A Neural Network Approach to Human Posture Classification and Fall Detection using RGB-D Camera

Alessandro Manzi[1], Filippo Cavallo[1] and Paolo Dario[1]

Abstract. In this paper, we describe a human posture classification and a falling detector module suitable for smart homes and assisted living solutions. The system uses a neural network that processes the human joints produced by a skeleton tracker using the depth streams of an RGB-D sensor. The neural network is able to recognize standing, sitting and lying postures. Using only the depth maps from the sensor, the system can work in poor light conditions and guarantees the privacy of the person. The neural network is trained with a dataset produced with the Kinect tracker, but it is also tested with a different human tracker (NiTE). In particular, the aim of this work is to analyse the behaviour of the neural network even when the position of the extracted joints is not reliable and the provided skeleton is confused. Real-time tests have been carried out covering the whole operative range of the sensor (up to 3.5 meters). Experimental results have shown an overall accuracy of 98.3% using the NiTE tracker for the falling tests, with the worst accuracy of 97.5%.

Keywords: human posture, neural networks, depth camera.

1 Introduction

In the recent years, the development of technologies strictly connected to humans increased exponentially. Nowadays, the advent of powerful mobile devices such as

1 Alessandro Manzi, Filippo Cavallo, Paolo Dario
The BioRobotics Institute, Scuola Superiore Sant'Anna, Piazza Martiri della Libertà, 33
56127 Pisa (Italy), e-mail: a.manzi@sssup.it

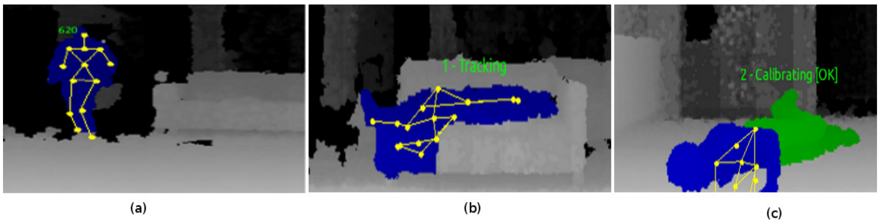


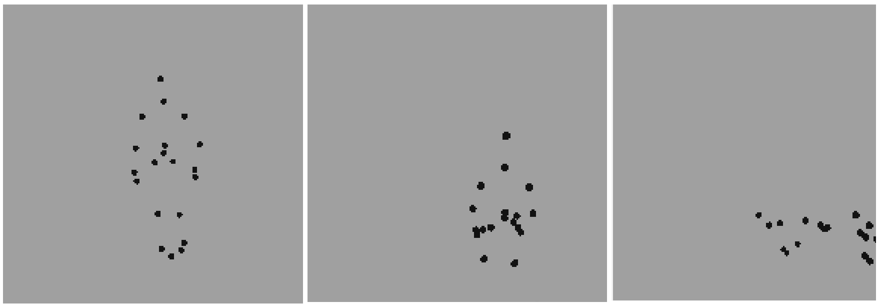
Fig. 1: Examples of worst skeleton detection. (a) The person is far from the sensor and at least two joints are missed. (b) The user is lying on a sofa and the skeleton is fused with the sofa. (c) The person falls down in front of the sensor and the output seems unusable.

smartphones and tablets are a reality, but in the near future smart home technologies will represent a huge market [1-3]. Distributed environmental sensors [4], robots [5], computers and wearable devices [6] will share the home environment with us. These kinds of smart systems need to be aware of humans in order to effectively interacts with them to address several tasks such as energy management or behavioral and health monitoring. In the context of home care application, especially if we consider elderly people, one of the most desirable feature is the ability to detect a falling event. Each year, one in every three older adults falls in their home [7], but less than half talk to their health-care providers about it [8]. Older adult falls lead to reduced functions and premature loss of independence, and oftentimes a fall may indicate a more serious underlying health problem. For these reasons, the importance of the fall detection to have a fast and quick reaction is crucial. In the past, video surveillance systems have been proposed to address this issue, but some of their limitations include the light conditions and the lack of privacy. The recent emergence of depth sensors, so-called RGB-D sensors (e.g. Microsoft Kinect, Asus Xtion, PrimeSense Carmine), has made it feasible and economically sound to capture in real-time not only color images, but also depth maps with appropriate resolution and accuracy. A depth sensor can provide three-dimensional data structure as well as the 3D motion information of the subjects/objects in the scene, which has shown to be advantageous for human detection [9]. Several works about human postures detection with the RGB-D sensors exploit the use of skeleton tracking algorithms for rapidly transforming persons depth information to spatial joints that represent the human figure [10, 11]. Unfortunately, when these methods are used for real world applications the output is not always stable and reliable (see Fig. 1). The reasons that reduce their performance depend on several factors. Among these, we have the distance between the person and the sensor, the occlusions that occur when people interacts with environmental objects and also sideways poses that hides some parts of the user that are not visible to the sensor.

The aim of this work is to develop a system, based on depth cameras, which is able to classify three human postures, including standing, sitting, and lying positions that reliable works in real conditions. In order to do that, we need to deal with the aforementioned problems that afflict the skeleton tracker methods. Therefore, an artificial Neural Network (NN) model is adopted to rely on its generalization ability and its robustness against noisy and missed data. As opposed to other similar works [12, 13], real-time tests, conceived to reproduce realistic and challenging situations for the tracker, covering the whole operative range of the sensor, have been carried out. During these experiments, the NN has been continuously fed with all the available joints generated by the skeleton tracker in order to analyse its robustness to unreliable and uncertain joints. At the end of the paper, an application for smart homes and a scenario that includes a domestic robot that integrates the trained NN for falling detection is presented. The paper is structured as follows. Sect. 2 presents the related work on human postures, while Sect. 3 gives an overview of the proposed system, describing the NN architecture, the dataset, the training and test phases. The real-time experiments are presented in Sect. 4, while the results are summarized in the Sect. 5. A falling event application example for smart environment is presented is Sect. 6 and Sect. 7 concludes the paper.

2 Related Work

The importance of detecting human postures, especially for recognize or prevent human falls, is addressed in various previous work. According to Yu [14], a system for the falling recognition must have three main properties, it has to be reliable, unobtrusive and has to preserve the privacy. Several proposed systems make use of wearable devices [15], such as accelerometers [16], gyroscopes [17] and RFID sensors [18]. However, these approaches are often cost prohibitive and they rely on the willingness of the subjects to wear devices, reducing the overall acceptability of the system. Non-invasive methods such as computer vision techniques have been extensively investigated. In [19], a 2-D human posture classification by means of a neural fuzzy network is presented. However, 2-D video based methods generally give not robust and inaccurate results, and they are influenced by the light conditions without providing an adequate privacy.



(a) (b) (c)

Fig. 2: Three examples of the samples extracted from the dataset: (a) standing, (b)sitting and (c) lying posture.

Recently, the advent of low-cost depth camera received a great deal of attention from researchers. This technology offers several advantages compared to standard video cameras. In addition to color and texture information, depth images provide three-dimensional data useful for segmentation and detection. Moreover, a system that uses only depth information is able to work in poor light conditions (high risk of falling accidents), providing privacy at the same time. In literature, several works address the problem of the posture detection using depth data. Some of these take into account the relation between the human and the ground [20], but, in order to perform floor segmentation, they often assume the floor as a large part of the scene and this assumption seems unrealistic in real home application. Silhouette extraction methods use the centroid as detector feature [19, 21], but the centroid is strong dependent on the posture and on the size of the user. Other works exploit proper skeleton tracking algorithms to use human joints as feature descriptor [22]. However, the depth data are usually affected by noise and the joints are not always available. As it has been pointed out by [11], depending on the quality of the segmented target and the level of occlusion, the skeleton trackers might not detect all the joints and their location cannot be totally reliable. A poor estimation of the skeleton joints occurs when the person is partially occluded or somewhat out of the image, or not facing the sensor (sideways poses provide some challenges regarding the part of the user that is not directly visible), or it is at the far end of the sensor range. For these reasons, it is common to find works based on skeleton tracker, limiting the test phases to samples that are free of excessive noise [12] or performing tests that allow to retrieve easily distinguishable human body features [13]. In order to deal with the aforementioned problems, a feed-forward NN, trained with all the available skeleton joints, is adopted to detect the target postures.

During the real-time tests, the NN is fed with the output of a tracking algorithm also when its
output is confused and not reliable. Tests have been carried out in order to analyse the behaviour of the NN at different distances, covering the whole range of the sensor.

3 System Overview

The proposed human posture detection relies on a skeleton tracker algorithm that is able to extract the joints of a person from the depth map. Among the most used skeleton tracker we can find the Microsoft Kinect SDK, which works with its namesake device, and the NiTE SDK [23], used in conjunction with the OpenNI framework [24] that is generic and runs both for Kinect and Asus Xtion or PrimeSense device. These software tools are similar and provide the 3D position of the skeleton joints combined with an additional confidence value for each of them. This datum can assume three values: “tracked” when the algorithm is confident, “inferred” when it applies some heuristics to adjust the position, and “untracked” when there is uncertainty. Both SDKs are affected by the same drawbacks when used in real world application. The undesirable conditions happen when the user is too close (< 1 m) or too farther from the sensor (> 3 m), or when the person assumes sideways poses and occlusions are present. In all these cases, the position of the calculated joints are not reliable and stable, so the associated confidence values are set as “untracked”. The Fig. 1 shows three examples of the aforementioned cases. A distant person produces fewer joints than usual, a human lying on a sofa confuses the tracker, while a person that falls down abruptly generates a messy output that seems unusable. Nevertheless, the “untracked” joints are anyway part of the whole skeleton and they have been used to analyse the robustness of the NN against noisy and uncertain values.

3.1 Dataset

The choice of the dataset samples for the training and the validation of the NN is a crucial step. Although there are some online RGB-D datasets about human performing daily activities, unfortunately very few of them contains people in lying position. For this work, the MSRDailyActivity3D dataset [25] has been chosen. It is recorded with the Kinect SDK and contains 10 subjects performing various activities at the distance of about 2 meters. For each frame of the video sequence, the position of 20 skeleton joints is stored in a text file. A total of 120 samples has been taken from these text files to build a set containing subjects in sitting, standing and lying position, equally subdivided (see Fig. 2).

3.2 NN Architecture

The aim of the NN is to detect the three different postures using as input the skeleton joints extracted by the tracker algorithm. The structure of the NN has three layers, with 60 input neurons (3 coordinates for each 20 joints, as provided by the text files of the MSRDailyActivity3D dataset) and 3 output neurons, whose values range from 0 to 1 according to the posture. The neuron number of the hidden layer needs to be minimized in order to keep the amount of free variables, namely the associated weights, as small as possible [26], decreasing also the need of a large training set. The cross-validation technique is adopted to find the lowest validation error as a function of the number of hidden neurons. As a result, an amount of 42 hidden units has been found as sufficient value. The activation function for the hidden and the output layer is the sigmoid, defined as:

$$y = \frac{1}{1 + e^{-2sx}}$$

where x is the input to the activation function, y is the output and s is the steepness (=0,5). The selected learning algorithm is the iRPROP- described in [27], which is an heuristic for supervised learning strategy and that represents a variety of the standard resilient back-propagation (RPROP) training algorithm [28]. It is one of the fastest weight update mechanisms and it is adaptive, therefore does not use the learning rate. The NN is developed in C++ using the Fast Artificial Neural Network Library [29].

3.3 Training, Validation and Testing Sets

In order to estimate the generalization performance of the NN and to avoid the over-fitting of the parameters, the dataset is randomly divided into a training set, to adjust the weights of the NN, a validation set, to minimize the over-fitting, and a testing set to confirm the predictive power of the network. There is no common splitting rule for the dataset. In the present work, we follow the procedure described in [30] in which is stated that the fraction of patterns reserved for the validation set should be inversely proportional to the square root of the number of free adjustable parameters. In our case, these sets are divided in 63, 21 and 36 samples respectively.

All the data are recorded at a distance of about 2 meters from the sensor. If the NN is trained with them, the network will produce better result only around 2 meters. To overcome this issue, a preprocessing step has been introduced. The NN is trained with normalized joints to ensure a depth invariant feature. Each joint vectors is normalized with the Euclidean norm:

$$\hat{j} = \frac{j}{\| j \|}$$

where j is the joint and $\hat{j}$ is the normalized joint.

During the learning phase, the Mean Squared Error (MSE) is separately computed for the training and for the validation set. To guarantee optimal generalization performance, this process is stopped when the validation error starts to increase, since it means that the NN is over-fitting the data [31]. The final errors of the process is $2 * 10$ -4 for the training and 0.028 for the validation. This process took 339 ms on a Intel Core 2 2GHz 32bit producing a 100% recognition rate on all the 36 samples of the testing set.

4 Real-Time tests

As expected, testing the NN with the samples of the dataset gives a True Positive Rate (TPR) of 100%, since the data are well acquired and free of excessive noise. To understand the real performance of the network, real-time experiments have been set up. Since the original dataset is built only with the Kinect SDK tracker, in order to prove the generalization power of the NN tests have been carried out with both Kinect SDK and NiTE tracker of the OpenNI framework. Although these two software behave in a similar way, they have a significant difference. The first one represents the human skeleton with 20 joints, while the latter uses only 15 joints. Therefore, to work with our trained NN, the input is preprocessed to fill the missed joints with the closest available, as depicted in the Fig. 3.

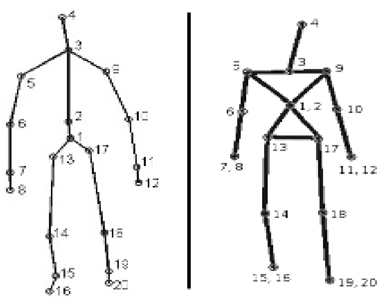


Fig. 3: Difference between skeleton representation. The Kinect SDK (left) uses 20 joints, while the NiTE SDK (right) only 15. The missing joints are replaced with the closest.

4.1 Experimental Setup

Two different kinds of experiments have been conducted. The first one is about the detection of the three human postures in daily life environment with a sofa, while the second tests how the network behaves when a person falls down. The output of the NN is "standing", "sitting", and "lying" according to the value of the output neuron that is closest to 1. To analyze the results, the outputs are compared with the actual posture of the person, but the intermediate poses between a posture and another are discarded, i.e. when the user is sitting down or standing up. All the tests run at 25 fps. Since the input of the NN is the skeleton data, which are extracted purely from depth maps, the light conditions do not influence the performed experiments.

4.1.1 Sit and Lie on a Sofa

This experiment has been conducted in a real living room with a sofa. The sensors (Kinect and Xtion) have been placed at 1 meters from the ground facing the sofa. A person, starting from the left, goes to the sofa, sits for a while, lies down on it, and then gets up again and goes away. The experiments have been carried out with 6 people (3 male and 3 female) at 3 different distances (3.5, 2.5 and 1.5 meters). This setup is intended to address the human trackers problems about the distance (Fig. 1 (a)) and the melting issue between human and objects (Fig. 1 (b)) as already mentioned in Sec. 2.

4.1.2 Falling Tests

Given the lack of available dataset containing falling people, we want to understand the ability of the NN to recognize a falling as a lying posture. Therefore, we set up a series of tests in which a man falls down to the side and to the front of the sensor (Fig. 1 (c)). The device is placed at 1 meter from the ground and the NN is fed with all the available joints, even if their confidence value is labelled as "inferred" or "untracked". In this way, the robustness of the NN against data uncertainty has been evaluated. Falling tests are divided in frontal and lateral to take into account also the self-occlusion of some parts of the body. They have been conducted in a kitchen environment with a person that falls down abruptly while its moving toward and sideways and repeated 5 times each. The frontal fall distance from the sensor is about 2 meters, while the distance of the lateral fall is about 3 meters.

Table 1. Confusion matrix of the Sofa experiments (NiTE)

	standing	**sitting**	**lying**
standing	100%	0%	0%
sitting	0%	100%	0%
lying	0%	2.8%	97.2%

Table 2. Confusion matrix for the falling tests (NiTE)
(a) Frontal Fall

	standing	**sitting**	**lying**
standing	100%	0%	0%
sitting	0%	100%	0%
lying	0%	6.7%	93.3%

Table 3. Confusion matrix for the falling tests (NiTE)
(b) Lateral Fall

	standing	**sitting**	**lying**
standing	98.9%	1.1%	0%
sitting	0%	100%	0%
lying	0%	4.9%	95.1%

Table 4. Accuracy (NiTE)

	Sofa	**Frontal Fall**	**Lateral Fall**
standing	100%	100%	99.4%
sitting	99.5%	97.6%	97.5%
lying	99.5%	97.6%	98.1%
overall	99.6%	98.4%	98.3%

5 Results

The experiment with the sofa has been conducted with 6 persons at different distances and two types of sensors, Kinect and Xtion, and trackers, Kinect SDK and NiTE respectively. The total number of analysed frame are 5214. The NN output with the Kinect SDK proves to be extremely robust and reliable, achieving a 100% for all the three postures. The output with the NiTE skeleton tracker is less reliable and it is summarized with the confusion matrix of the Tab. 1. As expected, lying is the most challenging posture to classify, since the skeleton tracker provides clearer output with the other two postures. It is worth to know that, in all the cases, the actual lying posture can be misclassified only as sitting, and that neither standing nor sitting is classified as lying. Considering the falling tests, the Kinect tracker yields a TPR of 100% for all the postures, while the NiTE is less reliable, but still satisfactory. The Tab. 2 contains the confusion matrices for these experiments and the results are consistent with the previous tests. To be thorough, since the person falls down quickly, there are not actual sitting posture. The Tab. 3 contains the accuracy calculated for the sofa and the falling experiments. In general, the real standing posture is always recognized even when the user is sideways, given that the output of the tracker is cleaner and reliable in this case. We have to point that these human trackers have been developed for natural interaction and gaming and the players must stand in front of the sensor. As expected, the lying posture is the most challenging to detect, but the rate of the false positive is always null and the actual lying posture is misclassified only with the sitting class and never with the standing. Since the NN is trained with a dataset built on Kinect tracker, using it produces excellent results. However, the use of the NiTE tracker does not involve bad effects. The most interesting result is about the falling test. The NN produces a TPR of 95.1% for the lateral test and 93.3% for the frontal. In particular, if we consider only the lying posture, the frontal fall test has a False Discovery Rate (FDR) of 0%, and the probability of the False Negative Rate (FNR) is 6.7%, while for the lateral fall test the FDR is 0% and the FNR is 4.9%.The overall accuracy is 98.4% and 98.3% respectively. Another important aspect to underline is that, for most of the cases, misclassification happens during postures transitions. These results make it feasible the use of the adopted NN for a falling event application that is described in the next section.

1

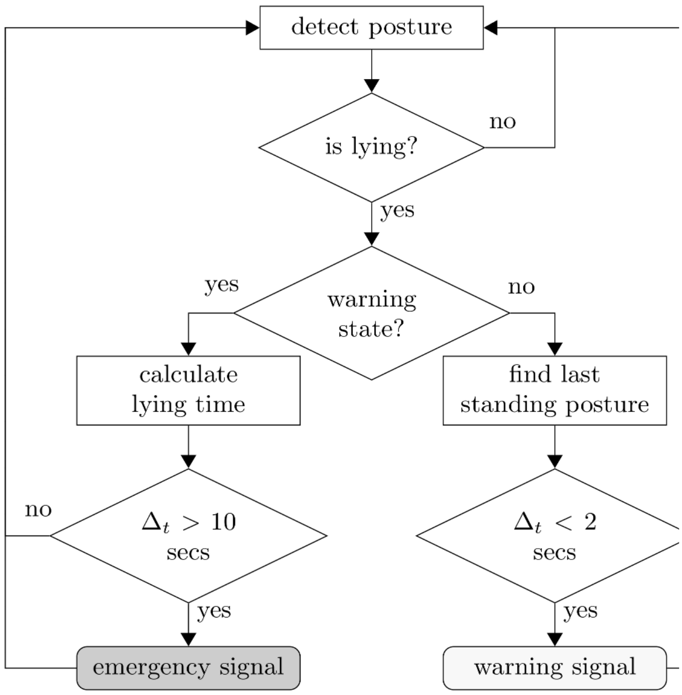


Fig. 4: The Fall Detector module reads the output of the neural network continuously. According to the posture and to appropriate threshold, it is able to send warning or emergency signals.

6 Fall Detector Application

Considering the results about the above experiments, a fall detector application has been developed. It is able to generate warning or emergency signals according to the NN output. The Fig. 4 outlines its flowchart. The event generator reads the outputs of the NN storing them with an associated timestamp. When a lying posture is detected and its internal state is not equal to warning, it finds the last standing detected posture and computes the delta time. As already stated by Fu et. al [32], if this value is less than 2 seconds, the system considers it as a falling event and generates a warning signal. The detector still continues to check the input and if the posture stands in lying position for more than 10 seconds, it sends also an emergency signal. Currently, we are simulating this system with a domestic robot, which is able to retrieve these kinds of events and react consequently. When the robot receives a warning signal from the falls detector, it moves to the area of interest and starts an interaction procedure with the person to ask him/her if an help is needed. If no answer is received it warns a specific person (i.e. a caregiver or relatives) through a video-call mechanism. If the robot receives an emergency signal, it starts soon an automatic video-call and at the same time it moves to the area of interest to provide as much information as possible to the caregiver.

7 Conclusion and Future Work

In this paper, a feed-forward artificial Neural Network to detect three target postures (i.e. standing, sitting and lying) by means of an RGB-D sensor is presented. The NN is trained with samples extracted from a public dataset recorded with the Kinect SDK, while the real-time tests are carried out both with the Kinect and the Asus Xtion Pro Live device using the Kinect proprietary skeleton tracker and the NiTE tracker respectively. The input data are preprocessed and normalized in order to be depth invariant, improving the results of the NN all along the field of view of the sensors. The output of these skeleton tracker algorithms in real world application is not always stable and accurate, especially when the user is not standing and parts of the human body are occluded by the person itself or by external objects. A series of real-time experiments, conceived to analyze the behavior of the trained NN in challenging situations, have been conducted. During these tests, the NN processes continuously the output of the skeleton tracker also when the joints are labelled as unreliable. Our results demonstrate its high robustness against the uncertainty of the data, achieving an accuracy of more than 98% for the falling tests. The NN, trained with a Kinect dataset, demonstrates its power of generalization also when it is fed with data produced by a different tracker (NiTE software). Following the results of the experiments, a fall detector application which integrates the NN is also presented. The proposed system runs in real-time and, since it is based only on depth maps that do not use color information it guarantees the privacy of the person and it is able to work also in poor light conditions. Further improvements can be obtained creating an ad-hoc database containing fallen people in a real environment. For our best knowledge, an RGB-D dataset of this type is not yet available, and it will concern one of our next works. In this way, it will be possible to train a model using more realistic data. Future developments will also focus on the development of a multiple depth cameras system, which covers areas of the home with high risk of fall, such for example bathroom and bedroom. Moreo-[3] ver, additional depth cameras give the possibility to estimate the user position in the home.

Acknowledgement
This work was supported in part by the European Community's 7th Framework Program (FP7 / 2007–2013) under grant agreement No. 288899 (Robot-Era Project) and grant agreement No. 601116 (Echord++ project).

References

[1]. Aquilano, M., Cavallo, F., Bonaccorsi, M., Esposito, R., Rovini, E., Filippi, M., Dario, P., Car-rozza, M. C. (2012, August). Ambient assisted living and ageing: Preliminary results of RITA project. In 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 5823-5826).

[2]. L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[3]. S. Suresh and P. Sruthi, "A review on smart home technology," in 2015 Online International Conference on Green Engineering and Technologies (IC-GET), pp. 1–3, IEEE, 2015.

[4]. Bonaccorsi, M., Fiorini, L., Cavallo, F., Saffiotti, A., & Dario, P. (2016). A Cloud Robotics Solution to Improve Social Assistive Robots for Active and Healthy Aging. International Journal of Social Robotics, 1-16.

[5]. Di Nuovo, A., Broz, F., Belpaeme, T., Cangelosi, A., Cavallo, F., Esposito, R., & Dario, P. (2014, October). A web based multi-modal interface for elderly users of the Robot-Era multi-robot services. In 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 2186-2191). IEEE.

[6]. Moschetti, A., Fiorini, L., Esposito, D., Dario, P., & Cavallo, F. (2016). Recognition of daily gestures with wearable inertial rings and bracelets. Sensors, 16(8), 1341.

[7]. A. Tromp, S. Pluijm, J. Smit, D. Deeg, L. Bouter, and P. Lips, “Fall-risk screening test: a prospective study on predictors for falls in community-dwelling elderly,” Journal of clinical epidemiology, vol. 54, no. 8, pp. 837–844, 2001.

[8]. J. A. Stevens, M. F. Ballesteros, K. A. Mack, R. A. Rudd, E. DeCaro, and G. Adler, “Gender differences in seeking care for falls in the aged medicare population,” American journal of preventive medicine, vol. 43, no. 1, pp. 59–62, 2012.

[9]. J. Sung, C. Ponce, B. Selman, and A. Saxena, “Human activity detection from rgbd images.,” plan, activity, and intent recognition, vol. 64, 2011.

[10]. G. Chen, M. Giuliani, D. Clarke, A. Gaschler, and A. Knoll, “Action recognition using ensemble weighted multi-instance learning,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 4520–4525, IEEE, 2014.

[11]. M. Munaro, S. Ghidoni, D. T. Dizmen, and E. Menegatti, “A feature-based approach to people re-identification using skeleton keypoints,” in Robotics and Automation (ICRA), 2014 IEEE International Conference on, pp. 5644–5651, IEEE, 2014.

[12]. O. Patsadu, C. Nukoolkit, and B. Watanapa, “Human gesture recognition using kinect camera,” in Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on, pp. 28–32, IEEE, 2012.

[13]. T.-L. Le, M.-Q. Nguyen, and T.-T.-M. Nguyen, “Human posture recognition using human skeleton provided by kinect,” in Computing, Management and Telecommunications (Com-ManTel), 2013 International Conference on, pp. 340–345, IEEE, 2013.

[14]. X. Yu, “Approaches and principles of fall detection for elderly and patient,” in e-health Networking, Applications and Services, 2008. HealthCom 2008. 10th International Conference on, pp. 42–47, IEEE, 2008.

[15]. “FATE, Fall Detector for the Elderly.” http://fate.upc.edu. Accessed: 2016-05-25.

[16]. A. Bourkeφ, C. N. Scanaillφ, K. Culhaneφ, J. OBrien, and G. Lyonsφ,5 "An optimum accelerometer configuration and simple algorithm for accurately detecting falls," 2006.

[17]. A. K. Bourke and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," Medical engineering & physics, vol. 30, no. 1, pp. 84–90, 2008.

[18]. M. Chan, D. Est`eve, C. Escriba, and E. Campo, "A review of smart homespresent state and future challenges," Computer methods and programs in biomedicine, vol. 91, no. 1, pp. 55–81, 2008.

[19]. C.-F. Juang and C.-M. Chang, "Human body posture classification by a neural fuzzy network and home care system application," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, vol. 37, no. 6, pp. 984–994, 2007.

[20]. Z.-P. Bian, L.-P. Chau, and N. Magnenat-Thalmann, "A depth video approach for fall detection based on human joints height and falling velocity," International Coference on Computer Animation and Social Agents, 2012.

[21]. C. Rougier, E. Auvinet, J. Rousseau, M. Mignotte, and J. Meunier, "Fall detection from depth map video sequences," in Toward useful services for elderly and people with disabilities, pp. 121–128, Springer, 2011.

[22]. L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," Image and Vision Computing, vol. 30, no. 3, pp. 217–226, 2012.

[23]. "NiTE GitHub." https://github.com/PrimeSense. Accessed: 2016-05-25.

[24]. "OpenNI GitHub." https://github.com/OpenNI. Accessed: 2016-05-25.

[25]. J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1290–1297, IEEE, 2012.

[26]. K. L. Priddy and P. E. Keller, Artificial neural networks: an introduction, vol. 68. SPIE Press, 2005.

[27]. C. Igel and M. H¨usken, "Improving the rprop learning algorithm," in Proceedings of the second international ICSC symposium on neural computation (NC 2000), vol. 2000, pp. 115–121, Citeseer, 2000.

[28]. M. Riedmiller and H. Braun, "Rprop-a fast adaptive learning algorithm," in Proc. of ISCIS VII), Universitat, Citeseer, 1992.

[29]. S. Nissen and E. Nemerson, "Fast artificial neural network library," Available at leenissen. dk/fann/html/files/fann-h. Html, 2000.

[30]. I. Guyon, "A scaling law for the validation-set training-set size ratio," AT&T Bell Laboratories, pp. 1–11, 1997.

[31]. D. Floreano and C. Mattiussi, Bio-inspired artificial intelligence: theories, methods, and technologies. MIT press, 2008.

[32]. Z. Fu, E. Culurciello, P. Lichtsteiner, and T. Delbruck, "Fall detection using an address-event temporal contrast vision sensor," in Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on, pp. 424–427, IEEE, 2008.