Human fall detection from acceleration measurements using a Recurrent Neural Network

T. Theodoridis[1], V. Solachidis[1], N. Vretos[1] and P. Daras[1]

[1] Information Technologies Institute, Centre for Research and Technology Hellas, Thessaloniki, Greece

Abstract— **In this work, a method for human fall detection is presented based on Recurrent Neural Networks. The ability of these networks to process and encode sequential data, such as acceleration measurements from body-worn sensors, makes them ideal candidates for this task. Furthermore, since such networks can benefit greatly from additional data during training, the use of a data augmentation procedure involving random 3D rotations has been investigated. When evaluated on the publicly available URFD dataset, the proposed method achieved better results compared to other methods.**

Keywords— **Human fall detection, Recurrent Neural Network, data augmentation, acceleration**

I. Introduction and related work

A fall is defined as an event that results in a person coming to rest inadvertently on the ground or floor or other lower level [1]. The people more susceptible to falls are usually the elders. The frequency of fall events is even higher in elders who suffer from chronic illnesses [2] (e.g. Parkinson, Arthritis, Osteoporosis). Moreover, in many cases, a fall may immobilize a person and make him/her unable to call for assistance. Thus, the presence of carers at home becomes necessary, resulting in increased expenses for the patient and the health-care system.

In this paper, a method that automatically detects a fall is presented. Although a fall detection system does not prevent the fall, the information that provides is valuable and can be used by both carers and medical professionals. The necessity for continuous presence by caregivers of chronic illness patients due to the risk of fall, can be relaxed if a fall detection system is installed, since, in the case of a fall, it can alert them in order to assist the patient. This technology can ease the life of caregivers (professional or patient relatives) and, at the same time, contribute to the decrease of the health-care system expenses. Regarding medical professionals, the detailed reports that the fall system can provide, give valuable information (e.g., frequency of falls per time of the day, increase/decrease of incidents, etc.), since they can be correlated with medication changes and, hence, contribute to the medication scheme definition.

Several methods that detect falls have been presented in the literature using a variety of sensors. The most common sensors that are used are accelerometers [3]-[8], RGB cameras [9, 10], depth or infrared cameras [11]. Other technologies such as floor-vibration sensors [12, 13] and Wireless Sensor Networks [14] have been employed as well.

The fall detection method presented in [15] uses acceleration measurements from two devices placed at the trunk and thigh of the users. A threshold value on the acceleration magnitude is used in order to determine if a fall has occurred or not. Two variations of this approach are presented: one that signals a fall event if the acceleration magnitude exceeds a certain threshold and another that signals a fall if the acceleration magnitude goes below a different threshold.

In [16] a fall detection method is proposed that uses data acquired from an accelerometer, placed near the pelvis region of the users, and depth cameras. The system assumes that there is no fall if the acceleration magnitude is below a certain threshold, regardless of the depth camera input. If the acceleration magnitude exceeds the threshold value, then the input from the depth camera is analyzed. The method detects the person from the depth image along with the floor plane equation, and then extracts features related to the person's body position (e.g., distance of body's centroid from the floor, ratio of the person's bounding box dimensions, etc). A Support Vector Machine (SVM) classifier, based on these features, produces the final decision.

The authors in [17] propose three different methods that rely on a Kinect device and two wearables placed to the person's wrist and waist. The methods use the body skeleton captured by Kinect and the wearable devices' acceleration and orientation. The best method of the three relies on the rapid downward movement of the spine base joint of the human body, on the distance of said joint from the floor and on the acceleration magnitude.

In [18] a variety of machine learning models has been tested on features extracted from the acceleration measurements of a wearable device and a mobile phone. The features extracted in the time domain include the mean, variance, kurtosis, etc., while the frequency domain features were the autocorrelation coefficients and the total spectral power in different frequency bands. The best results were obtained using a Decision Tree ensemble for both the wearable sensor and the mobile phone.

Despite the fact that, generally, the use of RGB or depth cameras increases the accuracy of fall detection methods compared to ones that use only accelerometers, cameras have two significant disadvantages: cost and limited coverage area. Thus, if we wish to apply a method that includes them in a house set-up, we have to install cameras in every room, increasing the total system cost. The proposed method detects falls using only data acquired from a body-worn accelerometer, keeping the total system cost low, and at the same time, having a large spatial range where the method can be applied. Additionally, the proposed method is capable of identifying falls without false positive detections, as indicated by the evaluation results on the UR Fall Detection (URFD) dataset [16]. In Section II the proposed method is described in detail. The experimental evaluation is illustrated in Section III, where the proposed method is compared not only to methods that use acceleration as the only modality, but also to ones that employ accelerometer and depth cameras. Finally, in Section IV conclusions are drawn.

II. Proposed Method

The proposed fall detection method takes advantage of the Recurrent Neural Networks' (RNNs) ability to process and encode the inherent information contained in sequential data. Traditional machine learning models, such as the Multi-layer Perceptron (MLP) or the Support Vector Machine (SVM), process their input without any notion of sequential order, and thus cannot take advantage of this information. Recurrent Neural Networks process their input in a sequential manner, accumulating more information after each time step about the sequence being presented to them. The Long Short-Term Memory variant of RNNs (LSTM [19, 20]), which is adopted in this work, further improves the basic RNN architecture, by enabling the network to retain information from many time steps back into the past, thus giving the network the ability to encode and learn longer sequences.

In order for a Recurrent Neural Network to process the input signal as a sequence, the sequence length n must be determined from the beginning. Then, the signal is divided into time windows of length n and the network processes each window independently. Fig. 1 shows the network architecture used in this work. Variables $\mathbf{X_1}, \ldots, \mathbf{X_n}$ denote the multi-dimensional input signal that spans n time steps. The first LSTM layer (first blue row) processes the input signal and produces an output at each time step $t = 1, \ldots, n$. The second LSTM layer, in the same way, processes the output of the first LSTM layer at each time step, but produces an output only at the last time step n. Then, a traditional feed-forward neural network (first cyan rectangle) processes the output of the second LSTM layer and, finally, a second feed-forward neural network produces the final decision Y, which in our case is the probability that a fall incidence has occurred. As with all supervised machine learning techniques, in order for the network to learn the output probabilities, besides the input signal at each time step, a label must also be provided, which indicates whether an incidence has occurred or not.

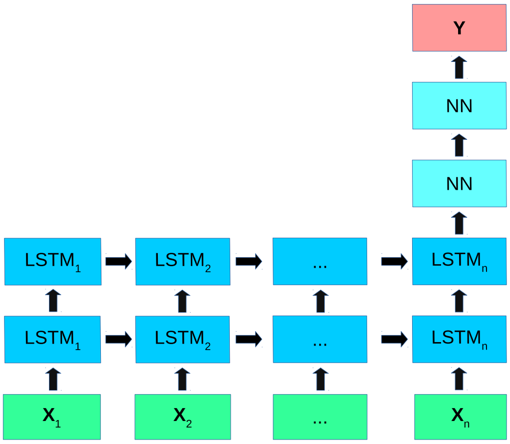

Fig. 1: Proposed model architecture.

Due to the beneficial impact of additional training data on model generalization and performance, the effectiveness of a data augmentation procedure involving random rotations has been evaluated as well. Given the acceleration vector $\mathbf{a}(t) = [a_x(t), a_y(t), a_z(t)]$ at time t, that contains the acceleration along the x, y and z axes of the device respectively, a new vector $\mathbf{a}^r(t)$ can be obtained by rotating $\mathbf{a}(t)$ by θ radians about the x axis, ϕ radians about the y axis and ψ radians about the z axis:

$$\mathbf{a}^r(t) = \mathbf{R}_z(\psi) \cdot \mathbf{R}_y(\phi) \cdot \mathbf{R}_x(\theta) \cdot \mathbf{a}(t) \tag{1}$$

where $(\cdot)$ denotes matrix multiplication, $\mathbf{R}_x(\theta)$, $\mathbf{R}_y(\phi)$ and $\mathbf{R}_z(\psi)$ are the rotation matrices about x, y and z axes respectively:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix}, \theta \in [0, 2\pi) \tag{2}$$

$$\mathbf{R}_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix}, \phi \in [0, 2\pi) \tag{3}$$

$$\mathbf{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \psi \in [0, 2\pi) \quad (4)$$

and $\mathbf{a}(t)$ is considered a 3×1 matrix in (1), in order for the multiplication to be well-defined.

III. Experimental Evaluation

A. The Dataset

The proposed method has been evaluated on the publicly available UR Fall Detection (URFD) dataset [16, 21], which contains measurements from an accelerometer, placed near the pelvis area of the human body, as well as depth images, and features extracted from those images, acquired from two Kinect cameras. In total, the dataset contains 40 sequences with activities of daily living and 30 sequences with falls.

The accelerometer data from the body-worn device at time t consist of the 3D acceleration $\mathbf{a}(t) = [a_x(t), a_y(t), a_z(t)]$ as well as the norm of the acceleration vector (also called the magnitude or in some cases the total sum vector):

$$\|\mathbf{a}(t)\| = \sqrt{a_x^2(t) + a_y^2(t) + a_z^2(t)} \quad (5)$$

Regarding the Kinect camera measurements, the authors of the dataset have provided features extracted from the Kinect depth images, such as the width to height ratio of the person's bounding box in the depth image, the height of the person's centroid, etc. The measurements from the accelerometer and the Kinect camera have been temporally synchronized, so that at each time step information from all sensors is available.

B. Parameter Selection and Evaluation Protocol

The proposed method, denoted as *LSTM-Acc*, consists of the network architecture presented in Section II that processes sequences of length $n = 30$, which corresponds to a time span of one second. The two LSTM layers and the first feed-forward layer of the network have 200 units each, while the last feed-forward layer has 2. The proposed method was trained and evaluated using only acceleration data. Furthermore, we augmented the training data with one rotated version of the original measurements by a random angle of $\theta, \phi, \psi \in [-10, 10]$ degrees about x, y and z axes, using the procedure discussed at the end of Section II. In doing so, we increase the number of available samples for training and also force the model to learn representations that are more robust to rotations. This second approach is denoted as *LSTM-Acc Rot*.

Two methods were chosen and implemented for comparison purposes. The first method was proposed in [15] and is denoted as *UFT*. It uses a threshold on the acceleration magnitude $\|\mathbf{a}(t)\|$ in order to determine if a fall has occurred or not. In training, the threshold is determined as the minimum of the magnitude peaks during the fall instances. In testing, the same threshold is used for separating fall instances from non-fall ones.

The second method was proposed in [16] and is denoted as *Acc + SVM-Depth*. It uses a threshold value of $3g$ on the acceleration magnitude $\|\mathbf{a}(t)\|$ in order to initiate a fall detection procedure, which consists of an SVM model that has been trained on the extracted depth features giving the final decision. The depth features were scaled so that they have zero mean and unit variance.

Regarding the *UFT* and *Acc + SVM-Depth* methods, there was no point in using the augmented dataset discussed previously, since they rely on the acceleration magnitude $\|\mathbf{a}(t)\|$, which is invariant to rotations.

The evaluation was performed using a 10-fold cross-validation procedure, in which the dataset is split into ten parts, nine of which are used for training and one for testing. The procedure is repeated ten times, so that all possible test parts have appeared once. Since this dataset consists of 40 sequences with activities of daily living and 30 with falls, each fold contained 4 sequences from the first group and 3 from the second. Finally, the results were evaluated on a sequence level using four metrics: accuracy, precision, sensitivity and specificity. The evaluation on sequence level means that for a non-fall sequence to be correctly classified, the models had to produce zero alerts during the whole sequence. On the other hand, in order for a fall sequence to be classified correctly, the models had to produce at least one fall alert, starting from one second before the beginning of fall and onward, not before.

C. Experimental Results

The experimental results on the URFD dataset are shown in Table 1. Starting from the simplest method of the four, *UFT*, it is evident that it produces the highest amount of false positives (detecting a fall when no fall has occurred), as it has the lowest specificity score (90%). Overall, it has the same accuracy as the *Acc + SVM-Depth* method, which however has higher specificity, but lower sensitivity. This means that the *Acc + SVM-Depth* method produces fewer false positives, but also finds fewer actual falls. Next, the proposed *LSTM-Acc* method is equal to or better than the previous methods in all evaluation metrics. It has the same sensitivity as the *UFT* method (96.67%), the same specificity as the *Acc + SVM-Depth* method (95%) and higher precision and accuracy than both. Lastly, the *LSTM-Acc Rot* approach has produced the best results. Even though it relies only on acceleration information, it has not produced a single false positive result (specificity = 100%), while the sensitivity of 96.67% corresponds to not detecting one fall event.

Table 1: The fall detection results (%) on the URFD dataset.

	LSTM-Acc	*LSTM-Acc Rot*	*Acc + SVM-Depth*	*UFT*
Accuracy	95.71	**98.57**	92.86	92.86
Precision	95.00	**100**	94.17	90.00
Sensitivity	**96.67**	**96.67**	90.00	**96.67**
Specificity	95.00	**100**	95.00	90.00

IV. CONCLUSIONS

In this work, a Recurrent Neural Network-based approach to fall detection has been presented. By leveraging the ability of such networks to process sequential data, as well as data augmentation in the form of random rotations of the input acceleration signal, the proposed method was able to find all but one fall event, while at the same time producing no false alarms when tested on the URFD dataset.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

ACKNOWLEDGEMENTS

This work was supported by the European Project: ICT4LIFE http://ict4life.eu/ Grant no. 690090 within the H2020 Research and Innovation Programme.

REFERENCES

1. World Health Organization WHO, Falls http://www.who.int/mediacentre/factsheets/fs344/en/. Accessed: 2017-09-19.
2. Kalache A, Fu D, Yoshida S, et al. *World health organisation global report on falls prevention in older age*. World Health Organisation 2007.
3. Yuwono M, Moulton B D, Su S W, et al. Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems *Biomedical Engineering Online.* 2012;11:9.
4. Bourke A K, Ven P, Gamble M, et al. Assessment of waist-worn tri-axial accelerometer based fall-detection algorithms using continuous unsupervised activities in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*:2782-2785 2010.
5. Liu S-H, Cheng W-C. Fall detection with the Support Vector Machine during scripted and continuous unscripted activities *Sensors.* 2012;12:12301–12316.
6. Kangas M, Vikman I, Wiklander J, et al. Sensitivity and specificity of fall detection in people aged 40 years and over *Gait & Posture.* 2009;29:571 - 574.
7. Koshmak G A, Linden M, Loutfi A. Evaluation of the android-based fall detection system with physiological data monitoring in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*:1164–1168 2013.
8. Abbate S, Marco A, Bonatesta F, et al. A smartphone-based fall detection system *Pervasive and Mobile Computing.* 2012;8:883 - 899.
9. Vishwakarma V, Mandal C, Sural S. Automatic detection of human fall in video *Pattern Recognition and Machine Intelligence.* 2007:616–623.
10. Rougier C, Meunier J, St-Arnaud A, et al. Robust video surveillance for fall detection based on human shape deformation *IEEE Transactions on Circuits and Systems for Video Technology.* 2011;21:611–622.
11. Mastorakis G, Makris D. Fall detection system using Kinects infrared sensor *Journal of Real-Time Image Processing.* 2014;9:635–646.
12. Rimminen H, Lindström J, Linnavuo M, et al. Detection of falls among the elderly by a floor sensor using the electric near field *IEEE Transactions on Information Technology in Biomedicine.* 2010;14:1475–1476.
13. Alwan M, Rajendran P J, Kell S, et al. A smart and passive floor-vibration based fall detector for elderly in *Proceedings of Information and Communication Technologies*;1:1003–1007 2006.
14. Wang Y, Wu K, Ni L M. Wifall: Device-free fall detection by wireless networks *IEEE Transactions on Mobile Computing.* 2017;16:581–594.
15. Bourke A K, Obrien J V, Lyons G M. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm *Gait & Posture.* 2007;26:194–199.
16. Kwolek B, Kepski M. Human fall detection on embedded platform using depth maps and wireless accelerometer *Computer Methods and Programs in Biomedicine.* 2014;117:489–501.
17. Cippitelli E, Gasparrini S, Gambi E, et al. An integrated approach to fall detection and fall risk estimation based on RGB-depth and inertial sensors in *Proceedings of the International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*:246–253 2016.
18. Alzubi H, Ramzan N, Shahriar H, et al. Optimization and evaluation of the human fall detection system in *Proceedings of SPIE 10008, Remote Sensing Technologies and Applications in Urban Environments*:1000816 2016.
19. Hochreiter S, Schmidhuber J. Long short-term memory *Neural computation.* 1997;9:1735–1780.
20. Gers F A, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM in *Proceedings of International Conference on Artificial Neural Networks*:850-855 1999.
21. UR Fall Detection Dataset http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html. Accessed: 2017-09-19.

Author: Thomas Theodoridis
Institute: Information Technologies Institute, Centre for Research and Technology Hellas
Street: 6th km Charilaou - Thermi Road
City: Thessaloniki
Country: Greece
Email: tomastheod@iti.gr