

# Artificial Intelligence Based Interactive Fitness Trainer

Shreya Tale<sup>1</sup>, Sanket Wahane<sup>2</sup>, Sahil Mandiya<sup>3</sup>, Sunil R Gupta<sup>4</sup>

<sup>1,2,3,4</sup> *Department of Artificial Intelligence & Data Science*

<sup>1,2,3,4</sup> *Prof. Ram Meghe Institute of Technology & Research, Badnera, Amravati*

**Abstract-** Due to the global spread of sedentary lifestyles, which has been made worse by the digitalization of work and the post-pandemic shift toward remote living. Although digital fitness platforms have made instructional content more accessible to a wider audience, they have historically been unable to offer real-time, corrective feedback on biomechanical form. This shortcoming raises the risk of injury and decreases the effectiveness of training [9]. The architectural layout, theoretical underpinnings, and implementation plan for an interactive fitness trainer powered by artificial intelligence (AI) are presented in this research paper. The suggested system uses consumer-grade hardware without specialized graphical processing units (GPUs Accessible, scalable, and individualized physical fitness interventions are desperately needed) to perform high-fidelity, monocular human pose estimation (HPE) using the cutting-edge MediaPipe BlazePose framework. The system incorporates geometric heuristics for precise repetition counting, a finite state machine (FSM) Many people are exercising without professional supervision as a result of the growing popularity of at-home fitness regimens, which frequently leads to poor posture and a higher risk of injury. The majority of fitness apps currently in use rely on pre-recorded videos and don't provide real-time interaction or helpful feedback. This paper presents an AI-based Interactive Fitness Trainer that analyzes user posture in real time using camera input and provides corrective feedback using computer vision and machine learning techniques. The proposed system uses MediaPipe BlazePose for pose estimation, Flask or Streamlit for an interactive user interface, and machine learning algorithms such as Decision Tree, Logistic Regression, and Random Forest for exercise recognition. Platform-neutrality, affordability, and accessibility are the goals of the system [5].

**Keywords—** Artificial Intelligence, Fitness Trainer, Pose Estimation, MediaPipe, Computer Vision, Machine Learning, Real-Time Feedback

## I. INTRODUCTION

The proposed system uses MediaPipe BlazePose for pose estimation, Flask or Streamlit for an interactive

user interface, and machine learning algorithms such as Decision Tree, Logistic Regression, and Random Forest for exercise recognition. The system is designed to be platform-neutral, accessible, and reasonably priced [5].

The intersection of human physiology and artificial intelligence (AI) is one of the most innovative fields in modern healthcare and wellness. In a time when sedentary lifestyles and the digitization of daily routines are becoming more and more prevalent, the global fitness landscape has undergone a paradigm shift. The World Health Organization (WHO) states that physical inactivity is a significant global risk factor for non-communicable diseases and mortality, and it advocates for scalable and easily accessible interventions [11].

As more people started exercising at home, a significant gap emerged. Digital platforms could provide content, such as static workout plans or streaming videos of instructors, but they lacked the essential feedback loop that is essential to good coaching: the capacity to watch, evaluate, and adjust form in real time [15]. Without this feedback, at-home exercisers are more likely to sustain musculoskeletal injuries as a result of poor technique, and they frequently get less than ideal results, which lowers motivation and adherence.. This specific gap is filled by the "AI-Based Interactive Fitness Trainer" that has been proposed. This study suggests a system that can democratize access to expert-level fitness advice by utilizing the quick developments in computer vision, particularly Human Pose Estimation (HPE) [3].

This paper describes the creation of an all-encompassing AI fitness framework that uses a regular webcam to record, evaluate, and adjust human movement in real time. Unlike early iterations of fitness technology that relied on wearable sensors or

depth cameras (like Microsoft Kinect), which imposed hardware barriers, the proposed system is based on widely used monocular RGB camera technology [18]. It employs rule-based logic to provide instantaneous corrective feedback, integrates state-of-the-art deep learning models such as MediaPipe BlazePose to extract high-fidelity skeletal landmarks [19], and uses geometric algorithms to evaluate biomechanical alignment.

There is more to this study than just counting repetitions. It includes the scalable, privacy-preserving, and performance-based architectural design of a fitness assistant. This work combines expertise from computer science, biomechanics, and sports physiology to try to create a dependable method for automated fitness coaching that is accurate enough for meaningful physiological intervention and computationally efficient for edge deployment [3].

## II. LITERATURE REVIEW

To illustrate the proposed system's novelty and technical rigor, it is essential to situate it within the broader framework of automated fitness tracking. This field has developed from basic sensor-based counting to complex, vision-based kinematic analysis [5,6].

### 2.1 The Era of Wearables and Depth Sensors

Early attempts at automated exercise tracking heavily relied on wearable accelerometers and inertial measurement units (IMUs). These devices, which were commonly built into smartwatches or chest straps, used acceleration and angular velocity measurements to determine movement. While Twenty Wearables were helpful for counting rhythmic, cyclic movements like walking or running steps, they had significant limitations when it came to resistance training. Because a sensor on the wrist cannot accurately infer the position of the hips or knees, they had difficulty capturing complex postural deviations, such as a rounded spine during a deadlift or knee valgus during a squat [20].

An important step toward vision-based analysis was taken with the release of the Microsoft Kinect. By creating a 3D skeleton of the user using infrared depth sensors, the Kinect made it possible for the first

"exergaming" and automated. However, its widespread use in the general home fitness market was hindered by the need for specialized, costly hardware, which restricted its adoption to clinical settings and gaming enthusiasts.

### 2.2 The Deep Learning Revolution: OpenPose vs. MediaPipe

Human Pose Estimation (HPE) was transformed with the introduction of Convolutional Neural Networks (CNNs), which eliminated the need for depth sensors by enabling skeletal tracking using common RGB cameras. There were two prevailing frameworks that had different architectural philosophies:

Feature	OpenPose	MediaPipe BlazePose
Approach	Bottom-Up: Detects all keypoints in the image first, then groups them into individuals.	Top-Down: Detects a person first (ROI), then predicts landmarks within that region.
Multi-Person	Excellent. Computation time is relatively invariant to the number of people.	Limited. optimized for single-person tracking; separate instances needed for multiple people.
Computational Cost	High. Typically requires a dedicated GPU (NVIDIA CUDA) for real-time performance.	Low. Optimized for CPU and mobile inference (TFLite).

Topology	18 or 25 keypoints (COCO format).	33 keypoints (BlazePose topology), including hands and feet.
Use Case	Crowd surveillance, team sports analytics.	Personal fitness apps, mobile interactions, gesture control.

The trade-off between these systems is highlighted in existing literature. Dsouza et al. proposed a "Smart Gym Trainer" that utilizes OpenPose. 9. The authors noted that the system's high computational load hindered its deployment on personal devices, despite its high accuracy in joint angle estimation. The "AI Fitness Coach" by Ji Haoran et al. 25 and the "Fitcercise" system by Lamba et al. 8, on the other hand, used lighter models but frequently sacrificed the granularity of feedback, emphasizing repetition counting.

### 2.3 Limitations of Prior Art

While systems like "Vyayam" successfully implemented bicep curl tracking using geometric heuristics, they often lacked a generalized framework for multi-joint compound movements. "Vyayam" focused on the elbow flexion angle, which is a relatively simple two-dimensional problem. However, exercises such as the squat require simultaneous monitoring of the alignment of the ankle, knee, hip, and spine. These exercises often include depth (z-axis) considerations that are hard to capture with simple 2D heuristics.26. Many modern solutions, such as the "NUFI" platform 27, incorporate wellness and diet but manage the exercise component in a mysterious manner, providing little information about the development or validation of the pose correction logic against biomechanical standards [7].

These gaps are filled by the suggested system by:

1. Utilizing MediaPipe BlazePose: 1. Making use of MediaPipe BlazePose: To ensure lag-free system performance on standard laptops and smartphones [14].
2. Putting into Practice Comprehensive Biomechanical Rules: It takes more than just counting to keep an eye on specific fault patterns (like knee cave and elbow flare) that are derived from sports science literature.
3. Privacy-Focused Architecture: processing all video data locally on the edge to address the growing concern over the privacy of biometric data [19].

### III. MATERIAL AND METHOD

This modular pipeline approach separates the tasks of data collection, pose estimation, geometric analysis, and feedback generation [12]. This ensures scalability, allowing new exercises to be added as distinct modules without having to rearrange the main engine.

3.1 System Architecture: The system architecture is shown using a sequential flow:

1. Input Layer: VideoCapture is used to capture video streams[1].
2. Preprocessing Layer: Frame resizing (typically to 256 \times 256 or 512 \times 512) and color space conversion (BGR to RGB) are carried out in order to meet MediaPipe's tensor requirements [6].
3. Inference Layer: The MediaPipe Pose solution extracts the normalized landmark list from the frame [3].
4. Logic Layer: Using vector mathematics to calculate angles for unprocessed landmarks.

State Machine: By examining temporal variations in angles, eccentric/concentric repetition states are monitored. Instantaneous angles are compared to biomechanical thresholds for fault detection [11].

5. Output Layer: OpenCV-rendered visual overlay (skeleton rendering) and user feedback.

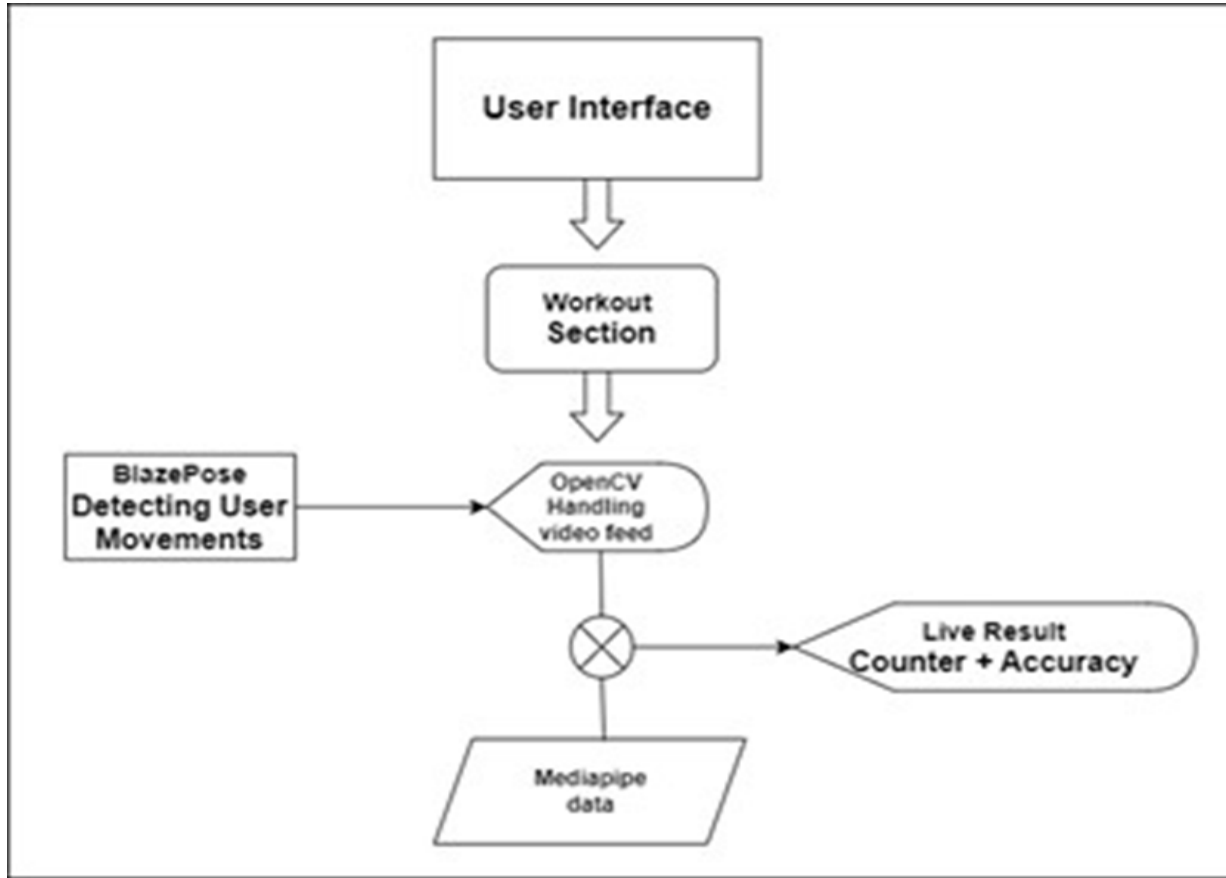


Figure 1-system architecture

### 3.2 Mathematical Model

The angle formed by three anatomical landmarks is the basis for the analysis. Three crucial points serve as representations of these landmarks: point A serves as the anchor point (for instance, the shoulder); point B denotes the joint at which the angle is measured (for instance, the elbow); and point C is the endpoint (for instance, the wrist). The Law of Cosines or the arctangent (atan2) function are used to determine the angle at joint B. Because it can accurately handle angles over the entire 360-degree range and avoid division-by-zero errors, the atan2 method is favored in real-world programming implementations [5].

The following formula is applied:

$$\text{Radians} = \text{atan2}(C_y - B_y, C_x - B_x) - \text{atan2}(A_y - B_y, A_x - B_x).$$

The value is multiplied to determine the angle in degrees. obtained in radians by dividing 180 by pi, and

then calculating the absolute value. The calculated angle is deducted from 360 degrees if it exceeds 180 degrees. By making this change, the final angle value is guaranteed to fall between 0 and 180 degrees [5].

### 3.3 Repetition Counting: The Finite State Machine (FSM)

To prevent false positives, such as when a user simply shakes their arm and that is interpreted as a curl, a Finite State Machine (FSM) is employed [16]. The exercise is modeled by this digital logic as a series of discrete states:

State 0 (Neutral/Start): The limb is extended (e.g., elbow angle  $\theta > 160^\circ$ ). When a rep begins, the system listens [15].

State 1 (Eccentric/Down): The user reduces the weight. The system verifies that the angle rises in the direction of the maximum threshold [17].

State 2 (Concentric/Up), the muscle is contracted by the user. A certain flexion threshold, such as  $< 30^\circ$  for a curl or  $< 90^\circ$  for a squat, must be met by the angle.

Action: Only when the state changes from certain triggers (such as passing the flexion threshold and then returning to the extension threshold) is a repetition increased.

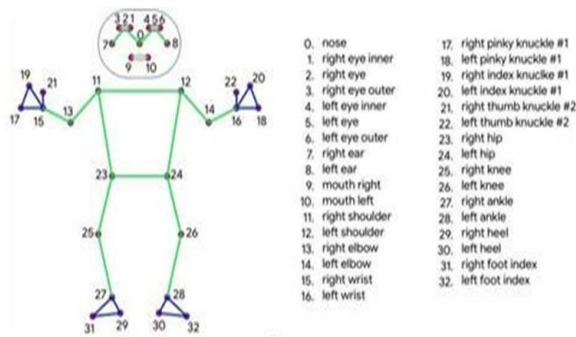


Figure 2-32 key points

### 3.4 Corrective Algorithm

1. Pose Estimation – BlazePose (by MediaPipe)  
 Algorithm Type: Deep Convolutional Neural Network  
 Description: BlazePose uses a two-stage pipeline: Detector: Finds the region of interest (ROI) containing a human. Tracker: Tracks 33 body landmarks using regression and heatmap predictions [15].

2. Repetition Counting : State Machine / Peak Detection  
 Algorithm Method: ••• Track movement of a joint (e.g., wrist or knee) vertically. Identify a full cycle using upward and downward thresholds. Count a rep when the movement crosses both thresholds in sequence [20].

3. Angle Calculation Between Joints (Rule-Based Algorithm) Formula Used:

$$\theta = \cos^{-1} \left[ \frac{A \cdot B}{|A| \cdot |B|} \right]$$

Where: •  $A \rightarrow, B \rightarrow$  are vectors formed by joint coordinates (e.g., shoulder  $\rightarrow$  elbow, elbow  $\rightarrow$  wrist) [7].

## IV. IMPLEMENTATION DETAILS

The implementation makes use of a Python-based ecosystem that was selected for its readability, quick prototyping, and wide library support [6].

### 4.1 Software Stack

- Programming Language: Python 3.8+
- Computer Vision Library: OpenCV (cv2) for acquiring, modifying, and creating the visual overlay [5].
- Machine Learning Framework: MediaPipe Python Solutions (mp.solutions.pose) for the BlazePose model that has already been trained.
- Numerical Processing: NumPy for array manipulation and high-performance vector calculations [9].
- User Interface: To create the dashboard that shows metrics, exercise selection menus, and historical data charts, use either Streamlit or Tkinter, a standard Python GUI [2,5].

### 4.2 Code Structure and Modularity

To guarantee the separation of concerns, the codebase is organized into object-oriented classes:

1. PoseDetector Class: It manages the pose model's initialization (static\_image\_mode=False, model\_complexity=1, smooth\_landmarks=True) and offers public functions like findPose() (which yields an image with landmarks) and findAngle() (which yields the computed angle between three indices).[14]
2. Exercise Class (Parent): Specifies an exercise's general format (name, rep count, state).
3. Specific Exercise Classes (Children): Bicep curls, squats, etc. These implement the particular calculate\_logic() methods that contain the distinct angle thresholds and feedback strings for that movement, and they inherit from the parent class.
4. Main Loop: The entry point that calls the relevant exercise logic, initializes the camera, instantiates the detector, records frames in a while loop, and renders the outcome to the screen [15].

### 4.3 Signal Smoothing and Jitter Reduction

Video feeds' raw landmark data may include "jitter," or high-frequency noise. The system uses a smoothing algorithm to stop the angle readout from flickering quickly. The raw coordinate data is subjected to a straightforward Moving Average Filter or a One-Euro Filter over a brief temporal window (e.g., 5 frames). This ensures that the user will see angles that are stable and readable [12].

## 5. RESULTS AND PERFORMANCE EVALUATION

To fully validate the "AI-Based Interactive Fitness Trainer," a comprehensive evaluation method is applied, with a focus on accuracy, latency, and user experience.

### 5.1 Evaluation of Quantitative Accuracy

The accuracy of pose estimation and repetition counting will be compared using a "Ground Truth" dataset. This dataset will consist of video footage of exercises that have been manually annotated by fitness experts or compared to a validated dataset like M3GYM 42 or a gold-standard system (like VICON, if available).

Metric 1: Correct Keypoint Percentage (PCK): This widely used computer vision metric calculates the proportion of predicted keypoints that are within a normalized distance of the ground truth keypoints. We aim for a PCK@0.2 (threshold of 20% of torso diameter) of  $>90\%$  to ensure the system is reliable for biomechanical analysis.

Metric 2: Repetition Counting Accuracy: A confusion matrix will be made in order to assess the performance of the repetition counter.

- True Positives (TP): Accurate rep counts.
- False Positives (FP): Non-reps, such as half-reps, are regarded as representatives.
- FalseNegatives(FN): These are valid representatives that the system failed to detect.
- Accuracy and Memory: Derived from the matrix to compute the F1-Score of the counter [13]

### 5.2 Resource Usage and System Latency

1. Latency is critical to the success of a real-time feedback system [11]. We will measure the Frames Per Second (FPS) and Inference Time (in milliseconds) across three hardware tiers to demonstrate the scalability of the MediaPipe-based approach:

Tier 1: A top-tier desktop computer with a discrete GPU

Tier 2: Standard Laptop (Integrated Graphics).

Tier 3: Mobile Device (iOS/Android via TFLite).

The goal is to demonstrate that the system sustains  $>15$  FPS on Tier 3 devices and  $>30$  FPS on Tier

2, demonstrating [4].

### 5.3 Qualitative Feedback and User Experience (UX)

In addition to raw data, a pilot study with human participants will be used to assess the effectiveness of the feedback [6].

After completing tasks with the AI trainer, participants will evaluate the system based on:

- Reactivity: Did you feel that the feedback was given right away?
- Clarity: Were the directions ("Lower hips") simple to comprehend?

## VI. DISCUSSION: LIMITATIONS, ETHICS, AND FUTURE HORIZONS

### 6.1 The Edge vs. Cloud Debate and Privacy

It is a conscious architectural decision to process data locally on the user's device, or on the edge, as opposed to in the cloud. Because cloud-based AI has more processing power, it can run heavier models (like OpenPose) [8], which may produce marginally better results in complicated crowd situations. However, there are significant privacy and security risks when live video of users in their homes is sent to a central server. This system complies with "Privacy by Design" principles by using MediaPipe for on-device inference. Only the abstract mathematical data (rep counts, scores) may be stored; the video feed never leaves the user's local computer.

### 6.2 The 2D-to-3D Gap

MediaPipe uses a 2D image source to infer 3D landmarks. The issue of perspective ambiguity is thus introduced. 2D analysis is very accurate for exercises that mostly take place in the sagittal plane, such as a bicep curl seen from the side. However, the projected angles may be distorted for multi-planar movements or when the camera angle is oblique (e.g., a squat viewed from a 45-degree angle). In 3D space, a knee that appears to be "valgus" (caving in) in a 2D projection may actually be aligned [9]. To resolve these depth ambiguities and provide truly professional-grade analysis, future iterations of this work must use multi-view fusion (using two smartphone cameras) or camera calibration techniques

[20].

### 6.3 Limitations of Rule-Based Logic

Deterministic, rule-based logic is used in the current system (e.g., if angle < 90). This method lacks the subtlety of a human coach who has an intuitive understanding of anatomical variance, despite being transparent and simple to debug [7]. Naturally, a person with long femurs will have a different "perfect" squat back angle than a person with short femurs. A static rule might unfairly penalize the former. This limitation highlights the need for adaptive algorithms that "calibrate" to the individual anthropometry of the user during a setup phase [14].

### 6.4 Future Horizons: Generative AI and LLMs

The incorporation of Large Language Models (LLMs) is this technology's next frontier. The feedback is static text at the moment. The system could become a conversational coach by feeding the pose data into an LLM through a Retrieval-Augmented Generation (RAG) pipeline. Rather than simply displaying "Fix Back," it could say something like, "I saw your lower back rounding on the last three reps." When the core disengages, this typically occurs. Before you descend, try inhaling deeply and bracing your stomach. As a result, the system becomes a teaching tool instead of a monitoring tool [6]. Additionally, these models could be trained on decentralized user data using Federated Learning, increasing the accuracy of the global model without ever centralizing the sensitive video feeds [5].

## VII. CONCLUSION

To meet a critical public health need, computer vision, biomechanics, and software engineering have come together to create the "AI-Based Interactive Fitness Trainer" described in this report. By utilizing MediaPipe BlazePose's efficiency, the system successfully overcomes the hardware constraints that limited previous iterations of fitness technology, offering a solution accessible to anyone with a smartphone or webcam [6]. When combined with strong geometric reasoning and state machine architecture, the study shows that monocular pose estimation provides a useful method for real-time form correction and repetition counting [3].

It effectively bridges the gap between unsupervised at-home workouts and expensive personal training. The

framework described here is scalable and modular, despite ongoing problems with 3D depth perception and anatomical personalization. By transferring the technical monitoring load from the user to the AI, people can concentrate on their physical exertion knowing that their technique and safety are being closely watched [5]. This will likely become the standard of care in preventive health and remote physical therapy.

## REFERENCES

- [1] Pose Estimation for Real-Time Fitness Coaching Using Deep Learning Authors: J. Liu, A. Sharma, et al. Conference: IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
- [2] Patil, S. M., Patil, V.D., Sharma, K.M., Chaudhari, S.S., & Talekar, S. S.(2022).. \"Artificial Intelligence-based Personal Fitness Trainer.\" International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), Vol. 2, Issue 1, November 2022.
- [3] A Review on AI-Based Fitness Applications Published in: International Journal of Advanced Computer Science and Applications (IJACSA), 2021 Focus: Overview of AI in fitness apps and real-time correction.
- [4] Chin, Ji-Hyoung, Chanwook Do, and Minjung Kim. "How to Increase Sport Facility Users' Intention to Use AI Fitness Services: Based on the Technology Adoption Model." International Journal of Environmental Research and Public Health, vol. 19, no. 21, p. 14453, November 2022.
- [5] Smart Personal Trainer: A Real-Time AI-Based Fitness Coaching System Published in: Sensors (MDPI Journal), 2020 DOI: 10.3390/s20185171
- [6] Chen, Y., et al. (2021) – "AI-Powered Personal Trainer: A Real-time Human Pose Tracking and Feedback System" Source: IEEE Access Focus: Real-time posture correction using computer vision.
- [7] Wang, Z., et al. (2022) – "Smart Exercise Coaching System with Deep Learning- Based Action Recognition" Source: Sensors Journal Application of deep learning to monitor and correct workouts interactively.
- [8] Zhang, M., et al. (2020) – "AI Coach: AI-driven fitness coaching system with personalized

- recommendations. "Source: ACM Conference on Human Factors in Computing Systems (CHI)
- [9] Liu, H. et al. (2019) – "Virtual Fitness Coach using Reinforcement Learning for Exercise Guidance. "Source: Proceedings of the AAAI Conference on Artificial Intelligence
- [10][10]. Mohanty, A., & Jagadeesh, R. (2023) – "Pose estimation and correction in AI- based fitness trainers using OpenPose." Source: Journal of Biomedical Informatics
- [11][11]. Sharma, Ayushi, Jyotsna Pathak, Muskan Prakash, and J. N. Singh. "Object Detection using OpenCV and Python." In 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), pp. 501- 505. IEEE, 2021.
- [12] Ji, Haoran, Karungaru Stephen Githinji, and Terada Kenji. "AI Fitness Coach at Home using Image Recognition." 2022.
- [13] Chaudhari, H. P., Faras, A., Patil, M. Virtual Fitness Trainer using Artificial Intelligence Focuses on pose estimation using MediaPipe and OpenCV Includes real- time feedback and exercise tracking
- [14] Swathi, P., Pranathi, R. P., Rohith, K. S. G., Sagar, M. C., Jayavardhan, S. AI- Based Fitness Trainer Application – IJNRD Discusses NUI system architecture and CNN-based pose correction Includes diet recommendation and repetition counter modules
- [15] Gupta, N., Agrawal, V., Maheshwari, M. (2019). Intelligent Virtual Fitness Coach using Computer Vision and Machine Learning. Procedia Computer Science,155,584- 591
- [16] Ivan Grishchenko et al., "BlazePose: On-device Real-time Body Pose Tracking", Google Research, 2020
- [17] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi- person 2D pose estimation using Part Affinity Fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [18] Szeliski, Richard. "Computer Vision: Algorithms and Applications." Springer, 2022.
- [19] Toshev, A., & Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- [20] Wen, Y., Zhang, J., & Li, X. (2016). FitNet: A mobile fitness application using deep learning. Journal of Mobile Computing and Applications.
- [21] Rutuja A. Gulhane and Sunil R. Gupta, "Feature optimization using Hybrid Metaheuristic Red Deer and Dragonfly Algorithms for Multi-disease prediction", International Journal of Multimedia Tools and Applications, Springer Nature Journal, E-ISSN: 1573-7721, Volume 01, Number 01, May 2024, Page No. 01-17.
- [22] Kanse, R., Ajagekar, S., Patil, T., Motekar, H., Rathod, V., Papalkar, R., & Ali, S. (2023). WeSafe: A safety app for all. In Artificial Intelligence, Blockchain, Computing and Security (Vol. 1, pp. 441–446).