OPEN ACCESS
Vietnam Journal of Computer Science
Vol. 8, No. 2 (2021) 199–214

DOI: 10.1142/S2196888821500081

World Scientific
www.worldscientific.com

Fall Detection in Elderly Care System Based on Group of Pictures

S. Sowmyayani*
Department of Computer Science
St. Mary's College (Autonomous)
Thoothukudi, Tamilnadu, India
sowmyayani@gmail.com

V. Murugan
Department of Computer Science
MSU Constituent College of Arts and Science
Kadayanallur, Tamilnadu, India
smv.murugan@gmail.com

J. Kavitha
Department of Computer Science
Wavoo Wajeeha Women's College of Arts and Science
Kayalpatnam, Tamilnadu, India
vsskavitha@gmail.com

Received 24 April 2020
Accepted 5 September 2020
Published 22 October 2020

Fall detection is a serious problem in elder people. Constant inspection is important for this fall identification. Currently, numerous methods associated with fall detection are a significant area of research for safety purposes and for the healthcare industries. The objective of this paper is to identify elderly falls. The proposed method introduces keyframe based fall detection in elderly care system. Experiments were conducted on University of Rzeszow (UR) Fall Detection dataset, Fall Detection Dataset and MultiCam dataset. It is substantially proved that the proposed method achieves higher accuracy rate of 99%, 98.15% and 99% for UR Fall detection dataset, Fall Detection Dataset and MultiCam dataset, respectively. The performance of the proposed method is compared with other methods and proved to have higher accuracy rate than those methods.

Keywords: Optical flow; group of pictures; foreground segmentation.

MSC 2000: 98U10, 54H30, 68U07.

*Corresponding author.

1. Introduction

Fall Detection (FD) in elders is a major issue in many countries. Smart homes are the solution that can be achieved socially and economically. It explores various problems of fall detection and prevention which lead to the suggestion of different fall detection systems to resolve them. Recently, some research has been done using image sensors to detect falls. The challenge in this system is if the camera is placed sideways, it will fail to include some objects. Therefore, the camera will be placed higher in the room to include all objects and to have a larger field of view.

Fall detection can be attained by wearable sensor-based system, ambient sensor based system and video camera based system. There are more researches in these three ways. All the researches are discussed in this section and the subsequent section. Lee and Mihailidis have developed a fall detection system[1] by analyzing the shape and the 2D velocity of the person. Nait Charif and McKenna have tracked the person using an ellipse, and analyze the resulting trajectory to detect inactivity outside the normal zones of inactivity.[2] Sixsmith and Johnson have classified falls with a neural network using 2D vertical velocity of the person.[3] But, 2D vertical velocity is insufficient to identify a real fall from a person sitting down abruptly. A fall is detected using the vertical and horizontal 3D velocities of the head of the person extracted from a monocular camera video sequence.[4]

Yu provides a survey on various fall detection methods.[5] He also provides some concepts associated with fall detection of elderly patients. Yu first identifies various falls from walking, sitting, standing, sleeping and then categorize various techniques of fall detection as wearable and context-aware system. Additionally, these techniques are classified according to activity recognition, posture analysis, proximity analysis, inactivity identification and 3D head motion analysis.

Perry *et al.* have presented a survey of real-time fall detection methods based on techniques that typically assess only the acceleration.[6] Hijaz *et al.* have presented a fall detection survey and categorize it into approaches based on ambient sensors, vision based sensors and kinematic sensors.[7] Delahoz and Labrador have presented a study of fall detection methods and prevention methods.[8] They categorize fall detection systems based on wearable and external sensors. They also address general aspects of fall detection systems based on machine learning such as features extraction, construction and selection.

Schwickert *et al.* have presented a review of fall detection techniques using wearable sensors.[9] Their survey has mainly focused to determine if the prior studies on fall detection use artificially induced falls in a laboratory environment or natural falls under real-world circumstances. Zhang *et al.* have provided a survey of fall detection methods that use vision sensors.[10] They have also implemented multiple public Fall Detection Datasets (FDDs) and categorize vision-based techniques using single or multiple RGB cameras and 3D depth cameras. In Ref. 11, an efficient fall detection system has been developed using Correlation factor and Motion History Image (CMHI) method for short).

In the proposed method, the video sequences are divided into Group of Pictures (GOP) according to changes in scene. The keyframes are introduced in each GOP. Optical flow is used to find large motion between GOP and the occurrence of fall within a GOP.

The remaining of the paper is organized as follows: Section 2 gives some works related to the research. Section 3 describes the proposed system architecture with the explanation of all the methods incorporated in the proposed method. Section 4 demonstrates with experiments followed by conclusion in Sec. 5.

2. Related Works

In this section, recent methods in fall detection are briefly explained whose results are compared in Sec. 4. A 3D ConvNet architecture is used to identify falls.[12] It consists of 3D Fall Detection Inception modules. It is a modified variant of Inflated 3D (I3D) architecture, which takes Structural Measurement Matrix (SMM) of video sequence as spatio-temporal data, obtained from compressive sensing frameworks, rather than video sequence as data, as in the case of I3D ConvNet.

The method used in Ref. 13 used an approximate ellipse that closely encapsulates the contours of the human body and temporal head position changes to detect human fall; falls were classified by the Support Vector Machine (SVM) algorithm. Reference 14 divides the human body into five partitions which corresponds to five partial occupancy areas. The area ratios are calculated for each frame and used as input data for fall detection and classification.

A method for detecting human fall is presented based on Recurrent Neural Networks.[15] The ability of these networks to process and encode sequential data, such as body-worn sensor acceleration measurements, makes them ideal candidates for this task. In addition, since such networks can greatly benefit from additional data during training, the use of a data increase procedure involving random 3D rotations was investigated. Another method is developed using Convolution Neural Networks (CNN) to determine whether a sequence of frames contains a falling person.[16] Optical flow images are used as input to the networks, followed by a three-step training process to model the video motion and make the machine scenario independent.

An embedded device using accelerometric data and depth maps are used for fall detection.[17] Elderly falls are detected with low computational cost. It uses depth-based inferring about the fall when the movement of the person is above a preset threshold. This method uses the SVM and K-Nearest Neighbor (KNN) classifiers.

Another method is established to accurately identify fall accidents based on differences in vision control in the human silhouette shape.[18] It implemented the transform curvelet and area ratios to classify human postures in images, and reduced the vector dimension of the function using the technique of differential evolution. Then, SVM established postures. It also used the Hidden Markov Model (HMM) to classify video sequences into events with no fall and fall.

A vision based fall detection system is implemented using Hourglass multi-task Convolutional Auto-Encoder (HCAE).[19] In this method, Hourglass Residual Units (HRUs) are inserted into the HCAE encoder to extract multiscale features through the expansion of neuron receptive fields. A multi-task mechanism is provided to improve the network's feature representativeness by completing an auxiliary frame reconstruction task while performing the main task of fall detection. Extended CORE9 was used for the detection of human behavior to obtain a qualitative spatial description of the video activity.[20] The spatial description of an activity obtained using Extended CORE9 along with the temporal information is used for identification of fall.

For indoor environments, a camera-based, real-time automated fall detection system is developed.[21] Background subtraction is applied to frames in the indoor environment for the detection of moving person. Then related geometric characteristics are extracted for classifying a person's fall from other everyday activities. SVM is applied to distinguish between a person's fall and other activities. VGG-16 net and a attention guided LSTM model are adopted for detecting falls in Ref. 22.

A method of fall detection is implemented in Ref. 23 by considering deformation of the human body in images. A person's silhouette is monitored and quantified using the shape analysis method. To detect a person's fall, Gaussian Mixture Model (GMM) is applied. Yun *et al.* detect a person's fall by observing human shape and when a person falls on the ground, there is a dramatic deformation.[24] This approach used view angle by shape analysis. Then, Riemannian manifolds are used for the camera's different views. The human shape is defined by calculating the velocity of the respective manifold points based on the Euclidean distance.

Ma *et al.* proposed a system focused on shape deformation and Extreme Learning Machine (ELM) approach for fall detection.[25] A low-cost camera is used to capture a person's images. Firstly, Curvature Scale Space (CSS) is extracted from the human silhouette in the images. Then, the ELM method is used to predict a person's fall from different activities.

In Ref. 26, fall detection was first achieved using a Multivariate Exponentially Weighted Moving Average (MEWMA) strategy based on the shape feature of five partial occupancy areas extracted from each frame, and then using an SVM classifier to further differentiate falls and fall-like behaviors. A system for detecting fall based on CNN is developed.[16] This method employs VGG-16 net to receive optical flow images as input and to decide if a sequence of frames contains a fall event.

In a video sequence, neighborhood frames have same pixels with very minute changes. From the literature, it is studied that in all the research, all the frames are used. In this research, it is decided to work only on some frames which have major differences. Such frames are called as representative frames or keyframes. By using only keyframes, the computation time will be reduced. Then, background is subtracted to identify only the elder person. Few literatures used optical flow images which identifies falls and non-falls accurately. Hence, in this research, optical flow is

used for identifying small and large movements in the frame. The movements are quantified for measuring the fall and non-fall.

3. System Architecture

In this paper, a new method is proposed to detect falls, which are one of the greatest risk for seniors living alone. The overall system architecture of the proposed method is shown in Fig. 1.

This section elaborates the architecture, algorithm and all the methods involved in the proposed method. This paper introduces keyframe based approach in identifying fall. For achieving the objective, this work uses four phases: Foreground Segmentation, Keyframe Extraction, Fall Identification and Motion Quantification. Initially, the video is partitioned into GOP based on changes in scene. A scene change identification algorithm is introduced based on correlation factor.[27] Then, Optical flow is calculated between keyframes of each GOP. If the optical flow identifies large motion, then there is a fall. Further, in order to identify if there is a

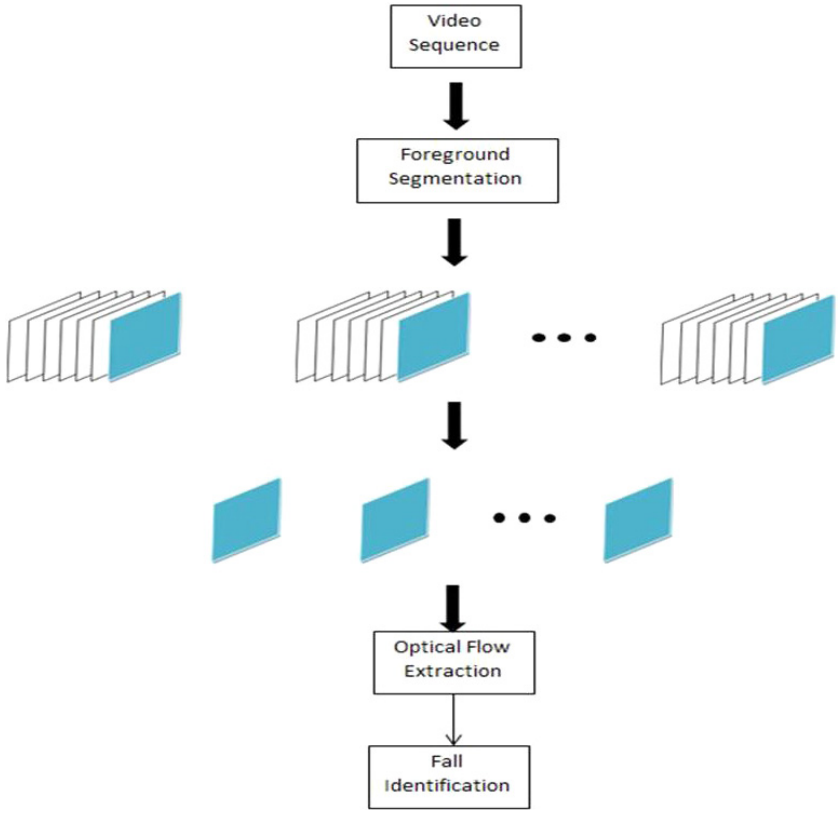

Fig. 1. System architecture.

Algorithm 1. Fall identification algorithm.

***Input*:** Video Sequence V

***Output*:** Fall or No Fall

***Steps*:**

1. Partition video V into frames $f_1, f_2, \ldots, f_n$
2. Segment Foreground from Background
3. Group the frames $G_1, G_2, \ldots, G_m$ where $G_1 = [f_1, f_2, \ldots, f_k]$
4. For each group G_1 to G_m
   - 4.1 Select Keyframes $k_1, k_2, \ldots, k_{m+1}$ where k_1 is the first frame f_1
5. End
6. For $i = 1$ to $m + 1$
   - 6.1 OF = Optical Flow$(k_i, k_{(i+1)})$
   - 6.2 if OF $> \alpha$ //large motion
     - For each frame in next group
       - OF = Optical Flow(f_x, f_y)
       - If OF $< \beta$ //very small or no motion
         - Response = fall
         - Exit
       - End
     - End
   - 6.3 End
7. End
8. Response = No Fall
9. End

movement after a fall, the next group is checked. If there is no movement, then fall is identified.

The complete fall detection process is given Algorithm 1. All the four phases involved in the proposed method are explained in this section.

3.1. *Foreground segmentation*

First, the moving person in the image needs to be extracted. For this purpose, Rougier *et al.*'s background subtraction method[4] is used, which gives good results on image sequences with shadows, highlights and high image compression.

3.2. *Keyframe extraction*

This phase is the key task of this research. After segmenting the human shape using the above-mentioned method, the frames in the video sequence are partitioned into GOP. Generally, video sequence consists of several scenes which can be grouped into individual GOP. In this work, scene can be recognized as human movement. When a

person changes his work, apparently there will be a change in the scene. Hence, the GOP is calculated according to the motion or work of the motion.

In scene change identification algorithm,[27] the video is partitioned into GOP. The GOP size is adaptive to the scene according to that algorithm. i.e. the number of frames in each GOP differs. As the GOP is constructed based on scene change, if there is a fall in the video sequence, substantially it will be in the next GOP. Hence, it is easy to identify the fall. Keyframe, also named as representative frame, is selected in each GOP for further processing.

For the first GOP, the first and the last frame are the keyframes in all the video sequence. For all other GOP, only the last frame is used as keyframe. If the number of GOP is m, then the number of keyframes will be $m + 1$. By using this phase, the number of frames in the video to be processed is narrowed. Thus, the computation time of the proposed method can be reduced.

3.3. *Fall identification*

Motion gives crucial information about fall because no serious fall occurs without a large movement. This subsection identifies the fall by checking the motion in the video sequence. Optical flow is commonly used to detect motion in a video sequence. In this work, optical flow is calculated between keyframes alone. Initially, the optical flow is calculated between the first frame and the first keyframe, as the first frame contains initial human shape. Depending on the value of the optical flow, fall is identified.

There are two scenarios.

- Identifying motion between keyframes (Inter GOP)
- Identifying motion between each frames within a GOP (Intra GOP)

When the optical flow between keyframes is high, then large motion is detected. The movement after this large motion can be observed in the next GOP. Optical flow is calculated between each frame of the next GOP.

If the optical flow between keyframes is less, then medium motion is detected. Now, the optical flow is calculated between keyframes.

The Inter GOP and Intra GOP motion is quantified by two parameters which are described in next subsection.

3.4. *Motion quantification*

To quantify the large, medium and small motion, two parameters α and β are used. To identify the sudden movement (i.e. large motion) between scenes or GOP, α is used. After identifying the sudden movement, the small motion is detected using β. In short, Inter GOP motion is quantified using α and Intra GOP is quantified using β. For this quantification, the magnitude of the optical flow is scaled to 0, no motion, and to 1, full motion.

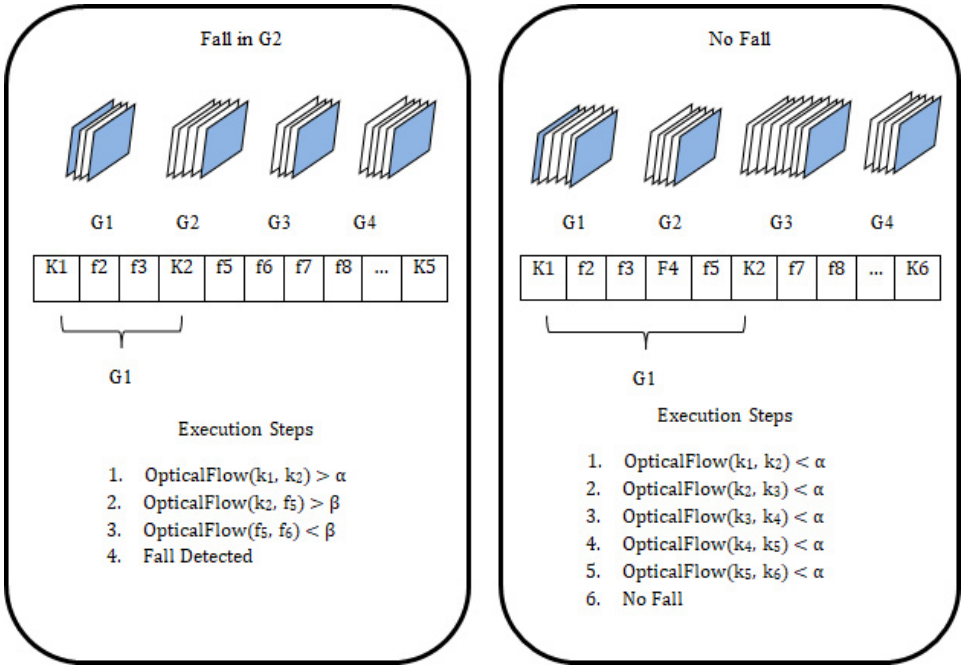

Fig. 2. Execution steps for fall and no fall identification.

The two scenarios, fall and no fall are explained with an example. Consider the two scenarios have 4 GOPs. The execution steps are shown in Fig. 2 and it is explained as follows.

When the optical flow between keyframes is greater than α, then large motion is detected. The movement after this large motion can be observed in the second GOP to identify whether the large motion is fall or no fall. For this, the optical flow is calculated between each frame of the next GOP. If it is lesser than or equal to β, then fall is detected. Otherwise, there is no fall.

If the optical flow between keyframes is lesser than α, then medium motion is detected. Now, the optical flow is calculated between keyframes.

In Fig. 2, the frames shaded with blue are the keyframes. In the first scenario of Fig. 2, the number of frames in the video sequence is 14. Only 5 frames are chosen as keyframes for further processing.

The value of α and β lies between

$$\text{Motion Parameter} = \left\{ \begin{matrix} 0.75 \leq \alpha \leq 1 \\ 0 \leq \beta < 0.75 \end{matrix} \right\}. \tag{1}$$

As large motion is identified using α, the value above 0.75 are chosen. Similarly, as small motion is identified using β, the value below 0.75 is chosen. Having assigned these values to α and β, the fall and no fall are detected like this.

$$f\begin{Bmatrix} \alpha > 0.95, & \text{large motion} \\ \alpha \leq 0.95, & \text{small/medium motion} \end{Bmatrix}, \tag{2}$$

$$f\begin{Bmatrix} \beta > 0.05, & \text{no fall} \\ \beta \leq 0.05, & \text{fall} \end{Bmatrix}. \tag{3}$$

Lack of motion after a fall is identified using β value.

4. Experimental Results

The proposed system is tested in UR Fall Detection (URFD) dataset (http://fenix.univ.rzeszow.pl/∼mkepski/ds/uf.html) the Multiple Cameras Fall Dataset (Multicam) (http://www.iro.umontreal.ca/∼labimage/Dataset/) and the FDD http://le2i.cnrs.fr/Fall-detection-Dataset?lang=fr). The URFD dataset contains 70 (30 falls + 40 activities of daily living) sequences. Fall events are recorded with 2 Microsoft Kinect cameras and corresponding accelerometric data. ADL events are recorded with only one device (camera 0) and accelerometer. Sensor data was collected using PS Move (60 Hz) and x-IMU (256 Hz) devices. The video sequence is captured with VGA resolution (640×480 pixels) on 11 bits (2048 levels of sensitivity) at 30 frames per second. This dataset consists of images with normal activities like walking, sitting down, crouching down and lying which were recorded in typical rooms, like office, classroom, etc. The selected image set consists of 402 images with typical ADLs, whereas 210 images depict a person lying on the floor.

FDD contains videos of four different locations with different actors. These available public datasets are recorded strictly in indoor environments with one person in a video. There are 40 fall videos (10 falls from each group such as a coffee shop, office, home and lecture room).

Multicam contains 24 performances (22 with at least one fall and the remaining two with only confounding events). Each performance has been recorded from eight different perspectives. The same stage is used for all the videos, with some furniture reallocation.

Figure 3 shows the GOP partition of some frames in a sample video sequence of UR FDD. The last row in the GOP has the fall which is correctly identified.

To examine the classification performances, the sensitivity, specificity, precision and classification accuracy are computed. Accuracy is the most important factor for any classification system. Sensitivity is defined as correctly classified video sequences to the sum of video sequences that are correctly and incorrectly classified. Specificity is the number of video sequences that are correctly classified as negative to the sum of video sequences that are correctly classified as negative and the video sequences that are wrongly classified as positive.

The precision or Positive Predictive Value (PPV) is number of video sequences that are correctly classified as positive to the sum of video sequences that are correctly classified as negative and the video sequences that are wrongly classified as

Fig. 3. Sample Images from a UR FDD. Each row represents a GOP.

positive. Thus, it shows how many of the positively classified falls were relevant. The formulae for computing accuracy, sensitivity, precision and specificity rates are presented below.

$$acc_r = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \times 100, \tag{4}$$

$$Sen_r = \frac{T_p}{T_p + F_n} \times 100, \tag{5}$$

$$Sp_r = \frac{T_n}{F_p + T_n} \times 100, \tag{6}$$

$$PPv = \frac{T_p}{F_p + T_p} \times 100, \tag{7}$$

where acc_r is the accuracy rate, Sen_r is the sensitivity rate and Sp_r is the specificity rate and PPv is the Positive Predictive Value. T_p, T_n, F_p, F_n are the True Positive, True Negative, False Positive and False Negative rates. The most important factor in any application is identifying best threshold values. In this research, two Motion Parameters α and β are used whose threshold values are set by several experiments on UR dataset. The optical flow is scaled to 0 to 1 which is divided as $[0, 0.25, 0.50, 0.75, 1]$. Accuracy is calculated at each level to identify the motion quantity. Figure 4 illustrates the study of motion values using optical flow and accuracy.

From Fig. 4, it is studied that, as the value of optical flow increases, the accuracy increases. After 0.75, there is a sudden increase in accuracy. Hence, it is set as threshold for quantifying the motion as slow and large.

Having assigned all the thresholds, the proposed method is tested on all other datasets. The results of the proposed method are compared with recent methods which are explained in Secs. 1 and 2 in terms of the above-defined formula. It is also compared with other methods.[28–30] Table 1 shows the comparison of the proposed method results with the recent methods for UR FDD.

Our recognition system classifies the fall and non-fall very accurately. A good accuracy rate of fall detection is obtained with a sensitivity of 99% and an acceptable rate of false detection with specificity 98.57%. Among the other methods, the SMM + 3D Convolution Inception Network achieves a highest accuracy of 100% and Visual Attention Guided 3DCNN achieves a higher accuracy of 99.27%. But both these methods need more computation time as it needs 3D convolution Neural Network. The proposed method obtain 99% accuracy rate which is lesser than the above-mentioned methods and higher than other methods with computation time of merely 5 s.

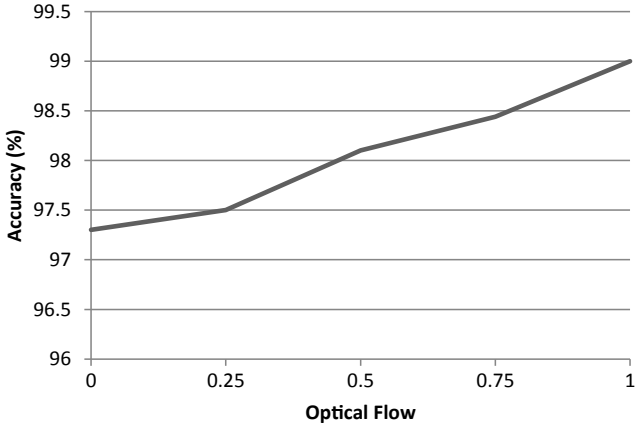

Fig. 4. Accuracy versus optical flow.

Table 1. Results obtained by the proposed method and recent methods for UR FDD.

Performance measure	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)
SMM + 3DConv Inception Network[12]	—	—	—	100
SVM + Depth Features[13]	100	80	83.3	90
SVM + Depth Features + acceleration[13]	100	96.67	96.77	98.33
[a]GLR–SVM[14]	100	0.94	0.94	96.66
LSTM–Acc[15]	96.67	95	95	95.71
[†]LSTM–Acc Rot[15]	96.67	100	100	98.57
OFCNN[16]	100	92		95
Sensor + KNN[17]	100	92.5	90.90	95.71
Sensor + SVM[17]	100	90	88.24	94.28
Curvelets – HMM[18]	—	—	—	96.88
HCAE – FD[19]	100	93	92.3	96.2
Extended core9[20]	93.33	95	—	94.28
Geometric Features – SVM[21]	98.15	97.10	—	—
CNN – LSTM – FD[22]	91.4	—	94.8	—
GMM[23]	95.40	95.80	—	—
Riemannian manifolds – Euclidian[24]	96.67	89.74	—	—
CSS – ELM[25]	99.93	91.67	—	—
MEWMA – FD[26]	100	94.9	93.6	96.7
CNN – FD[16]	100	92	—	95
Visual Attention Guided 3DCNN[28]	—	—	—	99.27
Shi-Tomasi – FD[29]	96.7	—	93.5	95.7
Area – FD[30]	98	89.4	83	94
Riemannian Manifolds[24]	96.77	89.77	—	94
Proposed Method	**99**	**98.57**	**98.57**	**99**

Notes: [a]GLR – SVM → Generalized Likelihood Ratio – SVM.
[b]LSTM-Acc Rot → LSTM – Accerleration Rotation.

After identifying the threshold, the range of β and α are set as (0–0.75) and (0.76–1) respectively. In the range of α, the motion is quantified as medium and large motion by fine-tuning the experiments. The graph is shown in Fig. 5. It is observed that the accuracy is gradually increases as optical flow increases. It reaches maximum

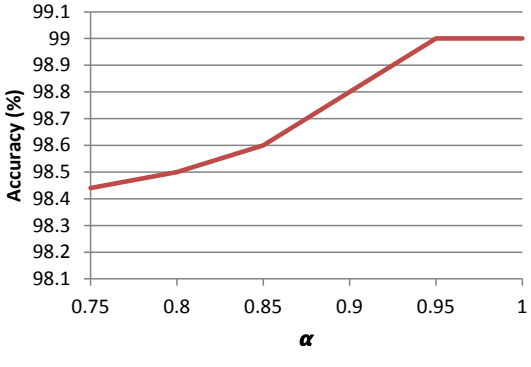

Fig. 5. Accuracy versus α.

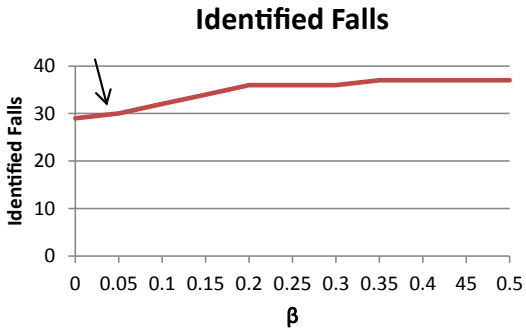

Fig. 6. Identified falls on various β values.

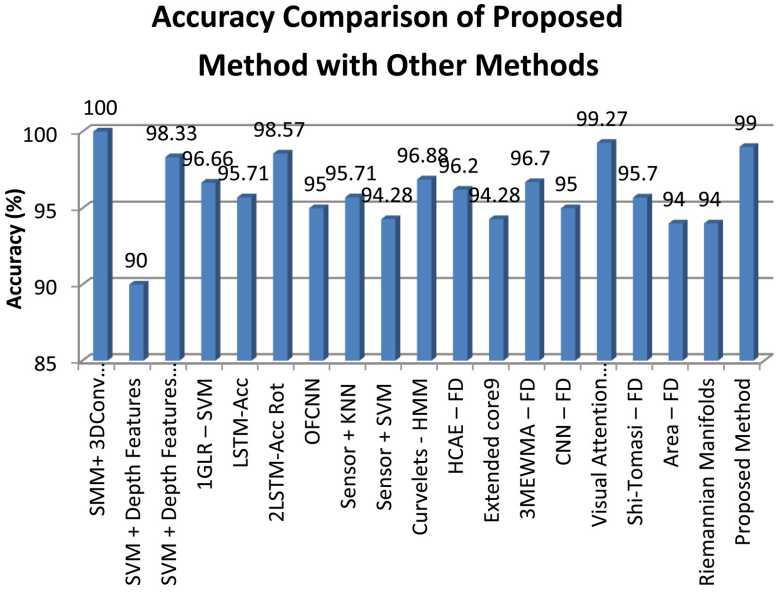

Fig. 7. Bar chart comparison of proposed method with recent methods.

Table 2. Results obtained by the proposed method and recent methods for FDD.

	Measure (%)		
Method	Sensitivity	Specificity	Accuracy
Kavya *et al.*[31]	87.5	—	87.5
CNN – FD[16]	99	97	97
Charfi *et al.*[32]	98	99.6	—
Curvelets – HMM[18]	—	—	97.02
Proposed Method	97.35	96	98.15

Table 3. Results obtained by the proposed method and recent methods for MultiCam dataset.

	Measure (%)	
Method	Sensitivity	Specificity
Wang *et al.*[33]	89.2	90.3
Wang *et al.*[34]	93.7	92
CNN – FD[16]	99	96
Proposed Method	99.21	97

at 0.95 and becomes constant after that. Hence, it is chosen as threshold in the α range. Next, the fall is identified using β (i.e small motion). The value of β is changed from 0 to 0.5 to identify the correct fall which is shown in Fig. 6. In the UR dataset, the number of falls is 30. From the Fig. 6, the number of identified falls is approximately nearer before $\beta = 0.05$. After that, the identified falls is greater than the actual falls. Hence the range 0–0.05 is chosen for identifying fall. Figure 7 shows the bar chart comparison of classification accuracy of all the methods for UR FDD.

The results of the proposed method are compared with recent methods on FDD which is shown in Table 2.

From Table 2, it is clear that the proposed method achieves accuracy of 98.15% which is higher than other methods. The results obtained by the proposed method for MultiCam dataset are also compared with other methods and it is shown in Table 3.

The accuracy of the proposed method is 99%. From Table 3, it is evident that the proposed method is better than other methods. From all the results, the proposed method proves to be better than other recent methods for all the given datasets.

5. Conclusion

In this work, a new method to detect elderly person falls is proposed. The video sequence is divided into GOP based on scene change. One of the frames in each GOP is selected as keyframe. Then, using optical flow, drastic changes are identified between GOP. If there are drastic changes in a GOP, lack of motion is identified again using optical flow in the next GOP. The performance of the proposed method is compared with recent methods and proves to be better than that by getting an accuracy of 99% in 5 s for UR FDD. Also, the accuracy of the proposed method is 98.15% and 99% for FDD and MultiCam datasets, respectively. The two thresholds were chosen manually on what is a large motion and small motion in a video sequence. However, an automatic method could be implemented to define the thresholds using a training dataset.

References

1. T. Lee and A. Mihailidis, An intelligent emergency response system: Preliminary development and testing of automated fall detection, *J. Telemed. Telecare* **11**(4) (2005) 194–198.

2. H. Nait-Charif and S. McKenna, Activity summarization and fall detection in a supportive home environment, in *Proc. of the 17th IEEE Int. Conf. on Pattern Recognition* (UK, 2004), pp. 323–326.
3. A. Sixsmith and N. Johnson, A smart sensor to detect the falls of the elderly, *IEEE Pervas. Comput.* **3**(2) (2004) 42–47.
4. C. Rougier, J. Meunier, A. St-Arnaud and J. Rousseau, Monocular 3d head tracking to detect falls of elderly people, in *Proc. Int. Conf. IEEE Engineering in Medicine and Biology Society* (USA, 2006), pp. 6384–6387.
5. X. Yu, Approaches and principles of fall detection for elderly and patient, in *Proc. 10th Int. Conf. Health Networking Applications and Services* (Singapore, 2008), pp. 42–47.
6. J. Perry, S. Kellog, S. Vaidya, J.-H. Youn, H. Ali and H. Sharif, Survey and evaluation of real-time fall detection approaches, *6th Int. Symp. High-Capacity Optical Networks and Enabling Technologies* (Egypt, 2010), pp. 158–164.
7. F. Hijaz, N. Afzal, T. Ahmad and O. Hasan. Survey of fall detection and daily activity monitoring techniques, in *Proc. Int. Conf. Information and Emerging Technologies* (Pakistan, 2010), pp. 1–6.
8. Y. Delahoz and M. Labrador, Survey on fall detection and fall prevention using wearable and external sensors, *Sensors* **14**(10) (2014) 19806–19842.
9. L. Schwickert, C. Becker, U. Lindemann, C. Marjechal, A. Bourke, L. Chiari, J. Helbostad, W. Zijlstra, K. Aminian, C. Todd, S. Bandinelli and J. Klenk, Fall detection with body-worn sensors: A systematic review, *Z. Gerontol. Geriat.* **46**(8) (2013) 706–719.
10. Z. Zhang, C. Conly and V. Athitsos, A survey on vision-based fall detection, in *Proc. of the 8th Int. Conf. on Pervasive Technologies Related to Assistive Environments* (ACM, 2015), pp. 1–7.
11. S. Sowmyayani and P. A. Jansi Rani, An efficient fall detection method for elderly care system, *World Acad. Sci. Eng. Technol. Int. J. Computer Inf. Eng.* **13**(3) (2019) 173–177.
12. R. Gupta, P. Anand, S. Chaudhury, B. Lall and S. Singh, Compressive sensing based privacy for fall detection, arXiv preprint arXiv:2001.03463 (2020).
13. B. Kwolek and M. Kepski, Human fall detection on embedded platform using depth maps and wireless accelerometer, *Computer Methods Prog. Biomed.* **117**(3) (2014) 489–501.
14. F. Harrou, N. Zerrouki, Y. Sun and A. Houacine, An integrated vision-based approach for efficient human fall detection in a home environment, *IEEE Access* **7** (2019) 114966–114974.
15. T. Theodoridis, V. Solachidis, N. Vretos and P. Daras, Human fall detection from acceleration measurements using a recurrent neural network. in *International Conference on Biomedical and Health Informatics* (Springer, 2017), pp. 145–149.
16. A. Núñez-Marcos, G. Azkune and I. Arganda-Carreras, Vision-based fall detection with convolutional neural networks, *Wirel. Commun. Mobile Comput.* **2017** (2017) 1–16.
17. M. Kepski and B. Kwolek, Embedded system for fall detection using body-worn accelerometer and depth sensor, in *Proc. of the IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Vol. 2 (IEEE, 2015), pp. 755–759.
18. N. Zerrouki and A. Houacine, Combined curvelets and hidden Markov models for human fall detection, *Multim. Tools Appl.* **77**(5) (2018) 6405–6424.
19. X. Cai, S. Li, X. Liu and G. Han, Vision-based fall detection with multi-task hourglass convolutional auto-encoder, *IEEE Access* **8** (2020) 44493–44502.
20. S. Kalita, A. Karmakar and S. M. Hazarika, February. human fall detection during activities of daily living using extended CORE9, in *2019 Second Int. Conf. Advanced Computational and Communication Paradigms (ICACCP)* (IEEE, 2019), pp. 1–6.

21. P. K. Soni and A. Choudhary, Automated fall detection from a camera using support vector machine, *2019 Second Int. Conf. Advanced Computational and Communication Paradigms (ICACCP)* (IEEE, 2019), pp. 1–6.
22. Q. Feng, C. Gao, L. Wang, Y. Zhao, T. Song and Q. Li, Spatiotemporal fall event detection in complex scenes using attention guided LSTM, *Pattern Recognit. Lett.* **130** (2020) 242–249.
23. R. Caroline *et al.*, Robust video surveillance for fall detection based on human shape deformation, *IEEE Trans. Circ. Syst. Video Technol.* **21**(5) (2011) 611–622.
24. Y. Yixiao and I. Y.-H. Gu, Human fall detection via shape analysis on Riemannian manifolds with applications to elderly care, *IEEE Int. Conf. Image Processing (ICIP)* (IEEE, 2015), pp. 3280–3284.
25. M. Xin *et al.*, Depth-based human fall detection via shape features and improved extreme learning machine, *IEEE J. Biomed. Health Informatics* **18**(6) (2014) 1915–1922.
26. F. Harrou, N. Zerrouki, Y. Sun and A. Houacine, Vision-based fall detection system for improving safety of elderly people, *IEEE Instrum. Meas. Mag.* **20**(6) (2017) 49–55.
27. S. Sowmyayani and P. Arockia Jansi Rani, Adaptive GOP structure to H.264/AVC based on Scene change, *ICTACT J. Image Video Process* **5**(1) (2014) 868–872.
28. N. Lu, Y. Wu, L. Feng and J. Song, Deep learning for fall detection: Three- dimensional CNN combined with LSTM on video kinematic data, *IEEE J. Biomed. Health Informatics* **23**(1) (2019) 314–323.
29. S. Bhandari, N. Babar, P. Gupta, N. Shah and S. Pujari, A novel approach for fall detection in home environment, in *Proc. IEEE 6th Global Conf. Consum. Electron. (GCCE)*, Nagoya, Japan, October 2017, pp. 1–5.
30. N. Zerrouki, F. Harrou, A. Houacine and Y. Sun, Fall detection using supervised machine learning algorithms: A comparative study, in *Proc. 8th Int. Conf. Model., Identification. Control (ICMIC)*, Algiers, Algeria, November 2016, pp. 665–670.
31. T. S. Kavya, Y. M. Jang, E. Tsogtbaatar and S. B. Cho, Fall detection system for elderly people using vision-based analysis, *Sci. Technol.* **23**(1) (2020) 69–83.
32. I. Charfi, J. Miteran, J. Dubois, M. Atri and R. Tourki, Definition and performance evaluation of a robust SVM based fall detection solution, in *Proc. 8th International Conference on Signal Image Technology and Internet Based Systems, SITIS 2012*, Italy, November 2012, pp. 218–224.
33. S. Wang, L. Chen, Z. Zhou, X. Sun and J. Dong, Human fall detection in surveillance video based on PCANet, *Multim. Tools Appl.* **75**(19) (2015) 11603–11613.
34. K. Wang, G. Cao, D. Meng, W. Chen and W. Cao, Automatic fall detection of human in video using combination of features, in *Proc. 2016 IEEE Int. Conf. Bioinformatics and Biomedicine, BIBM 2016*, China, December 2016, pp. 1228–1233.