AI-Powered Human Pose Estimation for Real Time Applications

[1] Dr. P. S. Naveen Kumar, [2]Shaik Adil Basha, [3]Shaik Shahul,
[4]Vavilapalli Ashok Kumar, [5]Mothukuru Sai Tharun
[1]Associate Professor, Dept. of CSE - AI & ML, St.Ann's College Of Engineering & Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India
[2,3,4,5]U. G Student, Dept. of CSE - AI & ML, St.Ann's College Of Engineering & Technology, Nayunipalli (V), Vetapalem (M), Chirala, Bapatla Dist, Andhra Pradesh – 523187, India

ABSTRACT

Human pose estimation plays a vital role in real-time applications such as fitness monitoring, healthcare analysis, and human–computer interaction. However, many existing systems depend on expensive sensors or complex deep learning models that increase computational cost and latency. This project presents an AI-powered human pose estimation system designed for real-time applications using standard webcam input. The system is developed using Python, Flask, OpenCV, and MediaPipe to efficiently detect and track human body movements without the need for specialized hardware. The proposed method identifies 33 skeletal keypoints and applies a geometric heuristic approach to compute joint angles using Euclidean distance for accurate pose classification. Actions such as standing, sitting, squatting, and T-pose are recognized in real time and visualized through an interactive dashboard. Experimental results show that the system achieves accurate pose detection with low latency and minimal computational overhead. The proposed solution reduces manual effort, eliminates intrusive wearable devices, and provides a secure, user-friendly, and efficient platform for real-time human pose estimation.

Key words:

Human Pose Estimation, MediaPipe, Computer Vision, Real-Time Applications, Action Recognition

INTRODUCTION

In recent years, computer vision techniques have been increasingly used to analyse human body movements for real-time applications. Human pose estimation focuses on identifying body posture and joint positions from images or video streams. Traditional pose analysis methods such as manual observation or wearable sensor-based systems are often time-consuming, costly, and uncomfortable for continuous use.

These approaches may also depend on trained professionals or specialized hardware, which limits their accessibility. To overcome these limitations, this project presents an AI-powered human pose estimation system that analyses body movements using a standard webcam. The system is developed using Python, Flask, OpenCV, and MediaPipe to detect skeletal keypoints and evaluate human posture in real time.

By relying on vision-based techniques, the system removes the need for external sensors or wearable devices. The main objective of the proposed system is to provide a simple, efficient, and contactless method for pose tracking with real-time visualization. This system can support applications such as fitness monitoring, posture analysis, and basic activity recognition.

LITERATURE SURVEY

Research studies indicate that traditional methods of body movement analysis, such as manual observation and basic video recording, are often slow and subjective, which limits their practical use in real-time applications. To address these issues, several researchers have proposed automated pose estimation techniques based on computer vision.

Bazarevsky et al. (2020) introduced the BlazePose model, which follows a detector–tracker architecture to achieve fast and accurate skeletal landmark detection suitable for real-time environments. Lugaresi et al. (2019) presented the MediaPipe framework, which enables the development of efficient cross-platform pipelines for processing live video streams with low latency. Other studies have shown that analysing joint angles and skeletal relationships can support effective pose classification while reducing computational complexity. Recent research in fitness and activity monitoring applications highlights the use of vision-based systems that operate without specialised hardware.

RELATED WORK

The system development process involved setting up a Python-based environment with Flask used as the web framework. OpenCV and MediaPipe libraries were integrated to support skeletal landmark detection and pose processing. Webcam input was captured using OpenCV to stream live video frames for analysis. The backend was structured using Flask to manage routing and real-time data handling.

Pose detection modules were implemented to extract skeletal keypoints from video input. Joint angles were calculated using geometric methods to support pose analysis. Simple classification logic was applied to recognise specific body postures. Front-end interfaces were developed using HTML and CSS to display pose visualisation. A dashboard was included to allow users to view real-time pose output. Testing was carried out to verify correct pose detection and system stability.

EXISTING SYSTEM

Existing human movement analysis systems mainly rely on manual observation or sensor-based technologies. In many fitness and rehabilitation settings, posture assessment is performed visually by trainers or therapists, which depends heavily on individual judgement. This approach often leads to inconsistent and subjective results. Some systems use wearable sensors or markers to capture body movements, but these devices are costly and uncomfortable for users. They also require careful placement and calibration before each session. Advanced motion capture systems provide accurate results but depend on specialised laboratory equipment and controlled environments. Such systems are not suitable for regular or home-based use. Additionally, most existing solutions lack real-time feedback and easy accessibility for general users.

PROPOSED SYSTEM

The proposed system introduces a real-time human pose estimation approach to analyse body movements using a standard webcam. The system is developed using Python along with the MediaPipe framework to detect and track skeletal keypoints from live video input. A web-based interface is implemented using Flask, allowing users to access the system through a browser without additional software installation. The pose estimation process follows a lightweight detector–tracker approach, which helps reduce processing delay and computational load. Joint angles are calculated using simple geometric methods to analyse posture and body alignment. Based on these calculations, the system identifies basic actions such as standing, sitting, squatting, and T-pose. Real-time visualisation is provided by overlaying skeletal landmarks on the video stream. The proposed system removes the need for wearable sensors or specialised hardware. It offers an accessible and efficient solution for movement analysis in real-time applications.

SYSTEM ARCHITECTURE

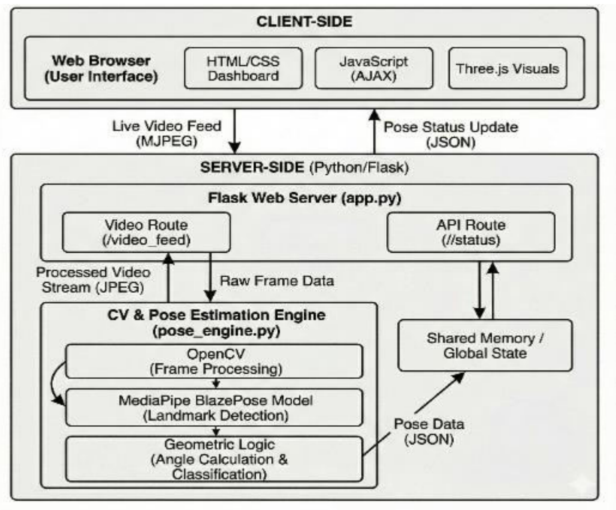

Fig:1 Ai-Powered Human Pose Estimation System Architecture

METHODOLOGY DESCRIPTION

CLIENT SIDE:

The Client Side acts as the user interface of the system and is developed using HTML, CSS, and JavaScript. It provides an interactive dashboard where users can view live video along with skeletal pose visualisation. The interface displays real-time pose status such as standing, sitting, or squatting without requiring page refresh.

API REQUEST/APIRESPONSE:

The system follows REST-based communication using HTTP GET requests. Live video is streamed from the server using an MJPEG stream, while pose status and classification results are sent as lightweight JSON responses. This approach ensures smooth synchronisation between video output and pose information.

SERVERSIDE:

The Server Side is implemented using Python and Flask, which handles routing, video streaming, and pose analysis logic. OpenCV is used to capture and process video frames, which are then passed to the MediaPipe BlazePose model for skeletal landmark detection.

LOGIC & PROCESSING:

The detected skeletal landmarks are processed using geometric calculations to determine joint angles. Based on these angle values, the system classifies different body postures in real time. This method avoids complex models and supports efficient real-time performance.

STATE MANAGEMENT:

To manage data flow efficiently, the system uses an in-memory data structure to store the latest pose information. This structure temporarily holds the current pose label and related values generated during processing. By storing only the most recent data, the system ensures fast access and real-time updates. The stored information is shared with the API to provide the latest pose status to the user interface without delay.

RESULTS AND DISCUSSION

SIGN IN/SIGN UP PAGE:

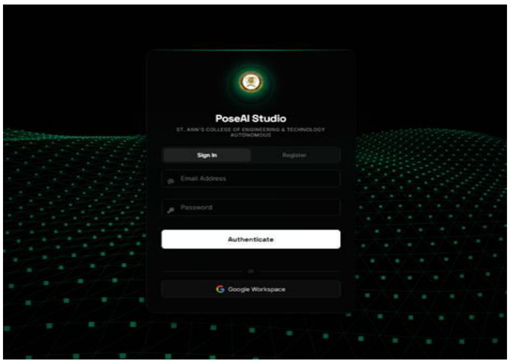

Fig :2 Sign In/Sign Up Page

Fig:2 shows the login page of the system, which is displayed when the application is accessed. This page is used for user authentication and ensures that only authorised users can enter the system. The interface is simple, allowing users to log in quickly without complexity.

HOME PAGE:

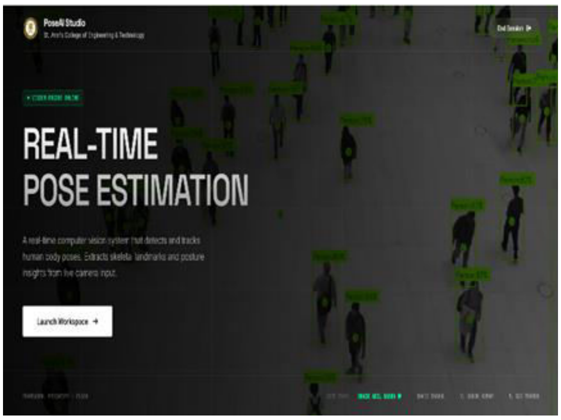

Fig :3 Home Page

After successful authentication, the user is redirected to the home page as shown in Fig:3. This page acts as the main landing interface and provides an overview of the system. It allows users to navigate to different features related to pose estimation.

USER DASHBOARD:

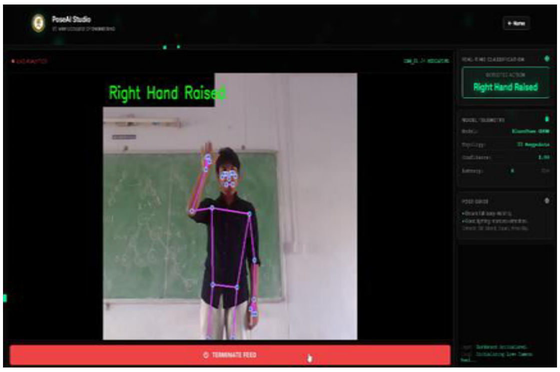

Fig 4: User Dashboard

Both administrators and students can access this dual login page for the attendance system. While administrators have a different login option below, students can enter their credentials. If new users don't have an account yet, the interface makes it easy for them to sign up.

REAL-TIME POSE DETECTION:

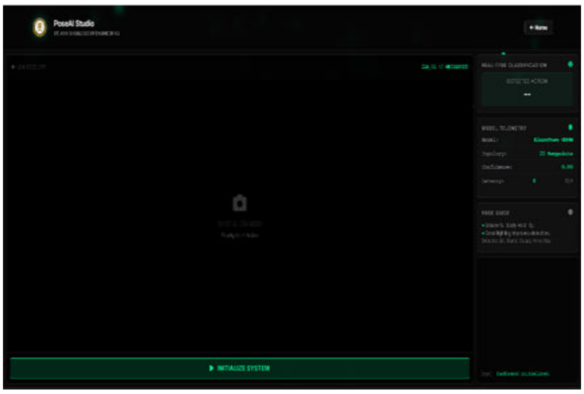

Fig 5: Real-Time Pose Detection

Fig:5 presents the real-time pose detection results of the system. The live video feed is displayed with skeletal landmarks overlaid using MediaPipe. The system accurately identifies body postures such as standing, sitting, and squatting. The results demonstrate smooth processing and minimal delay during real-time operation.

CONCLUSION

The AI-Powered Human Pose Estimation system successfully provides a practical solution for analysing human body movements in real time. The system uses computer vision techniques to detect skeletal landmarks without relying on manual observation or specialised hardware. By implementing Python, Flask, OpenCV, and MediaPipe, the system is able to track body posture accurately using a standard webcam. The results show that basic poses such as standing, sitting, and squatting are identified correctly with stable performance. The lightweight design ensures smooth operation on common devices without high computational requirements. Overall, the proposed system offers an accessible and efficient approach for real-time pose analysis and can be useful for applications such as fitness monitoring and basic movement assessment.

FUTURE SCOPE

In the future, this system can be improved by recognising a wider range of human movements and postures. The accuracy of pose detection can be refined by analysing movement patterns over time instead of relying only on single frames. The system can also be adjusted to work better in different lighting and background conditions. Additional features such as guided feedback for posture correction may be included. The application can be adapted for use in areas like physiotherapy, fitness training, and basic movement assessment. These enhancements can make the system more practical for everyday use.

REFERENCE

1. Bazarevsky et al., 2020 – Introduced BlazePose, a lightweight real-time human pose estimation model suitable for on-device skeletal tracking.
2. Lugaresi et al., 2019 – Proposed the MediaPipe framework for building efficient perception pipelines for real-time video processing.
3. Cao et al., 2017 – Presented OpenPose, a real-time multi-person 2D pose estimation method using part affinity fields.
4. Toshev and Szegedy, 2014 – Introduced DeepPose, an early deep learning-based approach for human pose estimation.
5. Bradski, 2000 – Described the OpenCV library used for real-time computer vision and image processing applications.
6. Simonyan and Zisserman, 2014 – Proposed a two-stream convolutional network for recognising human actions in video sequences.
7. Grinberg, 2018 – Explained the Flask web framework for developing lightweight Python-based web applications.
8. Yadav et al., 2021 – Reviewed multimodal human activity recognition techniques with emphasis on deep learning methods.