Fall Detection for Elderly Person Care Using Convolutional Neural Networks

Xiaogang Li*†‡§,Tiantian Pang*†‡§, Weixiang Liu*†‡§, Tianfu Wang*†‡§
*Health Science Center, Shenzhen University
†School of Biomedical Engineering, Shenzhen University
‡Guangdong Key Laboratory for Biomedical Measurements and Ultrasound Imaging
§National-Regional Key Technology Engineering Laboratory for Medical Ultrasound
Shenzhen 518060, China

Abstract—**Falls are one of the major leading causes of mortality for elderly people living alone at home, which can lead to severe injuries. Fall detection is the most important health care issue for the elderly. In computer vision domain, significant breakthrough technologies such as deep learning have been obtained for over five years. Deep learning belongs to computational methods that allow an algorithm to program itself by learning from training data. Convolutional neural networks (CNNs), a specific type of deep learning, have set the state-of-the-art image classification performance in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in recent years. In this paper, we present the use of convolutional neural networks for fall detection in video surveillance environment. CNN is directly applied to each frame image in the video to learn human shape deformation features that describe different postures of the human and determine if a fall occurs. Experimental results show that our proposed approach runs in real-time and achieves average accuracy of 99.98% for 10-fold cross-validation for fall detection. It is shown that the implemented CNN-based fall detection approach can be a promising solution for detecting falls.**

I. Introduction

Fall is a key health issue in the public health care domain. The risk of fall incidents for the elderly increases remarkably with age. Falls are the main cause of accidental death in older adults [5]. Due to the population explosion after the World War II, western countries face a serious problem, where majority of the population will be those over 65. As a consequence, it has become very important to develop intelligent, low-cost surveillance systems for fall detection especially for elderly persons living alone. Increased demand for security and health care has accelerated research in the fall detection. Many research work have been conducted in developing algorithms for automatically detecting falls. For an overview of such algorithms consult [19].

Many algorithms have been investigated for fall detection among older adults [15], [2]. These algorithms can be divided into three main aspects: ambience device-based, wearable sensor-based and computer vision-based algorithms. To be specific, Ambience device-based algorithms make use of ambience sensors that are installed on elders' active regions to detect if a fall occurs [32], [30]. But some of these ambience sensors are sensitive to noise signals, and then generate false alarms. Wearable sensor-based algorithms automatically detect a fall using sensors such as accelerometers that are continuously attached to the human body [20], [29], [25]. However, the elderly may forget to wear them. In addition, wearable devices may be burdensome. According to surveys, the elderly prefer non-wearable equipment [6]. Computer vision-based algorithms apply one or more cameras installed in the house of elders to fall detection, and give many information on the behavior of a person [9], [28], [22], [23], [3]. Besides, computer vision-based algorithms do not require the person to wear specialized devices. But there exists a challenge for vision-based fall detection, which is how to maximize the fall detection rate with minimal computational complexity. Many vision-based techniques normally require high processing power for real-time video processing which may not be practical for real-time practical deployment [24]. Another difficult problem for fall detection is to recognize a fall among all the daily life activities such as lying down, sitting down and crouching down activities similar characteristics to falls. These daily life activities often lead to false positives in fall detection systems [7].

In the field of computer vision and machine learning, deep learning techniques such as deep convolutional neural networks have obtained outstanding performances for image classification [11], [26], [31], [16]. Deep convolutional neural networks (CNNs) enable learning data-driven, highly representative, hierarchical image features directly from large amounts of annotated data and automatically learn abundant mid-level and high-level abstractions obtained from raw images [18]. In paper [1], in order to detect a fall, the extracted silhouette images (which are binary map images) are used to train the neural network that is a multilayer perceptron. Authors in [8] present a computer vision based fall detection approach, which utilizes restricted Boltzmann machine and deep belief network to analyse the postures in a smart home environment for detecting fall activities. Paper [14] proposes various time-frequency distributions and investigates the impact of different time-frequency representations on the performance of a fall detector that is based on two stacked auto-encoders and a softmax regression classifier. Similar work can also be found in [13]. In this paper, we apply convolutional neural networks to each frame image in the video from a camera to learn human

Corresponding author : Weixiang Liu, Email address : wxliu@szu.edu.cn.

978-1-5386-1937-7/17/$31.00 ©2017 IEEE

body deformation features (representations) for fall detection.

The remainder of this paper is organized as follows. Section II presents a description of the proposed method for fall detection using a CNN. Section III shows the results from our experiments. Section IV concludes this paper with discussion on the advantages and challenges of the proposed method.

II. Methods

A. Dataset Description

Our dataset used for this paper is selected from UR Fall Detection Dataset [1], which is publicly available [17]. The UR Fall Detection Dataset contains 40 activities of daily living (ADL) and 30 fall video sequences. Fall incidents are recorded with two Microsoft Kinect cameras namely camera 0 and camera 1. ADL incidents are recorded with only camera 0 parallel to the floor mounted. In this paper, we only use 28 ADL and 30 fall sequences of RGB images from camera 0, where the rest of 12 ADL sequences recorded at dark environment is excluded. According to the literature [21], a fall event can be decomposed into four phases: *a) pre-fall phase* corresponding to activities of daily living including lying down, sitting down and crouching down, *b) critical phase* implying the fall that is extremely short, *c) fall phase* where the person in this extremely short phase without motion lie down on the ground, *d) recovery phase* where the person is able to stand up alone or with the help of another person. In this paper, we define *phase a, phase b* and *phase d* as ADL. Hence we perform a binary classification, ADL and fall in our case. On the basis of the above analysis, for all selected ADL video sequences, we annotate each frame image as ADL. For all selected fall video sequences, we label each frame image from *phase a, b* and *d* as ADL, and tag each frame image from *phase c* as fall, where most of this person's body is exposed to the ground in this frame image, which is used as the labeling criterion. Finally, our dataset contains 8500 images consisting of 7733 images from ADL and 767 images from falls. Fig. 1 shows part of normal activities of daily living and simulated falls from our dataset.

B. Data Pre-processing

The images from our dataset have resolution of 640×480 pixels. We rescale all images to a fixed resolution of 227×227 pixels in our experiments, because the convolutional neural network used for our classification tasks takes as inputs 227×227 RGB images. Due to the selected video sequences recorded from different real-world environments, brightness between images may have difference. Concerning this, we compute an average image over all training images, and then all training and test images are subtracted by this average image. After removing the average image, we perform contrast normalization for each image. Concretely, every image is normalized by subtracting the mean and dividing by the standard deviation of its elements as used in [4].

C. Deep Convolutional Neural Network (CNN)

In our experiments, the convolutional neural network architecture we adopt is similar to AlexNet [16], which was the winning model in the ImageNet Large Scale Visual Recognition Challenge 2012. The AlexNet is one of the best CNNs today and achieves significantly improved performance over the other non-deep learning methods for many classification tasks. Its network architecture consists of five successive convolutional layers Conv-1...Conv-5 and three fully connected layers FC6...FC8 as Fig. 2 shows. Conv-1 and Conv-2 layers are followed by a response normalization layer and a max-pooling layer respectively. Conv-5 layer is followed by a max-pooling layer. The first two fully connected layers FC6 and FC7 have 4096 neurons. The last fully connected layer FC8 has 1000 neurons corresponding to 1000 classes. In order to reduce overfitting, the regularization technique called *dropout* was applied for FC 6 and FC7 layers [12].

In this paper, we modify AlexNet as the convolutional neural network used for our experiments. Specifically, we remove the fully connected layer FC7 and output layer FC8 of the AlexNet and then add a new fully connected layer fc7 followed by a new output layer fc8. The fc7 layer has 1024 neurons and the fc8 layer has 2 neurons corresponding to 2 classes namely ADL and fall. We also remove all the response normalization layers. This indicates our network is not composed of the response normalization layers. In our network, convolutional layers and max-pooling layers have the same geometry as that of AlexNet. But we have no use of the dropout technique [12].

Table I gives configuration information of our network. The network takes as inputs 227×227 RGB images. The details of each of the convolutional layers are given: *size of kernel* specifies the number of convolution filters and their receptive field size as $size \times size \times num$; *stride* and *pad* indicate the convolutional stride and the number of spatial padding zero, respectively; *feature maps size* refers to the number of the feature maps and each feature map size as $num - size \times size$. In each Max-pooling layer, *patch size* and *stride* indicate pooling window size and stride, respectively. The meaning of *feature maps size* in Max-pooling layers is the same as that in convolutional layers. For the network's input, $227 \times 227 \times 3$ input images are filtered by 96 kernels with size $11 \times 11 \times 3$ with a stride of 4 pixels in the first convolutional layer.

D. Training CNN

The network is trained to predict class labels, where the labels only are ADL and fall. We train the network using the MATLAB toolbox called "MatConvNet" [27]. Weight parameters are initialized using Gaussian distribution. Minibatch stochastic gradient descent is utilized to train the network, with batch size of 85. The initial learning rate is specified as 0.05 and multiplied by 0.1 after 20 epochs of training. Training is done for 40 epochs. Momentum is used to speed up learning. The value of this momentum is set to 0.9. In order to reduce overfitting, weight decay with a value of 0.0001 is adopted. Besides, *early stopping* algorithm, a regularization technique,

[1] http://fenix.univ.rzeszow.pl/mkepski/ds/uf.html

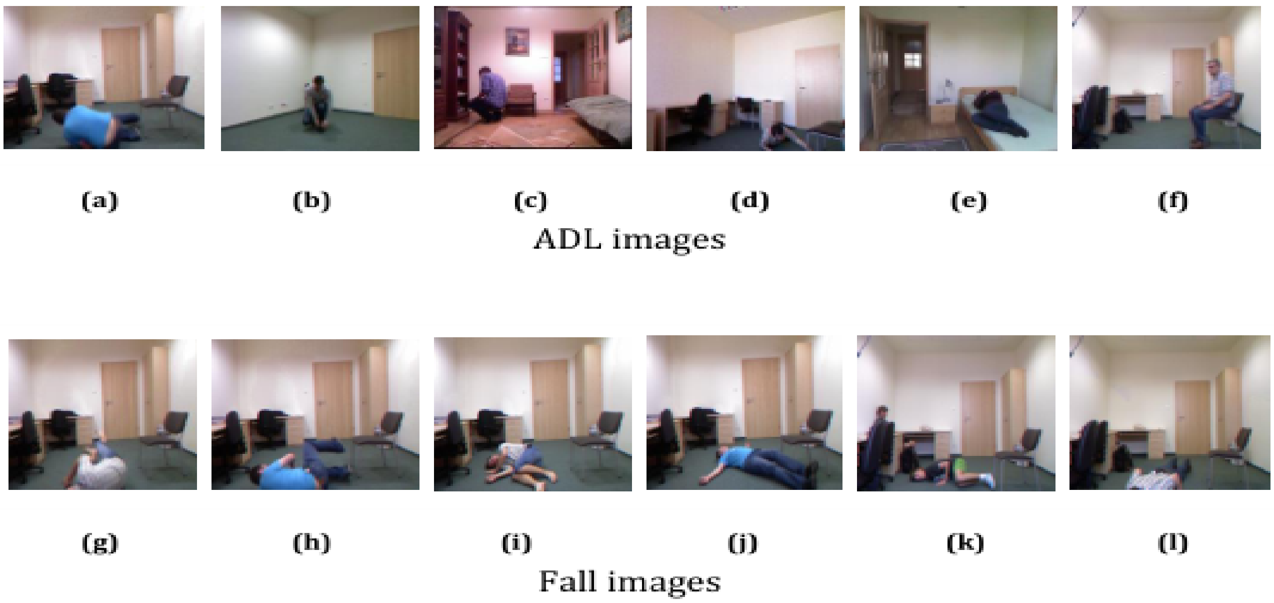

Fig. 1. Illustration of part of normal activities of daily living and simulated falls.

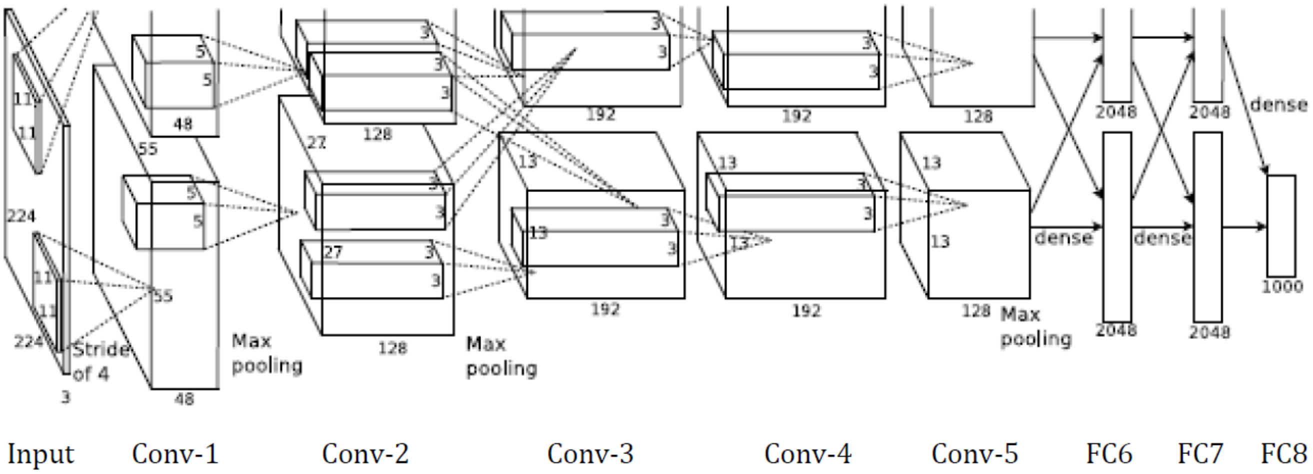

Fig. 2. AlexNet architecture (illustration taken from [16]).

also is used to stop the training before the network begins to overfit the training set.Informally, a model's *capacity* refers to its ability to fit all kinds of functions. Changing its capacity can be used to control whether this model is more likely to overfit or underfit [10]. In general, during training large models, Training error decreases steadily over time (epochs). However, generalization error starts to rise again, where generalization error has a U-shaped curve as a function of model capacity, as illustrated in Fig. 3. From this figure, we observe that the vertical red line corresponds to the model's optimal capacity, which indicates the generalization error is minimum for the current training epoch. In this paper, when training our network terminates, we return all parameters at the epoch with the lowest test error as our optimal model parameters. This strategy is well-known early stopping algorithm that is one of the most commonly used regularization techniques in deep learning.

III. Results

In this section, we evaluate the performance of the proposed approach by conducting 10-fold cross validation experiments on our dataset containing 8500 images. During training our network, we ensure that the network's initial weight parameters and hyperparameters are identical for each fold. In order to evaluate objectively our detection results in each fold, we calculate two criteria, which are *sensitivity* and *specificity* [21]:

$$Sensitivity = \frac{TP}{TP+FN} \tag{1}$$

$$Specificity = \frac{TN}{TN+FP} \tag{2}$$

TABLE I
OUR CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE.

ConvNet Configuration		
Input (227×227 RGB image)		
type	*size of kernel/stride/pad [feature maps - size]*	*patch size/stride [feature maps - size]*
Conv-1	$11 \times 11 \times$ 3/4/0 [96 - 55×55]	-
Max-pooling	-	$3 \times$ 3/2 [96 - 27×27]
Conv-2	$5 \times 5 \times$ 96/1/2 [256 - 27×27]	-
Max-pooling	-	$3 \times$ 3/2 [256 - 13×13]
Conv-3	$3 \times 3 \times$ 256/1/1 [384 - 13×13]	-
Conv-4	$3 \times 3 \times$ 192/1/1 [384 - 13×13]	-
Conv-5	$3 \times 3 \times$ 192/1/1 [256 - 13×13]	-
Max-pooling	-	$3 \times$ 3/2 [256 - 6×6]
FC6	4096 - neurons	
fc7	1024 - neurons	
fc8	2 - neurons	
-	Softmax loss	

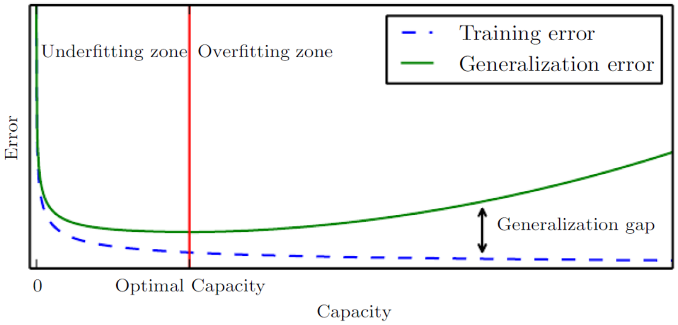

Fig. 3. Learning curves show how training and generalization error changes over the models capacity (illustration taken from [10]).

where true positive (TP) and true negative (TN) are the number of falls and ADL that are correctly detected respectively. The false positive (FP) and false negative (FN) are falsely detected respectively. Sensitivity(SE) refers to the capacity to detect a fall. Specificity(SP) denotes the capacity to detect only a fall. Besides, we also compute *accuracy* as follows:

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{3}$$

where accuracy is the ratio of TP and TN in the population. In the dataset used for the experiments, the fall and ADL datasets are unbalanced. Hence we use the geometric mean of sensitivity and specificity as one of criteria to measure the performances of the proposed approach. This criterion is independent of the size of the dataset. It can be calculated as $\sqrt{SE \times SP}$. For a perfect detection algorithm, the geometric mean has a value of 1. Due to carrying out 10-fold cross validation, there are 850 test images for each fold, and we compute the average detection time per image. Table II shows the experimental results that are obtained in 10-fold cross validation by the proposed approach.

As is shown in Table II, we can see, from Fold 1 to Fold 9, both sensitivity and specificity are equal to 100%. In Fold 10, the specificity is 99.87%. At the bottom of the Table II, we calculate average values for all performance criteria. The average sensitivity, specificity, geometric mean and accuracy are 100%, 99.98%, 99.99%, 99.98%, respectively. This means the network has great robustness for avoiding false alarms on our dataset. Although the dataset we use is very unbalanced (7733 ADL vs. 767 falls), the sensitivity, specificity and geometric mean have a very high value, which shows our convolutional neural network is still able to effectively separate falls from ADL. In Fold 10, the accuracy is 99.88%, which means there exists only a sample that is detected by mistake. Actually, this ADL sample is mistakenly detected as a fall by the network and is shown in Fig. 4. It will be discussed in Section IV.

It is difficult for majority of detection systems to detect a fall among all the daily life activities such as lying down, sitting down and crouching down activities similar characteristics to falls (for example, a large vertical velocity) [7]. For our problem, the challenge is to distinguish falls from lying postures belonging to normal activities in daily life. For example, a person without or with lying position is lying on the floor, sofa and bed. Our dataset contains about 560 images form these postures. According to above experimental results in Table II, our network is almost able to correctly distinguish all images, which shows the proposed method can effectively detect ADL from these postures from 560 images. In terms of computational time, the network takes 0.0234 s in average for each image to detect if a fall occurs meaning real-time feedback for fall detection is possible. Experimental results demonstrate that our network can maximize the fall detection rate with minimal computational complexity.

IV. Discussion and Conclusion

In this paper, we present the use of convolutional neural networks for automatic fall detection in video surveillance environment with high detection accuracy. CNN is directly applied to each frame image in the video to learn human shape deformation features to detect if a fall occurs. In our experi-

TABLE II
SUMMARY OF THE RESULTS IN EXPERIMENTS.

Folds	Sensitivity (%)	Specificity (%)	$\sqrt{SE \times SP}$ (%)	Accuracy (%)	Detection time per frame (s)
Fold 1	100.0	100.0	100.0	100.0	0.024
Fold 2	100.0	100.0	100.0	100.0	0.023
Fold 3	100.0	100.0	100.0	100.0	0.026
Fold 4	100.0	100.0	100.0	100.0	0.023
Fold 5	100.0	100.0	100.0	100.0	0.021
Fold 6	100.0	100.0	100.0	100.0	0.023
Fold 7	100.0	100.0	100.0	100.0	0.024
Fold 8	100.0	100.0	100.0	100.0	0.021
Fold 9	100.0	100.0	100.0	100.0	0.024
Fold 10	100.0	99.87	99.93	99.88	0.025
Average	100.0	99.98	99.99	99.98	0.0234

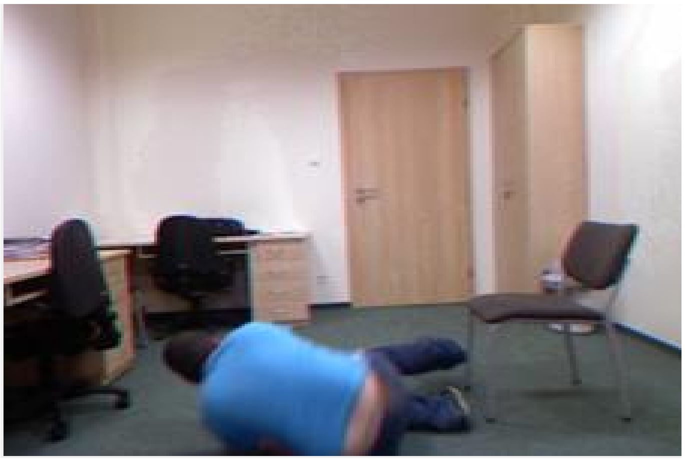

Fig. 4. A ADL sample that is mistakenly detected as a fall.

ments, we take advantage of 10-fold cross validation technique to evaluate the performance of the proposed approach. We do not provide an experimental comparison with other state-of-the-art approaches based on computer vision algorithms. The reasons are as follows. Firstly, as existing approaches do not disclose their own code and more details about their experiments, we cannot directly apply these approaches to our dataset. Secondly, existing approaches are usually evaluated using small datasets that are not used to train our CNN, where training the CNN need a large number of images. Hence we cannot compare the proposed approach with other existing approaches based on the same dataset. Experimental results indicate that our convolutional neural network can almost effectively detect ADL or fall events from each frame image and give a real-time feedback.

Although our approach achieves high detection accuracy, we suggest that sampling frequency of the surveillance videos should not be too high, because images closing to the *critical point* distinguishing ADL or fall from each frame image may not be correctly detected by our approach. As is shown in Fig. 4, this ADL sample is mistakenly detected as a fall. The reason may be that this ADL sample is very similar to some falls such as (h) in Fig. 1. Actually, this sample is from a fall video sequence. According to our labeling criterion, it is labeled as ADL. Taking into account the labeling criterion, we believe that relatively low video sampling frequencies help to improve the robustness of our approach. Besides, to further improve robustness, we need collect a large amount of elder's videos of activities in daily life at specific active regions, to train the convolutional neural network.

There is a challenge for the proposed approach. For example, changing backgrounds of the video such as shadows, moving objects may contribute to performance degradation of the approach. Occlusions and different colors of clothes may also have impact on detection results. In future work, we will conduct further extensive experiments to analyze our approach performance when backgrounds and foregrounds are changed. In the publicly available UR Fall Detection Dataset, all 30 fall video sequences were recorded in a scene with approximately the same background and same viewpoint. In this paper, the images we use from 30 falls almost have the same background, which may have influence on our approach. In order to evaluate our network's performance for falls detection in different viewpoints, future work include also using images captured from different viewpoints and backgrounds to train the convolutional neural network.

ACKNOWLEDGMENT

This work is partly supported by Shenzhen Fundamental Research Projects (Nos: JCYJ20160422113119640, JCYJ20140828152830610), and Shenzhen Science and Technology Program (No. CXZZ20140418182638764).

REFERENCES

[1] Laila Alhimale, Hussein Zedan, and Ali Al-Bayatti. The implementation of an intelligent and video-based fall detection system using a neural network. *Applied Soft Computing Journal*, 18(C):59–69, 2014.

[2] Kabalan Chaccour, Rony Darazi, Amir Hajjam El Hassani, and Emmanuel Andrs. From fall detection to fall prevention: A generic classification of fall-related systems. *IEEE Sensors Journal*, 17(3):812–822, 2017.

[3] Jia Luen Chua, Yoong Choon Chang, and Wee Keong Lim. A simple vision-based fall detection technique for indoor video surveillance. *Signal, Image and Video Processing*, 9(3):623–633, 2015.

[4] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. *Journal of Machine Learning Research*, 15:215–223, 2011.

[5] S Deandrea, E Lucenteforte, F Bravi, R Foschi, Vecchia C La, and E Negri. Risk factors for falls in community-dwelling older people: a systematic review and meta-analysis. *Epidemiology*, 21(5):658–668, 2010.

[6] George Demiris, Marilyn J Rantz, Myra A Aud, Karen D Marek, Harry W Tyrer, Marjorie Skubic, and Ali A Hussam. Older adults' attitudes towards and perceptions of smart hometechnologies: a pilot study. *Medical informatics and the Internet in medicine*, 29(2):87–94, 2004.

[7] N. El-Bendary, Q. Tan, F. C. Pivot, and A. Lam. Fall detection and prevention for the elderly: A review of trends and challenges. *International Journal on Smart Sensing & Intelligent Systems*, 6(3):1230–1266, 2013.

[8] Pengming Feng, Miao Yu, Syed Mohsen Naqvi, and Jonathon A. Chambers. Deep learning for posture analysis in fall detection. In *International Conference on Digital Signal Processing*, pages 12–17, 2014.

[9] Weiguo Feng, Rui Liu, and Ming Zhu. Fall detection for elderly person care in a vision-based home surveillance environment using a monocular camera. *Signal, Image and Video Processing*, 8(6):1129–1138, 2014.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*, 3(4):pgs. 212–223, 2012.

[13] Branka Jokanovic, Moeness Amin, and Fauzia Ahmad. Radar fall motion detection using deep learning. In *Radar Conference*, pages 1–6, 2016.

[14] Branka Jokanovic, Moeness G Amin, and Fauzia Ahmad. Effect of data representations on deep learning in fall detection. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016 IEEE*, pages 1–5. IEEE, 2016.

[15] S. S. Khan and J Hoey. Review of fall detection techniques: A data availability perspective. *Medical Engineering & Physics*, 39:12–22, 2016.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[17] B Kwolek and M Kepski. Human fall detection on embedded platform using depth maps and wireless accelerometer. *Computer Methods & Programs in Biomedicine*, 117(3):489–501, 2014.

[18] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Muhammad Mubashir, Ling Shao, and Luke Seed. A survey on fall detection: Principles and approaches. *Neurocomputing*, 100(2):144–152, 2013.

[20] Thuy-Trang Nguyen, Myeong-Chan Cho, and Tae-Soo Lee. Automatic fall detection using wearable biomedical signal measurement terminal. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5203–5206. IEEE, 2009.

[21] Norbert Noury, Anthony Fleury, Pierre Rumeau, AK Bourke, GO Laighin, Vincent Rialle, and JE Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007.

[22] 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 & Systems for Video Technology*, 21(5):611–622, 2011.

[23] Caroline Rougier, Jean Meunier, Alain Starnaud, and Jacqueline Rousseau. Fall detection from human shape and motion history using video surveillance. In *International Conference on Advanced Information NETWORKING and Applications Workshops*, pages 875–880, 2007.

[24] Caroline Rougier, Alain St-Arnaud, Jacqueline Rousseau, and Jean Meunier. Video surveillance for fall detection. In *Video Surveillance*. InTech, 2011.

[25] L Schwickert, C Becker, U Lindemann, C Marchal, A Bourke, L Chiari, J. L. Helbostad, W Zijlstra, K Aminian, and C Todd. Fall detection with body-worn sensors : a systematic review. *Zeitschrift Fr Gerontologie Und Geriatrie*, 46(8):706–719, 2013.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[27] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *ACM International Conference on Multimedia*, pages 689–692, 2015.

[28] Jared Willems, Glen Debard, Bert Bonroy, Bart Vanrumste, and Toon Goedem. How to detect human fall in video? an overview. In *International Conference on Positioning and Context Awareness*, 2009.

[29] GE Wu and Shuwan Xue. Portable preimpact fall detector with inertial sensors. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(2):178–183, 2008.

[30] Ahmet Yazar and A. Enis etin. Ambient assisted smart home design using vibration and pir sensors. In *Signal Processing and Communications Applications Conference*, pages 1–4, 2013.

[31] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. 8689:818–833, 2013.

[32] Yaniv Zigel, Dima Litvak, and Israel Gannot. A method for automatic fall detection of elderly people using floor vibrations and soundproof of concept on human mimicking doll falls. *IEEE Transactions on Biomedical Engineering*, 56(12):2858–2867, 2009.