Computer vision based fall detection by a convolutional neural network

Miao Yu
School of Computer Science
University of Lincoln
myu@lincoln.ac.uk

Liyun Gong
School of Computer Science
University of Lincoln
lgong@lincoln.ac.uk

Stefanos Kollias
School of Computer Science
University of Lincoln
skollias@lincoln.ac.uk

ABSTRACT

In this work, we propose a novel computer vision based fall detection system, which could be applied for the health-care of the elderly people community. For a recorded video stream, background subtraction is firstly applied to extract the human body silhouette. Extracted silhouettes corresponding to daily activities are applied to construct a convolutional neural network, which is applied for classification of different classes of human postures (e.g., bend, stand, lie and sit) and detection of a fall event (i.e., lying posture is detected in the floor region). As far as we know, this work is the first attempt for the application of the convolutional neural network for the fall detection application. From a dataset of daily activities recorded from multiple people, we show that the proposed method both achieves higher postures classification results than the state-of-the-art classifiers and can successfully detect the fall event with a low false alarm rate.

CCS CONCEPTS

• **Applied computing** → **Health informatics**;

KEYWORDS

healthcare, fall detection, convolutional neural network, classification

ACM Reference format:
Miao Yu, Liyun Gong, and Stefanos Kollias. 2017. Computer vision based fall detection by a convolutional neural network. In *Proceedings of 19th ACM International Conference on Multimodal Interaction, Glasgow, UK, November 2017 (ICMI'17),* 5 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There is an increasing number of elderly people due to the technologies development in the modern society. As shown in [3], the old-age dependency ratio (which means the number of people 65 and over relative to those between 15 and 64) in the European Union (EU) is expected to 54 percent by 2050. The topic of home care for elderly people is becoming a more and more important issue. One important issue for the home care is to detect whether an elderly person has fallen or not.

According to [6], 87% of all fractures of the elderly people group are caused by falls. An efficient fall detection system is essential for monitoring an elderly person and can even save his life in some cases. Different methods have been proposed for detecting falls. [9, 17] utilized acceleration sensors based method for fall detection. In their work, three axis acceleration sensors are attached to the subject's body in different positions and the dynamic and static acceleration components measured from these sensors were compared with appropriate thresholds to determine a fall. Y. Zigel et al. in [20] proposed a fall detection system based on both floor vibration and sound sensing. Temporal and spectral features were extracted from signals and a Bayes' classifier was applied to classify fall and nonfall activities. Although these methods may appear to be suitable for fall detection in an ideally simulated scenario, several problems do exist which prohibits them from the real home applications due to the following reasons: i). they are either inconvenient (elderly people have to wear acceleration sensors) ii). they are easily affected by noises (such as the TV sounds) in the environment (acoustic sensors and floor vibration sensors).

In order to overcome these problems, computer vision based fall detection techniques are adopted. In [14] and [15], the head's velocity information and the shape change information were extracted from video recording and appropriate thresholds were set manually to differentiate fall and non-fall activities. However these two methods produce high false detection rates (such as when a fast sitting activity was misclassified as a fall activity in [14]). In [2], multiple calibrated cameras were used to reconstruct the three-dimensional shape of people. Fall events were detected by analyzing the volume distribution along the vertical axis, and an alarm was triggered when the major part of this distribution was abnormally near the floor over a predefined period of time. With the development of the machine learning techniques, some of them have been applied for the computer vision based fall detection. Neural network and support vector machine based techniques have been applied for fall detection based on the posture recognition as in [8] and [19] respectively. Motion information were extracted from consecutive silhouettes as features in [1] to train a hidden Markov model (HMM) for classifying fall and non-fall activities.

The traditional machine learning methods, such as the support vector machine follows shallow learning paradigm.

ICMI'17, November 2017, Glasgow, UK

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

They can't represent the object of interest in a reasonably structural and hierarchical way and rely carefully chosen hand-crafted features. The limitations of the traditional machine learning methods are largely overcome by the deep learning, which has gained strong attentions from the academic community and been widely applied in different industrial applications (e.g., voice classification, object detection/classification) as shown in the recent survey [13] by Y. LeCun, Y. Bengio and G. Hinton. One type of the deep learning techniques is the convolutional neural network (CNN), which exploits convolutional layers to extract highly representative features and has been widely applied in image processing tasks. As shown in [11], the CNN achieves a much higher accuracy than the traditional machine learning method for the ImageNet object classification.

In this work, a CNN based method is used for fall detection. Firstly, human silhouette is extracted from the raw video stream. Extracted silhouettes for different classes of postures corresponding to daily activities (stand, sit, bend and lie) are pre-processed and used to construct a CNN for postures classification. Fall is then detected based on the posture classification results, that is, a fall event is triggered when a lying posture is detected in the floor region. *As far as we know, this is the first attempt for the CNN to be used in the fall detection application.* The organization of this paper is shown as follows: Section 2 presents the background subtraction method we use to extract the human silhouette. CNN construction for postures classification and fall detection is proposed for Section 3. Experimental results are shown in Section 4 and Section 5 gives the final conclusions and future works.

2 BACKGROUND SUBTRACTION

In visual surveillance, a common approach for discriminating moving objects from the background is detection by background subtraction. In this work, the codebook background subtraction method [10] is applied. Compared with other methods such as single-model based as in [7, 18] and mixture of Gaussians (MoG) method [16], there is no parametric assumption on the codebook model and it can achieve better performance by exploiting more comprehensive information from the color space information as in [10].

The codebook method is a pixel-based approach and initially a codebook is constructed for each pixel during a training phase. Assuming the training dataset $\mathbf{I}$ contains a number of N images: $\mathbf{I} = \{imag_1, ..., imag_N\}$, then for a single pixel (x,y) it has N training samples $imag(x,y)_1, ..., imag(x,y)_N$. From these N training samples, a codebook is constructed for this pixel, which includes a certain number of codewords. Each codeword, denoted by $\mathbf{c}$, consists of an RGB vector $\mathbf{v} = (\overline{R}, \overline{G}, \overline{B})$ and a 6-tuple $\mathbf{aux} = (\hat{I}, \check{I}, f, \lambda, p, q)$. Meanings of the six parameters in **aux** are described in Table 1.

The details of the training procedure are given in [10] and the trained codebooks of pixels are then used for background subtraction purpose. For an incoming colour frame $\mathbf{f}$, its pixel $\mathbf{f}(x,y) = (R(x,y), G(x,y), B(x,y))$ (a 3-dimensional vector) is determined as a foreground or background pixel by comparing $\mathbf{f}(x,y)$ with codewords in the codebook of this pixel. If $\mathbf{f}(x,y)$ is not matched with any codeword, then it is a foreground pixel;otherwise, it is taken as the background. We say $\mathbf{f}(x,y)$ matched the codeword $\mathbf{c}$ if the following two conditions are met:

Table 1: Meaning of the codeword components

$\hat{I}$	Maximum intensity represented by the codeword
$\check{I}$	Minimum intensity represented by the codeword
f	Number of times of the codeword being matched
λ	Maximum negative runtime length (MNRL)
p	The first frame in which this codeword was created
q	The last frame in which this codeword was matched

$$colordist(\mathbf{f}(x,y), \mathbf{c}) \leq \varepsilon$$
$$brightness(I, \langle \hat{I}, \check{I} \rangle) = true \quad (1)$$

where ε is a preset threshold value for comparison, I represents the norm of $\mathbf{f}(x,y)$, $\hat{I}$ and $\check{I}$ are the first two parameters of the 6-tuple **aux** vector of the codeword $\mathbf{c}$.

The $colordist(\mathbf{f}(x,y), \mathbf{c})$ measures the chromatic difference between two colour vectors, which can be calculated by:

$$colordist(\mathbf{f}(x,y), \mathbf{c}) = \sqrt{\| \mathbf{f}(x,y) \|^2 - \frac{\langle \mathbf{f}(x,y), \mathbf{v} \rangle}{\| \mathbf{v} \|^2}} \quad (2)$$

where $\mathbf{v}$ represents the RGB vector $\mathbf{v} = (\overline{R}, \overline{G}, \overline{B})$ of codeword $\mathbf{c}$, and $\| \cdot \|$ and $\langle \cdot \rangle$ denote respectively the Euclidean norm and dot product operations.

The $brightness(I, \langle \hat{I}, \check{I} \rangle)$ is defined as:

$$brightness(I, \langle \hat{I}, \check{I} \rangle) = \begin{cases} true & if\ I_{low} \leq \| \mathbf{f}(x,y) \| \leq I_{hi} \\ false & \text{otherwise} \end{cases} \quad (3)$$

where $I_{low} = \alpha \hat{I}$ and $I_{hi} = min\{\beta \hat{I}, \frac{\check{I}}{\alpha}\}$ are parameters which are set empirically.

Background subtraction examples are shown in Figure. 1. When a human object appears in the camera view, its silhouette region is extracted from the original video recording.

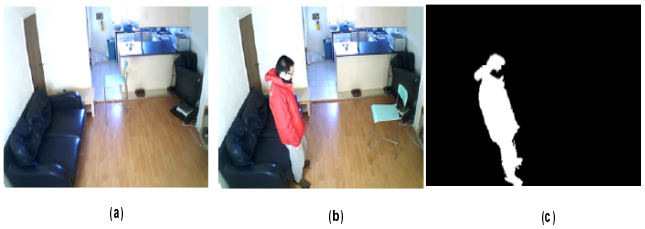

Figure 1: Background subtraction results. (a). the background scene (b). video frame with the object of interest (c). extracted human silloutte by the background subtraction

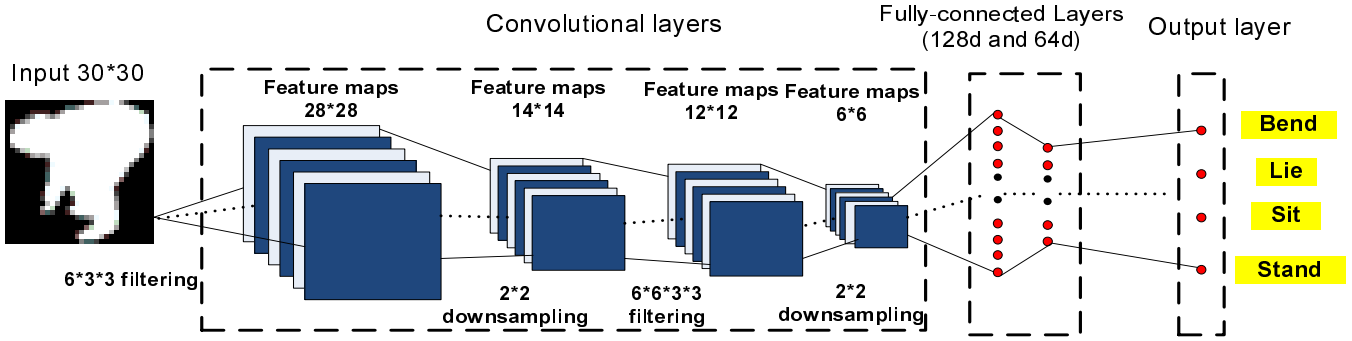

Figure 2: The structure of the developed convolutional neural network.

3 CNN FOR POSTURES RECOGNITION AND FALL DETECTION

The human silhouettes corresponding to different daily activities are extracted from video recordings by the aforementioned background subtraction method, which are further used to construct a CNN for the fall detection. For the CNN construction, firstly the minimum bounding rectangle (MBR) of every silhouette is extracted. All the MBRs are resized to be the same (30*30) for the CNN training, with the aspect ratio of the silhouette region being kept; besides, each pixel of the resized image is normalized to be within [0,1].

The structure of the trained CNN is shown as in Figure. 2. The extracted human silhouette is taken as the input of the convolutional layers, which exploit two sets of six filters for feature maps generations with *Relu* activation function being applied to obtain the final neurons outputs. A downsampling of 2*2 is applied to reduce redundant components for a more concise representation. The outputs after two convolutional layers are fed into two fully-connected layers, with 128 and 64 neurons respectively. Finally, an output layer with four neurons generates probabilities of activity classes, with a sigmoid function being taken as the activation function. For training the network, the categorical cross-entropy cost function is adopted. Based on the cost function, the root mean square propagation (RMSProp) method [12] is adopted for estimating the weights of the CNN, with batches of 64.

The trained CNN can then be applied for classifying different types of postures. When a lying posture is detected within the floor region, a fall is reported. For the floor region, it can either be marked manually or detected by an unsupervised method based on the foot positions [19].

4 EXPERIMENTAL RESULTS

In this part, we show the performance of our fall detection system in a real home environment. A USB camera was used for recording the real video sequence with the image size of 320×240, the recorded video sequence is processed by using VC++ 6.0 (with OpenCv library 1.0) for background subtraction and Keras [5] for the CNN training. 10 people were invited to attend the experiments for simulating different postures (bend, sit, lie and stand)and activities (both fall and non-fall).

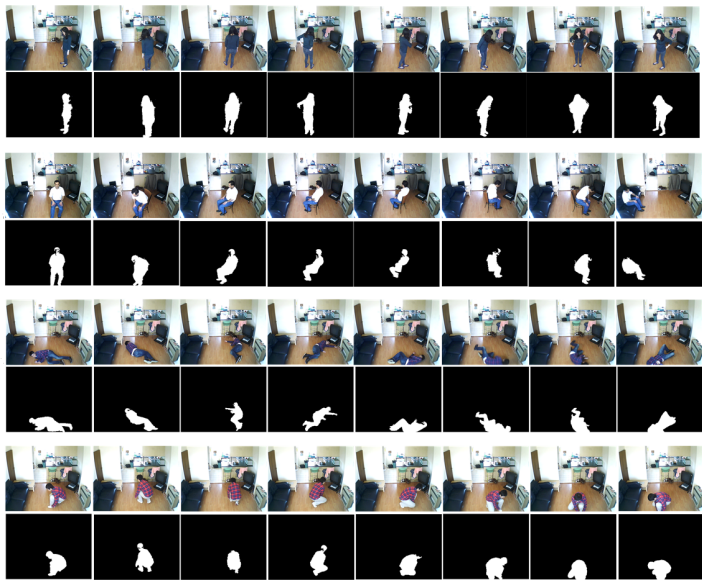

Figure 3: Posture samples simulated by different participates in different orientations.

4.1 Posture classification results comparison

A posture dataset containing 3216 postures (including 804 stands, 769 sits, 810 lies and 833 bends) were recorded for testing the developed CNN. As in [18], each person was asked to simulate postures in different directions so that the constructed classifier is robust to view angles. Some samples are shown in Figure. 3.

Table 2: Postures classification accuracies comparisons by different classifiers

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1-v-a	96.25%	94.12%	94.06%	**92.58%**	93.75%	**93.51%**	92.65%	96.31%	89.83%	90.74%
1-v-1	95.94%	94.43%	95.31%	91.94%	92.19%	92.04%	91.18%	96.62%	88.47%	90.74%
CNN	**96.88%**	**97.83%**	**96.56%**	**92.58%**	**95.31%**	92.04%	**93.24%**	**96.62%**	**93.56%**	**94.75%**

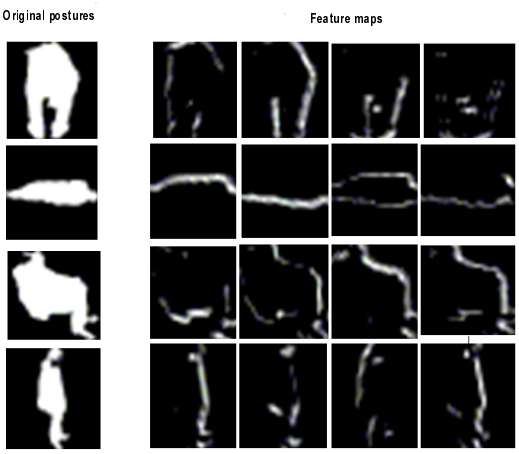

Figure 4: Original postures and selective feature maps extracted by the CNN.

Different classes of postures are used to construct the CNN structure shown in Fig. 2. For the trained CNN, firstly we show representative feature maps extracted by the CNN in Figure. 4, from which we can see that the CNN captures the silhouette edge as distinctive features for the posture classification. That coincides with our intuition that a binary human silhouette image and be well represented by its edge.

Secondly, comprehensive comparisons are made between the CNN and multi-class support vector machine [4] (including both the one-versus one (1-v-1) and one-versus-all (1-v-a) versions) for classifying the postures collected from every individual (P1-P10 as in Table 2). For testing the postures classification performance for a particular individual, postures of others are used for training. For the multi-class SVMs, the linear kernel is applied for both computational efficiency and the ability to separate high dimensionality features. The parameters of both the CNN and SVM based methods are tuned to be the optimal for a fair comparison. Comparison results are shown in Table 2, which show the advantage of the CNN. The CNN outperforms the SVM based methods for the majority of individual cases with higher classification accuracies.

4.2 Fall detection comparison

For evaluating this fall detection system, each person is asked to simulate different falling and non-fall activities in different directions. As mentioned previously, fall is detected when the lying posture is detected by the CNN within the ground region, as shown in Figure. 5.

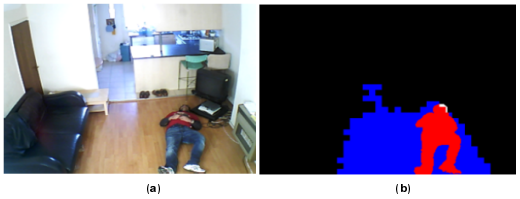

Figure 5: An example of simulated fall. The ground region of (a) is marked as blue as in (b) and the posture silhouette within the ground region is marked red.

The classification of fall and non-fall activities is shown in Table 3. We can observe that the proposed CNN based fall detection systems could accurately detect falls (with only one fall is not detected among the totally recorded falls) with a low false alarm rate (3 out of 310 non-falls are misclassified as falls). The misclassifications are attributed to the reason that the bend posture in some directions is viewed similar to the lie posture and we can adopt multiple cameras to capture different posture views for ameliorating it.

Table 3: Performance of the CNN based fall detection

Activity types	Numbers	Detected falls	Detected nonfalls
Falls	98	97	1
Walk around	92	0	92
Sit on sofa/chair	86	0	86
Bend	132	3	129

5 CONCLUSIONS

In this work, we proposed a novel CNN based computer vision fall detection method. A codebook background subtraction algorithm was adopted to extract the human silhouette region. Extracted silhouettes are pre-processed and applied to train a CNN, for both the postures classification and fall detection. Experimental results show the proposed CNN classifier both achieves better performance than the traditional ones for posture classification, as well as obtains a high fall detection accuracy. For the future works, we will exploit information from multi-modality sensors (such as audio and video sensors) together with deep learning techniques for developing a more robust fall detection system.

REFERENCES

[1] D. Anderson, J. Keller, M. Skubic, X. Chen, and Z. He. 2006. Recognizing Falls from Silhouettes. In *Proceedings of the 28th IEEE EMBS Annual International Conference*.

[2] E. Auvinet, F. Multon, A. Saint-Arnaud, J. Rousseau, and J. Meunier. 2011. Fall Detection With Multiple Cameras: An Occlusion-Resistant Method Based on 3-D Silhouette Vertical Distribution. *IEEE Transactions on Information Technology in Biomedicine* 15, 2 (2011), 290–300.

[3] G. Carone and D. Costello. 2006. Can Europe Afford to Grow Old? *Finance and Development* 43, 3 (2006).

[4] C. Chang and C. Lin. 2011. Libsvm: A library for support vector machine. *ACM Transactions on Intelligent System and Technology* 2, 3 (2011), 27:1–27:27.

[5] F. Chollet et.al. 2015. Keras. https://github.com/fchollet/keras. (2015).

[6] J. Halter, J. Ouslander, M. Tinetti, S. Studenski, K. High, S. Asthana, and W. Hazzard. 2009. Hazzard's Geriatric Medicine and Gerontology. *Sixth Edition, McGraw-Hill* (2009).

[7] T. Horprasert, D. Harwood, and L.S. Davis. 1999. A statistical approach for real-time robust background subtraction and shadow detection. *In Proc. IEEE ICCV* 99 (1999), 1–19.

[8] C. Juang and C. Chang. 2007. Human Body Posture Classification by a Neural Fuzzy Network and Home Care System Application. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans* 37, 6 (2007), 984–994.

[9] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jamsa. 2008. Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait and Posture* 28, 2 (2008), 285–291.

[10] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis. 2005. Real-time foreground-background segmentation using code-book model. *Real-Time Imaging* 11, 3 (2005), 172–185.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. 2012. ImageNet classification with deep convolutional neural networks. *In Proc. of Advances in Neural Information Processing Systems* 25, 1090–1098.

[12] Q. Le, J. Ngiam, A. Lahiri, B. Prochnow, and A. Ng. 2011. On Optimization Methods for Deep Learning. *28th International Conference on Machine Learning, Bellevue, WA, USA* (2011).

[13] Y. LeCun, Y. Bengio, and G. Hinton. 2015. Deep Learning. *Nature* 521 (2015), 436–444.

[14] C. Rougier and J. Meunier. 2010. 3D Head Trajectory using a Single Camera. *International Journal of Future Generation Communication and Networking, invited paper for the special issue on Image and Signal Processing* 3, 4 (2010), 43–54.

[15] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. 2007. Fall Detection from Human Shape and Motion History Using Video Surveillance. *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW), Niagara Falls, Ont., Canada.*

[16] C. Stauffer and W. Grimson. 1999. Adaptive background mixture models for real-time tracking. *Int. Conf. Computer Vision and Pattern Recognition, Fort Collins, CO, USA* (1999).

[17] P. Veltink, H. Bussmann, W. Vries, W. Martens, and R. Lummel. 1996. Detection of static and dynamic activities using uniaxial accelerometers. *IEEE Transactions on Rehabilitation Engineering* 4, 4 (1996), 375–385.

[18] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. 1997. Pfnder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 7 (1997), 780–785.

[19] M. Yu, A. Rhuma, S. Mohsen, , L. Wang, and J. Chambers. 2012. A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. *IEEE Transactions on Information Technology in Biomedicine* 6, 16 (2012), 1274–1286.

[20] Y. Zigel, D. Litvak, and I. Gannot. 2009. 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* 56, 12 (2009), 2858–2867.