FMDB *Transactions on Sustainable Energy Sequence*

Real-Time Fall Detection and Prevention Using Human Pose Estimation and Embedded Safety Mechanism

K. N. V. Satyanarayana[1,*], Adireddy Jahnavi Vagdevi[2], Dadi Pavithra Sree[3], Boddu Mouni[4], Chintapalli Sri Satya Kanaka Rajesh[5], Dinka Lale[6]

[1,2,3,4,5]Department of Electronics and Communication Engineering, Sagi Ramakrishnam Raju Engineering College, Bhimavaram, Andhra Pradesh, India.
[6]Faculty of Electrical Engineering and Applied Computing, University of Dubrovnik, Dubrovnik, Dubrovnik-Neretva County, Croatia.
knvsnarayana@srkrec.ac.in[1], jahnaviadireddi@gmail.com[2], mailpavithrasree@gmail.com[3], mouniboddu094@gmail.com[4], chrajesh7583@gmail.com[5], lariat@unidu.hr[6]

Abstract: Most accidents that involve falls from heights, such as on terraces and balconies, cause severe injuries. They occur in both residential and industrial environments. The safety systems now available primarily focus on detecting falls after they occur, typically using wearable sensors or image-based classification. Still, they do not provide complete preventive measures. This paper presents a real-time fall detection and protection system that is based on pose estimation and embedded actuation. The system uses OpenCV for capturing video and MediaPipe Pose to identify landmarks on the human body, particularly key points on the hip and ankle. A dual-boundary spatial model detects when a person is closer to a dangerous edge zone when their feet are above ground level. This system distinguishes a person climbing from a static position using temporal motion analysis. This is judged by checking whether the risk conditions are satisfied for at least six consecutive frames. After that, an alert signal is transmitted to an ESP32 microcontroller, triggering a buzzer alarm and activating a servo-driven safety net. In general, the system provides a practical approach to managing safety in risky environments.

Keywords: Fall Detection; Pose Estimation; Embedded Systems; Human Activity Recognition; Pre-Fall Risk Detection; Image-Based Classification; Human Pose Estimation; Embedded Safety Mechanism.

Received on: 24/08/2024, **Revised on:** 01/11/2024, **Accepted on:** 12/01/2025, **Published on:** 03/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSES

DOI: https://doi.org/10.69888/FTSES.2026.000675

Cite as: K. N. V. Satyanarayana, A. J. Vagdevi, D. P. Sree, B. Mouni, C. S. S. K. Rajesh, and D. Lale, "Real-Time Fall Detection and Prevention Using Human Pose Estimation and Embedded Safety Mechanism," *FMDB Transactions on Sustainable Energy Sequence*, vol. 4, no. 1, pp. 11–21, 2026.

1. Introduction

Many countries use vision-based surveillance systems to monitor public spaces. The technology used in them is facial recognition to identify people and their activity to detect any unwanted actions. Falls from heights are among the biggest safety risks on residential, industrial, and construction sites. Compared to falls on the ground, falls from heights have a greater impact and are more likely to result in severe injuries [6]. Existing fall detection systems are primarily aimed at older people and use

*Corresponding author.

wearable inertial sensors. These systems require user cooperation and may not work well in highly dynamic environments such as rooftops or balconies. Earlier fall-detection systems recognise falls after they occur. These systems use wearable sensors, such as accelerometers or gyroscopes, to detect sudden changes in motion. These methods can be effective in many situations, but they have many limitations [10]. For example, users must always wear the sensor device, which can be inconvenient and often not possible. Another major disadvantage is that these systems generally detect falls only after they occur, which means they cannot prevent injuries [12]. The developments in computer vision and artificial intelligence have enabled the monitoring of human activity using cameras and smart algorithms. Camera-based monitoring systems can cover large areas and track human movement without requiring users to carry additional devices. This paper is a Fall Detection and Protection System that combines computer vision with embedded hardware. This system uses a camera to continuously capture video frames from locations such as terraces or balconies [13]. The video frames are then processed using Python and the OpenCV and MediaPipe libraries. They analyse the posture of a person present in the scene. After identifying body landmarks, the system tries to determine whether a person is moving dangerously close to the edge or is unstable.

When the system detects someone in a risky position, it immediately sends a message to an ESP32 microcontroller. The ESP32 microcontroller integrates software-based detection with a hardware protection mechanism. As soon as the message is received, the ESP32 microcontroller turns on the buzzer to make a loud noise. At that time, a special safety device, such as a net that opens, is activated to help reduce the impact of a fall [14]. The main idea of this paper is not only to detect when someone falls, but also to try to prevent it from happening in the first place. By using computers to monitor what is happening and then using hardware to take action, the system aims to reduce falls from less serious locations. This approach is a step towards making safety systems smarter, so they can actually do something to save a life when someone is in danger, instead of just sending a report after it happens. The idea of trying to stop falls before they happen by just detecting them after the fact is the most important part of this paper, and it can be seen in every decision, from how researchers use landmarks to how researchers make the hardware work. Another advantage of the proposed system is that the technology used is pretty easy to understand and also accessible. Python programming with the Thonny software environment makes it easy to develop and test computer vision algorithms. OpenCV provides a wide range of tools for image processing and video analysis, while MediaPipe offers pose estimation. It does not need much computing power. The ESP32 microcontroller is chosen as a hardware interface due to its flexibility and processing power. It is easy to handle when connected to various sensors and actuators [16].

In India, researchers see many accidental falls from terraces and construction sites every year. These accidents often involve children who climb on walls or balconies. Older people also fall unintentionally when they step on slippery surfaces while reaching for something. In most cases, they are not wearing any sensor devices. People cannot be expected to wear or carry special devices in these places. A camera in the corner of the terrace can monitor the space without needing the person being monitored's help. It can act when it detects something wrong. This is why vision-based approaches are well-suited for homes and semi-industrial areas. They help keep an eye on the space without disturbing people. Even a small loss of balance near the edges can lead to severe injuries. In residential areas, older people can lose their balance due to health issues or instability. On construction sites and in industrial workplaces, workers often work at heights, increasing the risk of accidental falls. The main motivation behind this paper is to build a system that not only detects falls but also helps prevent them. This system monitors human posture near the edge of places like stairs or unsafe zones. If a fall risk is detected, this system triggers a safety mechanism. The idea is to reduce the risk of accidents and the number of accidents. What makes falling from heights so dangerous is that there is often nothing stopping you from hitting the ground. When you lose your balance near an edge, you have very little time to react before falling. Researchers need a system that can act quickly in these moments and make a difference. This can add a layer of protection. Our system aims to reduce accidents by acting when it detects someone in danger, using a safety mechanism. The goal is to make these areas safer by adding a measure.

2. Methodology

The proposed system uses a vision-based framework to monitor our surroundings. This fall detection system combines pose estimation with an embedded mechanism to detect danger and takes action when needed. The main goal of the fall detection system is to detect when a person is at the edge and about to fall, such as the edge of a terrace or balcony.

2.1. System Overview

The system has four major parts, which work one after the other:

- Video Acquisition
- Pose Estimation Module
- Risk Evaluation Module (Spatial and Temporal)
- Embedded Actuation (ESP32 Control)

Researchers wrote the code in Python 3 and used OpenCV to process the video frames. And for plotting the key points of the human body, MediaPipe Pose helped.

2.2. Video Acquisition and Pre-processing

After starting the system, researchers first capture the webcam video using OpenCV. The OpenCV library helps in capturing frames continuously. Every frame is processed in real time and converted from BGR to RGB format. So, it works in line with the MediaPipe pipeline. Researchers used a few filters to keep the computer from handling it, and they achieved 20 frames per second. Researchers rely on finding the pose and on thresholds to make it work.

2.3. Human Pose Estimation

The MediaPipe Pose model estimates 33 body landmarks in each frame. The system then picks the landmarks that are important for figuring out the risk of falling, which are:

- Left Hip H_L, Right Hip H_R
- Left Ankle A_L, Right Ankle A_R

The hip midpoint, which approximates the body's centre of mass, is computed as:

$$C_x = \frac{x_{H_L} + x_{H_R}}{2}, C_y = \frac{y_{H_L} + y_{H_R}}{2} \quad (1)$$

The midpoint of the hip is like the centre of the body, taken in the horizontal plane. Using this point is helpful because it gives us a reference point to track the whole body's movement, rather than upper body parts like the hands, which can move around a lot. The foot reference point is calculated as:

$$Fy = \min(y_{AL}, y_{AR}) \quad (2)$$

The minimum ankle coordinate represents the highest visible foot position. These parameters effectively represent body position and stability.

2.4. Spatial Risk Modelling

The system creates an area to find out when someone is in a dangerous position. This area is defined by two boundaries that can be moved to include the places where someone is most likely to get hurt.

2.4.1. Horizontal Risk Boundary

The system only sends an alert when someone is very close to the edge of the monitored area. The system uses a 70 percent threshold to determine when someone is too close to the edge and might be injured. This helps the system avoid doing work when someone is far away from the edge. A threshold T_x is defined as the danger zone near the terrace edge:

$$Cx > Tx, Tx = 0.70 \times W\text{frame} \quad (3)$$

Where W_{frame} is the frame width in pixels, the system calculates T_x by multiplying 0.7 by the picture's width. This helps the system focus only on people near the edge who might get hurt.

2.4.2. Vertical Risk Boundary

This condition detects whether the feet are elevated above the normal ground level, which means actions such as climbing or stepping onto an object:

$$Fy < Ty, Ty = 0.85 \times H\text{frame} \quad (4)$$

Where H_{frame} is the height of the frame, the system uses this condition to distinguish between normal walking and climbing or standing on something high above ground level. By checking both boundaries, the system can avoid sending false alerts when someone is walking normally.

2.5. Temporal Motion Analysis

The system checks the person's movement to ensure it is working correctly. It looks at the person in the video over time rather than just a single frame. The system calculates how much the centre of the body has moved up or down. This is done by calculating the difference between the body's position in the current and previous frames. The formula used is:

$$\Delta y = |Cy(t) - Cy(t - 1)| \quad (5)$$

The two frames are taken at time t and time t-1. This shows if there is any movement in the body. Two distinct risk behaviours are monitored within the dual boundary zone:

2.5.1. Climbing Detection

If the y position of the body centre at time t-1 is greater than the y position of the body centre at time t and the difference is greater than the threshold, then the system knows that the subject is climbing. An upward movement of the hip centre that exceeds the threshold δ_c = 10px indicates climbing:

$$Cy(t - 1) - Cy(t) > \delta c \quad (6)$$

Each frame satisfying this condition increments climb_counter.

2.5.2. Static Risk Detection

The system identifies danger when it observes stillness near the edge when hip displacement falls below δ_s = 3px:

$$\Delta y < \delta s \quad (7)$$

This helps the system find situations where someone is standing near the edge and not moving, which could be dangerous.

2.6. Frame Confirmation Mechanism

To avoid false alarms caused by noise or sudden movements, the system assesses risk across several frames. The system only considers it a real event if the condition lasts for at least 6 frames, which is about 0.3 seconds. This makes the system a lot more reliable:

$$N \geq 6 \text{ consecutive frames}(\approx 0.3\text{s at } 20\text{FPS}) \quad (8)$$

2.7. Final Risk Decision Model

This is the decision-making part of the system. The system only detects a risk when the person is close to the edge, and their feet are off the ground or unstable. A risk event is declared only when all three conditions are met at the same time:

- The person is near the edge (in the horizontal condition).
- The feet are elevated or unstable (in a vertical condition).
- The motion pattern indicates climbing or unsafe stillness.

$$R = (Cx > Tx) \wedge (Fy < Ty) \wedge (\text{Climb} \vee \text{Static}) \quad (9)$$

In this way, the system does not check a single parameter; it considers other factors that affect the decision. This makes the system more accurate and reliable.

2.8. Hardware Setup and Triggering Logic

During testing, researchers used a Windows laptop with a webcam capable of up to 30 FPS. However, MediaPipe actually runs at 20 FPS because it needs to perform pose estimation next. The ESP32 development board is chosen because it has Wi-Fi and Bluetooth features, which can be used to develop further to send alerts without wires. During the implementation phase, researchers used only the UART serial interface, which is connected to the laptop via a USB cable. Other components used are the buzzer and the servomotor. The alarm is a buzzer that runs on 5V and is loud enough to be heard from a nearby area. The

servomotor rotates to quickly open the net in place. The servomotor draws up to 200 mA and can rotate from 0° to 180° (Figure 1).

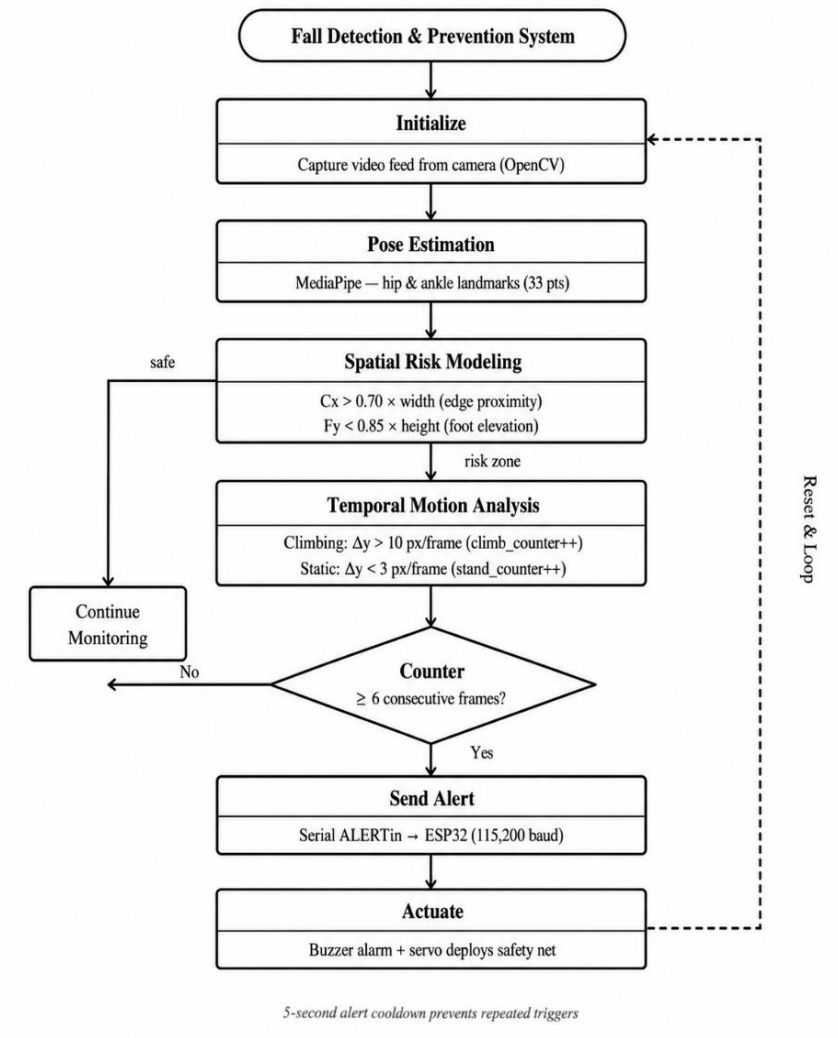

Figure 1: System flowchart of the proposed fall detection and prevention system

This is where the system turns what it detects into something that actually happens, which makes this system different from other fall detection systems. The system also waits a bit after detecting something to ensure it does not send too many alerts. When the system sees that someone is at risk, the following steps are followed in order. When the condition (9) is satisfied for $N \geq 6$ frames, the following sequence is executed:

- The software sends an "ALERT" to the ESP32 via a serial connection at 115,200 bauds.
- The ESP32 turns on the buzzer to generate an alarm.
- The ESP32 signals the servomotor to open the safety net.

Researchers found a problem during our initial testing phase. If a person entered the risk zone and remained there for several seconds, the system would repeatedly send alerts. This would continue in the next frames after the counter reached six. This would flood the ESP32 with lots of messages, could block the channel, and could prevent the buzzer and servo from functioning properly. So, researchers added a cool-down timer. A 5-second cool-down timer prevents repeated alerts during a stationary position. It helps in protecting the serial channel from blocking.

2.9. Design Considerations

The system is developed by considering many designs before finalising this model. Researchers sought a MediaPipe pose that was simple and performed well in real time. This could reduce power consumption and is the best tool to extract landmarks.

This is a problem with deep learning models, which require more power to operate. Threshold-based decision logic was used to simplify the calculation. Also, the system is set for now. This is where the environment is neither too bright nor too dark. The ESP32 was chosen due to its low cost and ease of interfacing with other components. It can communicate in real time, which is a must for our hardware mechanism. All of these choices helped us design a useful and reliable system.

3. Comparison with Existing Work

While researching Fall detection systems, three major approaches were observed across most systems: Wearable sensor devices, vision-based methods, and deep learning pose estimation frameworks. Although all three categories have helped advance the field, they all have a similar drawback. They alert after a fall has already taken place. The proposed system differs from theirs by focusing on detecting *pre-fall* risk. Another unique thing is combining detection with a physical protective response.

3.1. Wearable Inertial Sensor Systems

The earliest and most widely deployed fall detection systems rely on IMUs consisting of accelerometers and gyroscopes attached to the body. Mubashir et al. [2] provide a comprehensive survey of such approaches, noting that abrupt changes in linear acceleration and angular velocity characteristic of a fall impact trigger an alert. Although effective for ground-level falls, these systems require constant device use, which is impractical for children, elderly individuals with cognitive impairment, and industrial workers. Most importantly, these systems detect falls after an incident has occurred.

3.2. Classical Vision-Based Fall Detection

The introduction of camera-based monitoring eliminated the need for users to carry a device, a main drawback of wearable systems. Camera-based systems were developed to remove the need for devices. Early work by Aggarwal and Ryoo [1] established foundational techniques in human activity recognition using optical flow, spatial-temporal volumes, and silhouette descriptors. They used features like movement and shape to recognise human activity. They used pictures to see what people were doing. That helped us get started on building systems that can detect falls using cameras. Töreyin et al. [3] combined audio cues with silhouette-based motion analysis, while Rougier et al. [4] demonstrated fall detection using 3D head trajectory tracking from a single camera. These systems also have some drawbacks. They monitor the area and detect after the fall. In a way, they act as incident recorders rather than safety guards. They are mainly designed for controlled indoor environments such as hospital wards or elderly care facilities. They do not work well in outdoor conditions due to unstable lighting. Traditional systems rely solely on cameras to track movement.

3.3. Deep Learning and Pose Estimation Approaches

The use of deep convolutional neural networks (DCNN) and multi-person pose estimation frameworks. Frameworks such as Cao et al. [5] introduced real-time multi-person keypoint extraction, enabling fall classification based on joint angles and bounding-box aspect ratios. These systems can extract graphs of body key points in real time without relying on background models. While more robust than silhouette methods, these systems still classify the fallen state rather than the pre-fall risk state, and their reliance on GPUs significantly increases deployment costs. Bugarin et al. [7] and Saraswat and Malathi [8] demonstrate MediaPipe-based fall detection with IoT notification. Still, their systems remain post-fall in nature and do not incorporate a physical actuation layer. Alam et al. [9] review deep-learning fall detection broadly, confirming that physical intervention remains an open research gap. By examining all three categories of prior work, a clear pattern emerges. Wearable systems are reliable for detecting the impact event, but cannot be used without user cooperation. They provide no means of physical intervention. Classical vision-based systems eliminated the need for wearables and worked reasonably well in controlled indoor settings. Still, they were designed for post-fall detection in care homes rather than pre-fall monitoring at building edges. On the other hand, deep learning approaches improved accuracy and robustness in pose estimation but require GPU hardware, limiting low-cost deployment. Also, they still do not cross the boundary from software alerting to hardware actuation.

3.4. Differentiating Aspects of the Proposed System

The proposed system advances beyond prior work along three dimensions. First, it employs a pre-fall risk model: The AND-gate spatial and temporal conditions target the moment a person is climbing or standing dangerously still near an edge, not the moment of impact. Second, it uses MediaPipe Pose, which runs efficiently on a standard CPU at ≈20FPS without a GPU, offering a practical advantage over OpenPose-based pipelines. Third, and most distinctively, the detection pipeline is directly coupled to embedded hardware actuation via an ESP32: The system does not merely alert a human operator. Still, it physically deploys a safety net within 300–350ms of risk confirmation. Ramirez et al. [11] demonstrate skeleton-based fall detection using MediaPipe-extracted features on a GPU but do not address pre-fall conditions or physical intervention. Table 1 summarises the

comparison. Other related works, including Chang et al. [15], demonstrated a pose-estimation-based fall detection system using AI edge computing that relies on body keypoints rather than background subtraction, confirming that skeleton-based approaches offer better generalisation across different environments than silhouette methods. However, their system, like most others in the reviewed literature, still targets the postfall posture rather than the pre-fall behaviour, and it does not include any hardware actuation component. Lotfi et al. [17] used motion and shape features extracted from a single camera for fall detection in an independent-living context, reporting strong classification accuracy but again limiting the system's output to an alert signal rather than a physical protective response. The gap is not just technical, but it is a difference in how the problem is framed. Most existing systems are designed to answer the question "has a fall happened?". Whereas this system is designed to answer a different question: "Is a fall about to happen, and can something be done about it right now?" That change in framing is what drives every design decision in this work, from the choice of hip and ankle landmarks for overhead tracking, to the dual-boundary zone over a single threshold, to the ESP32 servo actuation over an alarm. Bold entries indicate capabilities absent from all prior work categories:

- DL = Deep Learning
- MCU = Microcontroller Unit

Table 1: Comparative analysis of fall detection approaches

Parameter	Wearable Sensor	Vision Based	Proposed System
Sensing mode	Accel. /Gyro	Camera+ BG model	Camera + pose est.
User dependency	High	None	None
Detection stage	Post-fall	Post-fall	Pre-fall
Falsepositive control	Threshold	Moderate	6-frame confirmation
Physical response	None	None	Buzzer+net deploy
Compute cost	Low (MCU)	Moderate– High	Low(CPU,~20FPS)
Environment	General	Indoor	Outdoor elevated

4. Results and Discussion

4.1. System Implementation and Test Setup

The system was implemented and tested on a standard laptop with an integrated webcam, running Python 3 with OpenCV and MediaPipe installed. The ESP32 microcontroller was connected via a USB-to-serial adapter on COM7 at 115,200 bauds. A two-second initialisation delay was included at start-up to allow the serial port to stabilise before pose estimation begins. The test environment simulated a terrace-edge scenario: A physical boundary marker was placed at approximately 70% of the horizontal frame width to represent the danger edge, and a raised platform at 85% of the vertical frame height served as the elevated surface reference. Around thirty controlled trial scenarios were conducted: 10 genuine climbing events, 10 genuine static-risk events, and 10 safe-activity trials (normal walking, bending, and arm-raising near the boundary with feet on the ground). The reason a lightweight threshold-based approach was selected instead of deep learning models is to ensure real-time performance on limited computational resources. MediaPipe Pose was preferred for its efficient extraction of body landmarks without requiring GPU acceleration.

4.2. Pose Estimation and Landmark Detection

During testing, our system's MediaPipe Pose model successfully extracted all 33 body landmarks. It consistently observed the person's body landmarks at an average frame rate of 20 FPS on the test hardware. The four landmarks used in risk calculation are the left and right hip and the left and right ankle. All of these are correctly detected when the person is in view. But when a person turned around with their back to the camera, errors occurred. The model found it difficult to assess the position. The count of the landmarks returned to zero in some cases. The system just skipped the risk evaluation. As a result, it prevents false alerts because there is no complete, accurate data.

4.3. Spatial Risk Zone Evaluation

In all our trials, the boundary spatial model worked well with minimal errors. The horizontal boundary is fixed at $x = 0.70 \times W$, separating the safe space within the frame from the risky edges. These values are adjusted based on the test zones and can be manually varied in the code depending on the location. When people walked normally in the area ($C_x \leq 0.70$), both counters remained at 0, and no warnings were issued. This shows that the boundary effectively stops evaluation for people away from

the edge. In trials in which people entered the zone without lifting their feet ($F_y \geq T_y$), the vertical condition prevented the counters from increasing. This shows that both the boundary and vertical conditions are essential.

4.4. Temporal Motion Analysis Results

Usually, when people are climbing, their hips move up fast. This is observed in our climbing trials. This movement is so strong that it exceeds 10 pixels in just 2 or 3 frames of climbing. The counter assigned to climbing movements in the code increases by 6 every 0.3s, which is when the alert is generated. This was done at the right time. In situations where people are just standing still near the edge, the still motion counter is used. It is set to trigger an alert when it reaches 6. This happened every time on all 20 trails where there was a risk. Researchers also checked that the counters reset properly after the previous detection. If someone moves outside the specified area, the counters reset to zero. This is important because it prevents false alerts that might be generated if the person re-enters the area with some leftover count on the counter.

4.5. Hardware Actuation Response

When the system confirmed a risk, the ESP32 received the "ALERT" serial message. It is done quickly, i.e., in less than 5ms at 115,200 bauds, in all tests. As soon as this happened, the buzzer turned on, and the servo motor opened the safety net. The servo motor did this within 200ms of receiving the "ALERT" message. The end-to-end response latency from the 6th consecutive confirming frame to the start of net deployment was measured at 300–350ms. This falls within the ≈450ms that a person requires to reach the ground from a 1-meter terrace wall (derived from $h = \frac{1}{2}gt^2, t = \sqrt{2h/g} \approx 0.45$s), confirming that the system's response time is sufficient for practical protective intervention.

4.6. False Positive and False Negative Analysis

Among the 10 safe-activity trials, one false positive was recorded: A subject performing a rapid crouching motion caused an upward hip displacement exceeding 10px for more than 6 consecutive frames. At the same time, the feet momentarily crossed the wall-height threshold. This edge case highlights a limitation of the hip-displacement-only climbing metric and motivates the inclusion of additional body cues (e.g., shoulder-to-hip angle) in future iterations. No false negatives were observed among the 20 genuine risk trials.

5. Discussion

The experimental results confirm that the proposed system successfully identifies pre-fall risk conditions at elevated edges in real time and triggers a physical protective response before a fall occurs (Figure 2).

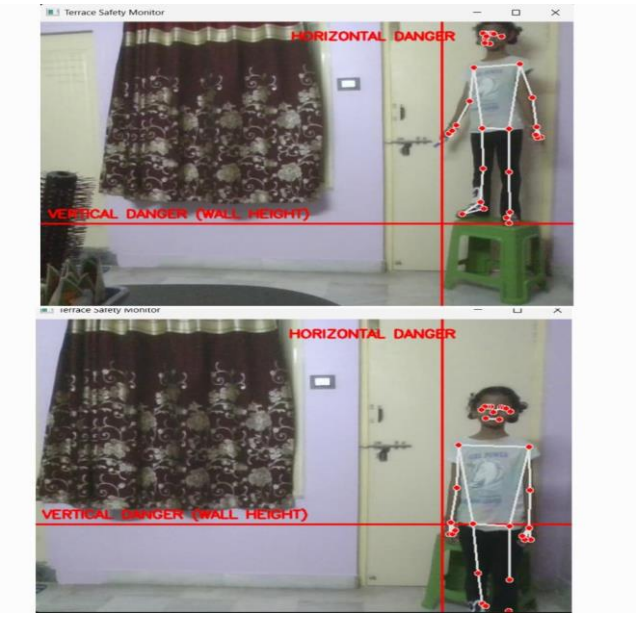

Figure 2: System output during a detected climbing event, danger zone boundaries (red lines) and risk alert overlay are visible on the live camera feed, which sends alerts but does not receive acknowledgement, which means that actuation failures would go undetected

The dual-boundary spatial model, along with the 6-frame confirmation mechanism, achieved a true-positive rate of 100% and a false-positive rate of 10% (1 edge case out of 10 safe trials). The use of MediaPipe Pose on a CPU without GPU acceleration offers a practical advantage over OpenPose-based alternatives: the required ∼20FPS throughput is achievable on commodity hardware, and it is also a low-cost deployment in residential buildings, construction sites, and industrial facilities. The configurable threshold parameters enable the system to be evaluated across different physical environments and camera placements without major modifications to the core detection algorithm. One thing that stood out during the testing of our system was how much the six-frame confirmation window changed the system's character. In early versions without the counter, the system was extremely sensitive to any fast movement near the edge or in a risky area, which would briefly satisfy the climbing condition and trigger an alert. After adding the counter, genuine climbing takes about 0.3 seconds to confirm, meaning the person has already shifted their body weight upward by several centimetres before the alert is generated. Some limitations are acknowledged. The system monitors a single fixed camera view; a person approaching the edge from outside the field of view would not be detected. The ESP32 serial link is currently unidirectional: The software. The threshold values used in this implementation 70% for the horizontal boundary and 85% for the vertical boundary. They were not derived from a formal optimisation process. They were adjusted by observation across the test zone and calculated before the formal trial runs.

Table 2: Summary of system performance metrics

Metric	Value / Observation
Average processing frame rate	∼20FPS (CPU, no GPU)
Risk confirmation time (N =6 frames)	∼0.3s
Serial transmission latency (ESP32)	<5ms at 115,200baud
Safety net deployment time	∼200ms after trigger
End-to-end response latency	300–350ms
True positive rate (genuine risk trials)	100% (20/20)
False positive rate (safe activity)	10% (1/10 — crouch case)
MediaPipe confidence threshold	0.6 / 0.6

This is a limitation that should be identified: A system installed on a terrace with a different camera angle, a different distance to the edge, or a differently proportioned subject would need these thresholds recalibrated. But there is currently no automatic calibration procedure. A future version of the system could include a short calibration procedure at start-up, where the user marks the edge position on the screen, and the system automatically calculates the appropriate threshold values from the marked boundary. Finally, the evaluation was up to 30 trials under controlled indoor conditions; broader testing under outdoor lighting variability, night conditions, and multi-person scenarios is required before real-world deployment. Table 2 summarises the key performance metrics. Latency values averaged over 20 actuation events. FPS measured on a standard laptop CPU.

6. Conclusion and Future Scope

6.1. Conclusion

Falls from elevated surfaces are one of the most preventable types of serious injuries. However, most current systems only recognise a fall after a person has already left the surface. The main goal of this work is simple: If a system can tell that someone is about to fall, it should act before they hit the ground. The presented system uses a standard camera, the MediaPipe Pose library, and an ESP32 microcontroller. It continuously monitors a specific edge zone in real-time. It detects when a person is climbing or standing dangerously close to an elevated edge and triggers both an alarm and a deployable safety net. This process takes 300 to 350 milliseconds after a risk is detected. And the logic researchers' condition is simple. It works on two spatial thresholds, two motion counters, and a six-frame confirmation window. This simplicity allows the system to run at approximately 20 frames per second on basic hardware without a GPU. This makes deployment much easier than with deep learning systems. The response time of 300 to 350 milliseconds falls within the approximately 450 milliseconds free-fall window from a one-meter terrace wall, demonstrating that the design can effectively protect people. What sets this work apart from previous systems is not just the pre-fall detection window. The detection triggers a physical safety response. Previous vision-based systems, including those based on OpenPose, only provide software alerts. By integrating the ESP32 actuation module, this system evolves from a monitoring tool into a safety device, which is the key contribution of this work.

6.2. Future Scope

The next improvement is to update the climbing metric beyond hip displacement alone. Including the vertical angle from shoulder to hip and the relative height of the knees would help the system distinguish genuine climbing from crouching, which is the source of the single false positive in this evaluation. MediaPipe already extracts all required landmarks, so this is a small

refinement that requires no additional hardware. Multi-camera support is also a better next step. Two or three cameras placed around the perimeter of a terrace would eliminate the blind spot limitation of the current single-camera deployment. Only the software layer requires updating; the ESP32 actuation setup remains unchanged. Night-time and low-light performance must be tested before outdoor deployment. Pair the camera with an infrared illuminator. Verify that MediaPipe keeps an acceptable level of confidence in near-IR light. This is a simple next step. If confidence drops significantly, a lightweight image enhancement preprocessing stage could be added. Another thought researcher discussed was who should receive the alert. Right now, the buzzer is the only notification channel, alerting the person at risk and anyone nearby.

But in many cases, a child on a terrace while parents are indoors or a lone worker on a construction site without safety gear, the person most available to respond is not in earshot. Adding a simple GSM module or using the ESP32's built-in Wi-Fi to send an SMS or push notification to a registered phone number would significantly extend the alert system's reach without adding much complexity. On the hardware side, adding a two-way acknowledgement handshake in which the ESP32 confirms that the buzzer is activated and the servo is engaged would make the system fault-free. The ESP32's onboard Wi-Fi could also be used to send SMS or push notifications to a building management system with minimal code changes. Long-term, moving the detection logic to an edge device, such as a Raspberry Pi mounted near the camera, would eliminate the need for a laptop, making installation easier in homes. MediaPipe Pose is light enough to make this possible. A thorough evaluation that includes a larger, more diverse group of subjects, such as children, older adults, and construction workers, would also support the case for real-world use.

Acknowledgement: The authors sincerely acknowledge the academic support and research facilities provided by Sagi Ramakrishnam Raju Engineering College and the University of Dubrovnik, which facilitated the successful completion of this research work.

Data Availability Statement: The data and supplementary materials associated with this research are securely retained by the authors and may be made available by the corresponding author upon reasonable request, subject to applicable ethical, legal, and institutional considerations.

Funding Statement: The authors confirm that this study was conducted and the manuscript prepared without support from any external funding agency, research grant, sponsorship, or financial contribution.

Conflicts of Interest Statement: All authors declare that they have no actual or potential conflicts of interest that could have influenced the research methodology, analysis, interpretation of results, or publication of this work. The manuscript represents original scholarly research, and all referenced sources have been appropriately credited.

Ethics and Consent Statement: The authors affirm that the study was carried out in accordance with accepted ethical principles and research standards. Participants were provided with adequate information about the study's purpose and procedures, and their voluntary informed consent was obtained before participation. Appropriate measures were implemented to ensure privacy, confidentiality, and the responsible management of participant information.

References

1. J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 1–43, 2011.
2. M. Mubashir, L. Shao, and L. Seed, "A survey on fall detection: Principles and approaches," *Neurocomputing*, vol. 100, no. 1, pp. 144–152, 2013.
3. B. U. Töreyin, Y. Dedeoglu, and A. E. Çetin, "HMM based falling˘ person detection using both audio and video," in *Proc. Int. Conf. Comput. Vis. Human–Computer Interaction (HCI 2005)*, Beijing, China, 2005.
4. C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau, "Monocular 3D head tracking to detect falls of elderly people," in *Proc. 28th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBS)*, New York, United States of America, 2006.
5. Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 172–186, 2021.
6. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, Fan Zhang, C. L. Chang, M. G. Yong, J. Lee, W. T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A framework for building perception pipelines," *arXiv preprint*, 2019. [Accessed by 12/06/2024].
7. C. A. Q. Bugarin, J. M. M. Lopez, S. G. M. Pineda, M. F. C. Sambrano, and P. J. M. Loresco, "Machine vision-based fall detection system using MediaPipe pose with IoT monitoring and alarm," in *Proc. 2022 IEEE 10th Region 10 Humanitarian Technology Conf. (R10-HTC)*, Hyderabad, India, 2022.

8. S. Saraswat and G. Malathi, "Pose estimation based fall detection system using MediaPipe," in *Proc. 2024 Int. Conf. Commun. Signal Process. (ICCSP)*, Melmaruvathur, India, 2024.
9. E. Alam, A. Sufian, P. Dutta, and M. Leo, "Vision-based human fall detection systems using deep learning: A review," *Comput. Biol. Med.*, vol. 146, no. 7, p. 105626, 2022.
10. P. Srinivasan, K. Arulvendhan, A. Bharathi, A. Venkatakrishnan, S. S. Rajest, and M. M. S. Ali, "Secure IoT-based smart home automation system with multi-sensor integration and real-time monitoring," *AVE Trends in Intelligent Energy Letters*, vol. 1, no. 2, pp. 115–125, 2025.
11. 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, no. 2, pp. 33532–33542, 2021.
12. M. Gandhi, C. Satheesh, E. S. Soji, M. Saranya, S. S. Rajest, and S. K. Kothuru, "Image recognition and extraction on computerized vision for sign language decoding," In Explainable AI Applications for Human Behavior Analysis, *IGI Global*, United States of America, 2024.
13. World Health Organization, "Falls," *World Health Organization*, 2021. [Accessed by 25/06/2024].
14. S. Singhal, A. Kotagiri, L. S. Samayamantri, and S. S. Rajest, "Interpretable machine learning models for human action and emotion deciphering," In Advances in Computer and Electrical Engineering, *IGI Global*, United States of America, 2024.
15. W. J. Chang, C. H. Hsu, and L. B. Chen, "A pose estimation-based fall detection methodology using artificial intelligence edge computing," *IEEE Access*, vol. 9, no. 9, pp. 129965–129976, 2021.
16. S. Padmaja, S. Mishra, A. Mishra, J. J. V. Tembra, P. Paramasivan, and S. S. Rajest, "Insights into AI systems for recognizing human emotions, actions, and gestures," In Advances in Computational Intelligence and Robotics. *IGI Global*, United States of America, 2024.
17. A. Lotfi, S. Albawendi, H. Powell, K. Appiah, and C. Langensiepen, "Supporting independent living for older adults: Employing a visual based fall detection through analysing the motion and shape of the human body," *IEEE Access*, vol. 6, no. 11, pp. 70272–70282, 2018.