Deep Learning for Posture Analysis in Fall Detection

Pengming Feng, Miao Yu, Syed Mohsen Naqvi, Jonathon A. Chambers

Advanced Signal Processing Group, Electronic and Electrical Engineering Department
Loughborough University, Loughborough, Leicester, UK
{p.feng, m.yu, s.m.r.naqvi, j.a.chambers}@lboro.ac.uk

ABSTRACT

We propose a novel computer vision based fall detection system using deep learning methods to analyse the postures in a smart home environment for detecting fall activities. Firstly, background subtraction is employed to extract the foreground human body. Then the binary human body images form the input to the classifier. Two deep learning approaches based on a Boltzmann machine and deep belief network are compared with a support vector machine approach. The final decision on the occurrence of a fall is made on the basis of combining the classifier output with certain contextual rules. Evaluations are performed on recordings from a real home care environment, in which 15 people create 2904 postures.

Index Terms— Fall detection, deep learning, Boltzmann machine, deep belief network, support vector machine (SVM), multiclass classification

1. INTRODUCTION

With the global problem of aging populations becoming more and more serious, there will be increasing numbers of elderly people who wish to stay at home but who are not healthy enough to fully take care of themselves on their own. So their safety is becoming a big challenge. In this case, it is really very important to design a system that can be used to generate an alarm when it detects the falling of human beings so that they can gain help from others without too many false alarms.

1.1. Current Fall Detection Techniques

Nowadays, different methods have been proposed for detecting falls; when classified by tools they use, such anomaly detection systems can be divided into the following three kinds: methods using wearable sensors, acoustic sensors and video cameras.

a) Wearable sensor-based system: Physiological measures of interest for use in rehabilitation are heart rate, respiratory rate, blood pressure, blood oxygen striation and muscle activity [1]. From the biological indicator of a human being extracted from the sensor, we can obtain indication of abnormality in the biological signal as a precursor to a fall. For example, Asada et al. [2] presented a ring sensor used for measuring blood oxygen saturation and heart rate. The sensor integrated a novel height sensor based on two manufactured microelectromechanical systems (MEMS) accelerometers for measuring the hydrostatic pressure offset of the photoplethysmographic (PPG) sensor relative to the heart. The mean arterial blood pressure was derived from the PPG sensor output amplitude by taking into account the height of the sensor relative to the heart [1]. If such a system detects abnormality in the heart rate, then the system may determine that a fall may have occurred.

b) Acoustic-based system: Some researchers use acoustic sensors, for example microphones, to detect falls based on the acquired audio signal, which can collect the audio signal from the environment. If the detected human being has fallen, there will be anomalous sounds; by detecting this kind of signal, we can get an alarm when the human being has fallen over. For example, in [3], Zigel et al. designed a fall detection system based on a floor vibration sensor and a microphone; the vibration and sound signals were obtained from the hardware equipment and temporal and spectral features were extracted from the obtained signals. A Bayes' classifier was then applied to classify fall and non-fall activities based on the extracted features.

c) Computer vision-based system: For computer vision-based systems, some researchers have extracted information from the video and set a threshold to determine whether there is a fall or not, some researchers have extracted some important features from the frames and perform classification of fall and non-fall events. For example, Yu et al. proposed a system which is based on posture recognition using a single camera to monitor the person in the room environment [4]. In this work, firstly, the features were extracted from an ellipse fitted to the human profile and a projection histogram, then a multiclass support vector machine was used for classification purpose, which is a scheme to combine multiple SVMs, it can be used to solve two-class classification problems, to achieve multiclass classification [5].

Though these three methods can all be used to detect falling, there still exist some advantages and shortcomings within the

three methods. The following table shows some details of the advantages and disadvantages of the detection methods:

Table 1: Comparisons of fall detection methods

METHODS	ADVANTAGES	DISADVANTAGES
Wearable sensors detection	By wearing sensors, the system can detect different kinds of biological indicator of elderly people to achieve fall detection.	It is not very convenient for elderly people to wear sensors all the time and they may even forget to wear them without others reminding them if they live alone.
Acoustic detection	Easy to detect abnormal sounds, convenient for old people, easy to transport data to a computer.	May be affected by many kinds of noises from the environment.
Video detection	Very convenient for old people and few facilities are used in the system with little interference from environment.	Computation complexity of processing video is high. Privacy of users maybe a problem.

In this work, we chose the method of using video, which is based on computer vision and exploits a camera/video and signal processing techniques that can utilize the real-time movement of the subject.

1.2. Proposed system

In this paper, we propose a novel computer vision system for fall detection which is based on posture analysis using a single camera to monitor an elderly person who lives alone at home. Firstly, a codebook background subtraction algorithm is applied to extract the area of the human body, then post processing is employed to improve the results, in this way, we can get the extracted foreground silhouette. Then these silhouettes are fed into deep learning classifiers (which are trained by a dataset containing silhouette from different postures) and particular silhouettes are classified as one of the postures in the room environment, from which we can decide which silhouettes represents falls. We chose to use the method of deep learning for classification because of the following advantages: it provides richer imformation about the observed data by capturing interactions of many different factors on different levels through hierarchical learning and benefits from larger training sets.

2. SYSTEM COMPONENTS

The system we design is formed with the following components: background subtraction, post processing, classification, and decision, the flow chart of the system is shown in Fig. 1:

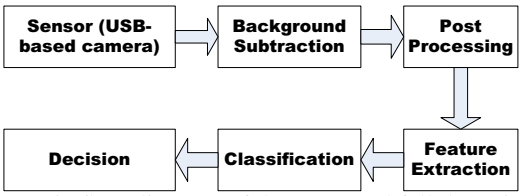

Fig. 1: Basic flow diagram of the proposed fall detection system.

As in the above figure, we show the system can be divided into five parts, the first part is the sensor part, in which we use a camera as a sensor to collect video and data of the whole environment, besides the body we want to detect. Then we come to the background subtraction part. In this part, because we only want information related to the human being, we should subtract the environment part in the images acquired from the camera; to achieve this, the first thing in the method we choose is to build a codebook [6] and train the codebook for the background, then we can extract the silhouette of the person. After the background extraction, we find the extracted person is affected by noise, which may be caused by the shadow of the body, to avoid this, we use post-processing to reduce such noise. In this system, this is achieved with a blob-merging method [7]. Lastly, we perform the classification of the incoming data, in this part, we use deep learning methods for classification, we use both deep belief networks and a restricted Boltzmann machine to analyse postures of humans for performing classification and we make comparison with a traditional SVM classifier. Then we can detect the falls of elderly person in the room environment.

2.1. Background subtraction

In the video based detection system, because there are still some unnecessary components in the environment which will influence the result we want; firstly we should discriminate moving objects from the background, we can use the method of background subtraction to achieve this. In this fall detection system, we use the codebook method to perform background subtraction for the frames from the incoming sequence. After measuring the sequence of images, a code book which contains each pixel of the training samples should be constructed for each pixel during the training phase. Inside each code book, there should be a number of code words. In each code word, as described in [4] and [6], let $\zeta = c_1, c_2, ...c_L$ represent the codebook for the pixel consisting of L codewords, for each codeword c_i which should contain an $\mathbf{RGB}$ vector $\mathbf{v} = (\bar{R}, \bar{G}, \bar{B})$ and a six-tuple vector $\mathbf{aux} = (\hat{\mathbf{I}}, \check{\mathbf{I}}, \mathbf{f}, \lambda, \mathbf{p}, \mathbf{q})$, the meaning of which can be seen below:

$\hat{\mathbf{I}}$: Maximum intensity that has been represented by the code word.

$\check{\mathbf{I}}$: Minimum intensity that has been represented by the code

word.
$\mathbf{f}$: Number of times that the code word has been used.
λ: Maximum negative runtime length (MNRL) in number of frames.
$\mathbf{p}$: The first frame in which this code word was used.
$\mathbf{q}$: The last frame in which this code word was used.

During the background phase, each incoming frame should compare colordist and brightness with any code word in the code book as below, the equation of colordist and brightness can be seen follow, which is described in [6]:

$$colordist(f(x,y),c) \leq \varepsilon \tag{1}$$

$$brightness(I, \langle \hat{\mathbf{I}}, \check{\mathbf{I}} \rangle) = true \tag{2}$$

$$colordist(f(x,y),c) = \sqrt{\|f(x,y)\|^2 - \frac{\langle f(x,y), v \rangle}{\|v\|^2}} \tag{3}$$

$$brightness(I, \langle \hat{\mathbf{I}}, \check{\mathbf{I}} \rangle) = \begin{cases} true & if \mathbf{I}_{low} \leq \|f(x,y)\| \mathbf{I}_{hi} \\ faulse & otherwise \end{cases} \tag{4}$$

The algorithm for codebook construction is given as the the algorithm below [6]:

Algorithm 1 Codebook Algorithm

1. L $\leftarrow 0^1, \zeta \leftarrow= \emptyset$ empty set
2. **for** t = 1 to N do
   (i) $\mathbf{x}_t = (R,G,B)$, $I \leftarrow \sqrt{R^2, G^2, B^2}$
   (ii) Find the codeword $\mathbf{c}_m$ in $\zeta = \{\mathbf{c}_i | 1 \leq i \leq L\}$ matching to $\mathbf{x}_t$ based on the two conditions (a) and (b)
   (a) $colordist(f(x,y),c) \leq \varepsilon$
   (b) $brightness(I, \langle \hat{\mathbf{I}}, \check{\mathbf{I}} \rangle) = true$
   (iii) If $\zeta \leftarrow= \emptyset$ or there is no match, then L $\leftarrow$ L + 1. Create a new codeword c_L by setting:
   $\mathbf{v}_L \leftarrow (R,G,B)$
   $\mathbf{aux}_L \leftarrow \langle I, I, t-1, t, t \rangle$.
   (iv) Otherwise, update the matched codeword $\mathbf{c}_m$, consisting of:
   $\mathbf{v}_m = (\bar{R}_m, \bar{G}_m, \bar{B}_m)$ and $\mathbf{aux}_m = (\hat{\mathbf{I}}m, \check{\mathbf{I}}m, \mathbf{f}m, m, \mathbf{p}m, \mathbf{q}m)$ by setting:
   $\mathbf{v}_m \leftarrow (\frac{f_m \bar{R}_m + R}{f_m}, \frac{f_m \bar{G}_m + G}{f_m}, \frac{f_m \bar{B}_m + B}{f_m})$
   $\mathbf{aux}_m \leftarrow \langle min\{\hat{\mathbf{I}}_m, I\}, max\{\mathbf{I}_m, I\}, \mathbf{f}_m + 1, max\{_m, t - q_m\}, \mathbf{p}_m, t \rangle$
   end for
3. For each codeword, wrap around λ_i by setting $\lambda_i \leftarrow max\{\lambda_i, (N - q_i + p_i - 1)\}$

2.2. Classification

An appropriate classifier is needed to analyse features and classify them into one of the four posture classes. In this work, we use both deep belief networks (DBNs) and restricted Boltzmann machines (RBMs) methods for classification. Deep learning methods use a hierarchical learning method when training the structure of the system, as described in [8], the hierarchical learning uses natural progression from low level to high level structure as seen in natural complexity, so it is easier to monitor what is being learnt and to guide the machine to better subspaces. For example, when the images are input in the system, the first layer of the system represents the $'$edges$'$ of the feature, the second layer represents the $'$object parts$'$ of the feature and the third layer represents the objects in the features [8].

The restricted Boltzmann machine (RBM) is a generative stochastic neural network that can learn a probability distribution over its set of inputs, which has a hidden layer and a visible layer, the structure of which is shown in Fig. 2:

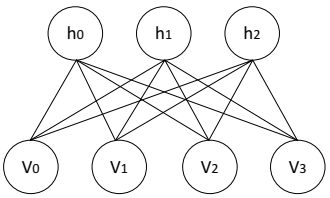

Fig. 2: Structure of restricted Boltzmann machine (RBM) with four visible units and three hidden units (no bias units), where h denotes the hidden units and v denotes the visible units, as it is shown in the figure, only hidden units and visible units are connected and there are no hidden-hidden or visible-visible connections, diagram taken from [8].

As described in [9], RBMs are trained to maximize the product of probabilities assigned to some training set $\mathbf{V}$, the algorithm used mostly to train the RBMs is to optimize the weight vector $\mathbf{W}$, which can be summarized as follows [10]:

Algorithm 2 RBMs Process Algorithm [10]

1. Take a training sample $\mathbf{v}$, compute the probabilities of the hidden units and sample a hidden activation vector $\mathbf{h}$ from this probability distribution.
2. Compute the positive gradient from outer product of $\mathbf{v}$ and $\mathbf{h}$.
3. Sample a reconstruction $\mathbf{v}'$ of the visible units from $\mathbf{h}$, then resample the hidden activations $\mathbf{h}'$ from this.
4. Compute the negative gradient from outer product of $\mathbf{v}'$ and $\mathbf{h}'$.
5. Let the weight update to $\omega_{i,j}$ be the positive gradient minus the negative gradient.

The DBNs used in this work go through two training phases [11]: a generative pre-training phase that initializes the weights to a good location in weight space and a discriminative fine-tuning phase in which the network is trained to perform a specific classification task. As shown in Fig. 3:

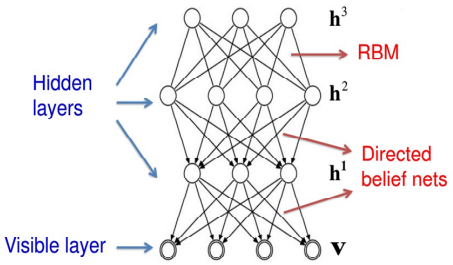

Fig. 3: Structure of deep belief network, where firstly construct the RBM then stack hidden layer on top of the RBM to form new RBM till all layers are trained, diagram taken from [8].

Firstly, the network is trained in an unsupervised fashion, and in this work, we trained the weight layer as a restricted Boltzmann machine (RBM), the RBM is constructed between an input layer and a hidden layer, then we stack another hidden layer on top of the RBM to form a new RBM, then continue to stack layers on top of the network until all hidden layers have been initialized. After this learning step, the DBNs can be further trained in a supervised way to perform classification.

The process of the DBNs is described as [12]:

Algorithm 3 DBNs Process Algorithm [12]

1. Train the first layer as an RBM that models the raw input $x = h^0$ as its visible layer.
2. Use that first layer to obtain a representation of the input that will be used as data for the second layer. Two common solutions exist. This representation can be chosen as being the mean activations $p(h^1 = 1|h^0)$ or samples of $p(h^1|h^0)$
3. Train the second layer as an RBM, taking the transformed data (samples or mean activations) as training examples (for the visible layer of that RBM).
4. Iterate (2 and 3) for the desired number of layers, each time propagating upward either samples or mean values.
5. Fine-tune all the parameters of this deep architecture with respect to a supervised training criterion, after adding extra learning machinery to convert the learned representation into supervised predictions for example, a linear classifier, then we can achieve classification.

2.3. Fall detection contextual rules

After analyzing the posture by classification, we can determine the posture of the human body in the room. Since most fall activities end up with a 'lie' posture and this 'lie' posture usually remains for a certain time due to the period of immobility of an elderly person after the fall, (besides, different from lying on the bed or sofa, the posture should be inside the ground region), we can determine if a fall activity happened by checking if the posture has met the following conditions:
1) The posture is classified as 'lie'.
2) The posture is inside the ground region.
3) The aforementioned two conditions are kept for a certain time, which exceeds a preset time threshold (we use 30s).
In order to detect the region of floor automatically, we can let the human being walk around the area and then by detecting the region in which he was walking, we can mark the area when he was standing, then we can get the region of floor. To mark the region, firstly, we have to build a dataset large enough for detecting, to achieve this, we just let the human being to walk around the room. As the training period above, we have the label of postures of a human being, so we can just mark the region when the label is 'standing, sitting or bending'. The region we obtained from the dataset is shown in Fig. 4:

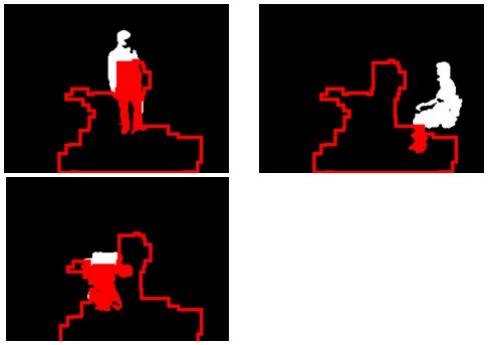

Fig. 4: Floor region detection, active area boundary marked in red.

As we can see from Fig 4, the region of floor is identified by the region of the feet area of human body. In this way, we can detect the floor area for fall detection.

3. EVALUATION RESULTS

In this part, we show the performance of our fall detection system. 15 people were invited to attend the experiment for simulating different postures and activities and 2904 postures (including 726 stands, 726 sits, 726 bends and 726 lies) are used to train the classification system, 294 postures (including 73 stands, 73 sits, 73 bends and 75 lies) are used to test the system.

3.1. Postures in room environment

The postures of the human in the room environment can be divided into four kinds: standing, sitting, bending and lying, which is shown in Fig. 5:

Fig. 5: Postures in room environment: a) standing; b) sitting; c) bending; d) lying on settee.

When the human in the room is lying on the floor for a period longer than a threshold, we can deduce the human has fallen, as shown in Fig. 6:

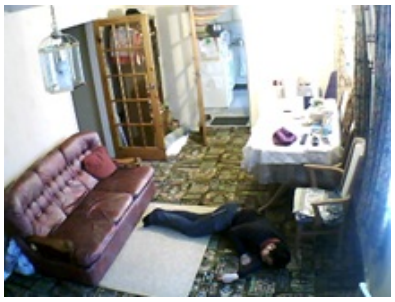

Fig. 6: Postures of falls of a human being in the room environment.

3.2. Background subtraction results

Some background subtraction results are shown in Fig. 7: From the figure above, we can obtain the posture of the human being in the room environment, which forms the input to the classifier. To make the results more accurate, we use postprocessing method to remove the noise in the frames, the results from the postprocessing are shown as Fig. 8. Then we can extract the area of the human being from the result from the result of postprocessing and use deep learning method to make classification.

Fig. 7: Background subtraction results, where the first row is the frames from the camera and the second row denotes the results from background subtraction.

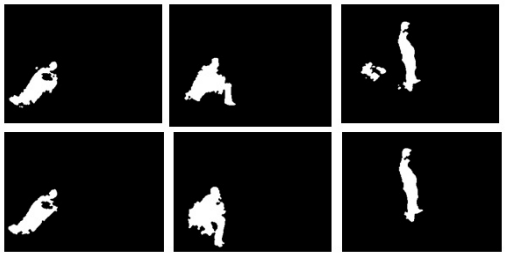

Fig. 8: Postprocessing results, where the first row is the frames from the background subtraction step and the second row denotes the results from postprocessing, where most of the noises are removed.

3.3. Posture analysis and classification results

In this work, we use a dataset with 3124 frames to tain the deep learning network. Examples of the dataset we use to tain the network are shown in Fig. 9 below: In this work, we use a dataset with 2904 frames to train the classification system, then we use a test dataset with 324 frames to test the system. Firstly, we used the RBM method to achieve classification, to determine the classification result, we try different numbers of hidden layers, the results are shown with a posture classification accuracy rate and a false fall detection rate (which including the no-fall detected to be a fall and a fall detected to be a non-fall) in Table **2** below: Then we tried different numbers of hidden layers of DBNs by using the toolbox [13], the results are shown in Table **3** below: and further improve-

Fig. 9: Postures used to train the deep learning classifier and making classification of different postures, where the first row denotes standing, the second row denotes sitting, the third row denotes bending and the last row denotes lying.

Table 2: Classification results accurate from RBM with different number of hidden layers.

Hidden layers number	100	200	500	1000
Classification rate	79.7%	82.8%	86%	81%
False detection rate	5.85%	6.45%	3.7%	5.4%

Table 3: Classification results accurate from DBNs with different number of hidden layers.

Hidden layers number	100-100	200-200	500-500	1000-1000
Classification rate	85.2%	83.7%	84.3%	83.3%
False detection rate	3.1%	4.9%	5.09%	4.94%

ment in classification performance may be possible with additional parameter training. In SVM method, after comparing different settings of parameters, we obtained the best accurate for posture classification is 81.79% with a false detection rate 7.4% when we set C equals to 1 and σ equals to 0.05. From the results above, we can obtain that the deep learning methods works better than SVM method and since when use deep learning methods, the deep learning methods can extract the features of frames in an unsupervised way, it is not necessary for us to extract the features such as ellipse fitting and projection histogram which are described in [4], the proposed system avoids the need for this additional feature extraction.

4. DISCUSSION AND CONCLUSION

In this work, we proposed a novel computer vision-based fall detection system based on posture analysis and employing deep learning methods for analysis and classification of postures. The background subtraction method is proposed to extract the foreground region and the results are improved by postprocessing. After which the features of the four postures in the room environment (standing, sitting, bending and lying) are extracted from a recording of a real home care environment and classified by using both restricted Boltzmann machine and deep belief network. When comparing with the traditional SVM method, we find that our fall detection system achieves better performance with a detection rate of 86% with a low false detection rate 3.7% and the results can be improved with additional parameter training. In order for deep learning methods to be successfully applied, more training data would be necessary to improve the accuracy of the system. Another way in which deep learning could be used as unsupervised feature extraction which may have then application in an online fall detection system.

5. REFERENCES

[1] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, "A review of wearable sensors and systems with application in rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 21, pp. 1–17, 2012.

[2] P. Corbishley and E. Rodriguez-Villegas, "Breathing detection: Towards a miniaturized, wearable, battery-operated monitoring system," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 1, pp. 196–204, 2008.

[3] Y. Zigel, D. Litvak, and I. Gannot, "A method for automatic fall detection of elderly people using floor vibrations and sound-proof of concept on human mimicking doll falls," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 12, pp. 2858–2867, 2009.

[4] M. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Transactions on Information Technologh in Biomedicine*, vol. 16, no. 6, pp. 1274–1286, 2012.

[5] M. Yu, Y. Yu, A. Rhuma, S. M. Naqvi, L. Wang, and J. Chambers, "A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1002–1014, 2013.

[6] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground-background segmentation using code-book model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[7] R. Gonzalez, "Digital image processing," *Third Edition, Pearson Education*, 2008.

[8] H. Lee, "Tutorial on deep learning and applications," *NIPS 2010 workshop on deep learning and unsupervised feature learning, Cheakmus*, 2010.

[9] S. Ilya and T. Tijmen, "On the convergence properties of contrastive divergence," *Proc. 13th Int'l Conf. on AI and Statistics (AISTATS), Cagliari*, 2010.

[10] G.E. Hinton, "Training products of experts by minimizing contrastive," *Neural Computation*, pp. 1771 – 1800, 2002.

[11] J. Huang and B. Kingsbury, "Audio-visual deep learning for noise robust speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7596 – 7599, 2013.

[12] G.E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, pp. 504 – 507, 2006.

[13] R. B. Palm, "Prediction as a candidate for learning deep hierarchical models of data," 2012.