COL812: Lab 3
Topic: Fall detection

Naman Jain (2020anz8848)

May 21, 2021

Problem definition

Falling is chronic problem for elderly population [1] which can cause fractures, trauma or in extreme cases death. Therefore it is imperative to device a solution for fall detection to improve the standard of living.

Fall can be cause due to various reason like -

- Losing balance due to weak muscles.
- Loss of consciousness.
- Stumble while walking, caused by slippery floor or hard friction object.
- Medical condition like seizures or stoke.

Human fall can be defined as involuntary action in which posture changes rapidly from sitting/standing position to lying pose or reclining position. According to Kellogg international group [2], fall is defined as - "unintentionally coming to ground, or some lower level not as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure". This definition clearly remove any external factor and include fall caused due medical condition like stoke or syncope.

Exploration which we will be doing is to detect the fall after it has occurred. There will be no effort put into fall prediction/forecast techniques. This means that whenever a fall is detected a notification/feedback will be generated to caregiver or emergency services. No fall prevention methods will be deployed like airbags.

Furthermore it is necessary to consider different sets of the fall. Fall models are discussed in N.Noury [3], (i) forward fall resulting in full body impact on the floor, (ii) while walking a person trips over loses balance but regains the standing posture and continue walking and (iii) person loses consciousness and end up in sitting position against a support (wall). Various fall scenarios for evaluation of fall detection techniques are mentioned in [4], Table 1 reference the same and show which type of scenarios will be considered in the case study. Few fall scenarios are depicted in Figure 1.

Generic fall detection system

Fall detector system consisting for three modules as shown in Figure 1 - (i) sensing block, (ii) processing block and (iii) feedback unit (user interface)

Sensing block contains various sensors which can be broadly classified into three categories -

- Wearable sensor - Numerous technique deploy such sensor due to small size, low power and portability. They have ability to detect inclination of the body or speed of movement. This consist of mercury tilt, accelerometer [5], gyroscope, ECG (Electocardiography), EEG (Electroencephalography) or EOG (Electromyography).

Type	Sub-type	Outcome	Proposed approach
Backward fall	Ending sitting	Positive	Yes
	Ending lying	Positive	Yes
	Ending in lateral position	Positive	Yes
	With recovery	Negative	Yes
Forward fall	On the knees	Positive	Yes
	With forward arm protection	Positive	No
	Ending lying flat	Positive	Yes
	With rotation, ending in the lateral right position	Positive	Yes
	With rotation, ending in the lateral left position	Positive	Yes
	With recovery	Negative	Yes
Lateral fall to the right	Ending lying flat	Positive	Yes
	With recovery	Negative	Yes
Lateral fall to the left	Ending lying flat	Positive	Yes
	With recovery	Negative	Yes
Syncope	Vertical slipping against a wall finishing in sitting position	Negative	No
Neutral	To sit down on a chair then to stand up	Negative	Yes
	To lie down on the bed then to rise up	Negative	Yes
	Walk a few meters	Negative	Yes
	To bend down, catch something on the floor, then to rise up	Negative	No
	To cough or sneeze	Negative	Yes

Table 1: Different scenarios under fall detection

- Ambient sensor - These sensor are used to detect change in environment, this is utilize to detect the fall. Examples are pressure sensor in the floor or smart tiles [6], RFID, Doppler Radar[7], microphone [8], ultrasonic sensor [9] and infrared sensor.
- Visual sensor - Recent studies are focusing on such sensor due to level of information captured and recent advancement in field of image processing and machine learning. This consist of RGB cameras[10] (USB webcam or mobile camera), infrared camera, depth camera (RGB-D or kinetic) and wearable cameras.
- Sensor fusion - Due to advent of internet of things, studies are using multiple sensors are used to record the data. Accelerometer, gyroscope and magnetometer are put in single device (IMU or smartphone) or kinetic sensor used in conjunction with smartphone.

After collection of data, it is passed onto processing block which employ either analytical model (Threshold) or machine learning to distinguish fall from normal human activity. Analytical model is straightforward to detect a particular human pose but fails to account for surrounding and transition features which lead to high false positive. Compared to this, machine learning based models like SVM, bagging, deep learning (CNN or LSTM), reinforcement learning can intuitively detect the fall from the recorded data especially from ambient and visual sensors.

Once the decision is made by processing unit, in case fall is confirmed the it is important to notify caregiver, hospital or emergency services with minimum latency. This is done in feedback unit. Implementation of such unit varies based on sensor and processing unit frequency of operation but generally before any alarm is raised it gives user sometime to cancel such alarm or notification in-case of false positive.

Most of the studies which employ visual sensor to detect fall did not demonstrate the ability to work in real time due to data intensive computation. Approaches deploying wearable sensor have higher percentage which given some kind of alarm or notification.

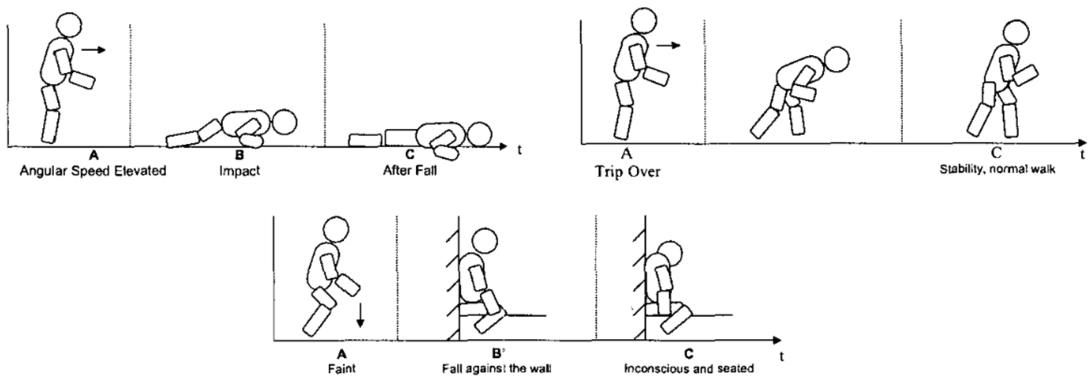


Figure 1: Fall scenarios

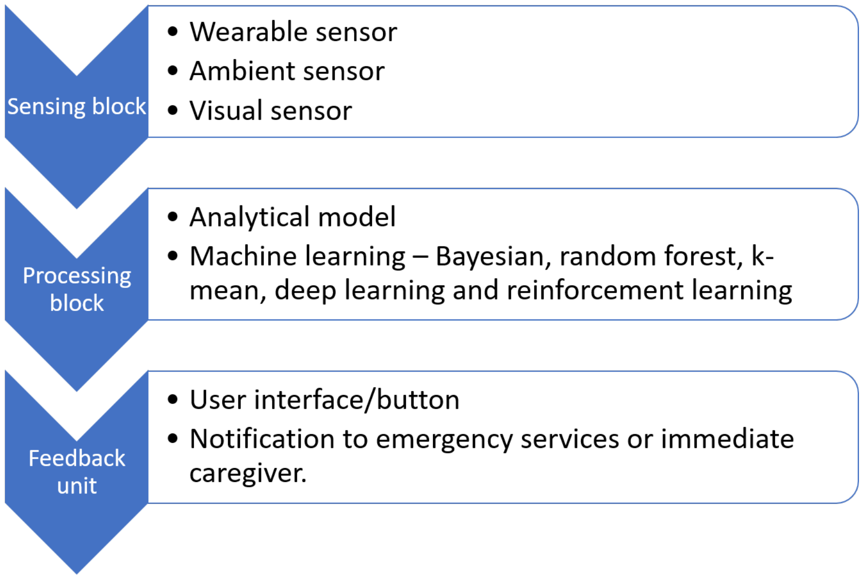


Figure 2: System overview

As it is difficult to collect data with respect to genuine fall where subject age is more than 65 years, most of the studies uses simulated data in which either a volunteer or professional stuntman perform various type of falls. Due to less funding or lack of time and privacy concerns, there is no robust common dataset is available to benchmark such activities. Few studies based on wearable device use a dataset provided by [11] but majority generate its own dataset. There is another dataset which contains RGB-depth images as well as accelerometer data [12]

Lastly, for evaluating a particular technique following cases are recorded -

- True positive (TP): When system detect fall when it has happened
- False positive (FP): When system detect fall when it has not happened
- True negative (TN): When system did not detect fall as normal activity is going on
- False negative (FN): When system did not detect fall even though it has happened

Mainly two criteria are defined using the above cases to evaluate performance of a technique -

- Sensitivity : This show a ability of a system to detect a fall

$$Sensitivity = \frac{TP}{TP + FN} \times 100$$

- Specificity: Ability to detect only a fall

$$Specificity = \frac{TN}{TN + FP} \times 100$$

For an ideal system, sensitivity and specificity should be 100%. But for real system, it is important to optimize sensitivity close to 100% as false negative are detrimental to the user of a system.

Approach used for case study

Approach is based on the study demonstrated in [12]. Flow and block diagram for proposed approach is shown in Figure 3 and Figure 4. In the approach, authors are using data from accelerometer as well as RGB-depth camera (kinect). UR Fall Detection Dataset [12] will be used.

The process can be split into three major blocks, (i) monitoring accelerometer data (ii) storing depth image from kinect and calculating background image and (iii) image processing with decision making combined. First block is responsible to collect the accelerometer data and provide potential fall decision, this will help save time and move on to next frame in further blocks. Second block is responsible to calculate the background image, this will help in extraction of human subject (foreground). This is separated from image processing module as it involves storing depth images at regular interval every 15^{th} image and stored in buffer size of 13. Last block is image processing module, (i) it consist of floor plane estimation, (ii) foreground extraction to get centroid and bounding box of human subject and (iii) decision making block is support vector machine (SVM) which take in distance of centroid, bounding box statistics, human height. All the blocks described here are shown in Figure 6 and CFDG is shown in Figure 7.

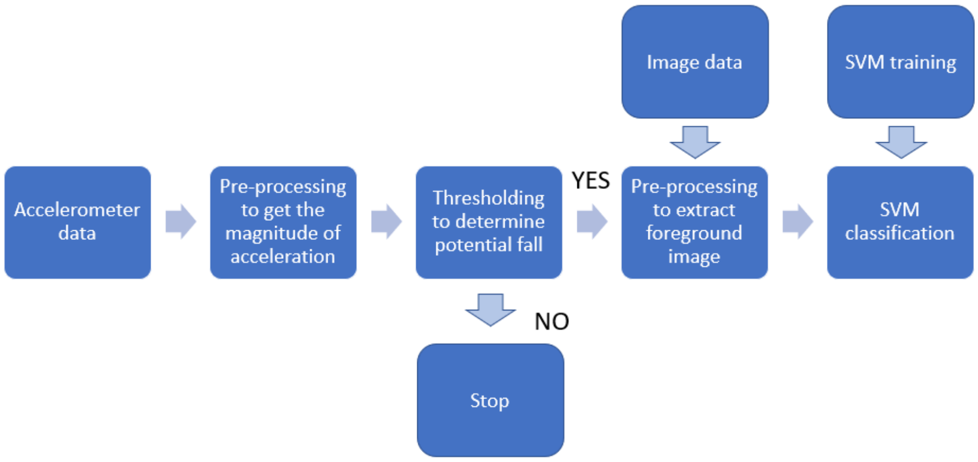


Figure 3: Flow diagram for fall detection

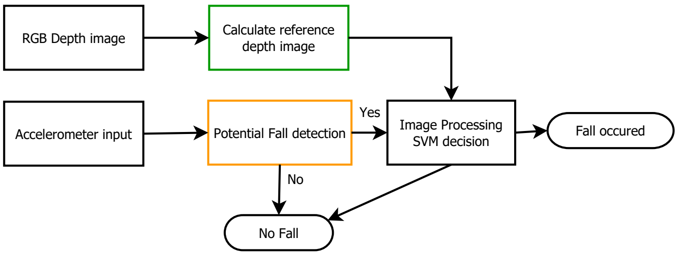
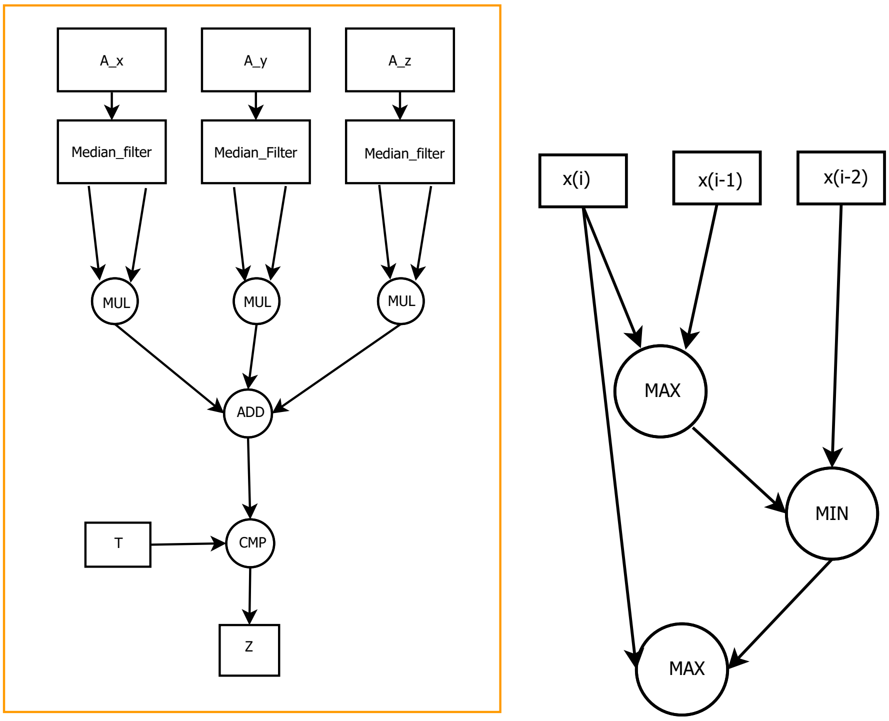


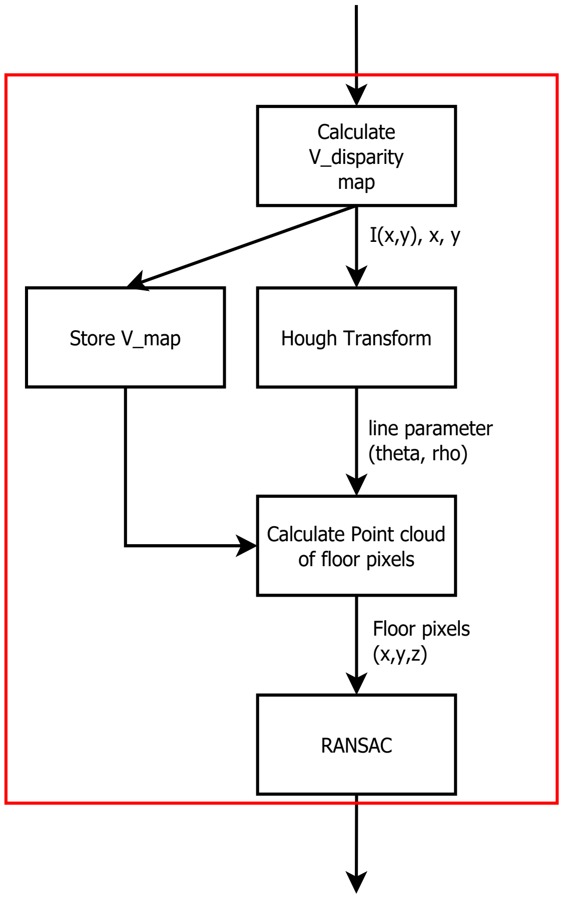
Figure 4: Block diagram for overall system



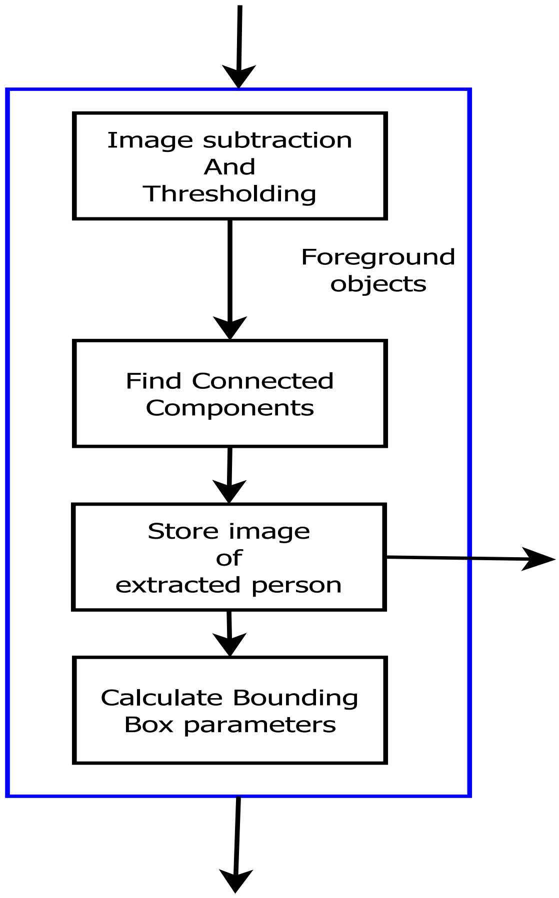
(a) Potential fall detector

(b) Median Filter size 1x3

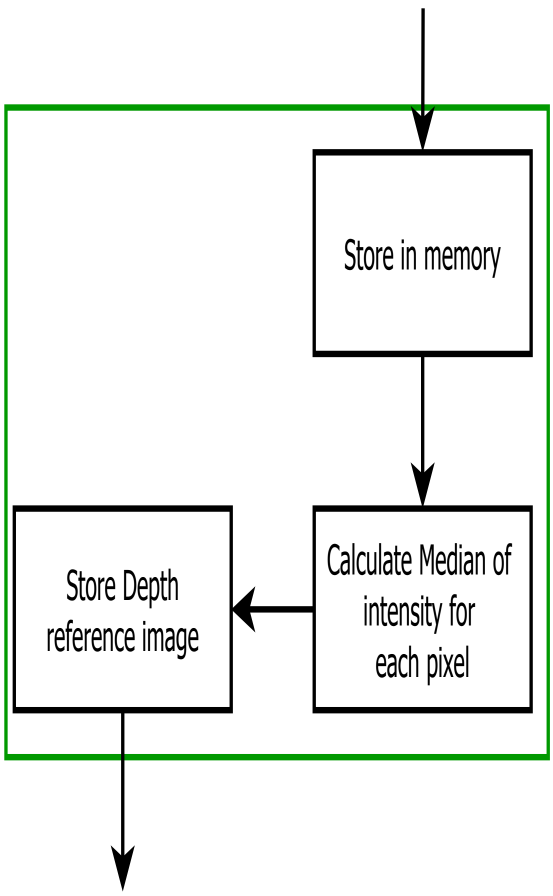
Figure 5: Accelerometer input processing block



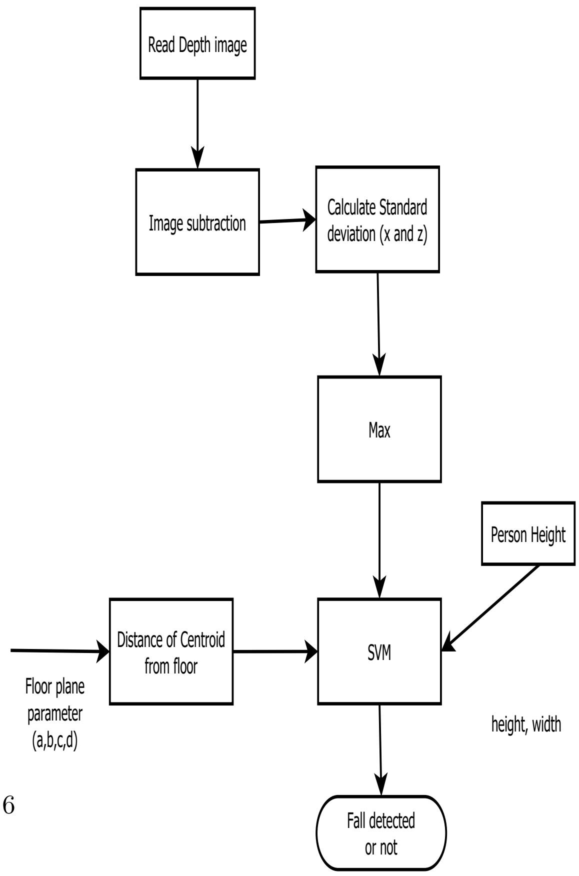
(a) Extract floor plane



(b) Bounding box extraction



(c) Background image extraction



(d) SVM fall detector

Figure 6: Image processing modules

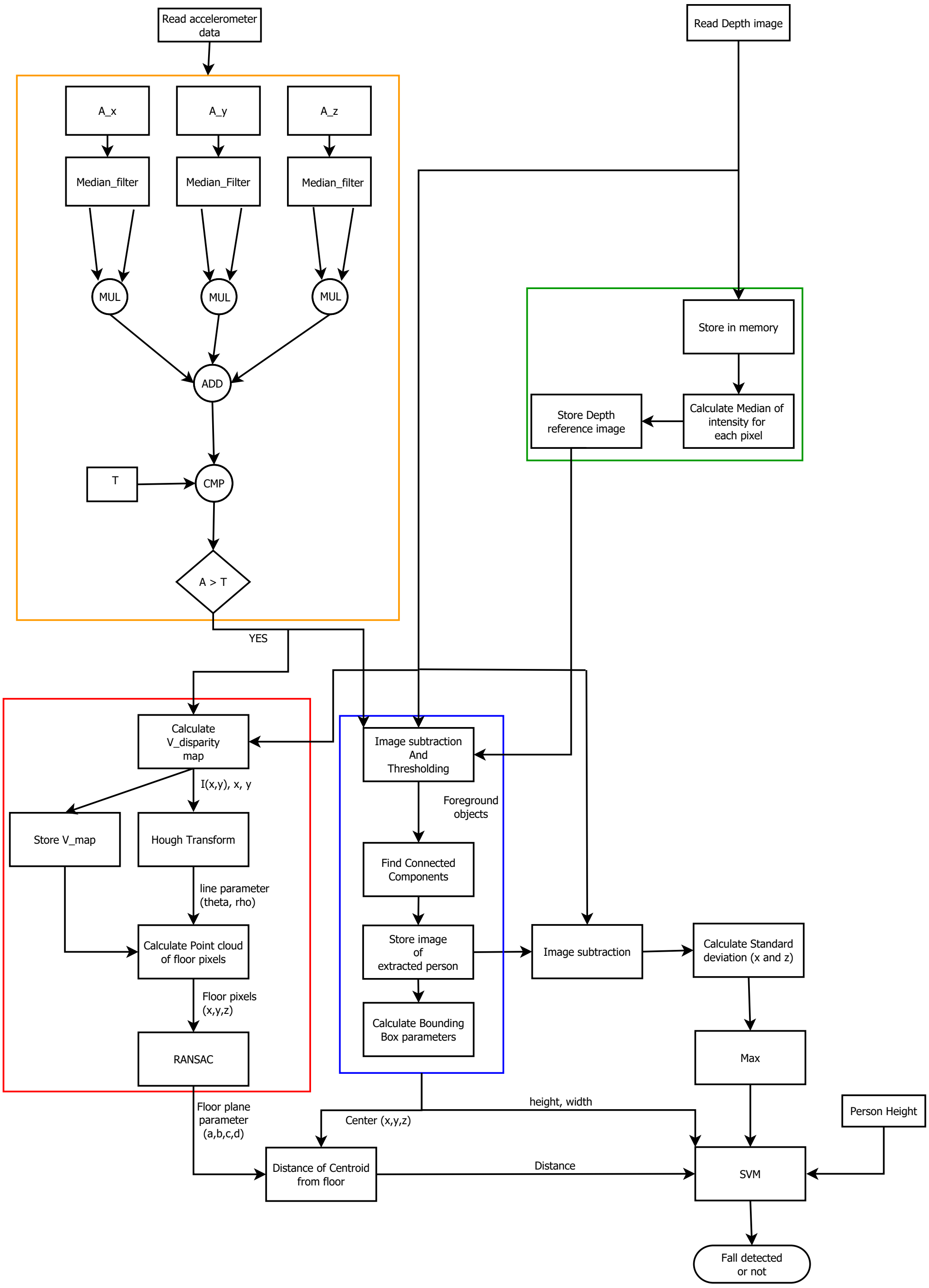


Figure 7: Overall fall detection system

Approach used in systemC modeling

As the approach described in [12] is intensive to be implemented in systemC under the given time frame. Thus, approach is trimmed down and following blocks are implemented in the systemC -

1. Depth image storing and calculated background image
2. Person extraction to get the centroid and bounding box width & height
3. Floor plane estimation
4. Distance result module to determine if fall has occurred or not

In addition to these blocks, an additional module is implemented to read and send the data to subsequent blocks. Module 1 gets the depth image data and stores it in buffer size of 13 and calculate the background image via checking median of each pixels along the buffer of images. This module which uses the highest amount of data. To accommodate this stack size of program is increase from default $16Kb$ to $16.7Mb$.

Further module 2 will take the background image and subtract it from current frame (depth image), then threshold it to get the binary image. On this binary image, erosion is performed to remove smaller connected components and dilation to restore size of bigger components. Finally number of connected component is determined and largest component with respect to area will be used. Using this centroid and bounding box height and width is determined. Module 3 will use the depth image to calculated the v-disparity map using Equation 1 where $b = 7.5cm$ is horizontal baseline and $f = 580pixels$is focal length of camera [12] , this will allow to get the line corresponding to floor pixels. Line extraction is done via applying hough transform on v-map. Using this line, points belonging to floor are determined in form of (x, y, z) to get a point cloud. After fitting the point cloud, floor plane $(ax + by + cz = 1)$ parameters (a, b, c) are determined which will be passed to further module. Module 4, determine the distance of centroid from the floor plane and the ratio of bounding box height and width. This will be used to determined if fall has happened or not. Whole systemC model is shown in Figure 8. For communicating and maintaining the synchronization *sc_fifo* is used with template class of *Mat* from opencv library.

$$disp = \frac{b * f}{depth} \tag{1}$$

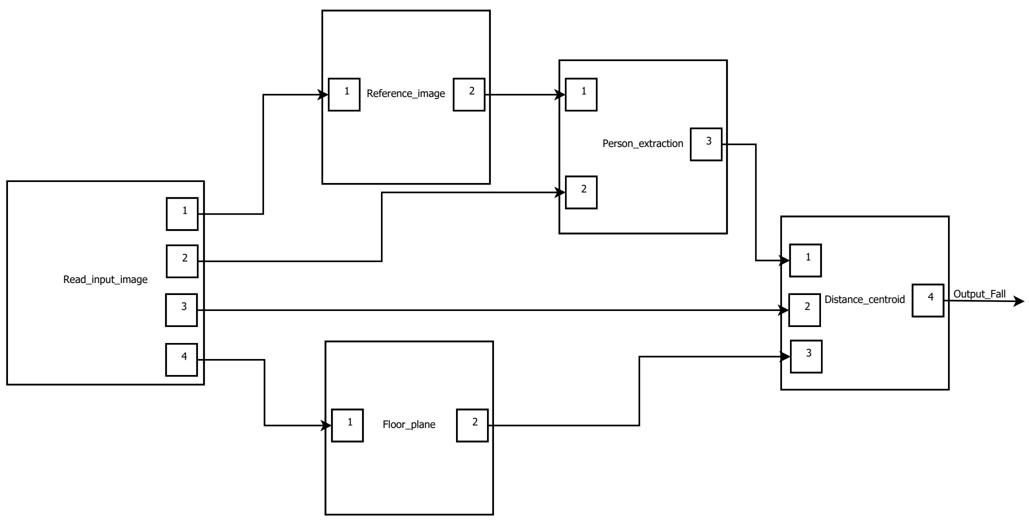


Figure 8: SystemC model overview

Results and testing

SystemC model is written in visual studio using external library opencv for using image processing algorithms. Output background image from module 1 is shown in Figure 9, it can be inferred that human subject is removed. Initially until the buffer has been filled to sufficient length, background image output will not be correct. To rectify this, in real world initial background image can be stored and updated online. Depth image data is stored as PNG16, therefore to get the actual depth need to use the Equation 2 as mentioned in [12], where $C_0$is camera related constant which is equal to 6000.

$$depth(mm) = \frac{C_0 * Pixel\ intensity(i,j)}{65535} \tag{2}$$

Output of module 2 is shown in Figure 10. As it can be seen from right image, human subject is extraction correctly. Qualitatively, it can be seen that when fall has occurred, bounding box ratio is reduced and distance from the floor is reduced. Intermediate images from floor plane are shown in Figure 11. From the v-map image it can be seen that there is slant/diagonal line which correspond to floor plane pixels and vertical line seen above is for the walls in the room. To avoid wall pixels original v-map is halved to use the bottom pixels only and the disparity value are used in range $0 - 50$ only.

Finally subsequent Figures shows the plot of distance of centroid from floor plane, bounding box ratio and change in Y coordinate of centroid with respect to each frame. When the bounding box ratio is less than 0.6 or $number\ of\ rows - y\_centroid$ is less than 90 then fall has occurred. For some dataset, after fall has happened ratio goes above 0.6 this is due to the incorrect extraction of human subject due to either subject not in the image or below the camera. Use of accelerometer data and second camera on ceiling will resolve such issues which is used in [12]. Signal output for various dataset is shown in Figure 16. Time access signifies frame number in each set of data.

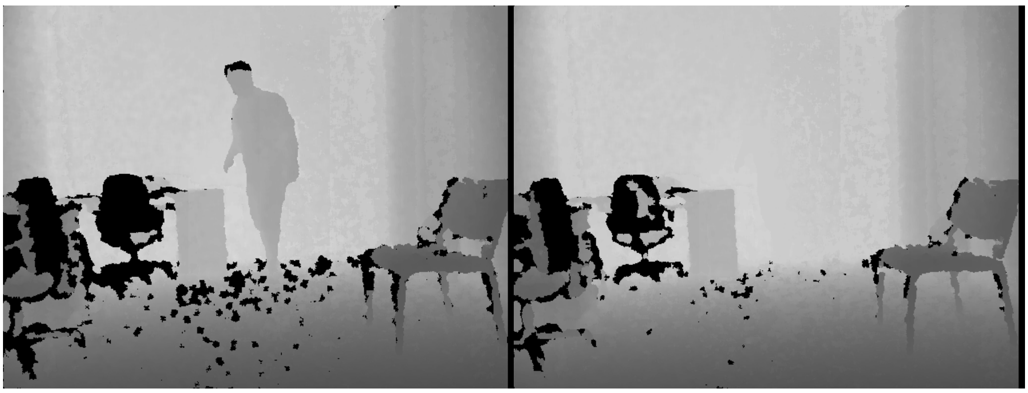


Figure 9: Comparing background image with current image on left

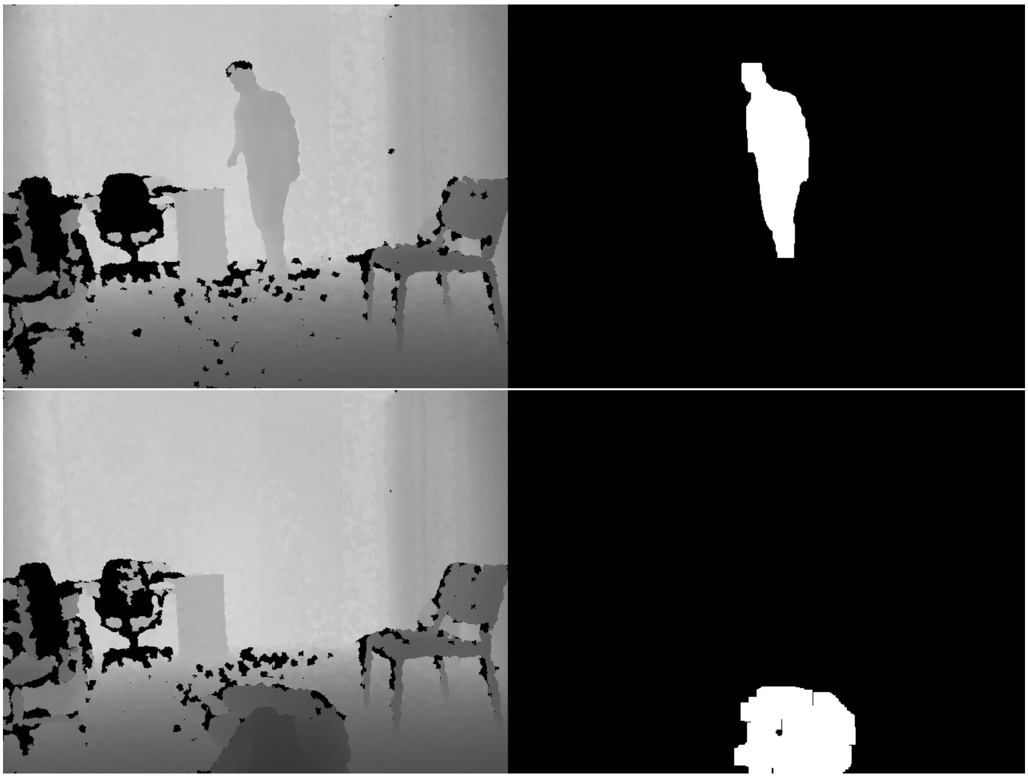


Figure 10: Extraction of human subject

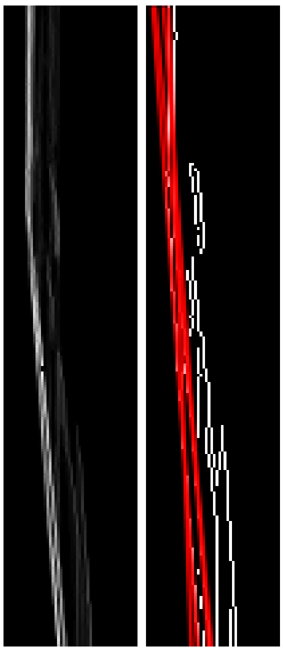


Figure 11: V-disparity map and extracted line

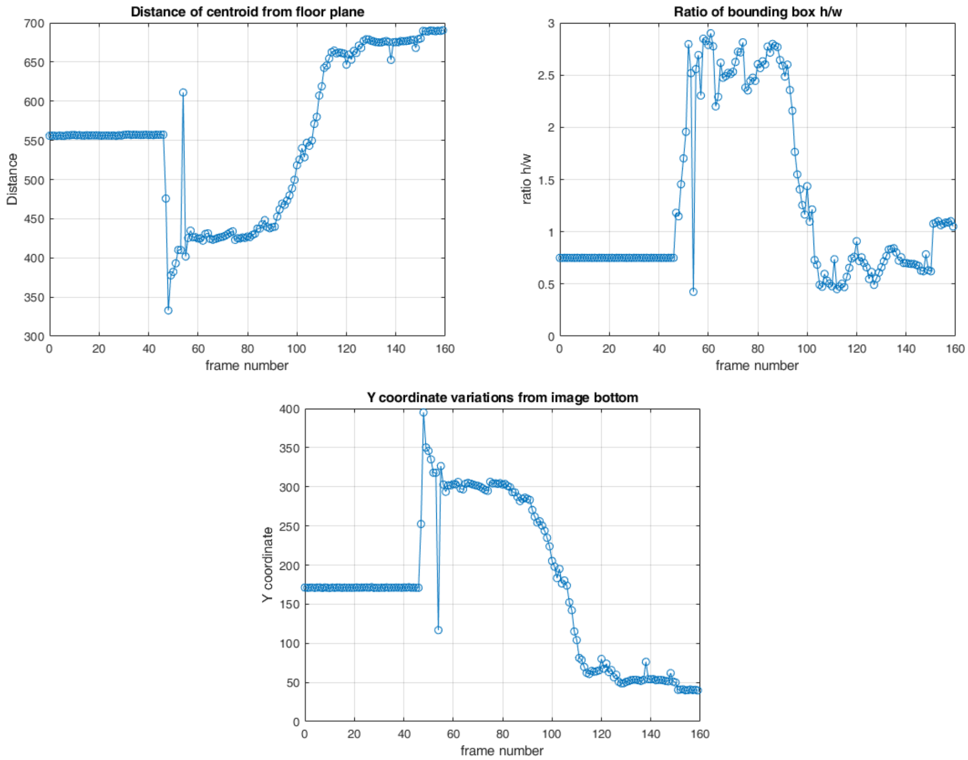


Figure 12: Final plot for dataset 1

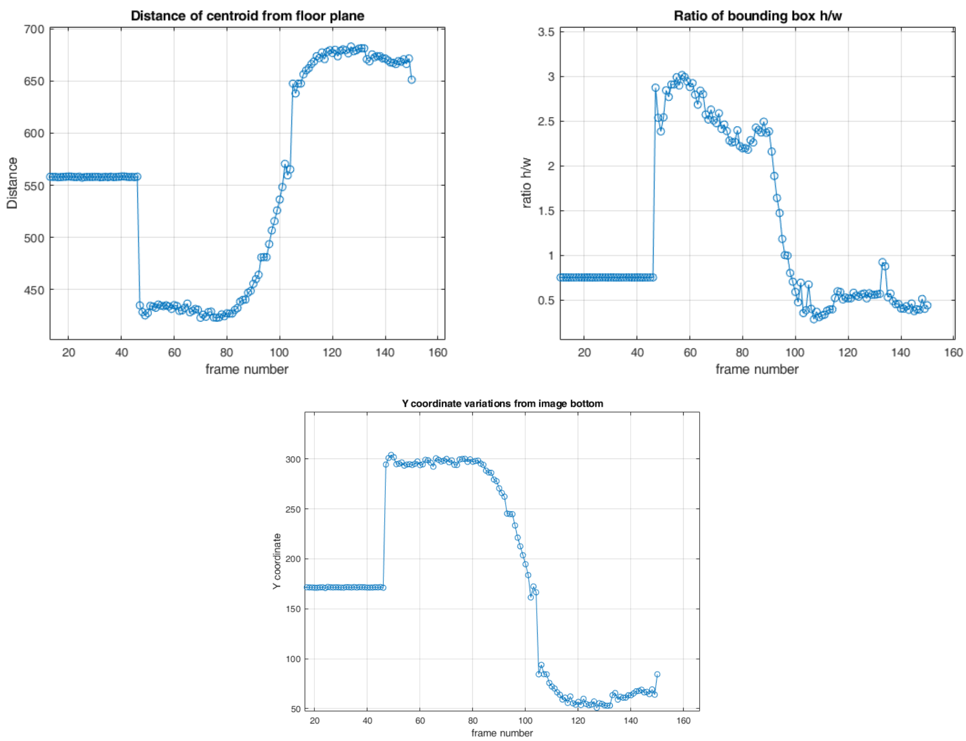


Figure 13: Final plot for dataset 2

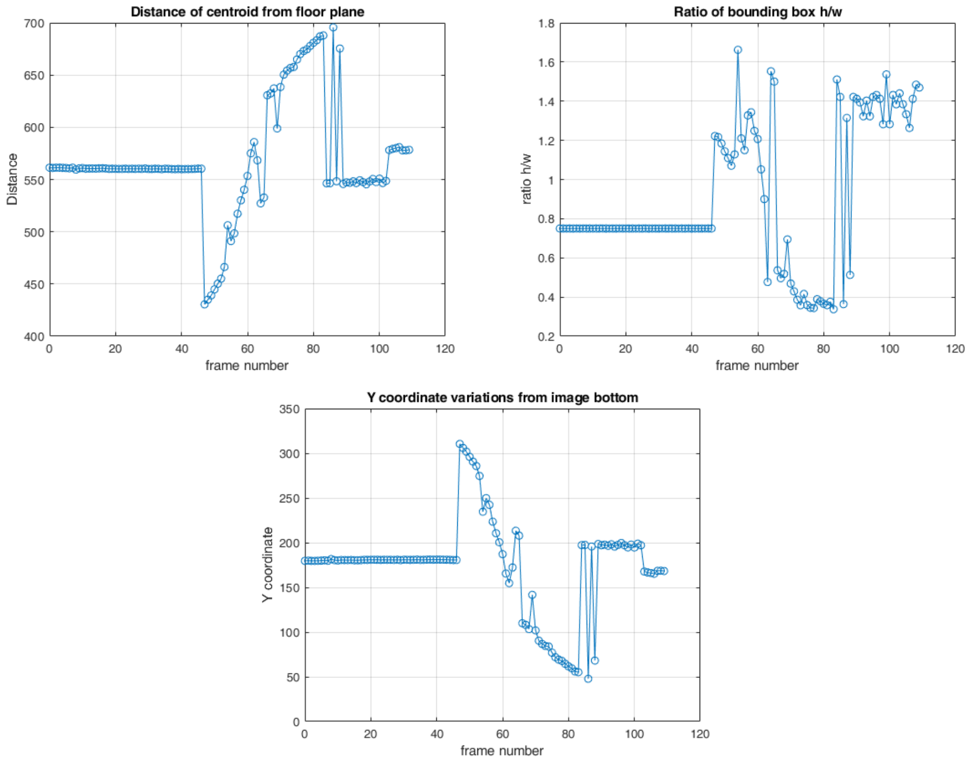


Figure 14: Final plot for dataset 3

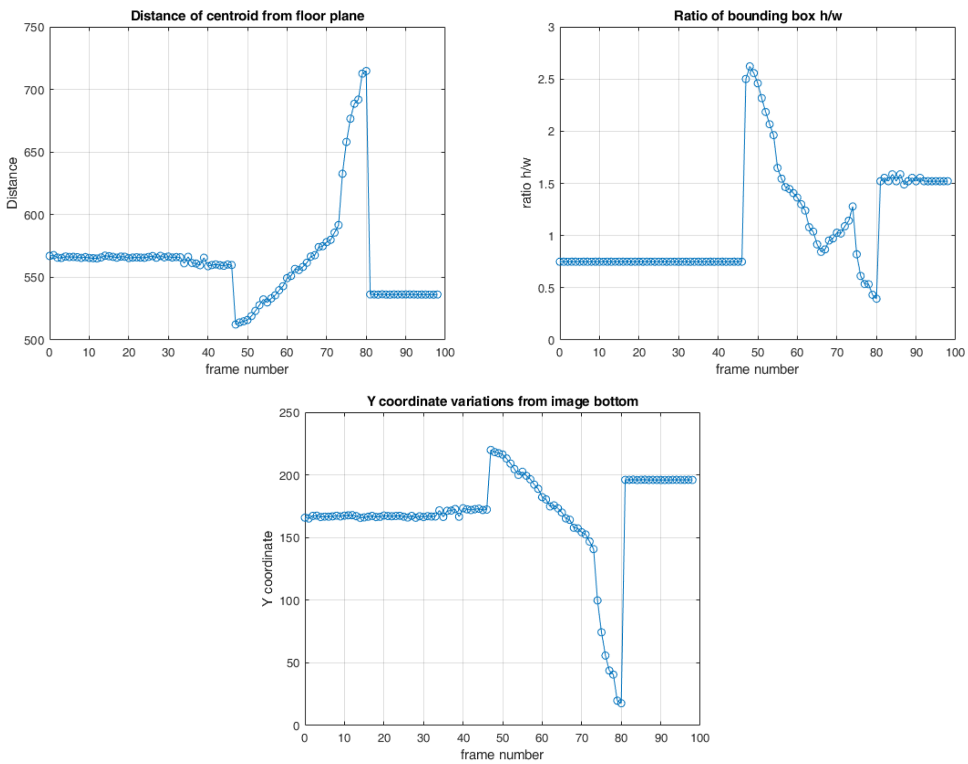


Figure 15: Final plot for dataset 4

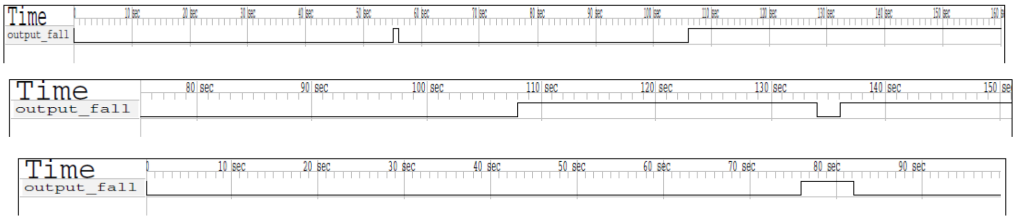


Figure 16: Output of fall detect signal

References

[1] "Falls." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[2] M. J. Gibson, "The prevention of falls in later life: a report of the kellogg international work group on the prevention of falls by the elderly," *Dan Med Bull*, vol. 34, no. 4, pp. 1–24, 1987.

[3] N. Noury, P. Barralon, G. Virone, P. Boissy, M. Hamel, and P. Rumeau, "A smart sensor based on rules and its evaluation in daily routines," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439)*, vol. 4. IEEE, 2003, pp. 3286–3289.

[4] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. Laighin, V. Rialle, and J. Lundy, "Fall detection-principles and methods," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 1663–1666.

[5] S. Garg, B. K. Panigrahi, and D. Joshi, "An accelerometer based fall detection system using deep neural network," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. IEEE, 2019, pp. 1–7.

[6] H. Rimminen, J. Lindström, M. Linnavuo, and R. Sepponen, "Detection of falls among the elderly by a floor sensor using the electric near field," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 6, pp. 1475–1476, 2010.

[7] B. Y. Su, K. Ho, M. J. Rantz, and M. Skubic, "Doppler radar fall activity detection using the wavelet transform," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 3, pp. 865–875, 2014.

[8] M. Cheffena, "Fall detection using smartphone audio features," *IEEE journal of biomedical and health informatics*, vol. 20, no. 4, pp. 1073–1080, 2015.

[9] D. Asbjørn and T. Jim, "Recognizing bedside events using thermal and ultrasonic readings," *Sensors*, vol. 17, no. 6, p. 1342, 2017.

[10] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Scientific reports*, vol. 11, no. 1, pp. 1–17, 2021.

[11] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, p. 198, 2017.

[12] B. Kwolek and M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Computer methods and programs in biomedicine*, vol. 117, no. 3, pp. 489–501, 2014.