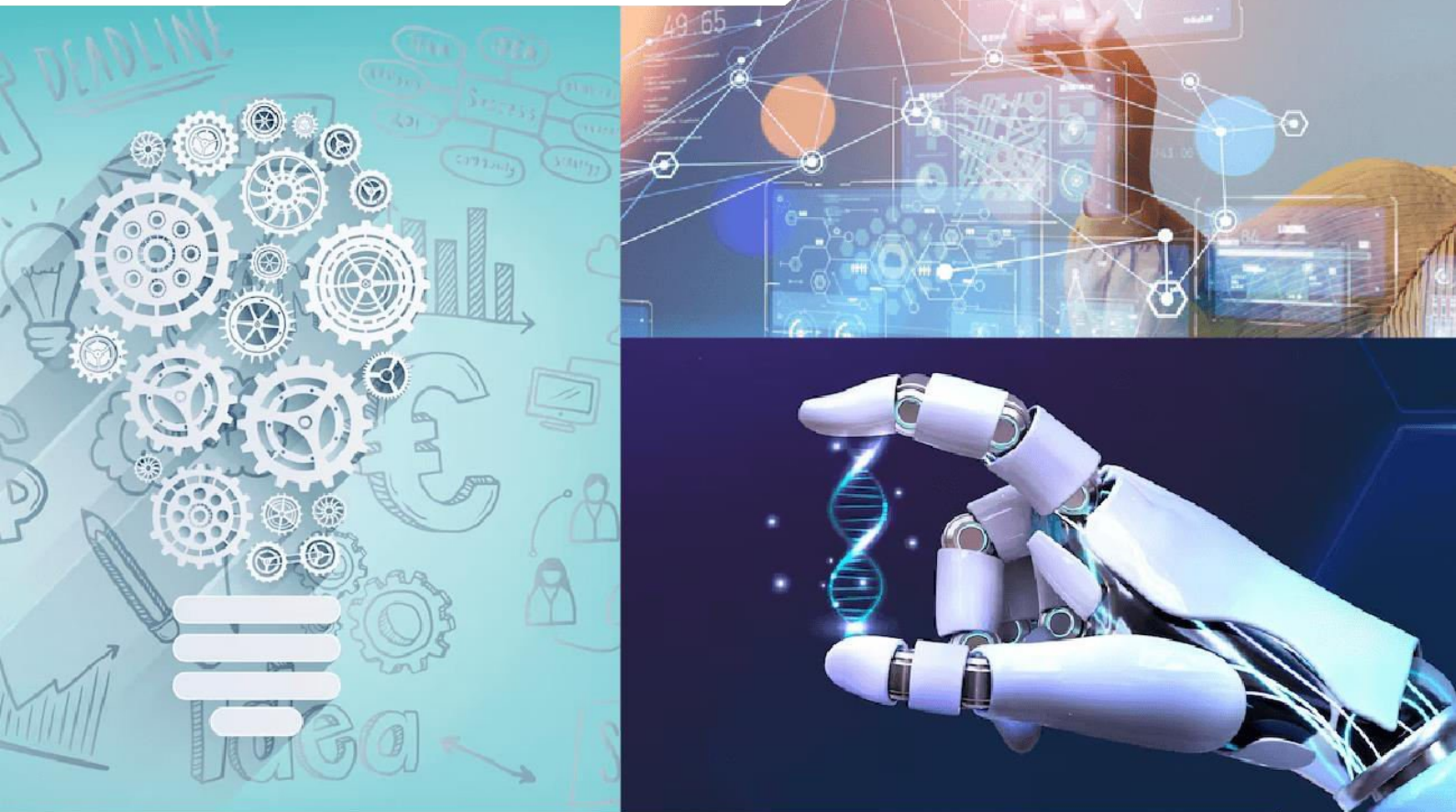




International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 2, March - April 2026

Impact Factor: 8.152



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



Human Fall Detection Using Temporal Pose Features and Random Forest Classification from Video Sequences

Ruchita Rajesh Joshi

Master of Computer Applications, CMR Institute of Technology, Bengaluru, India

ABSTRACT: Falls are a leading cause of injury among elderly individuals, making automated detection essential for safety monitoring. Existing systems depend on computationally heavy deep learning models or intrusive wearable sensors, limiting practical deployment. Several studies have explored pose estimation and Random Forest classifiers independently, but few combine temporal feature engineering and sliding window segmentation with lightweight machine learning for fall detection. The proposed framework utilizes MediaPipe pose estimation to extract 33 body landmarks from each video frame. Instead of single-frame analysis, the system models temporal motion patterns using sliding windows of 30 frames. From each window, 27 temporal features are computed. Group-wise video splitting and GroupKFold cross-validation prevent data leakage. Trained on the GMDCSA24 dataset, the model achieved 95.74% accuracy and supports real-time inference with alarms and automated email alerts.

KEYWORDS: Fall Detection, Temporal Feature Engineering, Mediapipe, Random Forest, Machine Learning, Elderly Safety

I. INTRODUCTION

Falls are one of the most serious health risks faced by elderly individuals living alone or in care facilities. According to the World Health Organization, falls are the second leading cause of accidental injury deaths worldwide, with adults above the age of 65 being the most vulnerable group [1]. A fall that goes undetected for even a short period can lead to severe physical injury, prolonged immobility, and in critical cases, death. This has motivated researchers to explore automated solutions using sensors, cameras, and machine learning that can detect falls quickly and trigger timely alerts without requiring constant human supervision [2], [3].

Over the past decade, researchers have proposed a wide range of fall detection approaches broadly grouped into three categories. The first category involves wearable sensor-based systems, where accelerometers and gyroscopes attached to the body capture motion signals to identify falls [4], [5], [6]. The second category includes ambient and radio frequency sensing methods that detect falls without physical contact, using technologies such as Wi-Fi channel state information [7]. The third and most widely explored category is vision-based fall detection, which uses cameras and computer vision algorithms to analyze human movement. Within vision-based systems, researchers have used convolutional neural networks [8], [9], deep learning pipelines combining object detection with recurrent models [10], pose estimation frameworks such as OpenPose and AlphaPose [11], [12], [13], two-stage classification methods [14], anomaly detection using graph convolutional networks [15], and IoT-integrated alert systems using MediaPipe [16], [17].

Despite strong results reported in many of these studies, several limitations remain that restrict their practical use in real-world deployment. Deep learning models such as 3D CNNs, Bi-LSTMs, and Vision Transformers achieve high accuracy but require large amounts of labeled training data [8], [10], [9], [3]. Wearable based systems depend heavily on user compliance, which is particularly unreliable for elderly individuals [4], [5], [6]. Many vision-based methods perform single-frame classification, which ignores the dynamic and temporal nature of a fall event [1], [11].

To address the identified limitations, this paper presents a vision-based human fall detection system built on temporal pose analysis. The proposed system is evaluated using the GMDCSA24 fall detection dataset. The system uses MediaPipe Pose to extract 33 skeletal body landmarks per frame, where each landmark carries x, y, z coordinates and a visibility score, forming a 132-dimensional representation per frame. Sliding window approach with a window size of 30 frames and a step size of 15, capturing approximately one second of continuous human movement per window. This temporal modeling is central to the proposed approach, as falls are inherently dynamic events. From each

window, 27 handcrafted temporal features are systematically extracted across six categories: statistical motion descriptors, inter-frame velocity, torso inclination angle, body aspect ratio, ground contact estimation, and movement jerk. A Random Forest classifier with 150 estimators is trained on these features. The system is deployed as a Streamlit web application called FALLGUARD AI, where users upload MP4 videos for fall detection and automated timestamped email alerts are triggered upon fall confirmation.

II. LITERATURE REVIEW

Human fall detection has become an important research problem, particularly for monitoring elderly people who live alone or in assisted care settings. This has pushed researchers to develop automated systems that can detect falls without requiring constant human supervision. Fall detection systems are broadly divided into three categories — wearable device-based, context-aware, and vision-based approaches. Wearable systems offer direct body motion capture, they depend heavily on user compliance, which is unreliable for elderly populations. Zafar and Zafar used a transformer model with self-attention on accelerometer and gyroscope data from the MobiAct dataset, achieving above 98% accuracy [4]. Liu et al. combined a coarse-fine CNN with a Gated Recurrent Unit to process IMU signals from multiple body locations, achieving 97.95% accuracy on the FallAID dataset [6]. The BlockTheFall system added blockchain security to wearable data and used K-means clustering to distinguish real falls from simulated ones, achieving 95.7% balanced accuracy [5]. Beyond wearables, Ji et al. proposed SiFall, which uses Wi-Fi channel state information and a self-supervised autoencoder to detect falls as signal reconstruction anomalies, achieving a reported 100% true positive rate [7]. Although these methods show strong results, vision-based systems have proven more practical as they require no body-worn hardware, a conclusion supported by comprehensive reviews of the field [3].

In contrast to sensor-based approaches, Vision-based deep learning methods have achieved impressive results by analyzing raw video frames directly. Nuñez-Marcos et al. used stacked optical flow images with a fine-tuned VGG-16 network, achieving a reported 100% sensitivity on the UR Fall dataset [8]. Alanazi et al. compressed 32 consecutive frames into four fused images and processed them through a four-stream 3D CNN, achieving 99.44% accuracy on the Le2i dataset [9]. For multi-camera environments, Shieh and Huang used background subtraction and Levenshtein distance matching to detect fall-like silhouette patterns, improving processing throughput by 2.1 times on DSP hardware [19]. Another study paired YOLO V4 with a bidirectional attention LSTM for person tracking and temporal classification, reaching 96.9% accuracy on the UR Fall dataset [10]. Moving away from supervised approaches, Galvão et al. trained an ST-GCN autoencoder purely on normal activities and flagged falls as reconstruction anomalies, achieving a reported 100% accuracy on UR-Fall [15]. Zaidi et al. similarly used tracking and sparse feature representations to isolate abnormal behavior in surveillance streams [20]. Despite strong performance, these methods are computationally heavy and require large labeled datasets, limiting deployment on everyday hardware.

As an alternative to computationally heavy CNN models, pose estimation-based methods have gained popularity as a middle ground between raw pixel analysis and wearable sensors. Mamchur et al. extracted 17 joint keypoints using PoseNet and classified fall sequences using an LSTM with a 50-frame sliding window, achieving 91.07% accuracy on combined benchmark datasets [1]. Chen et al. used OpenPose with two geometric rules — hip descent speed and body centerline angle — to detect falls without any learning model, achieving 97% accuracy on a custom dataset [11]. Beddiar et al. converted head-to-hip geometric vectors into macro-images and classified them using ResNet50 with an SVM, achieving 98.6% accuracy on the UR Fall dataset [21]. Ramirez et al. tested Random Forest, SVM, MLP, and KNN on AlphaPose skeletal coordinates, with Random Forest achieving 99.34% accuracy on the UP-FALL dataset [12]. Raza et al. compared multiple pose estimators and classifiers including ST-GCN and Vision Transformers, confirming that pose quality directly impacts classification accuracy [13]. Han et al. used a two-stage pipeline combining OpenPose with an SVM to first filter stable activities and then classify unstable movements, achieving 97.34% accuracy [14].

The reviewed literature shows that fall detection has advanced significantly across multiple sensing and learning paradigms. However, key gaps remain. Deep learning models achieve high accuracy but are computationally expensive and difficult to deploy on low-cost hardware. Wearable systems are impractical for users who cannot wear devices consistently. Pose-based methods using single frames or fixed geometric thresholds miss the dynamic temporal nature of falls. Studies using MediaPipe showed practical results but lacked structured temporal feature engineering across multiple frames. This paper addresses these gaps by proposing a lightweight system that combines MediaPipe pose estimation with 27 handcrafted temporal features extracted from 30-frame sliding windows and a Random Forest classifier.

III.METHODOLOGY

A. System Architecture Overview

The proposed system follows a sequential pipeline that takes video input and produces a binary fall or normal classification along with an automated alert. The overall architecture consists of five main stages. In the first stage, video frames are read one by one using OpenCV. In the second stage, MediaPipe Pose processes each frame and extracts 33 body landmarks. These landmarks are stored frame by frame and grouped into overlapping temporal windows in the third stage. In the fourth stage, 27 handcrafted temporal features are computed from each window and fed into a trained Random Forest classifier for prediction. In the final stage, predictions from consecutive windows are smoothed using a majority voting mechanism, and a confirmed fall triggers an audio alarm and an automated email alert. The complete pipeline from video input to alert generation is illustrated in Fig. 1.

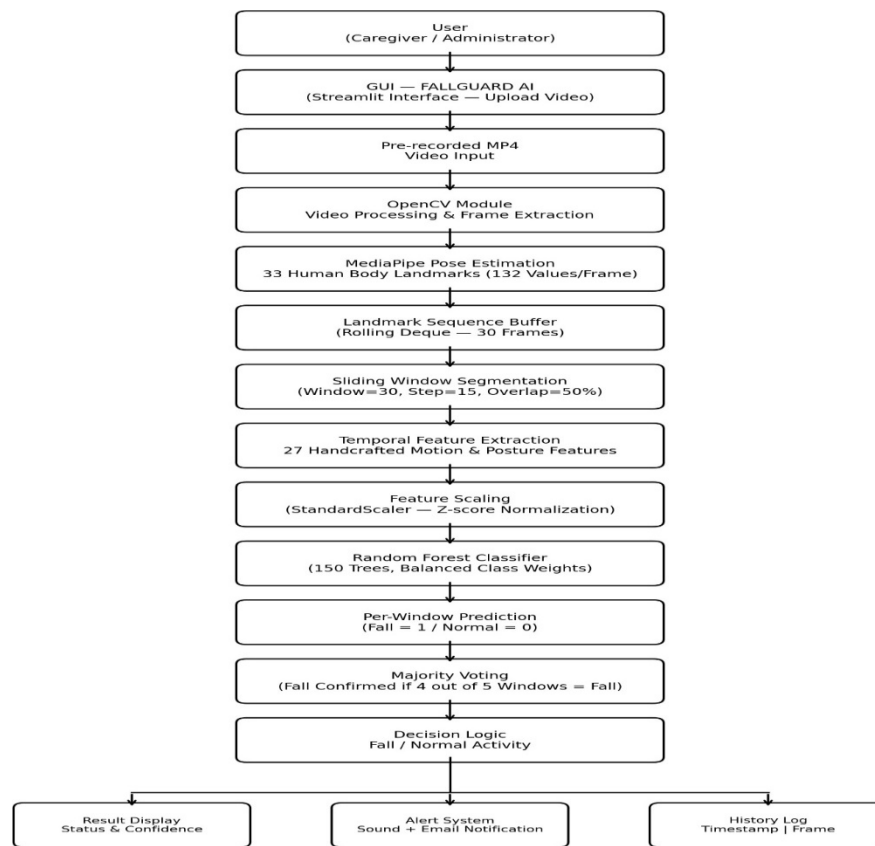


Fig. 1. System Architecture of the Proposed Fall Detection Model

B. Dataset Description

The proposed system in this paper is trained and evaluated on GMDCSA24. It contains videos from four subjects performing two categories of activities — falls and Activities of Daily Living. The fall category includes 81 video sequences covering different types of falls, while the ADL category contains 79 video sequences of complex daily activities such as sleeping, doing push-ups.

C. Pose Landmark Extraction using MediaPipe

The first stage of the proposed system involves capturing video frames from video files. OpenCV is used to read the frames sequentially. Each video frame is processed using MediaPipe Pose, a lightweight pose estimation library that detects 33 body landmarks in real time without requiring a GPU. Each landmark consists of four values: x coordinate, y coordinate, z depth coordinate, and visibility score. Thus, each frame is represented as a 132-dimensional feature vector. The landmarks cover key body joints including the nose, shoulders, elbows, wrists, hips, knees, and ankles, providing a comprehensive skeletal representation of the human.

D. Sliding Window Segmentation

Since falls are dynamic events that happen over a short period of time, analyzing a single frame is not enough to reliably detect them. To capture movement patterns over time, the extracted landmark sequences are divided into overlapping windows of fixed length. Each window contains 30 consecutive frames, which corresponds to roughly one second of video at 30 frames per second. A step size of 15 frames is used between consecutive windows, meaning each new window overlaps 50% with the previous one illustrated in Fig.2.

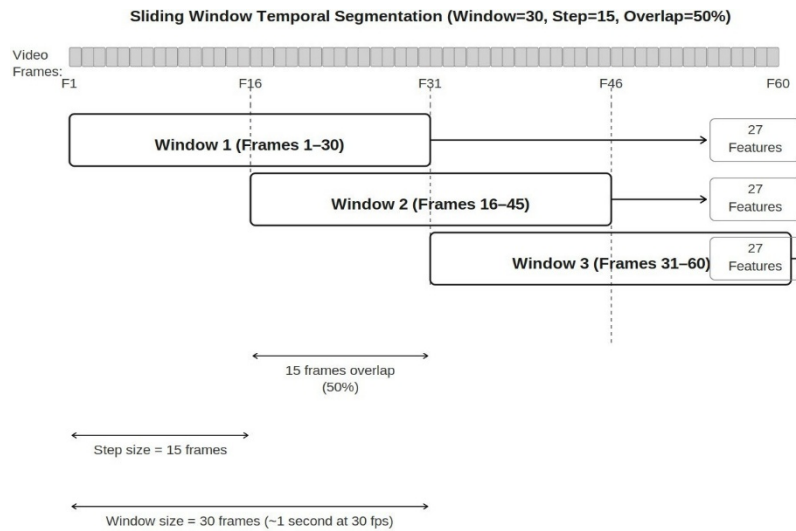


Fig. 2. Sliding Window Temporal Segmentation

E. Temporal Feature Extraction

The core contribution of this work is the extraction of 27 handcrafted temporal features from each window. These features are grouped into six categories as described below.

1) Statistical Motion Descriptors: For each of the four coordinate types — x, y, z, and visibility — the global mean, mean variation, average standard deviation and maximum standard deviation are computed across all landmarks in the window. This gives 16 features capturing body stability and movement intensity.

2) Inter-frame Velocity: The speed of body movement between consecutive frames is estimated by computing the Euclidean distance between corresponding landmark positions in adjacent frames as defined in(1):

$$d_t^j = \sqrt{\left(x_t^j - x_{t-1}^j\right)^2 + \left(y_t^j - y_{t-1}^j\right)^2} \quad (1)$$

where $x_t(j)$ and $y_t(j)$ are the coordinates of landmark j at frame t . Three velocity features are extracted — average, maximum, and standard deviation. Falls typically involve a sudden spike in velocity.

3) Torso Inclination Angle: The torso vector is computed as the difference between shoulder centre and hip centre, where each centre is the midpoint of the left and right joints. The angle with the vertical axis is calculated using the dot product formula as shown in (2)

$$\theta_t = \cos^{-1} \left(\frac{(T_t \cdot V)}{(|T_t| |V|)} \right) \quad (2)$$

where T_t is the torso vector and V is the vertical reference vector. Three features are derived — mean angle, standard deviation, and maximum angle. During a fall the torso angle increases significantly as the body tilts.

4) Body Aspect Ratio: The aspect ratio is computed as the ratio of vertical to horizontal span of all detected landmarks as defined in (3)

$$AR_t = \frac{\max(y_t^{(j)}) - \min(y_t^{(j)})}{\max(x_t^{(j)}) - \min(x_t^{(j)})} \quad (3)$$

When lying down after a fall the ratio decreases noticeably. Mean and standard deviation are used as features.

5) Ground Contact Estimation: The y-coordinates of both ankle landmarks are tracked to estimate proximity to the ground as defined in (4) :

$$G_t = \max(y_{\{ankleL\}}, y_{\{ankleR\}}) \quad (4)$$

Mean and minimum ankle y-values are extracted to identify collapsed body positions.

6) Movement Jerk: Jerk reflects how sudden and irregular a movement is, computed from the body centroid trajectory as shown in (5):

$$J = std(a_t), a_t = u_t - u_{\{t-1\}}, u_t = \|C_t - C_{\{t-1\}}\| \quad (5)$$

Where C_t is the centroid of all landmark positions at frame t , u_t is the centroid velocity, and a_t is the acceleration. Higher jerk values indicate abrupt motion such as falling.

F. Majority Voting for Fall Confirmation

To improve reliability and reduce false alarms, a majority voting mechanism is applied on top of the per-window predictions. Each window independently goes through the full feature extraction and Random Forest classification pipeline, producing its own fall or normal prediction. These predictions are collected across five consecutive windows, and a fall is confirmed only when at least 4 out of 5 windows independently predict a fall as defined in (6):

$$Fall = 1 \text{ if } \sum p_i \geq 4, \text{ else } 0 \quad (6)$$

where $p_i \in \{0, 1\}$ represents the prediction of the i -th window.

G. Model Training and Evaluation

The dataset is split at the video level so that all windows from a given video go entirely into either training or testing — never both. This prevents data leakage from correlated overlapping windows. An 80-20 stratified split resulted in 128 training videos and 32 test videos. Features are standardized using StandardScaler before training. A Random Forest classifier with 150 trees, balanced class weights, and bootstrap sampling is trained on these features. Five-fold GroupKFold cross-validation validates model stability before final evaluation on the held-out test set.

IV. RESULTS AND OUTCOME

A. Experimental Setup

The proposed fall detection system was evaluated using the GMDCSA24 dataset after extracting temporal features from pose landmark sequences. After applying sliding window segmentation with a window size of 30 frames and 50% overlap, a total of 2043 temporal windows were generated from 160 videos. Among these, 853 windows corresponded to fall activities while 1190 represented normal daily activities. To ensure reliable evaluation, the dataset was split at the video level rather than the window level. This prevents overlapping windows from the same video appearing in both training and testing sets. An 80–20 split resulted in 128 training videos (1635 windows) and 32 testing videos (408 windows).

B. Performance Evaluation

The performance of the proposed model was evaluated using standard classification metrics including accuracy, precision, recall and F1-score. These metrics are derived from the confusion matrix consisting of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). Table I shows the results of performance metrics of proposed system. Accuracy is defined in (7):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Precision is defined in (8) :

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

Recall is defined in (9) :

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

F1 Score is defined in (10):

$$F1 \text{ Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

TABLE I
PERFORMANCE METRICS OF PROPOSED FALL DETECTION MODEL

Metric	Value
Accuracy	95.74%
Precision	95.81%
Recall	93.90%
F1-Score	94.85%
ROC-AUC	0.990

The model achieved an overall accuracy of 95.74%, indicating that the majority of temporal windows were classified correctly-AUC score further indicates that the temporal features provide strong separation between fall and non-fall motion patterns.

C. Confusion Matrix Analysis

The confusion matrix obtained on the test dataset is shown in Table II. Since the model operates on temporal windows rather than entire videos, the results represent window-level classification performance. Each video produces multiple overlapping windows, which explains the larger number of evaluated samples.

TABLE II
CONFUSION MATRIX OF PROPOSED MODEL

	Predicted Normal	Predicted Fall
Actual Normal	1155	35
Actual Fall	52	801

The results show that 1155 normal windows were correctly classified while only 35 were incorrectly predicted as falls, indicating a low false alarm rate. Similarly, 801 fall windows were correctly detected while 52 fall windows were missed.

D. System Output Visualization

To further validate the practical performance of the proposed system, sample outputs obtained during testing are presented in this section. The system highlights the detected pose structure along with the classification result. Fig. 3. shows a sample frame where the system correctly detects a fall event. Fig. 4. shows a sample frame classified as normal activity

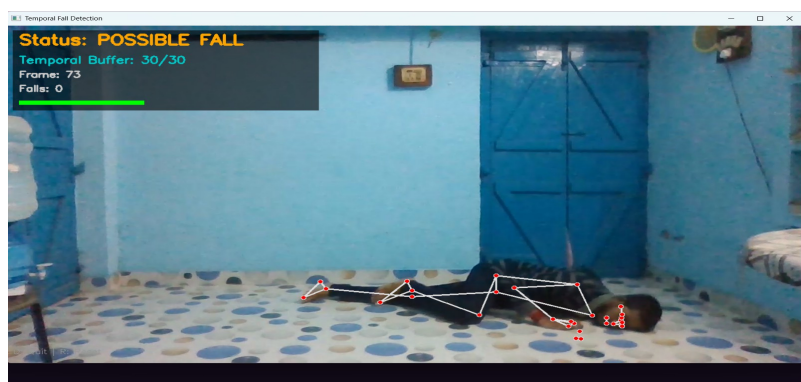


Fig. 3. System output showing detected fall event with pose landmarks

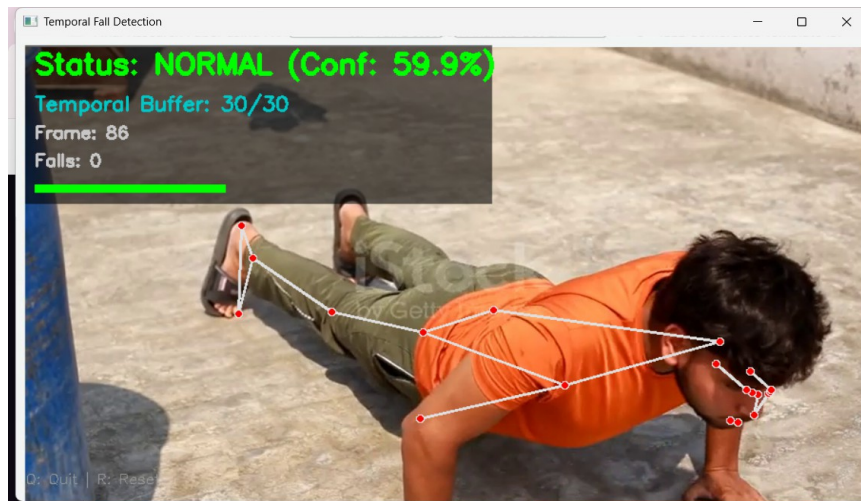


Fig. 4. System output showing normal activity classification.

V. CONCLUSION

This paper presented a vision-based fall detection system that uses temporal pose features extracted from video sequences and a Random Forest classifier for binary classification. The system processes overlapping windows of 30 frames to capture motion dynamics over time. From each window, 27 handcrafted temporal features were computed. These features were designed to capture the postural and kinematic changes that typically occur during a fall event. The system was trained and evaluated on the GMDCSA24 dataset. The proposed model achieved 95.74% accuracy. In future work, the system can be extended by incorporating larger and more diverse datasets to improve generalization. Adding more discriminative features such as optical flow or depth information could further reduce confusion between falls and similar daily activities. Exploring other lightweight classifiers or combining the current feature set with a small neural network may also help improve recall without significantly increasing computational cost.

REFERENCES

- [1] N. Mamchur, N. Shakhovska, and M. G. ml., "Person fall detection system based on video stream analysis," *Procedia Computer Science*, vol. 198, pp. 676–681, 2022.
- [2] K. Singh, A. Rajput, and S. Sharma, "Human fall detection using machine learning methods: A survey," *International Journal of Mathematical Engineering and Management Sciences*, vol. 5, no. 1, pp. 161–180, 2020.
- [3] E. Alam, A. Sufian, P. Dutta, and M. Leo, "Vision-based human fall detection systems using deep learning: A review," *Computers in Biology and Medicine*, vol. 146, p. 105626, 2022.
- [4] R. O. Zafar and F. Zafar, "Real-time activity and fall detection using transformer-based deep learning models for elderly care applications," *BMJ Health and Care Informatics*, vol. 32, no. 1, p. e101439, 2025.
- [5] B. Saha, M. S. Islam, A. K. Riad, S. Tahora, H. Shahriar, and S. Sneha, "Blockthefall: Wearable device-based fall detection framework powered by machine learning and blockchain for elderly care," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023, pp. 1412–1417.
- [6] C.-P. Liu, K.-C. Liu, C.-T. Chan, and Y. Tsao, "Deep learning-based fall detection algorithm using ensemble model of coarse-fine cnn and gru networks," in *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2023, pp. 1–5.
- [7] S. Ji, Y. Xie, and M. Li, "Sifall: Practical online fall detection with rf sensing," in *ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2022, pp. 1–15.
- [8] A. Nunez-Marcos, G. Azkune, and I. Arganda-Carreras, "Vision-based fall detection with convolutional neural networks," *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–16, 2017.
- [9] T. Alanazi, K. Babutain, and G. Muhammad, "A robust and automated vision-based human fall detection system using 3d multi-stream cnns with an image fusion technique," *Applied Sciences*, vol. 13, no. 12, p. 6916, 2023.
- [10] M. Agrawal and S. Agrawal, "Enhanced deep learning for detecting suspicious fall event in video data," *Intelligent Automation and Soft Computing*, vol. 36, no. 3, pp. 2654–2667, 2023.
- [11] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using openpose,"

Symmetry, vol. 12, no. 5, p. 744, 2020.

- [12] H. Ramirez, S. A. Velastin, I. Meza, E. Fabregas, D. Makris, and G. Farias, “Fall detection and activity recognition using human skeleton features,” *IEEE Access*, vol. 9, pp. 33 532–33 542, 2021.
- [13] A. Raza, M. H. Yousaf, W. Ahmad, S. A. Velastin, and S. Viriri, “Human fall detection using pose estimation: From traditional machine learning to vision transformers,” *Engineering Applications of Artificial Intelligence*, vol. 143, p. 109809, 2025.
- [14] K. Han, Q. Yang, and Z. Huang, “A two-stage fall recognition algorithm based on human posture features,” *Sensors*, vol. 20, no. 23, p. 6966, 2020.
- [15] Y. M. Galvao, L. Portela, J. Ferreira, P. Barros, R. A. D. A. Fagundes, and B. J. T. Fernandes, “A framework for anomaly identification applied on fall detection,” *IEEE Access*, vol. 9, pp. 77 264–77 274, 2021.
- [16] V. P. Kothari and P. S. Chakurkar, “Towards safer environments: A yolo and mediapipe-based human fall detection system with alert automation,” *MethodsX*, vol. 15, p. 103623, 2025.
- [17] T. Riahi, M. A. H. Bappy, and M. M. Islam, “Elderfallguard: Real-time iot and computer vision-based fall detection system for elderly safety,” *arXiv preprint*, 2025.
- [18] E. Alam, A. Sufian, P. Dutta, M. Leo, and I. A. Hameed, “Gmdcsa-24: A dataset for human fall detection in videos,” *Data in Brief*, p. 110892, 2024.
- [19] W.-Y. Shieh and J.-C. Huang, “Falling-incident detection and throughput enhancement in a multi-camera video-surveillance system,” *Medical Engineering and Physics*, vol. 34, no. 7, pp. 954–963, 2012.
- [20] S. Zaidi, B. Jagadeesh, K. V. Sudheesh, and A. A. Arlene, “Video anomaly detection and classification for human activity recognition,” in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, 2017, pp. 544–548.
- [21] D. R. Beddiar, M. Oussalah, and B. Nini, “Fall detection using body geometry and human pose estimation in video sequences,” *Journal of Visual Communication and Image Representation*, vol. 82, p. 103407, 2022.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 8.152