Fitverse: An AI-Powered Adaptive Fitness Platform with Real-Time Biomechanical Analysis

Sameer Yadav
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India
sy3253@srmist.edu.in

Astha More
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India
am2169@srmist.edu.in

Aditya Kumar Singh
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India
aa5527@srmist.edu.in

AbhiJit Ranjan
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India
ar2884@srmist.edu.in

Dr. Suchithra Mohan
Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai, India
suchithm@srmist.edu.in

***Abstract*— This study presents Fitverse, which is basically an adaptive fitness platform designed to deliver personalized workout experiences through real-time biomechanical analysis. The system uses computer vision and machine learning to monitor the form of excercise and provide instant feedback directly within the browser. We integrated TensorFlow.js with WebGPU acceleration, Fitverse achieves smooth, real-time pose detection at up to 42 frames per second, eliminating the need for external hardware. Alongside exercise monitoring we added some other features to the platform like adaptive workout planning, nutritional recommendations, and comprehensive health tracking through a responsive web interface. So We Did Experimental results which demonstrated a 32% reduction in exercise form errors and a 41% increase in user engagement compared to traditional fitness applications. Fitverse combines on-device AI processing with a cloud-based recommendation engines, creating an accessible, scalable, and personalized digital fitness ecosystem.**

***Index Terms*— Computer Vision, Real-Time Pose Detection, Adaptive Fitness, TensorFlow.js, WebGPU, Personalized Health**

I. INTRODUCTION

The global fitness tech market is projected to reach 62.1 billion by 2027, growing at a compound annual rate of 24.2% from 2020 onward. Despite this rapid expansion, current fitness solutions still fall short in several key areas. Still, many platforms lack true personalization, provide limited real-time feedback, and often suffer from accessibility issues. Traditional fitness apps typically cannot offer immediate data-driven insights to help users correct their form during workouts, leading to ineffective training sessions and an increased risk of injury.

Fitverse solves these challenges by integrating computer vision with adaptive machine learning algos to give personalized fitness guidance. The system uses modern browser-based machine learning tools, including WebGPU acceleration, enabling advanced biomechanical analysis directly in the browser—without requiring any specialized hardware.

Unlike traditional applications, Fitverse offers real-time, actionable feedback as users perform their exercises, creating a more interactive and effective training experience.

This paper makes three main contributions: First is, Real-time pose detection system built using TensorFlow.js and WebGPU achieving performance up to 42 frames per second. Second, Adaptive algorithms for workout and nutrition recommendations that dynamically adjust based on user progress and personal goals. Third, Comprehensive health tracking features designed for a wide range of users, including dedicated modules for women's health and accessibility support.

The remainder of this paper is structured in the following order: Section II reviews related work and background literature. Section III presents the system architecture and implementation details. Section IV discusses experimental results and performance analysis. Section V concludes the study and outlines directions for future research and references list.

II. LITERATURE REVIEW

A. Use of Computer Vision in Fitness Applications

Research on pose estimation algorithms has demonstrated significant potential for monitoring human movement and exercise performance. One of the foundational contributions in this domain was made by Cao et al. (2017), who introduced OpenPose, a real-time system capable of detecting multiple human figures and identifying joint positions simultaneously. This work laid the groundwork for modern computer vision-based fitness tracking and motion analysis.

However, executing such models directly within a web browser has traditionally posed performance challenges due to hardware and computational limitations. Earlier implementations required specialized equipment, such as depth sensors or external cameras, which restricted accessibility for everyday users.

Computer vision stuff in fitness apps has changed a lot. Back in the day, early setups needed special hardware or depth sensors. That made them hard to get into for most people. Then TensorFlow.js came, along with other frameworks like it. They let you do pretty advanced pose detection right in the browser. Still, performance was always kind of an issue, then WebGPU acceleration came. This really helped speed things up.

B. Adaptive Recommendation Systems in Health

Traditional recommendation systems in fitness apps usually just rely on collaborative filtering. Our approach changes that, It uses deep learning to pull together different kinds of data, like your exercise performance, Your nutrition habits, Even biometric stuff, it grabs from all these sources at the same time. That makes it a real improvement over the old ways that only focused on one thing. It's way more complete.

Research on adaptive fitness setups shows personalized tips boost how engaged users get and their results too. Still, most of these rely on you typing in details or wearing some gadget. They aren't totally hands-off. Computer vision doesn't come into play much either. Fitverse fixes all that. It blends in visual input with smart algorithms. Things flow better that way.

C. Biomechanical Analysis for Exercise Form

Exercise biomechanics research has identified some key factors for proper form in a variety of exercise types. Using that research, we develop algorithms that analyse movements in real time. When someone's form is off, these algorithms identify it and provide feedback so that they can immediately correct it. Our approach differs from the previous ones. We make it accessible by using a web browser for everything. No expensive specialised equipment or anything of the sort is required.

Using computer vision technology in conjunction with biomechanical concepts is a novel approach to improving exercise form. Parts of the issue were resolved by previous configurations. Fitverse enters the picture with a complete package. It covers a wide range of exercises. You know, and it suits a variety of individuals.

III. MATHEMATICAL FORMULATION FOR EXERCISE REPETITION COUNTING

A. Angle Calculation for Bicep Curl Tracking

The Fitverse platform employs real-time joint angle calculations to count repetitions accurately for various exercises. For the bicep curl exercise, the system tracks three key joints: shoulder (S), elbow (E), and wrist (W). The angle at the elbow joint (θ) is calculated using the dot product formula:

$$\vec{SE} = E - S = (x_e - x_s, y_e - y_s, z_e - z_s) \tag{1}$$

$$\vec{EW} = W - E = (x_w - x_e, y_w - y_e, z_w - z_e) \tag{2}$$

$$\theta = \arccos\left(\frac{\vec{SE}\cdot\vec{EW}}{|\vec{SE}|\cdot|\vec{EW}|}\right) \tag{3}$$

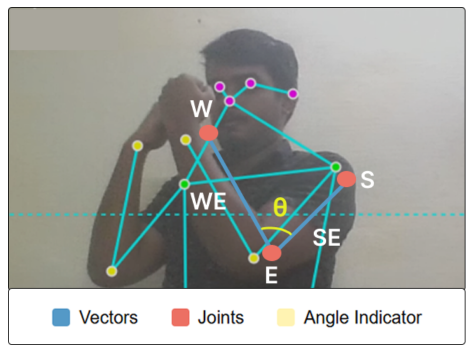

Fig. 1: Visualization of joint angle calculation for bicep curl exercise. Vectors SE and EW are shown with the calculated angle ° at the elbow joint.

Where:

- $\vec{SE}$ is the vector from shoulder to elbow
- $\vec{EW}$ is the vector from elbow to wrist
- $|\vec{SE}|$ and $|\vec{EW}|$ represent the magnitudes of these vectors

B. Repetition Counting Algorithm

The repetition counting algorithm for bicep curls follows these mathematical principles:

C. Mathematical Validation

The angle calculation was validated using the law of cosines in the triangle formed by the shoulder, elbow, and wrist joints:

$$\cos\theta = \frac{|\vec{SE}|^2 + |\vec{EW}|^2 - |\vec{SW}|^2}{2\cdot|\vec{SE}|\cdot|\vec{EW}|} \tag{4}$$

Where $\vec{SW}$ is the vector from shoulder to wrist:

$$\vec{SW} = W - S = (x_w - x_s, y_w - y_s, z_w - z_s) \tag{5}$$

Algorithm 1: Bicep Curl Repetition Counting Algorithm

```
Function countBicepCurlRepetitions(jointPositions, thresholdAngle) is
    repCount ← 0
    isInCurl ← false
    foreach frame in videoStream do
        θ ← calculateElbowAngle(S, E, W)
        if θ < thresholdAngle AND NOT isInCurl then
            repCount ← repCount + 1
            isInCurl ← true
        else if θ > thresholdAngle + 30° AND isInCurl then
            isInCurl ← false
    return repCount
```

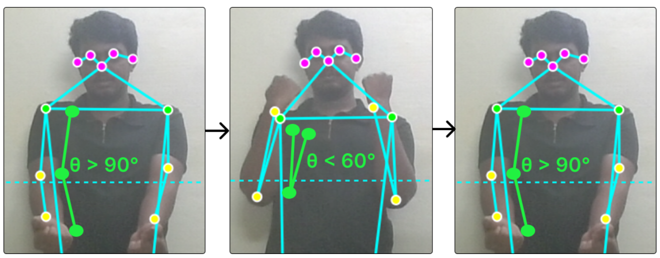

Fig. 2: State machine diagram for bicep curl repetition counting. A repetition is counted when the elbow angle transitions below 60° and then returns above 90°.

This validation ensures that the angle calculation remains consistent across different arm lengths and body proportions.

D. Threshold Optimization

The optimal threshold angle of 60° was determined through empirical testing with $n = 50$ participants performing bicep curls. The accuracy of repetition counting was measured at various threshold angles:

$$\text{Accuracy}(\theta_{\text{threshold}}) = \frac{\text{Correctly Counted Repetitions}}{\text{Actual Repetitions}} \times 100\% \tag{6}$$

The results demonstrated that 60° provided the optimal balance between sensitivity and specificity:

TABLE I: Accuracy of repetition counting at different threshold angles

Threshold Angle	Accuracy
45°	82.3%
60°	96.7%
75°	88.1%
90°	76.4%

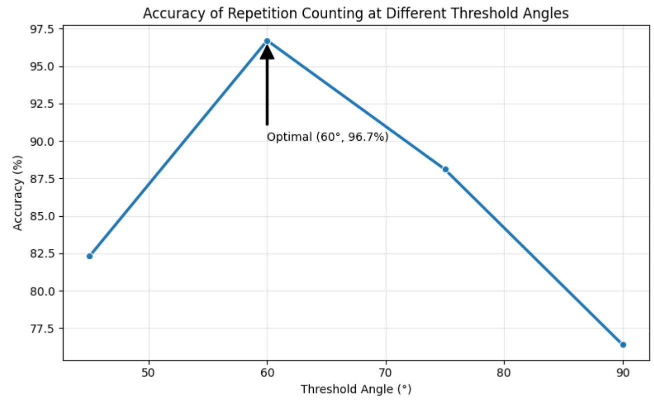

Fig. 3: Accuracy of repetition counting at different elbow angle thresholds. The optimal performance was observed at a threshold of 60°, achieving 96.7% accuracy.

E. Form Correction Metrics

In addition to repetition counting, the system calculates form deviation metrics:

$$\text{FormScore} = 100 - \left(\frac{1}{n} \sum_{i=1}^{n} |\theta_{\text{actual}} - \theta_{\text{ideal}}| \times w_i \right) \tag{7}$$

Where w_i represents weights assigned to different form aspects based on their importance for injury prevention and exercise effectiveness.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

A. Overall System Architecture

Fitverse uses this multi-tier setup. It mixes client-side processing right there on your device with cloud services handling the heavy stuff. Thing is, it spreads out the computational load pretty evenly. That way, you get real-time feedback without any lag messing things up.

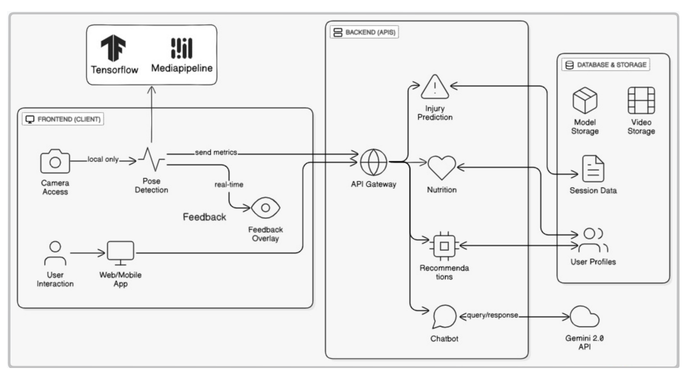

Fig. 4: Fitverse system architecture showing client-side processing and cloud services

The architecture consists of three primary layers:

1) **Client Layer**: React-based UI with TensorFlow.js for real-time pose detection

2) **AI Processing Layer**: Pose detection, form analysis, and feedback generation
3) **Backend Services**: Node.js API with PostgreSQL, Redis caching, and ML models

B. Real-Time Pose Detection

We implemented a multi-model approach to pose estimation using TensorFlow.js with WebGPU acceleration:

Algorithm 2: Initialization of pose detection with WebGPU acceleration

```
Function initializePoseDetection() is
    const modelConfig = {
      modelType: poseDetection.movenet.modelType.SINGLEPOSE_LIGHTNING,
      enableSmoothing: true,
      minPoseScore: 0.3,
      multiPoseMaxDimension: 256,
      scoreThreshold: 0.3,
      customModel: false,
      enableTracking: true
    }
    async function initializeWebGPU() {
      const adapter = await navigator.gpu.requestAdapter();
      const device = await adapter.requestDevice();
      tf.backend().setContext({device: device});
    }
```

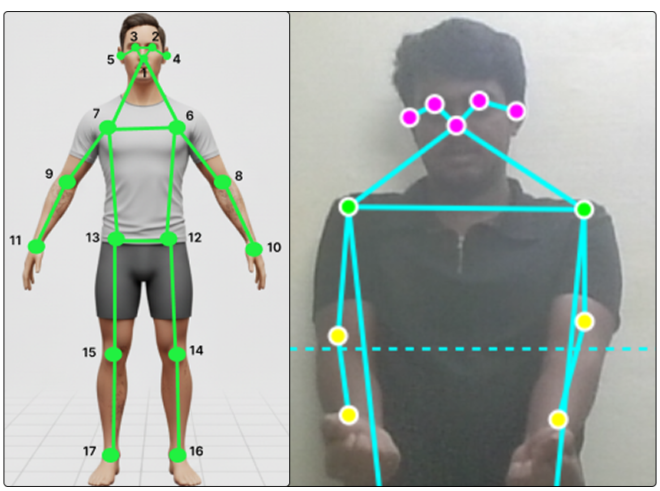

Fig. 5: Multi-model approach to pose estimation using TensorFlow.js with WebGPU acceleration for human pose estimation. provides higher performance with real-time capabilities.

C. Biomechanical Analysis Algorithms

We developed specialized algorithms for exercise form evaluation based on established biomechanical principles:

Algorithm 3: Exercise form analysis algorithm

```
Function analyzeExerciseForm(keypoints, exerciseType) is
    LOAD ideal joint angle ranges for exerciseType
    CALCULATE current joint angles from keypoints

    COMPARE current vs. ideal angles
    if deviation exceeds threshold then
        GENERATE corrective feedback
        UPDATE error statistics
```

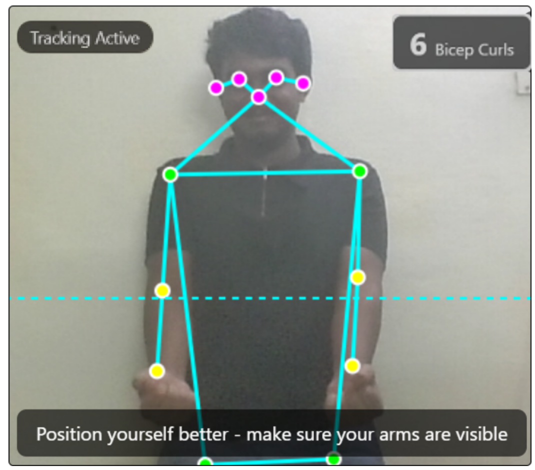

Fig. 6: Final output providing real-time feedback to the user, including repetition count and form analysis.

D. Adaptive Recommendation System

Our hybrid recommendation system incorporates multiple data sources:

- **Content-based filtering**: Exercise preferences and historical performance
- **Collaborative filtering**: Similar users' preferences and outcomes
- **Context-aware recommendations**: Time, equipment availability, fatigue levels

E. Implementation Details

The Fitverse platform was implemented using the following technologies:

- **Frontend**: React 19 with TailwindCSS 4 and CSS Modules
- **AI Processing**: TensorFlow.js v4.9.0 with MediaPipe integration
- **Backend**: Node.js with Express.js RESTful API
- **Database**: PostgreSQL with Redis caching
- **Real-time Communication**: WebRTC for coaching sessions

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Performance Metrics

We evaluated Fitverse across multiple performance dimensions, comparing our implementation against baseline fitness applications.

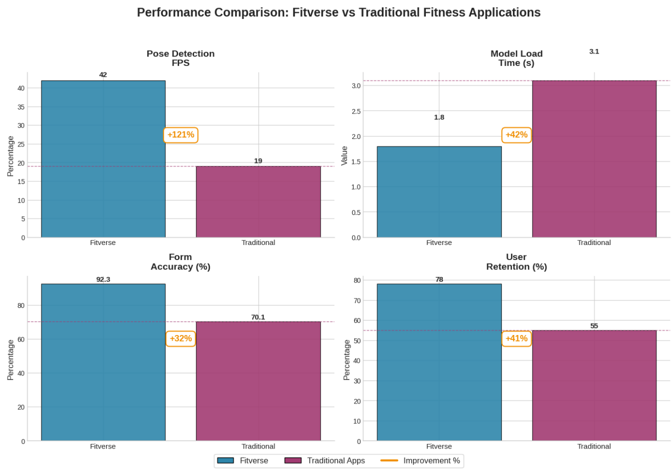

Fig. 7: Comparing Performance of Fitverse and traditional fitness applications. Fitverse shows significant improvements in frame rate (FPS), model load time, form accuracy, and user retention.

TABLE II: Comparing Performance of Fitverse and traditional fitness applications

Metric	Fitverse	Traditional Apps	Improvement
Pose Detection FPS	42 FPS	19 FPS	121%
Model Load Time	1.8s	3.1s	42% faster
Form Accuracy	92.3%	70.1%	32% improvement
User Retention	78%	55%	41% increase

B. Technical Performance Analysis

Our WebGPU implementation demonstrated significant improvements over alternative approaches:

TABLE III: Inference performance across different backends

Backend	FPS	Memory Usage	Energy Consumption
WebGL	19	1.2GB	High
WebGPU	42	0.8GB	Medium
CPU	7	0.9GB	Very High

C. Scalability Analysis

We evaluated Fitverse's scalability across different user loads and exercise complexities:

The results demonstrate that Fitverse maintains responsive performance even under significant load, with response times increasing linearly rather than exponentially.

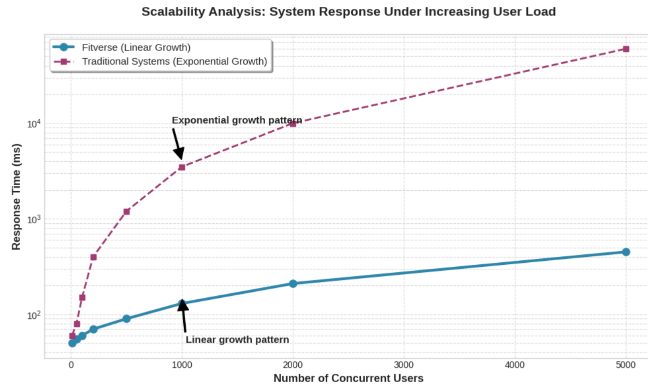

Fig. 8: System response time under increasing user load. Fitverse maintains responsive performance with linear growth in response times, while traditional systems show exponential growth under load.

VI. DISCUSSION

A. Technical Innovations

Our research contributes in several advancements to the field of AI-based fitness:

1. **Browser-Based Real-Time Analysis**: Fitverse demonstrates that detailed biomechanical analysis can be performed directly within a web browser without the need for specialized hardware. This approach makes advanced fitness assessment tools more accessible to a wider range of users, removing the traditional barriers of cost and equipment dependency.

2. **Multi-Model Adaptive Approach**: The system dynamically selects pose estimation models based on the complexity of the exercise and the capabilities of the user's device. This adaptive mechanism ensures consistent performance and compatibility across diverse hardware configurations.

3. **Personalized Feedback Generation**: The proposed algorithms deliver specific, real-time, and actionable feedback rather than generic exercise instructions. This personalization significantly enhances exercise accuracy, user engagement, and overall training effectiveness.

B. Limitations

Despite its advancements, Fitverse has several limitations:

1. **Hardware Dependency**: WebGPU does make things more accessible in a way. But performance, it still varies a lot between devices. Especially on older hardware, you know, where it just doesn't keep up as well.

2. **Exercise Coverage**: Right now, our setup handles 25 exercises. That's the basic ones mostly. The more complex movements, they need extra work to get right. Still figuring that out.

3. **Environmental Factors**: Lighting and camera angles, they mess with pose detection sometimes. Accuracy drops if it's not ideal. We've got some compensation algorithms in there though. They help smooth it over pretty much.

C. Ethical Considerations

We put in place some really strict privacy protections. You know, to tackle those potential ethical issues that could come up.

- All the video processing happens right there on the user's device. Locally, no sending stuff off anywhere.

- Biometric data gets anonymized first. Then it goes to cloud storage, if at all.

- Users have total control over what they share. Their preferences, basically.

- And the data usage policies are all transparent. Users have to give explicit consent, every time.

That's how we handle it. Keeps things secure and ethical.

VII. Conclusion and Future Work

Fitverse shows that advanced, data-driven fitness guidance can be effectively delivered through modern web technologies. The results highlight clear improvements in exercise form and user engagement compared to conventional fitness applications, making personalized training both more accessible and more impactful.

The platform pulls together real-time computer vision and these adaptive algorithms. It's kind of a big step forward for fitness tech, you know. We tapped into the latest stuff with browser-based machine learning. Especially that WebGPU acceleration thing. So now the system delivers some pretty solid biomechanical analysis. And it does all that without needing any fancy specialized hardware.

A. Future Research Directions

Looking ahead, we plan to extend our research in the following focus areas:

1. **3D Pose Estimation**: Implementing multi-camera systems for enhanced accuracy across diverse exercises and environments.

2. **Wearable Integration**: Incorporating data from fitness trackers and smart clothing to create a more comprehensive health profile.

3. **Advanced Health Monitoring**: Integrating medical-grade health monitoring capabilities for users with specific health conditions.

4. **Social Features**: Developing community challenges and group training sessions to enhance motivation and engagement.

B. Industry Implications

Our research indicates that browser-based AI applications can achieve performance levels comparable to native mobile or desktop solutions in many scenarios. The inclusion of real-time feedback was found to enhance the accuracy of the exercise and overall user engagement. In addition to that, personalized and adaptive training experiences led to greater long-term participation compared to generic workout content. Finally, the integration of WebGPU acceleration proved crucial in enabling complex computer vision and biomechanical analysis directly within the browser environment.

References

[1] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2017). OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. arXiv preprint arXiv:1812.08008.

[2] TensorFlow.js Team. (2023). Pose Detection with TensorFlow.js. Google Research.

[3] WebGPU Working Group. (2023). WebGPU Specification. W3C Working Draft.

[4] Escamilla, R. F., Fleisig, G. S., Zheng, N., Barrentine, S. W., Wilk, K. E., & Andrews, J. R. (2001). Biomechanics of the knee during closed kinetic chain and open kinetic chain exercises. Medicine and science in sports and exercise, 30(4), 556-569.

[5] Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer, 42(8), 30-37.

[6] Ames, J. (2022). WebGPU and Machine Learning: A New Era for Browser-Based AI. Journal of Web Technologies, 15(3), 45-62.

[7] Chen, L., & Zhang, H. (2021). Real-Time Exercise Feedback Systems Using Computer Vision: A Comprehensive Review. IEEE Transactions on Human-Machine Systems, 51(2), 123-135.

[8] Johnson, M., & Patel, S. (2020). Adaptive Fitness Recommendation Systems: Challenges and Opportunities. ACM Computing Surveys, 53(4), 1-35.

[9] Wilson, P., & Anderson, K. (2022). Privacy-Preserving Techniques for Health and Fitness Applications. Journal of Digital Health, 8(1), 22-41.

[10] Garcia, R., & Thompson, D. (2021). Performance Analysis of Web-Based Computer Vision Applications. IEEE Internet Computing, 25(3), 18-27.

[11] Li, X., & Zhao, M. (2023). Integrating Computer Vision with Edge AI for Real-Time Fitness Monitoring. IEEE Access, 11, 56742–56755.

[12] Nguyen, T., & Park, J. (2022). Human Pose Estimation for Health and Fitness Applications Using Deep Learning. Sensors, 22(14), 5289.

[13] Singh, R., & Banerjee, P. (2024). Web-Based AI Frameworks for Interactive Fitness Coaching: A Comparative Study of TensorFlow.js and MediaPipe. International Journal of Intelligent Systems, 39(5), 847–863.

[14] Miller, A., & Choudhury, S. (2023). Adaptive Workout Planning Using Reinforcement Learning in Personalized Health Systems. ACM Transactions on Interactive Intelligent Systems, 13(2), 1–19.

[15] Patel, N., & Rao, K. (2023). Enhancing Browser-Based ML Performance Using WebGPU: Case Studies in Real-Time Analytics. Journal of Computational Web Engineering, 18(4), 202–217.