AI Yoga Pose Detection for Tree Pose

1. Approach

1.1 Objective

The goal of this project is to create a real-time human pose detection system that provides feedback on posture and movement using a webcam. The system analyzes body angles (such as leg and spine angles) and tracks breathing patterns to give corrective suggestions. This system uses MediaPipe for pose detection, OpenCV for video capture and display, and NumPy for mathematical operations like angle calculations.

1.2 Key Features

- **Real-time Pose Detection**: Detect human body landmarks (joints like hip, knee, ankle, shoulder) using MediaPipe's Pose module.
- **Angle Calculation**: Compute angles between key body joints (e.g., leg angle, spine angle) to assess posture.
- **Posture Feedback**: Provide feedback on whether the user's leg or spine is correctly aligned or not.
- **Breathing Analysis**: Monitor chest movements to assess whether the user's breathing is steady or irregular.
- **Smooth Feedback**: Use buffers to smooth the calculated angles and breathing data to avoid erratic results.

2. Data Preprocessing

2.1 Pose Detection with MediaPipe

MediaPipe Pose is used to detect human body landmarks in the real-time video feed. The model identifies key body joints such as shoulders, elbows, wrists, hips, knees, and ankles.

Steps:

- **Video Frame Conversion:** The captured frame from the webcam is first converted from BGR to RGB as MediaPipe works with RGB images.
- **Pose Detection:** The pose.process() function is used to detect body landmarks.
- **Extracting Landmarks:** Once the landmarks are detected, the coordinates of specific body joints (like the left and right hips, knees, ankles, and shoulders) are extracted.

2.2 Angle Calculation

Angles between key body joints (such as the knee, hip, and ankle) are computed using trigonometry.

Steps:

- For each set of three points, the calculate_angle() function is used to compute the angle formed by those three points using the arctangent function. The angle is calculated in degrees.
- Example: The leg angle is calculated between the left hip, left knee, and left ankle.

Formula:

$$\text{angle} = \arctan\left(\frac{y_c - y_b}{x_c - x_b}\right) - \arctan\left(\frac{y_a - y_b}{x_a - x_b}\right)$$

This formula calculates the angle between points a, b, and c, where b is the midpoint.

2.3 Data Smoothing

To avoid erratic behavior in angle or breathing data, buffers (using collections.deque) are implemented for smoothing:

- **Leg Angle Smoothing**: The last 5 leg angles are stored in a deque buffer and averaged to give a smoothed angle.
- **Breathing Analysis Smoothing**: The vertical movement of the chest (measured from shoulder heights) is tracked in a buffer to detect the steadiness of breathing.

3. Model Architecture

The system does not use a traditional machine learning model but rather a pose detection pipeline integrated with logic for real-time feedback.

3.1 Pose Detection (MediaPipe)

The model architecture of MediaPipe Pose is based on deep learning models trained to detect human body landmarks in 2D from RGB images. It uses a Convolutional Neural Network (CNN) for feature extraction and joint detection.

- **Pose Estimation:** The MediaPipe Pose model takes an image as input and outputs the coordinates of various key points on the body (landmarks).

3.2 Posture Feedback System

The logic for posture feedback is implemented using:

- **Angle Calculation:** Using calculate_angle(), we calculate the angle of body parts like legs and spine.
- **Thresholds for Correct Posture:**
    - **Leg Angle:** The system checks if the leg is straight (between 165° and 180°) or if it's positioned incorrectly.
    - **Spine Angle:** The spine should have a certain alignment, and the feedback system checks if the angle falls within a certain range (165°-180°).
    - Breathing: The vertical movement of the chest is used to check if the user's breathing is steady (if the movement of the chest is significant enough).

3.3 Feedback Mechanism

The feedback system overlays messages on the video frame indicating:

- Whether the user's leg is straight or bent.
- Whether the user's spine is aligned.
- Whether the breathing is steady or irregular.

The feedback messages are color-coded:

- **Green:** Indicates correct posture or steady breathing.
- **Red:** Indicates incorrect posture or irregular breathing.

3.4 User Interface

The interface is created using OpenCV to display the webcam feed, along with:

- **Landmark visualizations:** Key body landmarks and connections are drawn on the frame.
- **Text feedback:** Posture and breathing feedback are displayed on the frame.

4. Results

4.1 Positional Feedback

- **Leg Angle**: The system correctly detects when the leg is straight (within a range of 165° to 180°) and when the leg is too bent (below 90° or above 140°).
- **Spine Alignment**: The system provides feedback on spine alignment when the angle between the shoulder, hip, and ankle falls outside the desired range (165° to 180°).

4.2 Breathing Analysis

- The system successfully tracks the chest's vertical movement to assess whether the breathing is steady. When the difference in the chest's vertical position exceeds a threshold (0.02 units), it provides feedback on steady breathing.

4.3 Real-Time Feedback

- The system provides instant feedback, displayed on the video feed, allowing users to adjust their posture and breathing in real-time.

4.4 Accuracy and Responsiveness

- The pose detection and angle calculations work reliably in good lighting conditions. However, the accuracy of the system depends on the webcam resolution and the pose's visibility.
- The buffer smoothing for angles and breathing helps avoid false feedback, especially in fast movements.

5. Next Steps

5.1 Improving Accuracy

1. **Lighting Conditions:** The system is sensitive to lighting; poor lighting might result in inaccurate pose detection. Using a better lighting setup or implementing techniques to handle low-light conditions could enhance the results.

2. **Higher-Resolution Cameras:** The accuracy of body landmark detection improves with higher resolution. A higher-quality camera could be used for better detection, especially in large-scale applications.

5.2 Enhancing Breathing Analysis

The current breathing analysis is based on vertical shoulder movement. Future work could incorporate more advanced techniques, such as detecting chest expansion using RGB-depth cameras or integrating sensor-based solutions (e.g., wearable sensors) for more accurate breathing monitoring.

5.3 Posture Training Mode

The feedback system can be expanded into a **posture training mode**, where users can follow specific exercises and receive feedback on their form, such as during yoga or physical therapy.