# PRIVACY PRESERVING ELDERLY FALL DETECTION USING KINECT DEPTH IMAGES BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS

**Sujatha Jamuna Anand[1,*], G. Arulselvi[2], D. Poornima[3]**

[1]*Loyola Institute of Technology, Chennai,*
[2]*Department of CSE, Annamalai University, Chidambaram.*
[3]*Department of CSE, Annamalai University, Chidambaram.*

## ABSTRACT

The increase in mortality rate of the elderly in the recent years is mainly due to the occurrence of falls. Ensuring safely of the elderly is a crucial task since they cannot be monitored constantly all the time. In this research, we propose a novel scheme for the fall detection of the elderly using depth videos acquired from two Kinect sensors. The two sensors include the frontal-mounted Kinect sensor and the depth-mounted Kinect sensor. Fall detection using Kinect depth maps offers a low-cost, reliable and privacy preserving solution. Using the depth maps, depth motion maps (DMM) are generated. From these maps, depth ConvNet features are extracted using deep convolutional neural network (DCNN) structure. In this research, two convolutional layers and two max-pool sub-sampling layers are employed. The obtained features have extremely high discriminative strength to distinguish between fall and non-fall actions. From the extracted features, fall detection is done using Extreme Learning Machine (ELM) classifier. The temporal frame length of the depth motion maps is varied, and the corresponding fall detection accuracy is obtained for the identification of optimal temporal length. Since, this framework utilizes only Kinect depth maps, the privacy issue involved in fall detection during the usage of RGB videos are eradicated. Evaluation was done using publicly available University of Rzeszow Fall Detection (URFD) dataset. Performance evaluation was done based on confusion matrix obtained. Metrics like specificity, precision, sensitivity, and F-score were evaluated from the obtained confusion matrix. Evaluation results clearly show the efficacy of the proposed fall detection framework by comparing with the other state-of-the-art fall detection works proposed in the literature.

**Keywords:** DCNN, URFD, ELM, Kinect and fall detection.

## 1. Introduction

The advancement in the field of Information and Communication Technology has led to the development of automatic systems for the safety of the elderly living[1]–[4]. Since, elderly people are extremely fragile they are prone to fall events[5]–[9]. It has been shown through research that one in every three elderly people (who live in US) are subject to fall in a year. This causes serious threat, since the fall event may eventually lead to serious injury or even to death if steps are not taken immediately. Hence, automatic fall detection is a recently trending research area. Recently Internet of Things (IoT) are widely being used for indicating the fall events to the caretakers [10]–[12]. The schemes for fall detection can be broadly categorized into two categories namely, the wearable sensor based[13][14][15][16][17] and computer vision[18][19][20][21][22] based schemes. The wearable sensor based schemes uses inertial sensors like accelerometers, gyroscopes, magnetometers, etc., that are worn by the elderly. However, the computer vision based schemes employ colour or infrared or thermal cameras to detect the fall events. Deep learning for the fall detection has become very popular in the recent days [12], [23]–[25][26][27]. Smart phones are also being used for the automatic detection of fall events [28]–[33].

Accelerometer and gyroscope sensors were used for the identification of fall from the activities of daily living (ADL) in [13].The data from the accelerometer sensor was used for identifying four types of static postures. Based on the linear acceleration and the angular velocity, the intentional transitions and fall events were identified. An algorithm for fall detection using a pre-defined threshold was proposed in [14]. The value of the threshold was determined using the acceleration value from the accelerometer sensor. The sensor was mounted on two regions namely, the thigh and the trunk of the individual. In this framework, eight different types of fall events were evaluated. Fall detection framework using MEMS accelerometer was proposed in [15]. In this work, the wearable device was worn around the waist. Here, falls were detected based on the orientation of the accelerometer signal.

Smart phone accelerometers were employed for fall detection in [16]. Here, the data was pre-processed using support vector machine (SVM) algorithm. The classification of fall was performed using a combination of kernel fisher discriminate (KFD) and k-nearest neighbour (k-NN) algorithm. A scheme for fall detection using accelerometer sensor was proposed in [17]. Here, feature extraction was performed for the signal window whenever a peak was identified. Different types of classifiers like neural network, SVM, decision tree, etc, were employed for classification. Though wearable sensors produce good fall detection results, they cause discomfort to the users, as they must be worn all the time.

Fall detection using RGB cameras based on tracking of head trajectories was proposed in [18]. The head tracking was performed using particle filtering technique. Based on the values of the 3D velocities, fall events were detected. A framework for fall detection using an omni camera was proposed in [19]. In this work, the personalized information about individuals like height, weight, etc., were loaded into the system to increase the sensitivity of fall detection. Fall detection was done based on the dimension of the silhouettes in the normal and the fall states. Fall detection using depth camera was proposed in [20]. Here, three types of features were used for fall identification that included area of the individual, shape of the individual and the distance between the head to the floor. Classification was done using a k-NN classifier.

Camera-based system for the identification of fall was proposed in [21]. In this work, the main steps involved were background subtraction followed by application of Kalman filter. The optical flow features were then extracted from the filtered frames. Finally, classification was done using k-NN classifier. Detection of elderly fall using Kinect images was proposed in [22]. Here, the falls were detected based on the tracking of body parts. Here, tracking was done using key joints. These key spots were identified based on a novel randomized decision tree algorithm.

Though RGB cameras offer a good solution for the fall detection, they have privacy issue. To avoid this, Kinect depth images are popularly being used for fall detection recently. Hence, in our research we propose a novel algorithm for fall detection using depth data.

The overall contributions of this paper are fourfold:

a) A novel scheme for fall detection using data from frontal and top view Kinect sensors.
b) A novel deep learning architecture for the generation of ConvNet features.
c) Performance evaluation and comparison with state-of-the-art techniques.

d) Comparison with other features and classification techniques.

## 2. Literature survey

A scheme for fall detection by using a combination of accelerometer and depth camera was proposed by Kwolek et al. (2014) in [34]. In this work,a new scheme for fall detection that has a low false alarm value was proposed. The instants of potential fall were indicated using the accelerometer signal. The accelerometer signal was constantly checked against a pre-defined threshold value. When the signal goes beyond the threshold the instance was captured. Using this instance, the depth maps associated to that instant were extracted. From the depth maps, the person delineation was done. Using the extracted regions, features were extracted. Finally, the classification was done using SVM classifier.

Kwolek et al. (2015) [35] proposed a scheme for fall detection using depth and inertial data. In this scheme, the potential fall was identified using accelerometer signal. Using the instance identified by the inertial sensor, the depth maps were downloaded from a circular buffer. From the extracted depth maps, features and point cloud data were extracted. Four features were extracted from the depth maps. The first feature was the ratio of the height to the width of the individual. The second feature was the height of the bounding box to the actual height of the individual. The third feature was the distance between the centroid to the floor. The last feature was the standard deviation of the centroid. Classification was done using k-NN classification algorithm. The outcome was compared with the SVM classifier. It was observed that the performance of the k-NN classifier was better than that of the SVM classifier.

Kong et al. [36] presented a technique for fall detection using Kinect data. In this work, the depth maps are obtained, and noise reduction was performed using noise reduction techniques. From the denoised depth maps, the histogram of oriented gradient features was extracted. Using the extracted features classification was done using SVM classifier. Using internet of things (IOT), the system was designed to send alert messages to the caretakers or to the hospital whenever a fall event was detected. In this work, evaluation was done using a new dataset that comprised of 3500 images. Here, fall alert messages were sent whenever the fall event was detected to prolong for 60s. If the duration was less than 60s, it was detected that the elderly person recovered from the fall, so the alarm was not sent to the caretaker.

Xu et al. [37] introduced a scheme for fall detection using Kinect V2 camera. Here, the skeletal tracker was employed for the identification of joints location. Using these locations, posture was recognized using neural network structure. From the identified postures, fall events were identified. If the identified posture was a lying state, then the previous posture was identified. The total duration of the lying posture was also determined. Using these values, fall events were detected. The classification was done using neural network architecture. This system attained a fall detection accuracy of about 97.3%. The main advantage achieved by this system was the privacy protection since it employed depth maps.

Li et al. [38] proposed a scheme in which the fall events were determined using convolutional neural networks. The classification of fall was done based on the analysis of spatio-temporal patterns. Here, three layers of fine tuning were done to increase the reliability of fall detection. Using a sliding window, the video snippets were detected and fed as input to the convolutional neural networks for the identification of fall events. Final detection was done using the SoftMax layer. This system employed the RGB images captured by cameras for fall analysis. Trials were conducted using five trainees. This system could identify fall within 0.5 seconds of the fall event.

Panahi et al. [39] presented a scheme for fall detection using Kinect images. In this framework, initially, the foreground images were obtained using background subtraction technique. Then, the depth maps were converted to black and white images. The morphological operations were then applied on the binarized images. Then, contour extraction was performed. Using the contour regions, feature extraction was done. From the extracted features, the system checks if the current action was lying on the floor. If the output was yes, then the system identifies the duration of the lying action. If the lying duration was high, then fall events were confirmed. This system achieved specificity of 97.5%.

A scheme for fall detection using deep learning was proposed in [40]. This scheme utilized neural network structure based on long short-term memory (LSTM). The proposed structure employed two LSTM structures. The first structure comprised of multi-class LSTM and the second structure involved a two-class LSTM. The general features extracted by the multi-class LSTM were transferred to the two-class LSTM using transfer learning wherein the weights were kept constant. The output of the two-class LSTM indicated the occurrence of fall event. This system attained a recall of 96.12%. The overall ROC achieved for the fall detection was 0.99. The main advantage of this scheme was that no hand-crafted features were generated instead the deep learning structure was used to generate the features for the classification.

Mastorakis et al. [41] proposed a scheme in which fall events were identified using infrared sensor of Kinect device. Fall events were detected based on the velocity computations and the inactivity details. The 3D bounding boxes were generated and the velocity was identified based on the width and the height of the box. The identified velocity was then used for the inactivity detection. The duration of inactivity is computed. If the length of this duration was greater than 2s, then the fall event was confirmed. Different types of fall like backward, forward, and sideways fall were analyzed in this system. The computational time of this scheme was very less around 0.3 to 0.4 ms.

## 3. Proposed Methodology

### 3.1. Proposed fall detection architecture

The depth video comprising of depth frames are acquired from the Kinect sensor. To increase the accuracy of classification, we have employed the depth maps from two Kinect sensors in this work. The first Kinect sensor captures the depth images from the front view and the second Kinect sensor captures the depth images from the top view. Using these depth images, the depth motion map (DMM) is generated. From the DMM, ConvNet features are extracted. These features are then classified using extreme learning machine (ELM) classifier. The output of classification is either fall or non-fall. The flowchart of the proposed framework is given in Figure 1.


**Figure 1. Flow chart of proposed fall detection framework**

### 3.2. Depth motion map(DMM) generation

Depth motion maps are computed as a sum of the depth information in a depth sequence. These maps represent the history of depth information for a particular window. The higher the window, the more is the information. In our work, every $M$ depth frames are represented in the form of a depth motion map $DMM^{M}$.

For a sequence of $M$ depth frames, say, $D_1, D_2, \ldots, D_M$, the depth motion maps are generated using the following equation,

$$DMM^{M} = \sum_{j=1}^{M-1} | D^{j+1} - D^{j} | \qquad (1)$$

Here, $DMM^{5}$ indicates that the depth motion map comprises of information from its previous 5 consecutive depth images. Similarly, $DMM^{9}$ indicates that the depth motion map comprises of information from its previous 9 consecutive depth images. Figure 2 shows the depth motion map of walking action.

Figure 2. DMM of a walking action

Figure 3 shows the depth motion map of fall action. From Figure 2 and Figure 3 we clearly observe that, the depth motion map of fall action has more distortion than that of depth motion map of walk action. These images were generated for $M$ =64.

The data in the depth motion map varies based on the value of $M$ . Hence selection of appropriate values of $M$ is essential for accurate fall detection. In the forthcoming sections we have identified the optimum value of $M$ for this research.

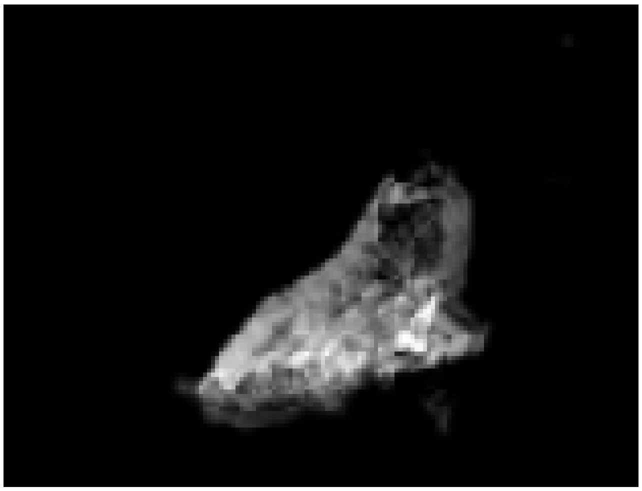

Figure 3. DMM of a fall action

### 3.3. Feature extraction using ConvNet

Feature extraction is done using the generated depth motion maps from both the Kinect sensors. For every $M$ consecutive depth images, DMM are generated. These DMM are employed for the extraction of ConvNet features. In this work,we have employed a four-layer deep learning network architecture for feature extraction. This is shown in Figure 4.The proposed architecture comprises of a convolutional layer with a kernel size of 5×5 followed by a max-pooling sub-sampling layer with a kernel size of 3×3. These two layers are followed by a convolutional layer with a kernel size of 5×5. Finally, there is another max-pooling sub-sampling layer with a kernel size of 3×3. The output of the final max-pooling sub-sampling layer generates the ConvNet features employed for fall detection. The depth motion maps from both the Kinect sensors are used for the generation of ConvNet features.

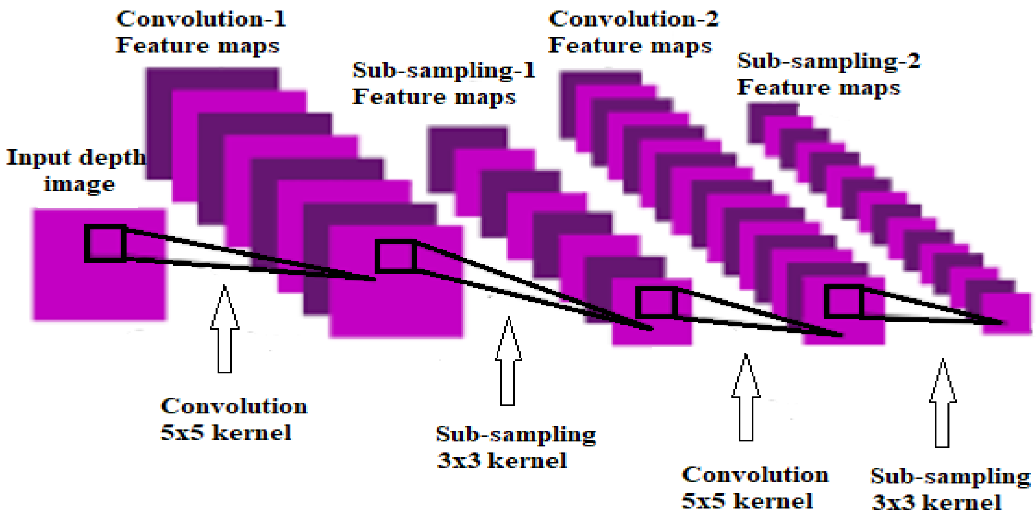


**Figure 4. ConvNet feature extraction**

### 3.4. Extreme learning machine (ELM) classifier

The generated ConvNet features are employed for classification using ELM classifier.The ELM classifier is a single hidden layer feed forward neural network. The ELM classifier is modelled using parameters such as threshold, weight and activation function. These parameters are iteratively optimized for maximum classification performance of the ELM classifier. In the learning process, the weights for the inputs are selected randomly whereas the weights for the outputs are calculated using analytical processing. For activating the hidden layer cells, non-linear functions like sigmoid and sinusoidal functions are employed.

## 4. Results and Discussion

### 4.1. Dataset description

The proposed system was evaluated using URFD dataset [34]. This dataset comprises of both inertial sensor data (accelerometer) and Kinect depth data. The depth data was acquired using Kinect Xbox 360. The sampling rate for the acquisition of Kinect data was 30 fps. Each depth map has a dimension of 480×640. This dataset included data from two different Kinect devices. The first Kinect device was installed in the front of the scene at a distance of 1m from the ground level. This provided the front view data. The second Kinect was installed at the top of the ceiling. This provided the top view data. This dataset has a total of 70 video sequences out of which 30 represents fall events and the rest 40 represents ADL events. The specifications of the URFD dataset is given in Table 1. The fall-based actions included fall actions from standing position and fall action from sitting position. The ADL actions included actions like sitting, object picking, crouching, quick lying, etc.

**Table 1. Specification of URFD dataset**

| Class | Action | Number of videos | Total |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Fall | Human fall | 30 | 30 |
| ADL | Crouch | 8 | 40 |
| | Bend | 7 | |
| | Sit | 9 | |
| | Lye | 19 | |

Figure 5 shows the sample images from the URFD dataset. In Figure 5,the top image shows the RGB image obtained from the frontal Kinect sensor during walking action. The bottom image shows the corresponding depth image obtained from the frontal Kinect sensor during walking action.

**Figure 5. RGB image and Depth map of walking action**

Figure 6 shows the sample images from the URFD dataset for fall action. In Figure 6, the top image shows the RGB image obtained from the frontal Kinect sensor during fall action. The bottom image shows the corresponding depth image obtained from the frontal Kinect sensor during fall action.

**Figure 6. RGB image and Depth map of fall action**

Figure 7 and Figure 8 shows the sample images from the URFD dataset obtained by the two Kinect sensors. Figure 7 shows the RGB and depth image pair captured by the frontal Kinect device.

**Figure 7. RGB and depth map images from front view Kinect sensor**

Figure 8 shows the RGB and depth image pair captured by the ceiling-mounted Kinect device. From Figure 8 we infer that, it provides complimentary information to that provided by the top-view Kinect sensor. Since data is captured from two different view, employing both of them increases the accuracy of classification.

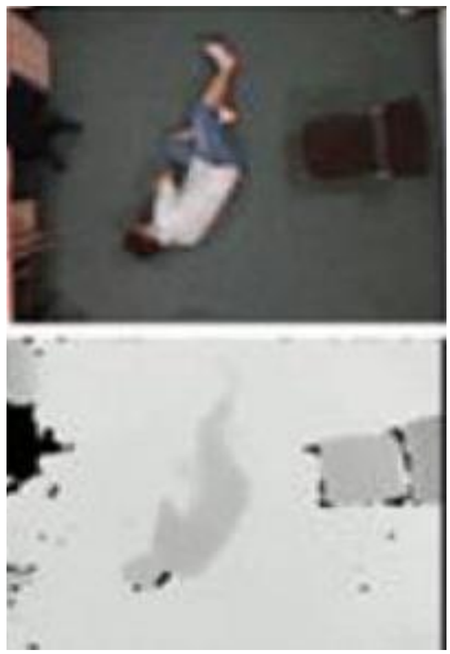

**Figure 8. RGB and depth map images from top view Kinect sensor**

### 4.2. Selection of frame length

The selection of frame length $M$ for the generation of depth motion map is very crucial. The frame length $M$ should have an optimum value. Very low value of frame length may not be capable of capturing the actions and very high length may have redundant information. Hence, its value must be optimized. To identify the optimum value of $M$, we have varied the value of $M$ and computed the accuracy of classification. For various values of $M$, the value of accuracy was computed and plotted in the form of a graph. This graph is shown in Figure 9. From Figure 9 we see that, the value of $M$ was varied from 5 to 70, with increments of 5. It was observed that maximum accuracy was achieved at $M$ =35. Hence, in our work we have used 35 consecutive frames for the generation of depth motion maps. The highest accuracy achieved was 97.14%.

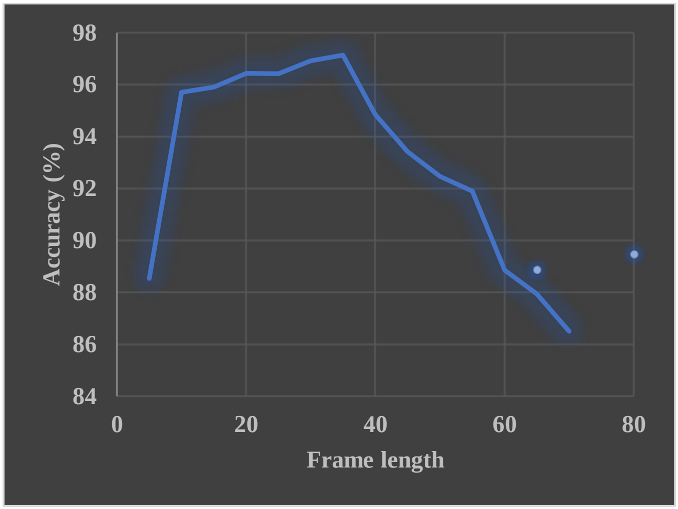


**Figure 9. Variation of accuracy with frame length**

## 4.3. Simulation results

We have used 10-fold cross validation technique. The dataset was divided into 10 parts wherein one part was employed for testing and the rest were used for training. This entire process was iterated 10 times. The final accuracy was the average of all the iterations. The classification performance was evaluated using five metrics namely accuracy, recall, specificity, precision and F-score.

Table 2 shows the results of the proposed deep learning based fall detection framework on the URFD dataset. We evaluated using three different cases. In the first case, only the frontal depth data was employed. In the second case, the top-view depth data was used. In the third case, both the frontal and the top-view data were employed.

When only the frontal Kinect sensor (Kinect-F) was employed, out of 30 fall events, 29 were detected correctly. However, one fall event was misclassified. Out of 40 ADL data, 36 were classified correctly and 4 were mis-classified. When the top-view Kinect sensor (Kinect-T) was used, 28 fall events were classified correctly, and 2 fall events were misclassified as ADL. Out of 40 ADL events, 35 were correctly classified as ADL and remaining 5 were misclassified as fall. However, when we employed data from both the sensors for classification, good classification results were obtained. We find from the Table 2 that, out of 30 fall events, all the 30 were correctly classified as fall. This makes the sensitivity of the classifier to be 100%. Among 40 ADL events, 38 were correctly classified and only 2 were misclassified as fall events. Thus, the proposed system achieved best results when data from both the sensors were employed.

**Table 2. Results of the proposed fall detection system**

| Sensor used | Fall/ADL | Total | Detected | Missed |
|---|---|---|---|---|
| Kinect-F | Fall | 30 | 29 | 1 |
| | ADL | 40 | 36 | 4 |

| | | | | |
|---|---|---|---|---|
| Kinect-T | Fall | 30 | 28 | 2 |
| | ADL | 40 | 35 | 5 |
| Kinect-F + Kinect-T | Fall | 30 | 30 | 0 |
| | ADL | 40 | 38 | 2 |

Using the obtained results, various metrics were evaluated.

**True Positive ($TP$):**

It indicates the number of fall events present and detected.

**True Negative ($TN$):**

It indicates the number of fall events not present and not detected.

**False Positive ($FP$):**

It indicates the number of fall events not present but detected as fall.

**False Negative ($FN$):**

It indicates the number of fall events present but not detected as fall.

From the above computed parameters, classification metrics like overall accuracy, recall, precision, specificity and F-score were calculated.

**Overall accuracy ($O_a$):**

Overall accuracy is computed as

$$O_a = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

**Recall ($R_e$):**

The recall refers to the sensitivity and is determined using

$$R_e = \frac{TP}{TP + FN} \tag{3}$$

**Precision ($P_r$):**

The precision is evaluated as

$$P_r = \frac{TP}{TP + FP} \tag{4}$$

**Specificity ( $S_p$ ):**

The specificity is given by

$$S_p = \frac{TN}{TN + FP} \tag{5}$$

F**-score ( $F_s$ ):**

The F-score is computed as

$$F_s = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

Table 3 shows the performance metrics of the proposed fall detection system. From Table 3 we find that the value of accuracy for Kinect-F is 92.86%. For the Kinect-F the accuracy is 90%. However, for a combination of Kinect-F and Kinect-T, the system produces an accuracy of 97.14%. The specificity for Kinect-F is 89.99%. For Kinect-T the specificity is 97.49%. However, for Kinect-F with Kinect-T, we obtained a high specificity of 94.99%. The precision obtained for Kinect-F alone is 87.87%. For Kinect-T, the precision value obtained is 84.84%. The combination of Kinect-F with Kinect-T, the framework achieved high precision of 93.74%. The recall value achieved for Kinect-F is 96.66%. For Kinect-T, the recall achieved is 93.33%. In this research, for combination of Kinect-F with Kinect-T, recall of 100% was attained. The F-score value achieved for Kinect-F is 92.05%. For Kinect-T, the F-score achieved is 88.88%. In this research, for combination of Kinect-F with Kinect-T, F-score of 96.76% is attained.

**Table 3. Performance metrics of the proposed fall detection system**

| **Performance metrics (%)** | **Kinect-F** | **Kinect-T** | **Kinect-F + Kinect-T** |
|---|---|---|---|
| Accuracy | 92.86 | 90.00 | 97.14 |
| Specificity | 89.99 | 87.49 | 94.99 |
| Precision | 87.87 | 84.84 | 93.74 |
| Recall | 96.66 | 93.33 | 100.00 |
| F-score | 92.05 | 88.88 | 96.76 |

Figure 10 shows the performance comparison of the proposed system when RGB images, Kinect-F depth images and combination of Kinect-F and Kinect-T depth images were used. The graph clearly shows that, when RGB images were employed the performance of classification was poor. However, when the Kinect-F depth images were used, the performance of fall detection improved. When both the depth maps from Kinect-F and Kinect-T sensors were employed, very high fall detection performance was achieved in this system.

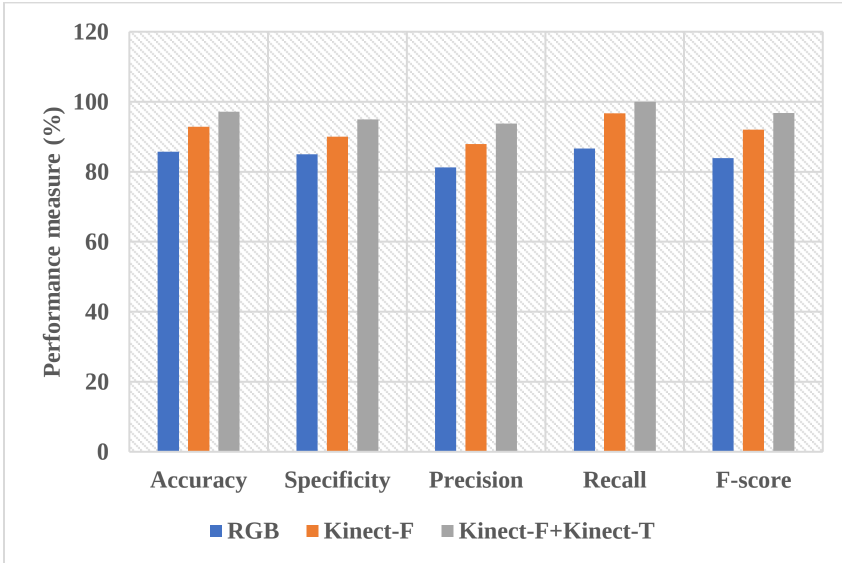


**Figure 10. Variation of performance measures**

Table 4 shows the comparison of performance when different combinations of features and classifiers were employed. When histogram of oriented gradient features was employed along with SVM classifier (HOG+SVM), the system achieved an accuracy of 78.57%. When Convnet features were classified using SVM (ConvNet + SVM), an accuracy of 84.29% was achieved. The histogram of oriented gradient features with extreme learning machine classifier (HOG+ELM), an accuracy of 88.57% was attained. The proposed system with ConvNet features with extreme learning machine (ELM) classifier (ConvNet+ELM), a very high accuracy of 97.14% was attained. This shows the efficacy of our system.

HOG+SVM achieved a specificity of 79.99%. The specificity of the ConvNet + SVM framework was 84.99%. The specificity of the HOG+ELM system was 87.49%. However, the proposed system with ConvNet+ELM attained a high specificity of 94.99%.

Similarly, HOG+SVM achieved a precision of 74.19%. The precision of the ConvNet + SVM framework was 80.64%. The precision of the HOG+ELM system was 84.37%. However, the proposed system with ConvNet+ELM attained a high precision of 93.74%.

Similarly, HOG+SVM achieved a recall of 76.66%. The recall of the ConvNet + SVM framework was 83.33%. The recall of the HOG+ELM system was 89.99%. However, the proposed system with ConvNet+ELM attained a high recall of 100%.

Similarly, HOG+SVM achieved a F-score of 75.40%. The F-score of the ConvNet + SVM framework was 81.96%. The F-score of the HOG+ELM system was 97.09%. However, the proposed system with ConvNet+ELM attained a high F-score of 96.76%.

**Table 4. Performance measure comparison of various fall detection schemes**

| Performance measure (%) | HOG+SVM | ConvNet+SVM | HOG+ELM | ConvNet+ELM |
|---|---|---|---|---|
| Accuracy | 78.57 | 84.29 | 88.57 | 97.14 |
| Specificity | 79.99 | 84.99 | 87.49 | 94.99 |
| Precision | 74.19 | 80.64 | 84.37 | 93.74 |
| Recall | 76.66 | 83.33 | 89.99 | 100.00 |
| F-score | 75.40 | 81.96 | 97.09 | 96.76 |

Table 5 shows the comparison of the proposed system with the state-of-the-art techniques proposed in the literature for fall detection. Bourke et al. [14] proposed a scheme for fall detection using wearable sensors. Using the signal from the accelerometer sensor, fall detection was performed. This system achieved an accuracy of 95%. The specificity value attained was 90%. The value of precision of this framework was 90.91%. The value of recall reached was 100%. The F-score of this fall detection system was 92.13%.

Kwolek et al. [34] proposed a scheme for fall detection using Kinect sensor. Using the signal from the Kinect sensor, fall detection was performed. This system achieved an accuracy of 90%. The specificity value attained was 80%. The value of precision of this framework was 83.3%. The value of recall reached was 100%. The F-score of this fall detection system was 94.34%.

Kwolek et al. [35] proposed a scheme for fall detection using both Kinect sensor and accelerometer. Using the signal from the accelerometer fall indication was given. The fall events were confirmed using Kinect sensor. This system achieved an accuracy of 94.28%. The specificity value attained was 89.99%. The value of precision of this framework was 88.23%. The value of recall reached was 100%. The F-score of this fall detection system was 93.74%.

The proposed system achieved an accuracy of 97.14%. The specificity value attained was 94.99%. The value of precision of this framework was 93.74%. The value of recall reached was 100%. The F-score of this fall detection system was 96.76%.

**Table 5. Comparison of fall detection performance with state-of-the-art techniques**

| Reference | Data used | Accuracy (%) | Specificity (%) | Precision (%) | Recall (%) | F-score (%) |
|---|---|---|---|---|---|---|
| Bourke et al. [14] | Acc. | 95.00 | 90.00 | 90.91 | 100.00 | 92.13 |
| Kwolek et al. [34] | Depth | 90.00 | 80.00 | 83.30 | 100.00 | 94.34 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Kwolek et al. [35] | Depth+Acc. | 94.28 | 89.99 | 88.23 | 100.00 | 93.74 |
| Proposed | Depth | 97.14 | 94.99 | 93.74 | 100.00 | 96.76 |

### 4.4. Computational time comparison

Table 6 shows the comparison of time complexity. The proposed system was evaluated using RGB images, depth images from frontal sensor alone and using a combination of frontal and top-view sensor. We inferred that, the average time for classifying one fall event was 1450 ms when RGB images were used. For Kinect-F sensor, the time was only 120 ms. when both the Kinect F and Kinect-T were used the time was 155 ms. From this analysis we find that the depth images consume less time than the RGB images. Though the time for using a combination of frontal and top-view sensor is higher than that of a single sensor, the accuracy is too high. Hence, we employed data from both he sensors in our work.

**Table 6. Comparison of computational time**

| **Method** | **Average computational time (ms)** |
|---|---|
| RGB | 1450 |
| Kinect-F | 120 |
| Kinect-F + Kinect-T | 155 |

## 5. Conclusion

A new scheme for fall detection of the elderly using Kinect depth data was proposed in this research. Using depth data from the frontal-mounted Kinect sensor and the depth-mounted Kinect sensor, depth motion maps (DMM) were generated. From these maps, depth ConvNet features were extracted using deep convolutional neural network (DCNN) structure. In this structure, two convolutional layers and two max-pool sub-sampling layers were employed.From the extracted features, fall detection was done using Extreme Learning Machine (ELM) classifier. The temporal frame length of the depth motion maps was varied, and the corresponding fall detection accuracy was obtained for the identification of optimal temporal length. It was found that maximum accuracy was achieved for a frame length of 35 frames. The efficacy of the proposed system was validated using performance evaluation. This system achieved an accuracy of 97.14%. The time taken for classifying one action was only 155 ms. Thus, this scheme has a very low time complexity. Since, this scheme employs only depth maps, the privacy issue is solved. The proposed scheme was compared using other combinations of features and classification techniques like HOG+SVM, ConvNet + SVM and HOG+ELM. It was observed that, the proposed scheme attained best results compared to all other combinations.

**References**

[1] N. Noury *et al.*, "Fall detection - Principles and methods," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2007, pp. 1663–1666.

[2] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 2005, vol. 7 VOLS, pp. 3551–3554.

[3] X. Yu, "Approaches and principles of fall detection for elderly and patient," in *2008 10th IEEE Intl. Conf. on e-Health Networking, Applications and Service, HEALTHCOM 2008*, 2008, pp. 42–47.

[4] F. Bagalà *et al.*, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS One*, vol. 7, no. 5, 2012.

[5] M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, pp. 144–152, 2013.

[6] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors (Switzerland)*, vol. 14, no. 7. pp. 12900–12936, 2014.

[7] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait Posture*, vol. 28, no. 2, pp. 285–291, 2008.

[8] M. Tolkiehn, L. Atallah, B. Lo, and G. Z. Yang, "Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2011, pp. 369–372.

[9] N. El-Bendary, Q. Tan, F. C. Pivot, and A. Lam, "Fall detection and prevention for the elderly: A review of trends and challenges," *Int. J. Smart Sens. Intell. Syst.*, vol. 6, no. 3, pp. 1230–1266, 2013.

[10] T. Nguyen Gia *et al.*, "Energy efficient wearable sensor node for IoT-based fall detection systems," *Microprocess. Microsyst.*, vol. 56, pp. 34–46, 2018.

[11] N. Mozaffari, J. Rezazadeh, R. Farahbakhsh, S. Yazdani, and K. Sandrasegaran, "Practical fall detection based on IoT technologies: A survey," *Internet of Things*, vol. 8, p. 100124, 2019.

[12] X. Kong, Z. Meng, N. Nojiri, Y. Iwahori, L. Meng, and H. Tomiyama, "A HOG-SVM Based Fall Detection IoT System for Elderly Persons Using Deep Sensor," in *Procedia Computer Science*, 2019, vol. 147, pp. 276–282.

[13] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Proceedings - 2009 6th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2009*, 2009, pp. 138–143.

[14] A. K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial

accelerometer fall detection algorithm," *Gait Posture*, vol. 26, no. 2, pp. 194–199, 2007.

[15] J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy, "Wearable sensors for reliable fall detection," *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 7 VOLS, pp. 3551–3554, 2005.

[16] T. Zhang, J. Wang, P. Liu, and J. Hou, "Fall Detection by Embedding an Accelerometer in Cellphone and Using," *J. Comput. Sci.*, vol. 6, no. 10, pp. 277–284, 2006.

[17] S. B. Khojasteh, J. R. Villar, C. Chira, V. M. González, and E. de la Cal, "Improving fall detection using an on-wrist wearable accelerometer," *Sensors (Switzerland)*, vol. 18, no. 5, pp. 1–28, 2018.

[18] C. Rougier and J. Meunier, "Demo : Fall Detection Using 3D Head Trajectory Extracted From a Single Camera Video Sequence," *J. Telemed. Telecare*, vol. 11, no. 4, pp. 2–3, 2005.

[19] S. G. Miaou, P. H. Sung, and C. Y. Huang, "A customized human fall detection system using omni-camera images and personal information," *Conf. Proc. - 1st Transdiscipl. Conf. Distrib. Diagnosis Home Heal. D2H2 2006*, vol. 2006, pp. 39–42, 2006.

[20] M. Kepski and B. Kwolek, "Fall detection using ceiling-mounted 3D depth camera," *VISAPP 2014 - Proc. 9th Int. Conf. Comput. Vis. Theory Appl.*, vol. 2, pp. 640–647, 2014.

[21] K. De Miguel, A. Brunete, M. Hernando, and E. Gambao, "Home camera-based fall detection system for the elderly," *Sensors (Switzerland)*, vol. 17, no. 12, 2017.

[22] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, "Fall detection based on body part tracking using a depth camera," *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 2, pp. 430–439, 2015.

[23] P. Feng, M. Yu, S. M. Naqvi, and J. A. Chambers, "Deep learning for posture analysis in fall detection," in *International Conference on Digital Signal Processing, DSP*, 2014, vol. 2014–January, pp. 12–17.

[24] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Systems with Applications*, vol. 105. pp. 233–261, 2018.

[25] 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. Heal. Informatics*, vol. 23, no. 1, pp. 314–323, 2019.

[26] B. Jokanovic, M. Amin, and F. Ahmad, "Radar fall motion detection using deep learning," in *2016 IEEE Radar Conference, RadarConf 2016*, 2016.

[27] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. H. Ngu, and C. C. Rivera, "Smartfall: A smartwatch-based fall detection system using deep learning," *Sensors (Switzerland)*, vol. 18, no. 10, 2018.

[28] I. N. Figueiredo, C. Leal, L. Pinto, J. Bolito, and A. Lemos, "Exploring smartphone sensors for fall detection," *mUX J. Mob. User Exp.*, vol. 5, no. 1, 2016.

[29] P. Tsinganos and A. Skodras, "A smartphone-based fall detection system for the elderly," in

*International Symposium on Image and Signal Processing and Analysis, ISPA*, 2017, pp. 53–58.

[30] S. Madansingh, T. A. Thrasher, C. S. Layne, and B. C. Lee, "Smartphone based fall detection system," in *ICCAS 2015 - 2015 15th International Conference on Control, Automation and Systems, Proceedings*, 2015, pp. 370–374.

[31] T. Vilarinho *et al.*, "A combined smartphone and smartwatch fall detection system," in *Proceedings - 15th IEEE International Conference on Computer and Information Technology, CIT 2015, 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2015 and 13th IEEE International Conference on Pervasive Intelligence and Computing, PICom 2015*, 2015, pp. 1443–1448.

[32] M. Ashfak Habib, M. S. Mohktar, S. Bahyah Kamaruzzaman, K. Seang Lim, T. Maw Pin, and F. Ibrahim, "Smartphone-based solutions for fall detection and prevention: Challenges and open issues," *Sensors (Switzerland)*. 2014.

[33] S. Abbate, M. Avvenuti, F. Bonatesta, G. Cola, P. Corsini, and A. Vecchio, "A smartphone-based fall detection system," *Pervasive Mob. Comput.*, vol. 8, no. 6, pp. 883–899, 2012.

[34] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, 2014.

[35] B. Kwolek and M. Kepski, "Improving fall detection by the use of depth sensor and accelerometer," *Neurocomputing*, vol. 168, pp. 637–645, 2015.

[36] X. Kong, Z. Meng, N. Nojiri, Y. Iwahori, L. Meng, and H. Tomiyama, "A HOG-SVM Based Fall Detection IoT System for Elderly Persons Using Deep Sensor," *Procedia Comput. Sci.*, vol. 147, pp. 276–282, 2019.

[37] Y. Xu, J. Chen, Q. Yang, and Q. Guo, "Human posture recognition and fall detection using kinect V2 camera," *Chinese Control Conf. CCC*, vol. 2019–July, no. c, pp. 8488–8493, 2019.

[38] S. Li, H. Xiong, and X. Diao, "Pre-impact fall detection using 3d convolutional neural network," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2019–June, pp. 1173–1178, 2019.

[39] L. Panahi and V. Ghods, "Human fall detection using machine vision techniques on RGB–D images," *Biomed. Signal Process. Control*, vol. 44, pp. 146–153, 2018.

[40] A. Shojaei-Hashemi, P. Nasiopoulos, J. J. Little, and M. T. Pourazad, "Video-based Human Fall Detection in Smart Homes Using Deep Learning," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2018–May, pp. 0–4, 2018.

[41] G. Mastorakis and D. Makris, "Fall detection system using Kinect's infrared sensor," *J. Real-Time Image Process.*, vol. 9, no. 4, pp. 635–646, 2014.