally start off all the weights with small random values. Usually, we take them from a flat distribution around zero [-smwt, +smwt], or from a Gaussian distribution around zero with standard deviation smwt. Choosing a good value of smwt can be difficult. Generally, it is a good idea to make it as large as you can without saturating any of the sigmoid. We usually hope that the final network performance will be independent of the choice of initial weights, but we need to check this by training the network from a number of different random initial weight sets. The initial weights were generated using the functions:

- rands: return values between -1 and 1;
- midpoint: is a weight initialization function that sets weight (row) vectors to the center of the input ranges.
- initzero: initializing all the weight to zero.
- Nguyen-Widrow: the standard function in Matlab for initializing the weights

The best performance is obtained with the Nguyen-Widrow function [23].

TABLE II. DIFFERENT WAYS OF INITIALIZING THE WEIGHTS

Initial Weights	Num of Epochs	Performance			
		Training(lastepoch)	Validation		
			Sensitivity	Specificity	Accuracy
rands	4	0.00593	100	98.39	99.19
Midpoint	11	0.00849	100	98.93	99.46
initzero	11	0.00849	100	98.93	99.46
Nguyen-Widrow	7	0.00307	100	96.59	98.29

C. *Choosing the learning rate*

Choosing a good value for the learning rate η is constrained by two opposing facts:

- If η is too small, it will take too long to get anywhere near the minimum of the error function
- If η is too large, the weight updates will over-shoot the error minimum and the weights will oscillate, or even diverge.

The best performance was obtained for the learning rate of 0.001 (Table III). For the types of learning rates—fixed and adaptive learning rates—best performance was achieved with adaptive learning rate (Table IV). We tried to vary the number of hidden units from our network. We tried with 4, 8, and 12 hidden neurons and observed that the best performance was obtained for 4 neurons in the hidden neurons (Table V). The best performance was achieved with 4 neurons. The number of layers was also modified searching for the optimal network. It turned out to be that one hidden layer was enough for achieving good performances (Table VI). The best performance was obtained for 1 hidden layer.

TABLE III. RESULTS OBTAINED FOR DIFFERENT LEARNING RATES

Learning rate	Num of Epochs	Goal	Training (lastepoch)	Validation		
				Sensitivity	Specificity	Accuracy
0.001	**14**	**0.01**	**0.00979**	**100**	**98.36**	**99.10**
0.01	10	0.01	0.00985	100	97.75	98.55
0.1	11	0.01	0.0098	100	98.29	99.15
10	25	0.01	0.0099	98.61	98.95	98.78

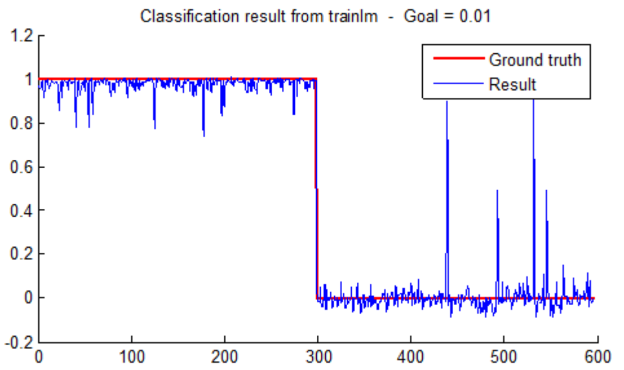

Figure 10. Classification result from trainlm (Lr = 0.001).

TABLE IV. RESULTS FOR DIFFERENT TYPES OF LEARNING RATES

Learning rate 0.1	Num of Epochs	Goal	Training (lastepoch)	Validation		
				Sensitivity	Specificity	Accuracy
Fixed	20	0.01	0.00911	100	97.56	98.12
Adaptive	**14**	**0.01**	**0.00921**	**100**	**98.29**	**99.15**

TABLE V. RESULTS FOR DIFFERENT TYPES OF HIDDEN UNITS

#of Neurons	Num of Epochs	Performance			
		Training (lastepoch)	Validation		
			Sensitivity	Specificity	Accuracy
4	14	0.00923	100	98.29	99.15
8	11	0.00959	100	98.29	99.15
12	12	009610.	100	97.27	98.63

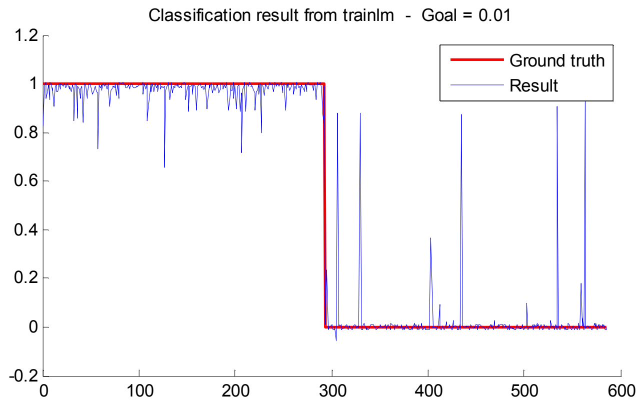

Figure 11. Classification result from trainlm (4 neurons 2 layers).

TABLE VI. RESULTS FOR DIFFERENT NUMBER OF HIDDEN LAYERS

# of Hidden Layers	Num of Epochs	Goal	Performance			
			Training (lastepoch)	Validation		
				Sensitivity	Specificity	Accuracy
1	**11**	**0.01**	**0.00959**	**100**	**98.29**	**99.15**
3	9	0.01	0.0097	100	96.55	98.01
5	12	0.01	0.09612	100	96.27	97.54

 ISBN: 978-1-61208-237-0

D. *Fall detection*

For fall incidents, the inputs (speed, ratio, angle, and heart rate) have to satisfy certain thresholds:

- 650 < Speed < 700
- 0.15 < Ratio < 1.5
- 0 < Angle < 45
- 70 < Heart rate < 110

TABLE VII. TESTING THE NETWORK

Sensitivity	**Specificity**	**Accuracy**
100	97.58	**99.15**

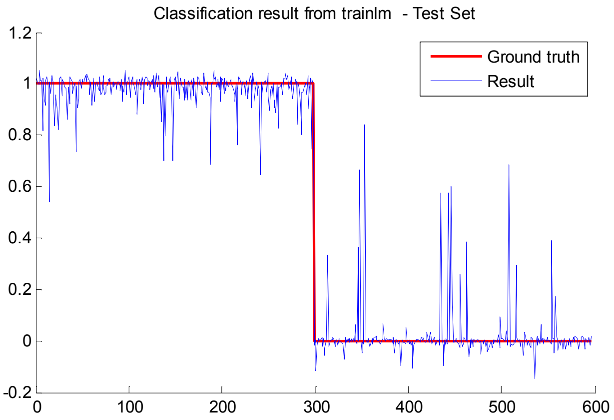

Figure 12. Classification result from trainlm (Test Set).

TABLE VIII. FALL DETECTION

Speed	**Ratio**	**Angle**	**Heart rate**	**Class**
666	0.46	15	104	Fall
689	0.99	5	70	Fall
698	0.19	26	91	Fall
650	0.56	16	86	Fall
674	0.42	17	99	Fall
583	4.4	60	110	No Fall
328	1.12	20	90	No Fall
147	0.32	45	86	No Fall
423	2.03	50	77	No Fall

V. CONCLUSION AND FUTURE WORK

Fall-related injuries have been among the five most common causes of death amongst the elderly population. Falls represent 38% of all home accidents and cause 70% of deaths in the 75+ age group. Early detection of a fall is an important step in avoiding any serious injuries. An automatic fall detection system can help to address this problem by reducing the time between the fall and arrival of required assistance. In an eldercare context, false alarms can be expensive. Too many false alarms could result in a loss of trust, or worse, loss of use of the system. However, missing a single fall is the worst-case scenario. Identifying an acceptable false-alarm rate and understanding the conditions in which many false alarms occur is of vital use for the long-term success of an automated system. Healthcare video surveillance systems are a new and promising solution to improve the quality of life and care for the elderly, by preserving their autonomy and generating the safety and comfort needed in their daily lives. This corresponds to the hopes of the elderly themselves, their families, the caregivers, and the governments. The positive receptivity for video surveillance systems suggests that this technology has a bright future for healthcare and will advantageously complement other approaches (e.g., fixed or wearable sensors, safer home modifications, etc.) by overcoming many of their limitations. Better performances and results can be obtained by implementing neural network architecture when different methods of acquiring data are combined (wearable devices + webcam images). The presented work may be extended and enhanced, in a later phase, to include multiple webcams and other parameters that could help to address this problem by reducing the risk of false alarms and improving the time between the fall and the alarm.

VI. REFERENCES

[1] N. Noury, T. Hervé, V. Rialle, G. Virone, and E. Mercier, “Monitoring behavior in home using a smart fall sensor and position sensors,” in IEEE-EMBS. Microtechnologies in Medicine & Biology, October, Lyon-France, 2000, pp. 607–610.

[2] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, “Fall Detection - Principles and Methods,” 29th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, Lion (France), August 2007, pp. 1663–1666.

[3] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, “Fall Detection - Principles and Methods,” 29th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society, Lion (France), August 2007, pp. 1663–1666.

[4] A. Yamaguchi, “Monitoring behavior in home using positioning sensors” Int. Conf. IEEE-EMBS, Hong-Kong, 1998; 1977-79.

[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, Journal for Physiological Measurement (IOPScience), Vol. 25, 2004.

[6] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti and N. H. Lovell, Barometric Pressure and Triaxial Accelerometry Based Falls Event Detection, IEEE Transactions on Neural Systems and Rehabilitation Engineering, Vol. 18, pp. 619-627, 2010.

[7] M.A. Estudillo-Valderrama, L.M. Roa, J. Reina-Tosina and D. Naranjo-Hernandez, Design and Implementation of a Distributed Fall Detection System - Personal Server, IEEE Transactions on Information Technology in Biomedicine, Vol. 13, pp. 874-881, 2009.

[8] T. Tamura, T. Yoshimura, M. Sekine, M. Uchida and O. Tanaka, A Wearable Airbag to Prevent Fall Injuries, IEEE Transactions on Information Technology in Biomedicine, Vol.13, pp. 910-914, 2009.

[9] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, Wearable Sensors for Reliable Fall Detection, 27th IEEE Annual Conference of Engineering in Medicine and Biology (EMBS), pp. 3551-3554, 2005.

[10] C. C. Wang, C. Y. Chiang, P. Y. Lin, Y. C. Chou, I. T. Kuo, C. N. Huang and C. T. Chan, Development of a Fall Detecting System for the Elderly Residents, 2nd IEEE International Conference on Bioinformatics and Biomedical Engineering, ICBBE, pp. 1359-1362, 2008.

[11] X. Zhuang, J. Huang, G. Potamianos and M. Hasegawa-Johnson, Acoustic Fall Detection Using Gaussian Mixture Models and GMM Super-Vectors, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.69-72, 2009.

[12] M. Alwan , P. J. Rajendran, S. Kell, D. Mack , S. Dalal, M. Wolfe, and R. Felder, A Smart and Passive Floor-Vibration Based Fall Detector for Elderly, IEEE International Conference on Information & Communication Technologies (ICITA), pp. 1003-1007, 2006.

[13] Y. Zigel, D. Litvak and I. Gannot; A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound Proof of Concept on Human Mimicking Doll Falls, IEEE Transactions on Biomedical Engineering, Vol. 56, pp. 2858-2867, 2009.

[14] B. U. Toreyin, E. B. Soyer, I. Onaran, and A. E. Cetin, Falling Person Detection Using Multi-sensor Signal Processing, 15th IEEE Signal Processing and Communications Applications Conference, SIU & EURASIP Journal on Advances in Signal Processing, vol. 8, 2007 & 2008.

[15] R. Cucchiara, A. Prati, and R. Vezzani, “A multi-camera vision system for fall detection and alarm generation,” Expert Syst. J., vol. 24(5), 2007, pp. 334-345.

[16] C. H. Nait and S. J. McKenna, “Activity summarisation and fall detection in a supportive home environment,” Int. Conf. on Pattern Recognition (ICPR), 2004.

[17] J. Tao, M. Turjo, M. F. Wong, M. Wang and Y. P. Tan: Fall Incidents Detection for Intelligent Video Surveillance, Fifth IEEE International Conference on Information, Communications and Signal Processing, pp. 1590-1594, 2005.

[18] H. Foroughi, B. S. Aski, and H. Pourreza, Intelligent Video Surveillance for Monitoring Fall Detection of Elderly in Home Environments, 11th IEEE International Conference on Computer and Information Technology (ICCIT), pp. 219-224, 2008.

[19] S. G. Miaou, P. H. Sung and C. Y. Huang, A Customized Human Fall Detection System Using Omni-Camera Images and Personal Information, 1st Trans-Disciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2), pp. 39-42, 2006.

[20] C Rougier, J Meunier, A St-Arnaud, and J Rousseau, Robust Video Surveillance for Fall Detection Based on Human Shape Deformation, IEEE Transactions on Circuits and Systems for Video Technology (CSVT), Vol. 21, pp. 611-622, 2011.

[21] G. Miller, P. Todd, and S. Hedge, " Designing neur al networks using genetic algorithms," Proceedings of the International Confe rence on Genetic Algorithms 1989.

[22] H. Demuth, M. Beale, and M. Hagan. “Neural Network Toolbox 5, User’s Guide,” Version 5, The MathWorks, Inc., Natick, MA, revised for version 5.1.

[23] H. Demuth and M. Beale. “Neural Network Toolbox, User’s Guide,” Version 4, The MathWorks, Inc., Natick, MA, revised for version 4.0.4.