WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-BASED FALL DETECTION AND ALERT SYSTEM

Mrs. R. Pushpakumari[1],Mr. B. Arunalex[2], Ms. J. Heera[3], Mr. A. Mohammed Riyas[4]

Department of Artificial Intelligence and Machine Learning
Sri Shakthi Institute of Engineering and Technology, Coimbatore

ABSTRACT :

Elderly fall incidents represent a major health risk, often leading to severe injuries or delayed medical response. Traditional fall-monitoring systems rely on wearable sensors or manual surveillance, both of which are inconvenient and prone to human error. This paper presents VISIONGUARD, an AI-driven fall detection system that utilizes computer vision and pose-estimation techniques to monitor human posture in real time. By leveraging Mediapipe's skeleton-landmark detection and custom fall-logic algorithms, the system identifies abnormal posture transitions and sends instant alerts through a lightweight web interface. The proposed solution eliminates the need for wearable devices, operates efficiently on standard hardware, and offers a cost-effective approach to continuous elderly monitoring. Experimental results demonstrate reliable detection accuracy in common fall scenarios, highlighting its applicability in home-care, hospitals, and assisted-living environments.

INTRODUCTION

Falls among elderly individuals are a major cause of injury, disability, and hospitalization worldwide. With rising life expectancy, the need for intelligent healthcare support systems has become critical. While several technological solutions exist—such as wearable sensors, smartwatches, and manual CCTV monitoring—these approaches suffer from practical limitations including battery dependency, discomfort, poor user acceptance, and the requirement for continuous human supervision.
Recent advancements in artificial intelligence, particularly computer vision and pose-estimation, have enabled the development of automated fall-detection systems without the need for physical sensors. By detecting human body keypoints, interpreting posture orientation, and analyzing motion patterns, AI systems can accurately distinguish normal activities from falls.

VISIONGUARD is designed to be a non-intrusive, camera-based system that operates in real time and automatically triggers alerts during fall events. The project integrates AI, backend automation, and a user dashboard, making it a complete full-stack smart-healthcare solution.

USER INTERACTION PHASE

The system begins with user interaction via a simple web dashboard. Caregivers or administrators can access the live camera feed, monitor posture status, and view fall alerts. The interface is designed for minimal user involvement, ensuring accessibility for non-technical users. A single button initiates the monitoring process, and alerts are displayed on-screen with automated notifications, ensuring quick response.

INPUT PROCESSING PHASE

The camera feed is captured through a Flask backend and processed frame-by-frame. Each frame undergoes preprocessing steps such as resizing, normalizing, and conversion to an appropriate format for pose estimation. The processed frames are then passed to the Mediapipe model, which extracts body keypoints and maps them to a skeletal representation.

EXPENSE EXTRACTION PHASE

During the analysis phase, the extracted pose-landmarks are evaluated to determine the user's posture. Angles, body-relative distances, and vertical orientation are calculated. The fall-detection logic checks for rapid downward movement, horizontal body posture, or loss of skeletal stability. These calculations ensure accurate differentiation between actual falls and benign actions such as sitting or lying gently.

RESULT PHASE

The outcome of the detection algorithm is classified as either **Normal Activity** or **Fall Detected**. When a fall event is identified, the system logs the event, highlights the skeleton on the dashboard, and updates the status indicator. These results are stored for analysis, enabling monitoring of frequency

and pattern of falls.

OUTPUT PHASE

The output phase delivers the final system responses based on the processed fall-detection results. This includes generating real-time alerts, updating the dashboard, and displaying visual indicators such as detected pose, fall status, and timestamps. The system ensures that the output is clear, timely, and actionable so that caregivers or administrators can quickly respond to emergency events.

EXISTING METHOD

Existing fall-detection systems mostly rely on wearable sensors, threshold-based acceleration analysis, or traditional computer vision techniques. While these methods offer basic detection, they often suffer from accuracy issues, require physical devices to be worn, or fail in low-light and cluttered environments. Many systems also lack real-time alert mechanisms and are difficult to integrate with modern IoT or cloud-based platforms.

PROPOSED METHOD

The proposed method uses a camera-based AI system integrating Mediapipe pose estimation and custom fall-detection logic to identify abnormal movements automatically. The system processes live video frames, extracts skeletal keypoints, analyzes posture transitions, and triggers alerts instantly when a fall is detected. This approach eliminates wearable devices, improves accuracy, supports remote monitoring, and provides continuous real-time visual feedback.

IMPLEMENTATION PROCESS

Phase 1: Initial Setup & Environment Configuration
his phase includes installing Python, configuring Flask, integrating Mediapipe, and setting up the camera pipeline. The environment is tested for frame capture, model loading, and latency evaluation.

Phase 2: Frontend Development
A clean HTML/CSS interface is created to display:

- Live video stream
- Status indicator
- Alert messages

Phase 3: Backend Development
Flask routes handle video streaming, pose detection, and fall classification logic.
Modules include:

- Camera handler
- Pose-detection engine
- Alert manager
- Event logger

Phase 4: Testing
Fall scenarios (forward, backward, lateral) and non-fall scenarios (sitting, bending, lying slowly) are tested. False positives and false negatives are analyzed. Camera angle and lighting variations are evaluated.

Phase 5: Deployment
The system is packaged for deployment on Windows/Linux. Future versions include deployment on Raspberry Pi for IoT usage.

SYSTEM ARCHITECTURE

–System architecture consists of:

1. Input Layer – Camera feed acquisition
2. Processing Layer – Pose detection, skeletal mapping
3. Analysis Layer – Fall logic evaluation
4. Backend Layer – Flask server handling stream and alerts
5. Presentation Layer – Web dashboard with visualization
6. Alert Layer – Output notifications

METHODOLOGY

1. Capture real-time frames from camera

2. Preprocess frames
3. Extract pose landmarks using Mediapipe
4. Compute body orientation and key angles
5. Apply fall-detection rules
6. Generate alerts on detection
7. Display results on user dashboard

TECHNOLOGY STACK

1. Frontend: HTML, CSS, JavaScript
2. Backend: Python Flask
3. AI/ML Layer: Mediapipe Pose, OpenCV
4. Hardware: Webcam / external camera
5. Tools: VS Code, Python 3.12

RESULT

The system successfully detected falls in controlled scenarios with high reliability. Skeleton-based analysis proved consistent even in varying lighting conditions. The dashboard provided real-time output with minimal latency. Experimental observations confirmed the system's capability for continuous monitoring without requiring wearable equipment.

ADVANTAGES

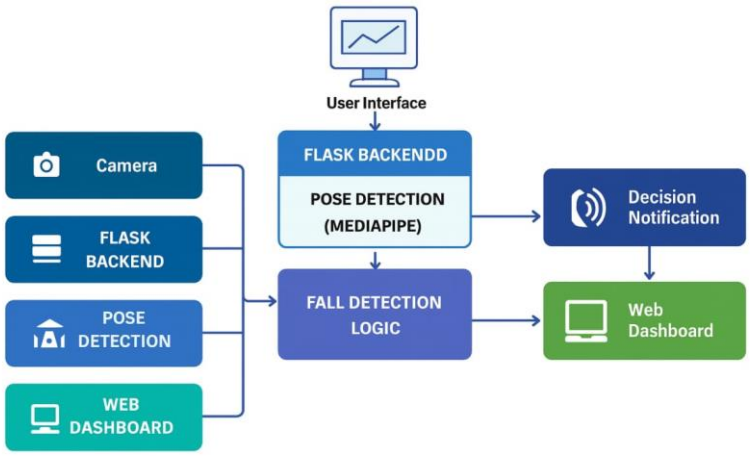
1. No wearable devices required
2. Real-time detection
3. Low-cost hardware
4. Runs on basic computing systems
5. Non-intrusive elderly monitoring
6. Fully automated alert system
7. Dashboard for caregivers

CONCLUSION

VISIONGUARD demonstrates a practical and efficient AI-based solution for elderly fall detection using computer vision. By integrating pose estimation with fall-logic algorithms, the system provides accurate and real-time monitoring without sensors. The lightweight design enables easy deployment in homes and hospitals. This work establishes a strong foundation for future healthcare automation using vision-based AI.

DIAGRAM:

VISIONGUARD: AI-BASED FALL DETECTION AND ALERT SYSTEM



REFERENCES

1. Google Developers Official Documentation.
2. The Pallets Project Official Documentation.
3. OpenCV Official Documentation.
4. W3C Web Technologies Documentation.
5. Python Software Foundation Official Documentation.
6. Bootstrap Official Documentation.