Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140

Volume 14 Issue 03s, 2025

Real-Time Fall Detection Using MLP, OpenCV, and IoT Integration: Development of the FallNex Smart Application

[1]Aastha Choudhari, [2]Diya Chilmulwar, [3]Dr. Pallavi Wankhede, [4]Saurabh Wankhede, [5]Dishant Kewat

[1,2,3,4,5] Dept. of Computer Engineering, St. Vincent Pallotti College of Engineering & Technology, Nagpur, India
Email: [1]choudhariaastha185@gmail.com, [2]diyachilmulwar@gmail.com, [3]pwankhede@stvincentngp.edu.in, [4]swankhede228@gmail.com, [5]dishantkewat2510@gmail.com

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

Fall detection, real-time monitoring, machine learning, OpenCV, IoT

Abstract

Falls among older adults pose a serious health risk, as delayed assistance can lead to severe injuries or fatal outcomes. To address this challenge, we propose FallNex, a real-time fall detection system that integrates a machine learning model, lightweight visual monitoring, and IoT-based hardware. A Multi-Layer Perceptron (MLP) model performs the primary fall prediction using motion data collected from wearable sensors. To minimize false alarms, a simple OpenCV-based monitoring module evaluates frame differences from a camera feed to verify whether significant activity occurred at the moment of a suspected fall. The proposed hardware unit communicates with the FallNex mobile application to deliver instant alerts, maintain fall logs, and provide post-fall guidance through an integrated chatbot. Experimental results demonstrate that FallNex achieves high accuracy with reduced false positives while maintaining real-time responsiveness, making it suitable for home-based and assisted-living environments.

Introduction

Falls are among the most pressing health risks faced by older adults, contributing to an estimated 684,000 deaths globally each year [1]. Beyond the immediate physical injuries such as fractures or head trauma, fall incidents often lead to long-term psychological effects including fear, mobility loss, and reduced independence. These consequences negatively affect the quality of life of elderly individuals and increase the demand for continuous monitoring by caregivers. With the growth of the global aging population, there is a rising need for automated fall detection systems that can provide timely alerts without requiring constant human supervision.

Traditional fall detection approaches typically rely on wearable sensors, camera-based solutions, or Internet-of-Things (IoT) infrastructures. While wearable devices are convenient and lightweight, they often struggle with comfort issues, inconsistent user compliance, and false alarms caused by rapid but harmless movements. Vision-based systems offer rich contextual understanding but face challenges such as privacy concerns, sensitivity to lighting conditions, and computational complexity. IoT-based systems improve communication and emergency response, yet require stable connectivity and effective integration of data from different sources

To address these challenges, we propose FallNex, a hybrid fall detection platform that combines the strengths of sensor data, machine learning classification, lightweight camera verification, and IoT communication. By integrating multiple confirmation layers, FallNex aims to reduce false alarms, improve detection reliability, and ensure

rapid emergency response while maintaining low computational cost.

Existing Systems

Wearable Sensor-Based Systems

Wearable devices equipped with accelerometers and gyroscopes are widely used for fall detection due to their low cost and portability. These devices continuously capture motion data, enabling recognition of sudden accelerations or posture changes. Earlier systems used simple threshold-based rules, where a fall is detected if acceleration exceeds a predefined limit. However, thresholding suffers from high sensitivity, generating false alarms during fast but normal activities such as running, jumping, or sitting abruptly.

To overcome these limitations, researchers introduced machine learning models such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Random Forests to classify fall events based on patterns in motion signals [2]. Deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) further improved recognition by learning temporal dependencies in sensor data. Despite improvements, wearable systems remain vulnerable to false positives and rely heavily on consistent user adherence

Vision-Based Systems

Camera-based solutions analyze human posture, motion trajectory, and environmental context to identify fall incidents. Early methods relied on handcrafted features such as silhouettes, motion energy images, and optical flow. Modern systems employ advanced deep learning models including CNN-LSTM networks, which extract spatial and temporal patterns from video sequences [7]. In addition, pose estimation frameworks like OpenPose generate skeletal keypoints to infer body orientation while preserving some privacy [8].

Although these methods offer high accuracy, they require significant computational resources and are sensitive to variations in lighting, camera angle, and occlusion. More importantly, continuous video recording raises privacy concerns, limiting deployment in homes or care institutions

IoT and Hybrid Systems

IoT-based systems combine sensors, communication modules, and cloud servers to enable real-time monitoring and remote assistance [9]. These frameworks offer robust connectivity and centralized data management. Hybrid systems that integrate sensor readings with vision output can further enhance detection reliability. However, they face deployment challenges such as network delays, data privacy issues, and increased maintenance costs. A balanced, lightweight hybrid solution is therefore desirable.

Methodology

Dataset and Preprocessing

This work utilizes a publicly available Kaggle dataset containing labeled fall and non-fall activities. The dataset includes sensor readings representing various movement intensities and orientations. Preprocessing steps included noise filtering, normalization, and feature scaling to ensure uniformity across samples. The dataset was divided into training and testing subsets to evaluate generalization performance [3], [5]. Understanding data variability is essential because fall patterns differ significantly from normal activities, making preprocessing a critical step in achieving accurate classification.

Machine Learning Model (MLP)

A Multi-Layer Perceptron (MLP) neural network was selected due to its efficiency, low computational cost, and strong performance on structured sensor data. The network includes an input layer corresponding to sensor features, multiple hidden layers with non-linear activation functions, and an output layer representing fall or non-fall classification. The model was trained using backpropagation with stochastic gradient descent, optimizing loss functions such as cross-entropy. Evaluation metrics—including accuracy, precision, recall, and F1-score—were used to assess reliability. To improve robustness, ensemble learning methods such as LightGBM were also explored in the broader design context, as they have shown accuracy levels exceeding 98% in similar sensor-based applications [7], [10].].

OpenCV-Based Cross Verification

OpenCV is incorporated as a lightweight visual monitoring module rather than a classification mechanism. When the MLP or hardware component detects a potential fall, the OpenCV module evaluates the corresponding video frames using simple frame-difference analysis. This approach detects noticeable scene changes without requiring complex algorithms such as pose estimation or object detection. By confirming whether visible motion aligns with the suspected fall event, the module helps reduce false alarms caused by abrupt but harmless movements in sensor data. This ensures accuracy while preserving privacy and computational efficiency.

Hardware Device Integration

The hardware subsystem consists of a compact wearable device equipped with a 3-axis accelerometer and gyroscope. The

microcontroller preprocesses raw sensor signals by filtering noise and computing derived metrics such as resultant acceleration. These processed values are transmitted wirelessly to the mobile application for further evaluation. The hardware module ensures continuous monitoring even in the absence of camera feed, making the system reliable in both indoor and outdoor scenarios [1], [2].

FallNex Mobile Application and Chatbot

The FallNex mobile application serves as the primary interface for users and caregivers. It displays real-time activity status, logs historical fall events, and sends emergency alerts when necessary. A built-in HealthBot chatbot offers immediate guidance following a fall, including first-aid suggestions and prompts to call emergency contacts. The application ensures seamless communication between the hardware, machine learning module, and cloud services

System Architecture

Data Acquisition

The wearable device records accelerometer (Ax, Ay, Az) and gyroscope (Gx, Gy, Gz) data. Resultant acceleration is calculated as:

$$A = sqrt(A_x^2 + A_y^2 + A_z^2) \quad (1)$$

A sudden spike in A typically indicates abnormal motion consistent with a fall

Data Processing and Fall Detection

The microcontroller filters noise and forwards sensor values to the MLP for fall classification. Tilt angle θ is also computed to evaluate posture. A fall is detected when

$$A > Athreshold \quad and\ \theta > \theta threshold$$

Cloud Integration and Alerting

Firestore serves as a real-time communication bridge between hardware and the mobile application. Upon detecting a fall, user details, timestamps, and location data are uploaded, and emergency alerts are dispatched instantly via FCM or SMS.

Visual Monitoring Using OpenCV

A lightweight OpenCV module performs frame-difference analysis on the camera feed surrounding the fall event. If a substantial visual change is detected, the fall alert confidence increases. This module does not classify falls; it simply validates motion corresponding to sensor-based detection.

Application and Backend Communication

The FallNex mobile application provides a user-friendly interface for monitoring live data, viewing fall history, and responding to alerts. A FastAPI backend ensures seamless interaction between the database, machine learning model, and front-end services.

HealthBot Integration

The integrated HealthBot guides users with actionable prompts such as “CALL” or “NEED HELP.” If the user does not respond within a predefined interval, the system automatically contacts the assigned emergency number.

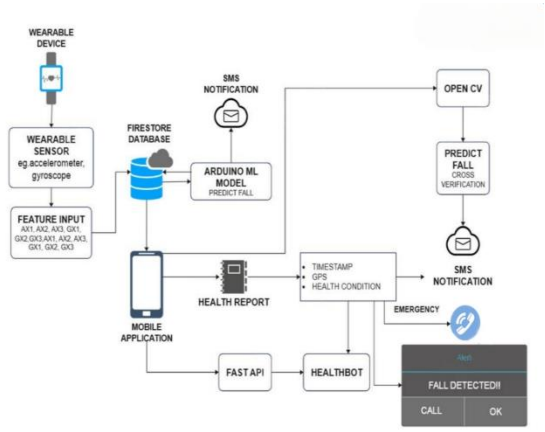

Fig. 1. System Architecture

Results

Model Accuracy and Loss Analysis

The training and validation curves in Fig. 2 illustrate the learning behavior of the MLP model over multiple epochs. The accuracy graph shows a consistent upward trend, where both training and validation accuracy steadily improve before stabilizing. This indicates that the model successfully learns to differentiate between fall and non-fall activities without significant overfitting.

Similarly, the loss plot shows a gradual decline for both training and validation sets, confirming effective convergence of the model. The closeness of the two curves reflects strong generalization, meaning the model performs consistently on unseen data.

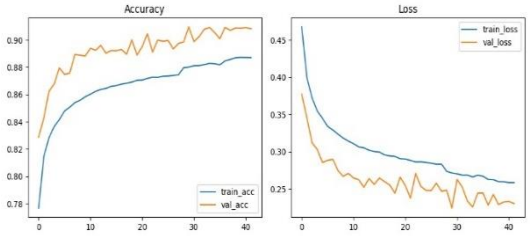

Fig. 2. Model Accuracy Graph

Dataset Distribution and Statistical Visualization

The dataset visualizations provide insight into activity patterns, distribution of motion intensities, and class balance. The top histogram displays the distribution of fall durations, revealing natural variation and realistic activity complexity. The following bar graphs show the

number of samples per activity and average signal magnitude for each activity label.
These visualizations confirm that the dataset contains diverse movements with sufficient representation across classes. This diversity contributes to improved model learning and reduces bias in the classifier.

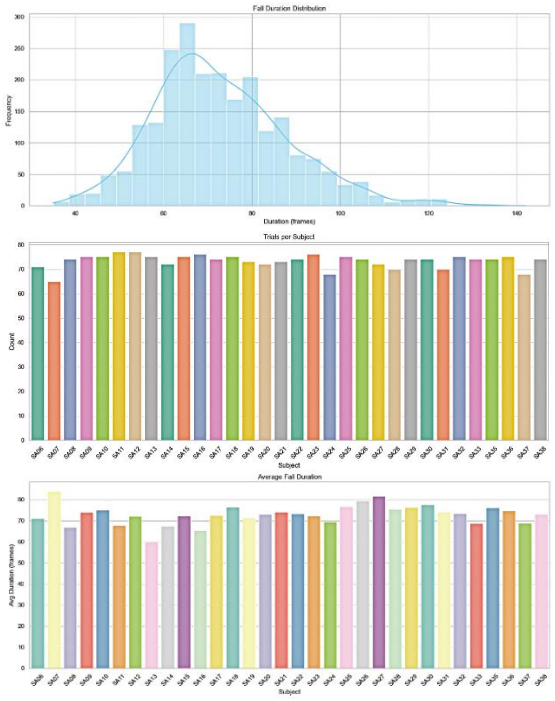

Fig. 3. Dataset Visualization

Hardware Sensor Data Visualization
Fig. 4 presents real-time accelerometer and gyroscope readings collected from the wearable device during various movements. The accelerometer plot displays relatively stable readings during normal activities, while the gyroscope plot exhibits more variability due to rotational movements.
During fall events, sudden spikes and irregular patterns are clearly visible, differentiating abrupt falls from routine movements. These observations validate the reliability of the hardware sensors and support their suitability for real-time fall detection..

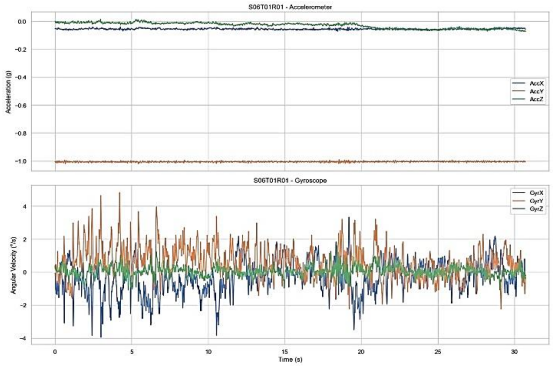

Fig. 4. Hardware Data Visualization

Clustering Analysis of Motion Features
Clustering results shown in Fig. 5 depict how accelerometer magnitude and gyroscope magnitude can be grouped to identify distinct motion patterns. The plot reveals well-separated clusters, representing normal activities and fall events. Cluster centroids are marked with red "X", and the detected fall impact points are highlighted with star markers.
The clear separation among clusters demonstrates that sensor features are highly discriminative, enabling the machine learning model to accurately classify fall events.

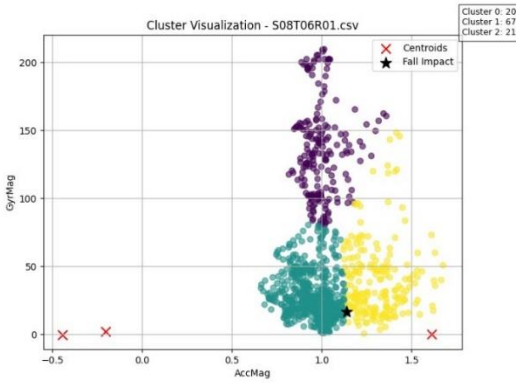

Fig. 5. Data Clustering

LightGBM Classification Performance
To benchmark performance, a LightGBM model was trained and evaluated on the same dataset. The classification report shown in Fig. 6 indicates:

Precision: 0.95–0.96

Recall: 0.94–0.97

F1-score: 0.95–0.96

Overall Accuracy: 0.96

These results demonstrate that LightGBM performs exceptionally well on sensor-based fall detection and aligns with modern research that reports similar levels of accuracy. The high precision and recall values indicate that misclassification is minimal, and the model is reliable for real-world deployment.

[LGB] Test classification report:

	precision	recall	f1-score	support
0	0.96	0.97	0.97	46387
1	0.95	0.94	0.95	29360
accuracy			0.96	75747
macro avg	0.96	0.96	0.96	75747
weighted avg	0.96	0.96	0.96	75747

Fig. 6. Precision Table

Conclusion

FallNex presents a comprehensive real-time fall detection system that integrates machine learning, hardware-based sensing, and lightweight visual monitoring. The MLP classifier performs primary detection, while OpenCV provides secondary motion verification to reduce false positives. The IoT-enabled mobile application ensures instant alert delivery and supports users through an integrated HealthBot. Experimental results demonstrate that the system is reliable, efficient, and applicable for home-based elderly care. Future improvements include expanding datasets, integrating additional deep learning models, and conducting real-world field testing for enhanced robustness.

References

World Health Organization, “Falls: Key facts,” 2023.

R. Igual, C. Medrano, and I. Plaza, “Challenges, issues and trends in fall detection systems,” BioMedical Engineering OnLine, vol. 12, no. 66, pp. 1–17, 2013.

K. Vavoulas et al., “The MobiFall dataset: Evaluation of fall detection algorithms,” MobiHealth, 2013.

C. Rougier et al., “Fall detection from motion history using video surveillance,” AINAW, 2007.

L. Martínez-Villaseñor et al., “UP-Fall Detection Dataset,” Sensors, 2019.

Y. Chen et al., “Vision-based fall event detection,” Pattern Recognition Letters, 2020.

Z. Cao et al., “OpenPose: Real-time multi-person pose estimation,” IEEE TPAMI, 2021.

A. Salimi et al., “Human fall detection using pose estimation,” Sensors, 2022.

Sunkara, S. P. (2025). Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems. International Journal of Electrical Engineering and Technology (IJEET), 16(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001

P. Wankhede et al., “A comprehensive review of fall detection approaches,” IJIRCCE, 2025.

Published research on LightGBM and MLP ensemble fall detection