

PosePerfectAI: AI-Based Yoga Pose Detection & Personal Trainer

Guide: Ms. Rohini Wankhede

Ritik Jaiswal¹, Alisha Md Naz Baddulla Sheikh ,Ganesh Londhe¹, Sakshi Deshkar¹, Prashant Singh Chauhan¹

Department of Computer Technology,
Priyadarshini College of Engineering, Nagpur, Maharashtra, India

Abstract - PosePerfectAI is an AI-based mobile application developed to analyze yoga poses using computer vision. The application allows users to select a yoga pose and perform it in front of a mobile camera. The captured frame is sent to a Python-based AI engine through a FastAPI REST API, where OpenCV is used for image processing and MediaPipe is used to detect human body landmarks. Based on the detected landmarks, the system evaluates the user's posture and provides a pose score, feedback, status, and hold-time information. The mobile application is developed using React Native and TypeScript to provide a simple and user-friendly interface. The current implementation supports the analysis of TREE, WARRIOR, and CHAIR poses. PosePerfectAI aims to make yoga practice more accessible by providing posture-related guidance through a mobile device.

Key Words: Yoga Pose Detection, Computer Vision, MediaPipe, OpenCV, FastAPI, React Native.

1.INTRODUCTION

Yoga is a widely practiced form of exercise that helps improve flexibility, balance, strength, and overall physical fitness. Performing yoga poses with proper body posture is important because incorrect posture can reduce the effectiveness of the exercise. However, beginners may find it difficult to understand whether they are performing a pose correctly without guidance from an instructor.

With the development of computer vision and artificial intelligence, cameras can be used to analyze human body movements and provide posture-related information. Pose estimation techniques can detect important body points, known as landmarks, from an image or camera frame. This information can then be used to study the position of different parts of the human body during a yoga pose.

PosePerfectAI is an AI-based mobile application

developed to analyze yoga poses using computer vision. The application allows the user to select a yoga pose and use the mobile camera while performing it. The captured frame is sent to a Python-based AI engine through a FastAPI REST API. OpenCV is used for image processing, while MediaPipe is used to detect human body landmarks. The detected landmarks are then analyzed to evaluate the selected yoga pose.

The system provides a pose score along with feedback and status information. It also provides hold-time information as part of the analysis result. The current implementation

includes analysis for TREE, WARRIOR, and CHAIR poses. The mobile application is developed using React Native and

TypeScript, allowing the AI analysis to be connected with a mobile user interface.

The current system has successfully implemented the camera-to-AI analysis pipeline for individual captured frames. The next stage of development is to extend this pipeline toward continuous camera-based tracking so that the score and feedback can be updated while the user is performing the pose.

2. PROBLEM STATEMENT

Many people practice yoga without continuous guidance from an instructor. Beginners may have difficulty understanding whether their body posture is correct while performing a yoga pose. Manual checking of posture also requires a trainer and may not always be available. A simple camera-based system can help by analyzing the user's posture and giving basic feedback.

The problem addressed in this project is to develop a mobile application that can analyze selected yoga poses using computer vision and provide useful posture-related information to the user. The system should detect body landmarks from the camera input, analyze the selected pose, and provide a score, feedback, status, and

hold-time information.

PosePerfectAI is developed to connect a mobile camera with a Python-based AI engine. The mobile application sends the captured frame to the FastAPI server, where OpenCV and MediaPipe are used for image processing and pose landmark detection. The analyzed result is then returned to the mobile application and displayed to the user.

The current system supports analysis of the TREE, WARRIOR, and CHAIR poses. The project also aims to extend the current single-frame analysis toward continuous camera-based tracking for live pose feedback.

3. PROJECT OBJECTIVES

The main objective of PosePerfectAI is to develop a mobile-based system that uses computer vision to analyze yoga poses and provide posture-related feedback to the user.

The main objectives of the project are:

1. To develop a mobile application that allows users to select and practice different yoga poses.
2. To use the mobile camera to capture the user's body posture during yoga practice.
3. To use MediaPipe for detecting human body landmarks from the captured camera input.
4. To use OpenCV for image processing before and during pose analysis.
5. To develop a Python-based AI engine that analyzes the detected body landmarks and evaluates the selected yoga pose.
6. To connect the React Native mobile application with the Python AI engine using a FastAPI REST API.
7. To provide the user with a pose score, feedback, status, and hold-time information after analysis.
8. To support the currently implemented yoga poses: TREE, WARRIOR, and CHAIR.
9. To extend the current captured-frame analysis toward continuous camera-based pose tracking for live feedback.
10. To provide a simple and user-friendly system that can assist users while practicing yoga.

4. LITERATURE REVIEW

Several studies have been carried out on yoga pose detection and analysis using computer vision, artificial intelligence, deep learning, and human body landmarks. These studies show that computer vision can be used to recognize yoga poses and analyze body posture. Some

of the important studies related to this project are discussed below.

S.-C. Yeh and C.-K. Yang [1] proposed a system titled *Yoga Pose Recognition and Motion Analysis for a Home-Based Fitness Monitoring System* in 2025. The study focused on recognizing yoga poses and analyzing body motion for home-based fitness monitoring. The use of computer vision and AI helps in monitoring the user's movements during exercise.

D. Santhadevi et al. [2] presented *Yoga Pose Detection and Correction Using MediaPipe Landmarks and a Deep Multi-Layer Perceptron Classifier* in 2025. The study used MediaPipe landmarks along with a Deep Multi-Layer Perceptron classifier for yoga pose detection and correction. The approach focuses on using body landmark information to identify poses and provide posture-related feedback.

R. Bansal et al. [3] proposed *DeepYoga: Enhancing Yoga Practice with a Real-Time Yoga Pose Detection System* in 2024. The study used deep learning and computer vision techniques for real-time yoga pose detection. The work focuses on using intelligent techniques to assist users during yoga practice.

I. Chaudhary et al. [4] presented *Real-Time Yoga Pose Detection Using OpenCV and MediaPipe* in 2023. The study used Python, OpenCV, and MediaPipe for detecting yoga poses from camera input. This work is closely related to PosePerfectAI because it demonstrates the use of MediaPipe and OpenCV for yoga pose analysis.

From these studies, it can be observed that computer vision and body landmark detection are useful for yoga pose recognition and posture analysis. The reviewed works provide a foundation for developing systems that can assist users during yoga practice. PosePerfectAI uses a similar computer-vision approach with MediaPipe and OpenCV, while connecting the AI processing system with a React Native mobile application through FastAPI. The system is designed to provide a pose score, feedback, status, and hold-time information to the user.

5. PROPOSED SYSTEM

PosePerfectAI is proposed as a mobile-based yoga pose analysis system that combines computer vision with a simple mobile interface. The main purpose of the system is to help users understand their yoga posture by analyzing camera input and providing basic feedback.

The proposed system consists of two main parts: a **React Native mobile application** and a **Python-based AI engine**. The mobile application is responsible for the

user interface, pose selection, and camera input. The AI engine is responsible for processing the captured frame and analyzing the user's body posture.

When the user selects a yoga pose and starts the camera, the application captures an image or frame of the user's posture. The frame is sent to the Python AI engine through the FastAPI REST API. OpenCV is used for image processing, and MediaPipe is used to detect the body landmarks from the input.

The detected landmarks are analyzed according to the selected yoga pose. The system then generates information such as the selected pose, score, feedback, status, and hold time. This result is returned in JSON format through the API and displayed in the React Native application.

The currently implemented system supports **TREE**, **WARRIOR**, and **CHAIR** pose analysis. The current implementation uses captured-frame analysis, and the system is being developed further toward continuous camera-based tracking for live pose feedback. This proposed architecture separates the mobile application from the AI processing engine, making the two components easier to develop and test independently.

6. SYSTEM ARCHITECTURE

The system architecture of PosePerfectAI consists of a mobile application and a separate Python-based AI engine. The mobile application is developed using React Native and TypeScript, while the AI processing is handled using Python. FastAPI is used as the communication layer between the mobile application and the AI engine.

When the user selects a yoga pose, the camera is opened through the mobile application. The user performs the selected pose in front of the camera and an image or frame is captured. This captured input is sent to the FastAPI server for analysis.

The FastAPI server receives the image and passes it to the Python AI engine. OpenCV is used for image processing, while MediaPipe detects the body landmarks from the captured frame. These landmarks are then analyzed according to the selected yoga pose.

After the analysis, the system generates the pose score and related feedback. The result also contains the status and hold-time information. The processed result is returned to the mobile application in JSON format and displayed on the user interface.

The architecture keeps the mobile application and AI engine as separate components. This allows the user interface and AI processing to be developed and tested

independently. The current system supports **TREE**, **WARRIOR**, and **CHAIR** pose analysis.

The architecture also makes it easier to update the AI engine without making major changes to the mobile application. This modular design helps in testing each part of the system separately and improving the system in future.

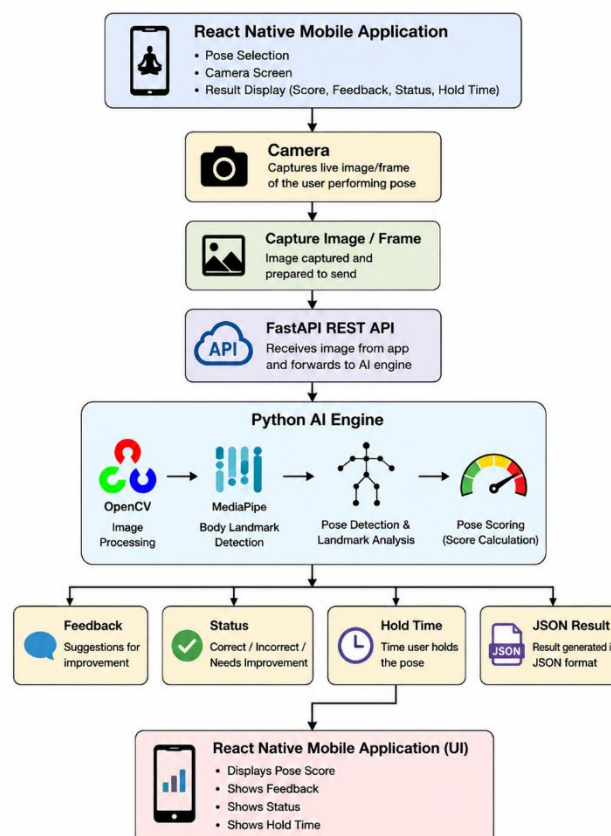


Figure 1: System Architecture of PosePerfectAI

7. RESEARCH METHODOLOGY

The methodology of PosePerfectAI is based on the integration of a mobile application, camera input, a REST API, and a Python-based computer vision system. The system is designed to capture the user's yoga posture and analyze the selected pose using body landmarks.

7.1 User Pose Selection

The process starts with the React Native mobile application. The user selects a yoga pose from the available poses and opens the camera screen. The selected pose is used by the AI engine during the analysis.

7.2 Image Capture

After opening the camera, the user performs the selected yoga pose in front of the camera. The application

captures an image or frame containing the user's posture. The current implementation uses captured-frame analysis as the first stage of the camera-to-AI pipeline.

7.3 Image Processing

The captured frame is sent to the Python AI engine through the FastAPI REST API. OpenCV is used for handling and processing the image before pose analysis.

7.4 Pose Landmark Detection

MediaPipe is used to detect human body landmarks from the input frame. These landmarks represent important points of the user's body and provide the information required for posture analysis.

7.5 Pose Analysis and Scoring

The detected landmarks are analyzed according to the selected yoga pose. The AI engine evaluates the pose-related information and generates a score. The analysis also generates feedback, status, and hold-time information.

7.6 Result Transmission

After the analysis is completed, the result is returned from the Python AI engine through the FastAPI API in JSON format. The React Native application receives the result and displays the information to the user.

The current implementation has been tested for TREE and WARRIOR pose analysis, while CHAIR pose analysis is also present in the system. The methodology is currently based on captured-frame analysis, with continuous camera-based tracking being the next development stage.

8. IMPLEMENTATION

The implementation of PosePerfectAI is divided into two main components: the React Native mobile application and the Python-based AI engine. Both components work together through a FastAPI REST API. This separation allows the mobile application to handle the user interface and camera operations, while the Python engine performs the pose analysis.

8.1 Mobile Application

The mobile application is developed using React Native and TypeScript. It provides the user interface for selecting a yoga pose, opening the camera, and viewing the analysis result. The application also includes navigation between the different screens of the system. React Native Vision Camera is used to access the mobile device camera. The application supports both front and back cameras, and the user can switch between them. Camera permission is also handled by the application.

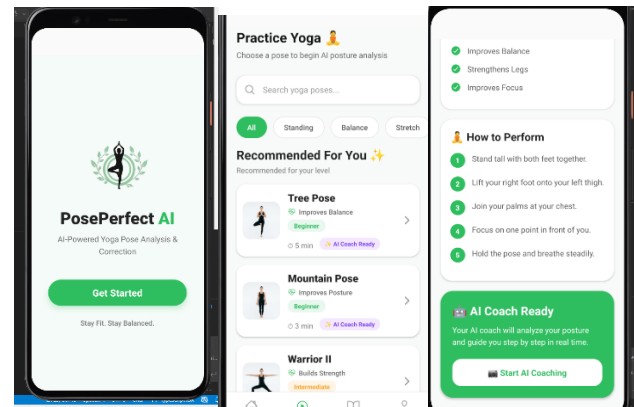


Figure 2: PosePerfectAI Mobile Application Interface

8.2 FastAPI Server

FastAPI is used as the communication layer between the React Native application and the Python AI engine. The server receives the captured image and selected pose from the mobile application and sends the request for analysis.

The implemented API provides endpoints including /, /health, /result, and /analyze-frame. The /analyze-frame endpoint accepts an image and pose name for analysis. The API returns the analysis result in JSON format.

8.3 Python AI Engine

The AI engine is developed using Python. OpenCV is used for image processing, while MediaPipe is used for detecting human body pose landmarks. The detected landmarks provide the input required for evaluating the selected yoga pose.

The AI engine processes the captured frame and generates information including the pose name, score, feedback message, status, and hold time.

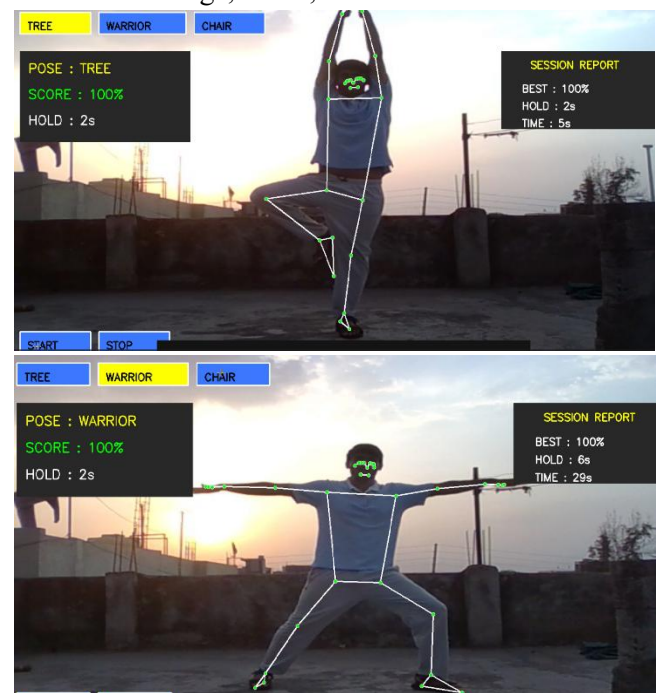


Figure 3: AI-Based Yoga Pose Analysis Output

8.4 Pose Analysis

The current implementation contains analysis for three yoga poses: **TREE**, **WARRIOR**, and **CHAIR**. **TREE** pose analysis has been successfully tested through the API, and **WARRIOR** analysis has also been tested. **CHAIR** pose analysis is present in the system.

The analysis result is returned to the mobile application and displayed to the user. For example, the result can contain the selected pose, score, feedback key, status, and hold time.



Figure 4: Camera-to-AI Pose Analysis Result

8.5 Current Camera-to-AI Pipeline

The current implementation follows a captured-frame pipeline. The camera captures an image, the image is sent to the analyze-frame endpoint, the Python AI engine processes the input, and the resulting information is returned to the React Native application.

This implementation provides the basic connection between the mobile camera and the AI engine. The next development stage is to extend the captured-frame process into continuous camera-based tracking for live pose analysis.

9. WORKING OF THE PROPOSED SYSTEM

The working of PosePerfectAI involves the interaction between the mobile application, camera, FastAPI server, and Python-based AI engine. The complete process starts when the user selects a yoga pose and continues until the analysis result is displayed.

1.Application Start: The user opens the PosePerfectAI mobile application. The home screen provides options to start a practice session and explore available yoga poses.

2.Pose Selection: The user selects a yoga pose from the available options. The current system supports **TREE**, **WARRIOR**, and **CHAIR** poses for analysis.

3.Pose Information: Before starting the practice, the application provides information about the selected pose, including its benefits, difficulty level, recommended duration, and instructions for performing the pose.

4.Camera Start: The user starts the practice session and positions themselves in front of the camera. The user should keep the body visible within the camera frame

for proper landmark detection.

5.Frame Capture: The camera captures an image or frame of the user performing the selected yoga pose.

6.API Communication: The captured frame and selected pose are sent from the mobile application to the FastAPI /analyze-frame endpoint.

7.Image Processing: The Python AI engine receives the captured frame. OpenCV is used for image processing, and MediaPipe is used to detect the body landmarks.

8.Pose Analysis: The detected body landmarks are analyzed according to the selected yoga pose. The system evaluates the posture and generates a pose score.

9.Result Generation: After the analysis, the system generates information such as the pose name, score, feedback, status, and hold time. The result is returned through the FastAPI server in JSON format.

10. Result Display: The React Native application receives the result and displays the analysis information to the user. The user can view the score, feedback, status, and hold-time information.

The current implementation provides a captured-frame analysis pipeline between the mobile application and the Python AI engine. The system supports analysis of **TREE**, **WARRIOR**, and **CHAIR** poses. Continuous camera-based tracking for live pose feedback is considered as the next stage of development.

10. EXPERIMENTAL SETUP AND RESULTS

The PosePerfectAI system was tested using camera input to check the working of the yoga pose analysis pipeline. The testing focused on the communication between the mobile application and the Python-based AI engine, pose landmark detection, pose analysis, and generation of the final result.

The system was tested using the currently supported **TREE**, **WARRIOR**, and **CHAIR** poses. During testing, the user performed the selected pose in front of the camera, and the captured frame was processed by the Python AI engine. MediaPipe detected the body landmarks, and the detected landmarks were used for pose analysis.

The analysis result included the selected pose, pose score, hold time, and session information. The results were successfully displayed by the system for the tested poses.

During the available test cases, the system produced the following outputs:

Pose	Score Shown	Hold Time Shown	Result
TREE	100%	2 s	Pose detected and analyzed
WARRIOR	100%	2 s	Pose detected and analyzed
CHAIR	100%	1 s	Pose detected and analyzed

The screenshots of these outputs show the detected body landmarks along with the pose name, score, hold time, and session report. These results demonstrate that the implemented pipeline can process the captured input and return pose-analysis information to the user.

However, these individual test outputs should **not be considered as an overall accuracy measurement** of the system. A larger and systematic test dataset would be required to calculate metrics such as accuracy, precision, recall, and processing time.

The current testing is based on captured-frame analysis. Continuous camera-based tracking and live feedback can be evaluated separately after that functionality is completed.

11. ADVANTAGES AND LIMITATIONS

11.1 Advantages

The main advantages of the PosePerfectAI system are:

- Mobile-Based:** The system is designed as a mobile application, making it convenient for users to practice yoga.
- Computer Vision-Based Analysis:** The system uses MediaPipe and OpenCV to analyze the user's body posture from camera input.
- Pose Score:** The system provides a score to give the user an indication of the analyzed pose.
- Posture Feedback:** The system provides feedback related to the performed pose.
- Hold-Time Information:** The system records and displays information about the time for which the user holds the pose.
- Multiple Yoga Poses:** The current system supports TREE, WARRIOR, and CHAIR pose analysis.
- Separate AI Engine:** The Python-based AI engine is separated from the mobile application and communicates through a FastAPI REST API.
- Simple User Interface:** The application

provides pose information, instructions, practice options, and analysis results through a simple interface.

11.2 Limitations

The current implementation also has some limitations:

- Captured-Frame Analysis:** The current camera-to-AI pipeline is based on captured-frame analysis rather than continuous frame-by-frame analysis.
- Limited Number of Poses:** The current AI analysis is implemented for a limited set of yoga poses, mainly TREE, WARRIOR, and CHAIR.
- Camera Conditions:** The quality of pose detection can depend on factors such as camera position, lighting, and whether the user's body is clearly visible.
- Limited Experimental Data:** The current testing is based on individual test cases. A larger dataset and systematic testing would be required for detailed performance evaluation.
- Live Tracking:** Continuous real-time pose tracking and live feedback are planned as a future development stage.

These limitations identify areas that can be improved in future versions of PosePerfectAI. The current project documentation also identifies continuous camera-based tracking as the next development stage.

12. FUTURE SCOPE

PosePerfectAI can be further improved by adding more features and improving the existing pose analysis process. The current system provides captured-frame analysis, and the main future development is to make the analysis continuous so that users can receive feedback while performing a yoga pose.

The following improvements can be considered in future versions:

- Real-Time Pose Tracking:** The system can be extended to continuously process camera frames and update the pose score and feedback while the user is performing the pose.
- More Yoga Poses:** Additional yoga poses can be added to increase the range of exercises supported by the application.
- Improved Feedback:** More detailed posture feedback can be provided based on the detected body landmarks.
- Performance Evaluation:** A larger number of test cases can be conducted with different users and conditions to evaluate the system using

measurable performance metrics.

5. **Improved Hold-Time Tracking:** The hold-time feature can be further developed to provide more consistent tracking during continuous pose practice.
6. **Better Camera-Based Analysis:** The system can be optimized to handle different camera positions, lighting conditions, and user distances more effectively.
7. **Mobile AI Optimization:** In the future, parts of the AI processing could be optimized for mobile devices to reduce the dependency on communication between the mobile application and the external Python server.

These improvements can make PosePerfectAI more useful for regular yoga practice and provide a better user experience while keeping the system simple and accessible.

13. CONCLUSION

PosePerfectAI is an AI-based mobile application developed to assist users in analyzing yoga poses using computer vision. The system combines a React Native mobile application with a Python-based AI engine through a FastAPI REST API. OpenCV is used for image processing, while MediaPipe is used to detect human body landmarks for pose analysis.

The current implementation provides a complete captured-frame analysis pipeline from the mobile camera to the Python AI engine and back to the mobile application. The system currently supports TREE, WARRIOR, and CHAIR pose analysis and provides information such as pose score, feedback, status, and hold time.

The developed system shows how computer vision can be combined with a mobile application to provide posture-related information during yoga practice. The current results demonstrate the working of the implemented pose-analysis pipeline for the tested poses. However, further testing with more users and test cases is required before making broader performance claims.

The next major development stage is continuous camera-based pose tracking, which can allow the score and feedback to be updated while the user is performing the pose. With further development and systematic testing, PosePerfectAI can be extended to support more yoga poses and provide improved live posture feedback.

14. REFERENCES

- [1] S.-C. Yeh and C.-K. Yang, "Yoga pose recognition and motion analysis for a home-based fitness monitoring and health management system," *Signal, Image and Video Processing*, vol. 19, article 841, 2025, doi: 10.1007/s11760-025-04436-6.
- [2] D. Santhadevi, S. Yerigera, S. Aaditya, and B. R. K. Reddy, "Yoga pose detection and correction using MediaPipe landmarks and a deep multi-layer perceptron classifier," in *Proceedings of the 1st Engineering Data Analytics and Management Conference (EAMCON 2025)*, Advances in Engineering Research, pp. 332–351, 2025, doi: 10.2991/978-94-6463-978-0_30.
- [3] R. Bansal, R. Sharma, P. Jain, R. Arora, S. Pal, and Vishal, "DeepYoga: Enhancing Practice with a Real-Time Yoga Pose Recognition System," *Engineering, Technology & Applied Science Research*, vol. 14, no. 6, pp. 17704–17710, 2024, doi: 10.48084/etasr.8643.
- [4] I. Chaudhary, N. T. Singh, M. Chaudhary, and K. Yadav, "Real-Time Yoga Pose Detection Using OpenCV and MediaPipe," in *2023 4th International Conference for Emerging Technology (INCET)*, Belgaum, India, pp. 1–5, 2023, doi: 10.1109/INCET57972.2023.10170485.