

AUTOMATED FALL DETECTION SYSTEM FOR ELDERLY PEOPLE BASED ON SENSOR DATA AND VISION DATA

Indu Chawla¹
Archana Purwar
Khushi Chauhan
Vandita Chauhan

Received 08.01.2025
Revised 25.03.2025
Accepted 11.05.2025

Keywords:

Fall detection, Machine learning, Sensor data, VISION data, Elderly people

A B S T R A C T

Falls are one of the most crucial health risks faced by the elderly. Sudden falls may craft serious danger to elderly people and can cause injuries or even long-term disability. By developing an automated fall detection system, the aim is to mitigate these risks and improve the overall well-being of elderly individuals. With the presence of wearable devices, important health parameters can be extracted for healthcare assessment. Along with this, non-wearable devices like cameras also help in monitoring any unexpected fall by capturing and analyzing image data. In this paper, we analyze sensor data and image data as extracted from wearable and non-wearable devices respectively for accurate fall detection. This research emphasizes use of threshold based and machine learning models for sensor data. For image-based fall-detection, fall events are extracted using CNN and YOLO. In summary, the current work offers a promising way for elderly care promoting a safer environment.



© 2025 Published by Faculty of Engineering

1. INTRODUCTION

Falls among older adults are particularly worrisome due to their profound effects on health and quality of life. These incidents frequently result in serious injuries, reduced mobility, and increased medical costs. In order to address this challenge, we address the critical need for accurate detection of fall incidents. In case of an unintentional fall, system can identify such incident and care taker or healthcare provider can provide immediate help by reducing the severity of injury and facilitate timely treatment (Huerta et al., 2024; Vasoya et al., 2023). This also provides a reassurance to the person

taking care of them that their loved one is being supported and monitored in events of their absence.

In order to address the demand for real time and low-cost system for fall identification, a wide range of techniques have so far been proposed and applied for automatic fall detection (Jiang et al., 2024). These can be broadly categorized as based on wearable and non-wearable devices. Current technologies, particularly wearable devices, show potential for solving the problem of fall detection. The sensors are generally embedded in devices like belt or wrist band or walking stick etc. which can be worn around human body. The sensors values are monitored and whenever there is any change in the body

¹ Corresponding author: Indu Chawla
Email: indu.chawla@jiit.ac.in

orientation and hence variation in these sensors value, a fall can be detected. For example, a gyroscope measures angular velocity and angular acceleration along with other parameters. An accelerometer sensor which is also one of the commonly used sensors provides acceleration measurements (Durgun, 2023; Van Thanh et al., 2019). These are the basic sensors which are now a days embedded in all smart phones.

Recently, much attention has been given to accuracy improvement in fall identification or segregating fall incidents from non- fall incidents. Different machine learning algorithms have been proposed which helps in classification of fall incidents from the continuously received real time data as received from sensors devices (Musci et al., 2020, Liang, 2017). In threshold-based activities, the observed values from sensors embedded in wearable devices are compared with the selected threshold values. If the values are beyond the threshold value boundary, then a fall is notified. Threshold-based approaches are generally simpler and less computationally intensive. The determination of threshold value plays an important role in accuracy of fall detection output. Authors have examined and proposed an effective method for selection of threshold value (Razum et al., 2018). We have also chosen to apply the same method for selecting effective threshold value.

Automated fall detection has also been proposed based on non-wearable devices or vision-based systems which use surveillance cameras installed at different places (Alam et al., 2022). These cameras record the real time movements and the vision-based algorithms continuously monitor this data and report fall incidents as and when some changes are reflected. As, sometimes it is difficult for elderly people to wear these devices all the time or the devices become inoperative due to limited battery power. In contrast with this vision-based system are effective, however people sometimes also hesitate to use them for privacy and security purpose.

We have worked to enhance the effectiveness of fall detection systems while accommodating the unique needs and traits of older adults. Developing robust and efficient solutions can enhance early fall detection, enable swift intervention, and ultimately improve the safety and quality of life for older adults. In this paper, we explore two approaches by using data from two sources i.e. sensor-based data and image data. The proposed work intends to identify falls among the elderly by developing comprehensive and robust solutions tailored specifically to their needs. This will be achieved by leveraging cutting-edge sensor technology, sophisticated signal processing techniques, and advanced machine learning algorithms (Peng et al., 2019).

By preventing falls and reducing fall-related injuries, our system can contribute to significant cost savings for healthcare systems, insurers, and caregivers. The main contributions of the paper can be summarized as below:

- Examining the performance of fall detection using sensor data from wearable devices using selected machine learning algorithms.
- Automated fall detection using vision-based data by applying convolutional neural network and different variants of Yolo.
- Analyze the performance of sensor-based approach in detection of fall and segregating fall from daily living activities.
- Analyze the effectiveness of two popular methodologies for vision-based fall detection

The rest of the paper is structured as follows. Section 2 depicts the related work as proposed in the literature. Section 3 elaborates the description of the dataset used for experimentation and portrays the proposed methodology for automated fall detection. Section 4 explains about the performance evaluation of proposed approaches. Finally, Section 5 concludes the paper and section 6 highlights the scope for future work.

2. LITERATURE SURVEY

Automating fall detection among elderly using machine learning techniques based on sensor data and image data has been receiving attention in current research scenario. Fall detection can be performed using data from both sensors and images (Galvão et al., 2021). The two primary approaches for data collected from sensors are: machine learning techniques and threshold-based techniques. Machine learning techniques analyse patterns in the sensor data to detect falls. In contrast, threshold-based techniques involve defining suitable threshold-based algorithms for the sensor data. By setting appropriate thresholds, the system can effectively identify fall events based on the sensor readings. For image data, Convolutional Neural Network and object detection methods like You Only Look Once can be used for the data collected from images. These methods can detect falls by effectively analysing the visual information.

2.1 Techniques used for sensor-based data

Machine learning leverages mathematical models on datasets to analyse, categorize, and uncover new insights, allowing systems to automatically learn and improve based on the training data provided (Sabry et al., 2021). In recent years, the application of machine learning in fall detection has grown, providing intelligent methods to identify falls based on an individual's activity patterns (Luna-Perejon et al., 2021). Conversely, a system that is overly cautious may fail to trigger alerts during actual falls. The ability of these algorithms to self-learn plays a crucial role in achieving accurate results. In fall detection, the machine learning model classifies outcomes into falls and rest activities of daily life. In (Anishchenko, 2018) it has been proposed employing

deep learning and transfer learning techniques to detect falls using sensor-generated data in realistic settings, aiming to address the limitations associated with simulated datasets under controlled conditions. Support Vector machine has also been widely used for this task. In (Liang & Usaha, 2017), time series analysis was performed using Support Vector Machine, while frequency domain analysis was carried out with lifting wavelet transform. The frequency-based approach achieved 99.7% accuracy in fall detection, whereas the accuracy of SVM based time domain analysis was 97.6%. The study assessed several algorithms, including

decision trees, k-nearest neighbours, decision tree ensembles, multilayer perceptron. Experimental results revealed that the decision tree ensemble provided the highest performance among the algorithms evaluated. Javed et al, 2021 proposed sensors embedded in smart phones. Author used different sensors like accelerometer, magnetometer, and gyroscope for human activity recognition. Huerta et al, (2024) applied several machine learning algorithms for identifying fall and daily life activities using data from gyroscope and accelerometer. Figure 1 depicts the flowchart for sensor-based data methodology.



Figure 1. Flowchart for sensor-based data methodology

The use of threshold values involves use of several factors for detecting falls, such as being able to understand the difference between daily living activities and falls. In (Luna-Perejon et al., 2021) a technique has been described by analysing the changes in the accelerometer readings across the three-coordinate axis. It effectively differentiates between abrupt movements and actual falls. In (Althobaiti et al., 2020) a system was created utilizing a triaxial accelerometer, incorporating a method based on cumulative acceleration and rotation angle criteria. Additionally, sensor fusion methods have been evaluated in various scenarios involving sensors beyond just accelerometers. For example, Zhao et al. (2021) suggest that customising the design can increase performance as acceleration profiles differ based on individual physique. It has also been observed that an accelerometer can be combined with an HRV sensor and this can effectively help in fall detection. Each analysis uses thresholds and is conducted separately. A fall is confirmed if both analyses indicate its occurrence. The study reported an accuracy ranging from 96.79% to 99.98% in differentiating falls from similar activities. The authors propose that tailoring the design could enhance performance, as acceleration profiles vary with individual body types. This study primarily uses data generated by accelerometers and gyroscopes as sensors.

2.2 Techniques for image-based data

In addition to wearable devices, there is contribution of non-wearable devices. Non-wearable device like camera capture events for surveillance purpose. So, this can also reflect upon the fall events (Mobsite et al., 2023, Gutiérrez et al., 2021; Iqbal et al., 2017). In contrast to traditional threshold-based systems, which may face challenges with adaptability and accuracy, the image data captured through cameras has also been utilized for detection of possible fall events (Hussain et al., 2020). CNN and YOLO can autonomously learn intricate patterns and features from image data. This capability makes them more resilient and effective in managing a variety of scenarios. In (Alam et al., 2022), the model trained using convolutional neural networks was subsequently adapted for fall detection using transfer learning techniques. However, this method can be less reliable due to variations in lighting of surroundings.

The UP-Fall Detection dataset enables equitable comparisons and offers new opportunities for further investigation. YOLO variants have shown promise in real-time fall detection, with future studies focusing on exploring different deep learning models and pose estimation methods to improve performance. Figure 2 depicts the flowchart for image-based detection.



Figure 2. Flowchart for image-based detection

3. METHODOLOGY

3.1 Dataset

The results of the study are evaluated using an experimental dataset called UP-Fall, which includes 12 tasks carried out by 17 healthy young adults. There are seven human activities and events like forward fall using hands and knees, falling while sitting, falling backwards while standing or sitting. The UR-Fall dataset is an extensive set of fall detection recordings, featuring 70 sequences that include 30 fall incidents. Table 1 depicts the sample data of subjects.

Table1. Data of subjects considered during experiment.

Subject	Age	Height (m)	Weight (kg)
1	18	1.70	99
2	20	1.70	58
3	19	1.57	54
4	20	1.62	71
5	21	1.71	69
6	22	1.62	68
7	24	1.74	70
8	23	1.75	88
9	23	1.68	70
10	19	1.69	63

3.2 Dataset pre-processing

Before applying machine learning algorithms, performing pre-processing is required which delivers better results than using machine learning algorithms on raw data directly. The subsequent step involves feature selection through techniques such as random forest and ANOVA analysis to retain most relevant features. Then the dataset is divided into training and testing datasets. Cross-validation entails splitting the dataset into several segments in which training of model is done using some segments and some are used for evaluating. It aims to show how the model works in case of new data. For the sensor data from the UP-FALL dataset, both machine learning and threshold-based methods are employed, while image-based algorithms are applied for the UR-FALL dataset.

3.3 For sensor-based data

We employed six machine learning methods including Decision Tree, SVM, KNN, XGB Classifier, LightGBM Classifier, and Naïve Bayes. The Decision Tree method constructs a tree-like structure from the training data, where internal nodes denote features, branches indicate decision criteria, and leaf nodes signify the final results. Classification and regression tasks can be both performed by decision trees. Using Naïve Bayes, the probability of a fall based on sensor data collected from individuals is calculated. The algorithm assumes that features (sensor measurements) are independent of each other, simplifying probability calculations. By analyzing the likelihood of a fall given the observed sensor data, the most probable class (fall or not fall) is assigned to each data point. The XGBoost works by building an ensemble of decision trees and iteratively improving their performance. XGBoost excels in handling complex datasets with high dimensionality and non-linear relationships between features. By optimizing a specific objective function, such as binary classification for fall detection, XGBoost finds the best split points in the data to maximize predictive accuracy. LightGBM is engineered to be highly efficient and scalable, making it well-suited for large datasets with many dimensions. It employs a unique method known as Gradient-based One-Side Sampling (GOSS) to accelerate training by focusing on the most informative samples when constructing decision trees. Additionally, it uses Histogram-based Gradient Boosting (HGB) to enhance computational efficiency by converting continuous feature values into discrete bins. LightGBM also supports various objective functions and evaluation metrics, allowing flexibility in model optimization for fall detection tasks.

To train a fall detection model using kNN and SVM, we started by gathering and pre-processing the dataset, which consists of video frames or sensor data depicting both fall and non-fall events. Next, extracted pertinent features from the processed data, such as mean, variance, or signal magnitude area from sensor data, and key point

movements or motion vectors from video frames. For SVM, we have chosen an appropriate kernel function (such as linear, polynomial, or RBF) and train the classifier to find the hyperplane that optimally separates fall and non-fall instances, maximizing the margin between the classes.

For kNN, perform cross-validation to determine the optimal k number of neighbours, and train the model by retaining the training samples and their associated labels. During inference, classify new instances by assigning them the label that appears most frequently in the feature space among the k-nearest neighbours. Optimize the models according to their performance, and subsequently implement them in real-time fall detection systems. These systems can then monitor and predict fall events continuously while maintaining low computational demands.

The optimal threshold values for the algorithms were established using the Receiver Operating Characteristic (ROC) curve. Evaluate specificity and sensitivity values; to minimize the distance between the curve and the point (0,1), the appropriate thresholds are selected from the curve. Two methods for determining threshold values for the algorithms were explored.

$$SVM(n) = \sqrt{(a_x(n)^2 + a_y(n)^2 + a_z(n)^2)} \quad (1)$$

The Euler angle (THETA) represents the angle between the direction of gravitational field of Earth and y depicts the vertical axis of the device. This is calculated using the acceleration values $a_x(n)$, $a_y(n)$ and $a_z(n)$ for the n-th sample along the respective sensor axes. The Atan2 function, which computes the arctangent and yields results within the range of $[-\pi, \pi]$, is used for this calculation.

$$Theta(n) = \text{atan2} \left(\frac{\sqrt{(a_x(n)^2 + a_z(n)^2)}}{a_y(n)} \right) \quad (2)$$

As detailed in (Razum et al., 2018), this algorithm is used for fall detection. For the THETA angle and the SVM algorithm, we use a two-step procedure. Initially, we track the THETA angle over a 0.5-second period. If the THETA value exceeds a set threshold, we then evaluate the SVM values within the same timeframe and compare them to the SVM feature threshold. A fall is recognized if the SVM value exceeds its specified threshold within this timeframe. We have assumed here a fall results in a substantial change in body angle, which is then accompanied by a sudden increase in the SVM feature. In the alternative method that involves both SVM and THETA, the procedure is slightly altered. First, the SVM values for each sample are checked against the SVM threshold. If this threshold is exceeded, the following samples are then analysed for the THETA feature. The rationale is to first detect the initial ground contact and then observe the subsequent change in body orientation. If the THETA angle exceeds its set threshold, an

additional condition is applied to minimize false positives. Specifically, the average SVM value within a 1-second period surrounding the sample where the THETA angle surpasses the threshold is computed. If this average SVM value falls below a specified limit, it suggests that the individual is stationary, confirming fall detection.

$$threshold = \sqrt{(I_1^2 + I_2^2 + I_3^2)} \quad (3)$$

Additionally, in this the algorithm has been enhanced to include data from infrared sensors to improve fall detection and account for various scenarios. Threshold values are determined from infrared sensor readings, labelled as I1, I2, and I3. A fuzzy logic algorithm is then used for further distinguishing between falls and non-falls.

Table 2. Heuristic rules for case 1.

Label 1	Label 2	Category
0	0	Not Fall
0	1	Fall
1	0	Fall
1	1	Fall

Table 3. Heuristic rules for case 2.

Label1	Label2	Category
0	0	Not Fall
0	1	Not Fall
1	0	Not Fall
1	1	Fall

Two labels are created: label1 is based on IMU sensor threshold values and label2 is based on infrared sensor threshold values. If both label1 and label2 are 0, the event is considered not a fall; otherwise, it is classified as a fall. Table 2 and table 3 illustrate the above rules.

3.4 For image-based data

Firstly, a Convolutional neural network (CNN) model specifically for image classification task was created. The Convolutional Layer uses many filters on the image that is inputted or the output from preceding layers to identify features such as patterns, textures, and edges. Each filter in the Convolutional Layer processes the input to generate a feature map, which highlights different characteristics of the image. The Activation Layer adds non-linearity to the network. Rectified linear unit is a function which enables the network to capture and model complex relationships. To prevent overfitting, we use pooling layer which decreases computational demands. Following multiple convolutional and pooling layers, Fully Connected Layers are employed to perform reasoning at a higher level, where every neuron is linked to all neurons in the preceding layer. This layer

consolidates the extracted features to produce the final predictions. To avoid overfitting, a Dropout Layer is commonly employed, which randomly deactivates a portion of neurons during training to encourage the network to develop more generalizable features. The Output Layer produces the final prediction by applying activation functions, such as SoftMax for handling multiple classes or Sigmoid for binary outcomes.

In this work, CNN has been successfully applied to analyse images to identify whether a person has experienced a fall. Data pre-processing includes normalizing pixel values to a range of 0 to 1 by dividing by 255, and applying augmentation methods like rotation, flipping, and zooming to enhance the variety within the dataset. Accurate labelling of images, such as "fall" and "no fall," is crucial for effective model training. The CNN starts with convolutional layers, where filters move across the input data to identify features like edges, textures, and patterns. The convolution operations are executed and activation functions like ReLU are applied to incorporate non-linearity into the model. Subsequently, pooling layers are employed to decrease spatial dimensions while preserving key features. The feature maps are converted into a 1D vector and fed into fully connected layers for further processing. During the training of the CNN, a loss function like categorical cross-entropy for multi-class tasks, is utilized along with an optimizer Adam to improve accuracy. Training involves running the model for multiple epochs with an appropriate batch size, while a validation set is employed to track performance and fine-tune hyperparameters. The model generates a final feature vector, which is then processed through a softmax activation function to yield probability scores for various classes, such as "fall" or "no fall." Figure 3 depicts the architecture of convolutional neural network.

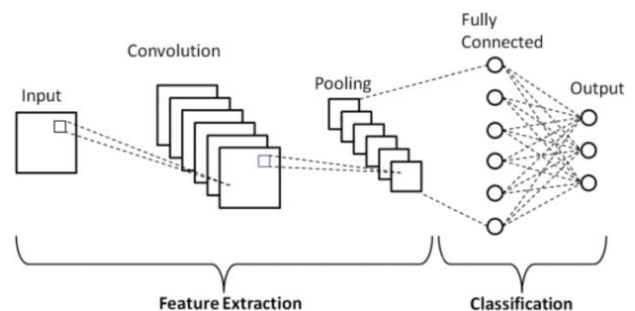


Figure 3. Architecture of Convolution Neural Network

3.5 YOLO (You only look once)

YOLO is a single shot detector object detection algorithm. It is used in wide number of applications including video and image analysis. Here, we use YOLO for image analysis task to identify the fall incidences. We begin by setting up the YOLO network to partition each input image into cells. Cell is tasked with prediction of bounding box and probabilities for objects which have centres inside the cell area. Anchor boxes are introduced

to enhance the accuracy of bounding box predictions. The network's design includes multiple convolutional layers that extract advanced features from the images, ultimately leading to detection layers that generate bounding boxes, confidence scores, and class probabilities. The model was trained on a labelled dataset with annotated falls and non-falls. A composite loss function, integrating localization loss, confidence loss, and classification loss, was employed.

The network was then fine-tuned through backpropagation with optimizers such as Adam or SGD, adjusting weights across several epochs to reduce the loss. We also used data augmentation methods during training to improve the robustness of the data once the image is segmented into a grid, algorithm estimates bounding boxes and class probabilities for objects present in each grid section. In this paper, different variants of the YOLO algorithms are analysed for identifying the best performing model. YOLOv1, YOLOV3 and YOLOv7 models have been used for fall identification.

4. PERFORMANCE EVALUATION

This study employs a set of performance metrics to assess the performance of fall detection techniques as discussed in section 3. The evaluation metrics offer critical insights about the model's effectiveness. Furthermore, sensitivity, or recall, measures how well the model identifies actual positive cases, such as falls, amongst all the instances that are true positive. In contrast, specificity assesses the model's capability to accurately recognize negative instances, such as non-falls, out of all the true negative cases. Ultimately, the F1-score uses both precision and recall for effectively evaluating the model, particularly in scenarios with uneven distributions of positive and negative cases. Utilizing these metrics aims to provide a thorough insight into the advantages and limitations of each fall detection approach being evaluated.

4.1 Performance evaluation of machine learning methods

We applied selected six machine learning algorithms to the UP-Fall Dataset

Gaussian Naïve Bayes, Extreme Gradient Boosting (XGB), Decision Tree (DT), Light Gradient Boosting Machine (LightGBM), Support Vector machine (SVM) and K-means are applied for identification of fall incidents. Table 4 depicts a comparative analysis among these algorithms in terms of four evaluation parameters viz. precision, recall, F1 score and accuracy. As it is evident from the corresponding values of precision, recall, F1 score and accuracy that the decision tree achieves highest values among all six algorithms in terms of precision, recall, F1 score and accuracy. Hence, it can be concluded that the decision tree turned out to be the best method for distinguishing fall and non-fall activities in this case.

Table 4. Comparative Analysis of Machine learning Algorithms.

Algorithm	Precision	Recall	F1-score	Accuracy
Gaussian NB	81	70	65	82
XGB	97	96	97	97
Decision Tree	99	98	99	98
LightGBM	97	97	98	66
SVM	92	92	91	92
K-Means	96	95	67	94

4.2 Performance evaluation of threshold-based methods

The two algorithms were assessed, then confusion matrices were produced. Performance metrics were derived from these matrices, and for both the processes ROC curve had been generated as shown in Figure 4 and Figure 5. The curves revealed that the Theta & SVM method outperformed the others. The accuracy achieved with the fuzzy characteristics applied to the Theta & SVM algorithm was 54%, whereas applying fuzzy characteristics to the SVM & Theta algorithm resulted in an accuracy of 61%.

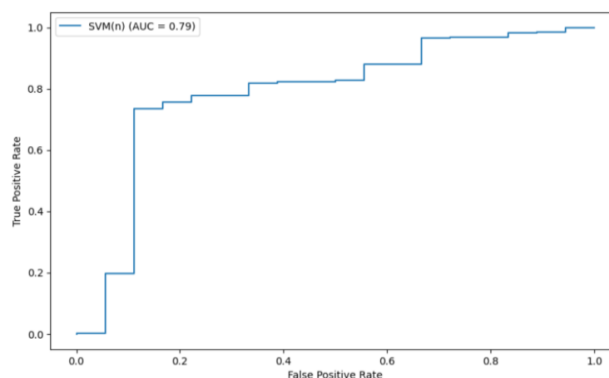


Figure 4. ROC curve for SVM (n)

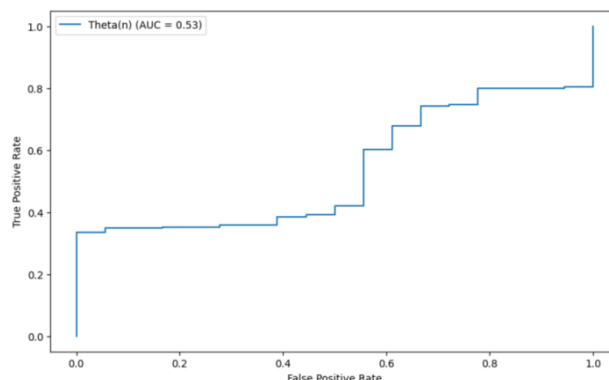


Figure 5. ROC curve for Theta (n)

4.2 For image-based data

By leveraging the UR fall dataset, different variants of YOLO models have been applied. Three variants of YOLO models have been used for fall identification. Among these, YOLOv7 is the recent among the object detection models. The improvements proposed in the

model YOLOv7 contributed towards more accurate results for fall detection. When compared with YOLOv1 and YOLOv3, YOLOv7 outperforms the first two models in terms of accuracy. The findings depicted in Table 5 indicate that for detecting falls in real time YOLOv7 proved to be the most effective model.

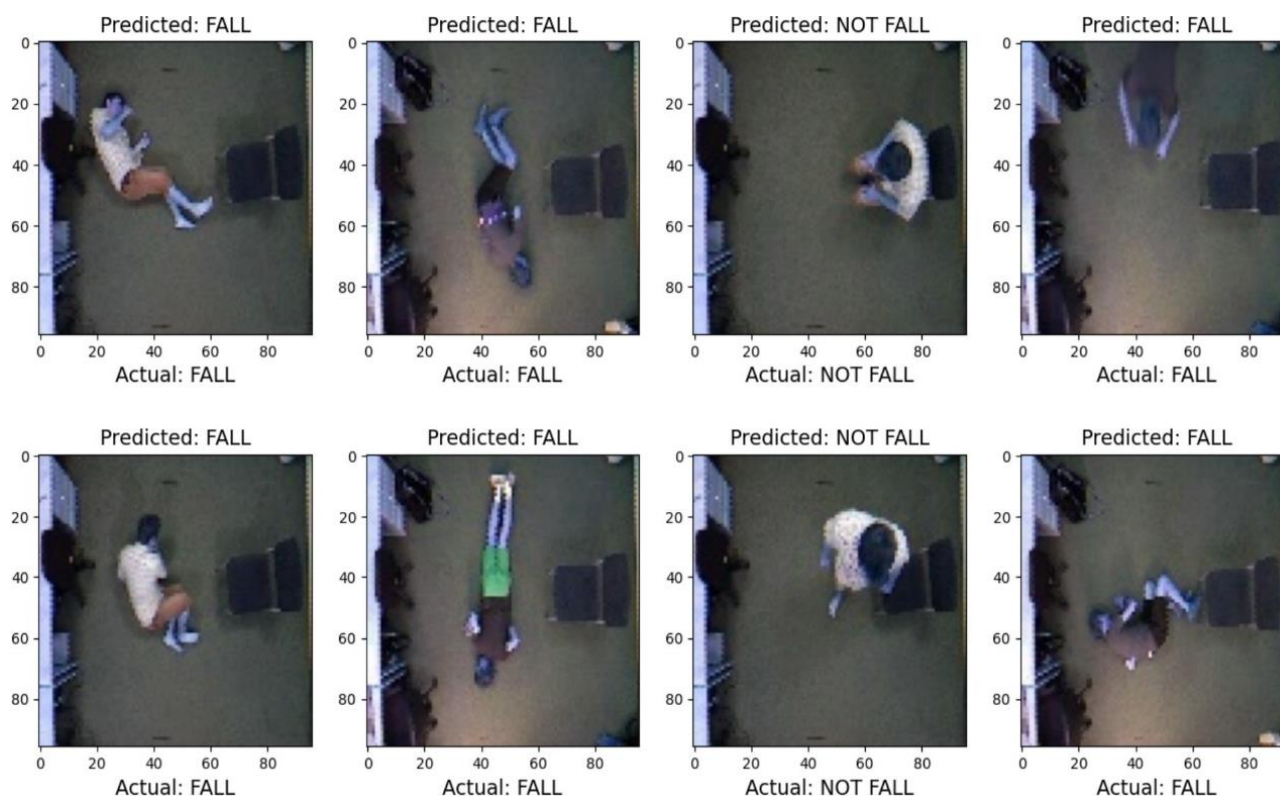


Figure 6. Predicted labels using Convolutional Neural Network

Table 5. Performance of YOLO Model.

Model used	Recall	Precision	F1-Score
YOLOv1	96.1	94.2	93
YOLOv3	93	95	94.1
YOLOv7	91	94	97

This figure displays a comparison between the labels predicted by the CNN and the true labels for a set of images. The predicted labels reflect the CNN's classification output, whereas the actual labels correspond to the ground truth annotations for those images.

This comparison is essential for assessing the model's effectiveness. A grid of images is shown in Figure 6, with each image labelled with both the predicted and true labels. This visual format aids in detecting trends in the model's predictions, such as frequently misclassified classes or specific features contributing to errors.

5. CONCLUSION

Quick and accurate fall detection is important for providing timely medical support to the deceased. This paper presents a detailed comparison of fall detection system for elderly people developed by using sensor data and vision-based dataset. For sensor-based data, two approaches were followed. First, selected machine learning models were applied to determine the most effective ones for fall identification. Secondly, we implemented threshold-based algorithms in fall detection to boost prediction accuracy. These algorithms apply specific thresholds to make decisions based on set criteria, thereby enhancing the accuracy of fall identification. Machine learning techniques applied in this paper mark a notable improvement over previous method. By employing advanced methods and performing thorough empirical assessments, these approaches have the potential to enhance fall detection and activity monitoring systems. The YOLO family algorithms have been applied and have given significant results for efficient fall detection. All YOLO algorithms have proved well but YOLOv7 has surpassed others in terms of fall detection.

6. FUTURE DIRECTIONS

In future, there is potential to implement real-time monitoring and instant notifications to caregivers or emergency responders. As fall detection systems become more widespread, safeguarding user privacy and addressing ethical issues will be crucial. Future developments could focus on privacy-preserving methods for data handling, storage and analysis. New approaches can be proposed for securing informed consent and preserving user autonomy. Beyond fall detection, we aim to incorporate effective alert systems to quickly notify caregivers or emergency services.

Additionally, integrating with video surveillance systems could enhance sensor-based detection by offering visual verification of falls and providing contextual information, such as the environment and incident severity.

Nevertheless, these solutions encounter several obstacles, such as differentiating falls from everyday activities, accommodating variations in individual movement patterns, and considering environmental influences on sensor data. Furthermore, ensuring that these systems are practical and acceptable to users is crucial for their broad adoption and effectiveness in real-world scenarios.

References:

- Alam, E., Sufian, A., Dutta, P., & Leo, M. (2022). Vision-based human fall detection systems using deep learning: A review. *Computers in Biology and Medicine*, 146, 105626. <https://doi.org/10.1016/j.combiomed.2022.105626>
- Althobaiti, T., Katsigiannis, S., & Ramzan, N. (2020). Triaxial accelerometer-based falls and activities of daily life detection using machine learning. *Sensors*, 20(13), 3777. <https://doi.org/10.3390/s20133777>
- Anishchenko, L. (2018, May). Machine learning in video surveillance for fall detection. In *2018 Ural Symposium on Biomedical Engineering, Radio Electronics and Information Technology* (pp. 99-102). IEEE. <https://doi.org/10.1109/USBREIT.2018.8384560>
- Durgun, Y. (2023). Fall detection systems supported by TinyML and accelerometer sensors: An approach for ensuring the safety and quality of life of the elderly. *International Scientific and Vocational Studies Journal*, 7(1), 55-61. <https://doi.org/10.47897/bilmes.1299289>
- Galvão, Y. M., Ferreira, J., Albuquerque, V. A., Barros, P., & Fernandes, B. J. (2021). A multimodal approach using deep learning for fall detection. *Expert Systems with Applications*, 168, 114226. <https://doi.org/10.1016/j.eswa.2020.114226>
- Gutiérrez, J., Rodríguez, V., & Martín, S. (2021). Comprehensive review of vision-based fall detection systems. *Sensors*, 21(3), 947. <https://doi.org/10.3390/s21030947>
- Huerta, E. B., Juárez, E. M., Caporal, R. M., & Urbina, E. V. (2024). Fuzzy logic and machine learning algorithms for detection and classification of falls and activities of daily living. *International Journal of Combinatorial Optimization Problems and Informatics*, 15(4), 42. <https://doi.org/10.61467/2007.1558.2024.v15i4.497>
- Hussain, Z., Sheng, Q. Z., & Zhang, W. E. (2020). A review and categorization of techniques on device-free human activity recognition. *Journal of Network and Computer Applications*, 167, 102738. <https://doi.org/10.1016/j.jnca.2020.102738>
- Iqbal, U., Milan, A., & Gall, J. (2017). Posetrack: Joint multi-person pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2011-2020). <https://doi.org/10.48550/arXiv.1611.07727>
- Javed, A. R., Faheem, R., Asim, M., Baker, T., & Beg, M. O. (2021). A smartphone sensors-based personalized human activity recognition system for sustainable smart cities. *Sustainable Cities and Society*, 71, 102970. <https://doi.org/10.1016/j.scs.2021.102970>
- Jiang, Z., Al-Qaness, M. A., Dalal, A. A., Ewess, A. A., Abd Elaziz, M., Dahou, A., & Helmi, A. M. (2024). Fall detection systems for Internet of Medical Things based on wearable sensors: A review. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2024.3421336>
- Liang, H., & Usaha, W. (2017, September). Fall detection using lifting wavelet transform and support vector machine. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)* (pp. 877-883). IEEE. <https://doi.org/10.15439/2017F405>
- Luna-Perejón, F., Muñoz-Saavedra, L., Civit-Masot, J., Civit, A., & Domínguez-Morales, M. (2021). AnkFall—Falls, falling risks and daily-life activities dataset with an ankle-placed accelerometer and training using recurrent neural networks. *Sensors*, 21(5), 1889. <https://doi.org/10.3390/s21051889>
- Mobsite, S., Alaoui, N., Boulmalf, M., & Ghogho, M. (2023). Semantic segmentation-based system for fall detection and post-fall posture classification. *Engineering Applications of Artificial Intelligence*, 117, 105616. <https://doi.org/10.1016/j.engappai.2022.105616>

- Musci, M., De Martini, D., Blago, N., Facchinetti, T., & Piastra, M. (2020). Online fall detection using recurrent neural networks on smart wearable devices. *IEEE Transactions on Emerging Topics in Computing*, 9(3), 1276-1289. <https://doi.org/10.1109/TETC.2020.3027454>
- Peng, Y., Peng, J., Li, J., Yan, P., & Hu, B. (2019). Design and development of the fall detection system based on point cloud. *Procedia Computer Science*, 147, 271-275. <https://doi.org/10.1016/j.procs.2019.01.253>
- Razum, D., Seketa, G., Vugrin, J., & Lackovic, I. (2018). Optimal threshold selection for threshold-based fall detection algorithms with multiple features. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics* (pp. 1513-1516). IEEE. <https://doi.org/10.23919/MIPRO.2018.8400272>
- Sabry, F., Eltaras, T., Labda, W., Alzoubi, K., & Malluhi, Q. (2022). Machine learning for healthcare wearable devices: The big picture. *Journal of Healthcare Engineering*, 2022(1), 4653923. <https://doi.org/10.1155/2022/4653923>
- Van Thanh, P., Tran, D. T., Nguyen, D. C., Duc Anh, N., Nhu Dinh, D., El-Rabaie, S., & Sandrasegaran, K. (2019). Development of a real-time, simple and high-accuracy fall detection system for elderly using 3-DOF accelerometers. *Arabian Journal for Science and Engineering*, 44, 3329-3342. <https://doi.org/10.1007/s13369-018-3496-4>
- Vasoya, H., Bhattasana, H., & Mishra, R. G. (2023, May). A review of elderly fall detection systems using artificial intelligence. In *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 541-546). IEEE. <https://doi.org/10.1109/ICICCS56967.2023.10142753>
- Zhao, C., Zeng, W., Hu, D., & Liu, H. (2021). Robust heart rate monitoring by a single wrist-worn accelerometer based on signal decomposition. *IEEE Sensors Journal*, 21(14), 15962-15971. <https://doi.org/10.1109/JSEN.2021.3075109>

Indu Chawla

Jaypee Institute of Information
technology
Noida,
India
indu.chawla@jiit.ac.in
ORCID 0000-0002-5333-9185

Archana Purwar

Jaypee Institute of Information
technology
Noida, India
archana.purwar@jiit.ac.in
ORCID 0000-0001-8824-4859

Khushi Chauhan

Accenture India,
Bangalore
India
chauhankhushi2018@gmail.com
ORCID 0009-0003-5746-7605

Vandita Chauhan

Oracle India,
Pune
India
vanditachauhan171@gmail.com
ORCID 0009-0008-3098-8028
