International Conference
on
Artificial Intelligence in Engineering Systems
(ICAIES-2025)

Organised by :

Rajeev Institute of Technology

Hassan, Karnataka, India.

26-27th December 2025

Fitverse: An AI-Powered Adaptive Fitness Platform with Real-Time Biomechanical Analysis (PAPER ID - CS 475)

Authors:

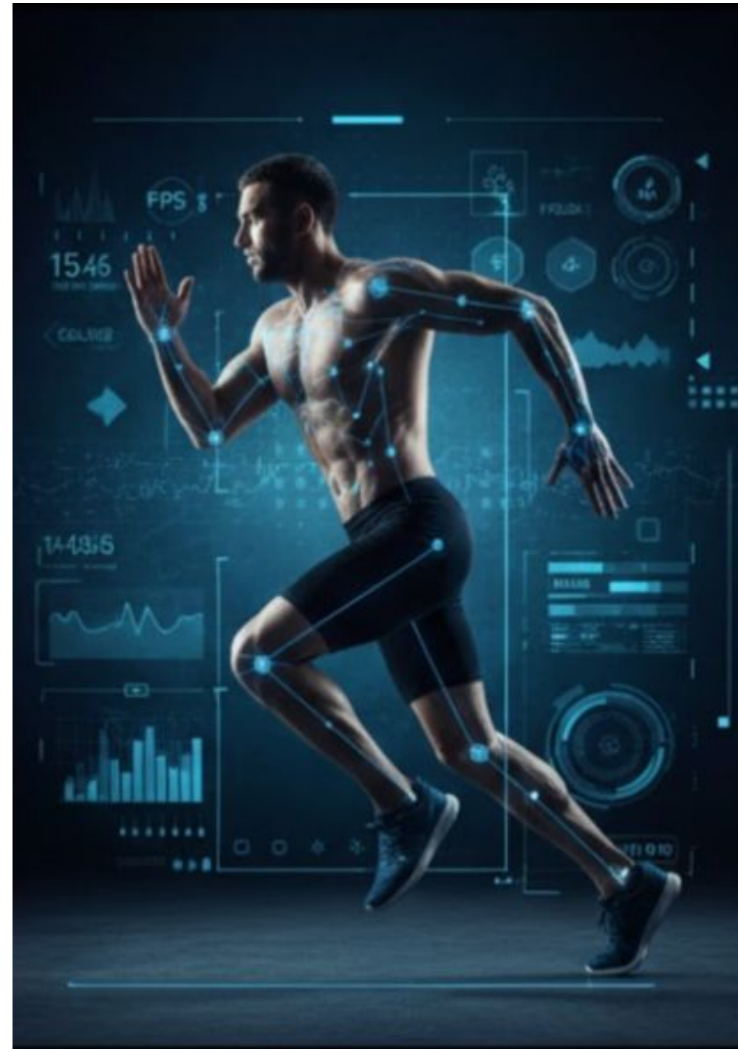
- Aditya Kumar Singh
- Sameer Yadav
- Astha More
- Abhijit Ranjan
- Dr. Suchithra Mohan

Institution: SRM Institute of Science and Technology

INDEX

ABSTRACT:

- Fitverse is an AI-driven adaptive fitness platform that analyzes body movement using real-time biomechanical tracking.
- Powered by TensorFlow.js and WebGPU, it performs fast, in-browser pose detection at up to 42 FPS.
- Provides instant exercise form correction, personalized workout recommendations, and dynamic nutrition guidance.
- Demonstrated a 32% improvement in form accuracy, reducing common posture and alignment mistakes.
- Increased overall user engagement by 41% through interactive, real-time feedback.
- Runs completely in the browser, requiring no external sensors, wearables, or additional hardware.



INTRODUCTION:

Fitness tech market reaching **$62.1B** by 2027, but has limitations

Current apps lack:

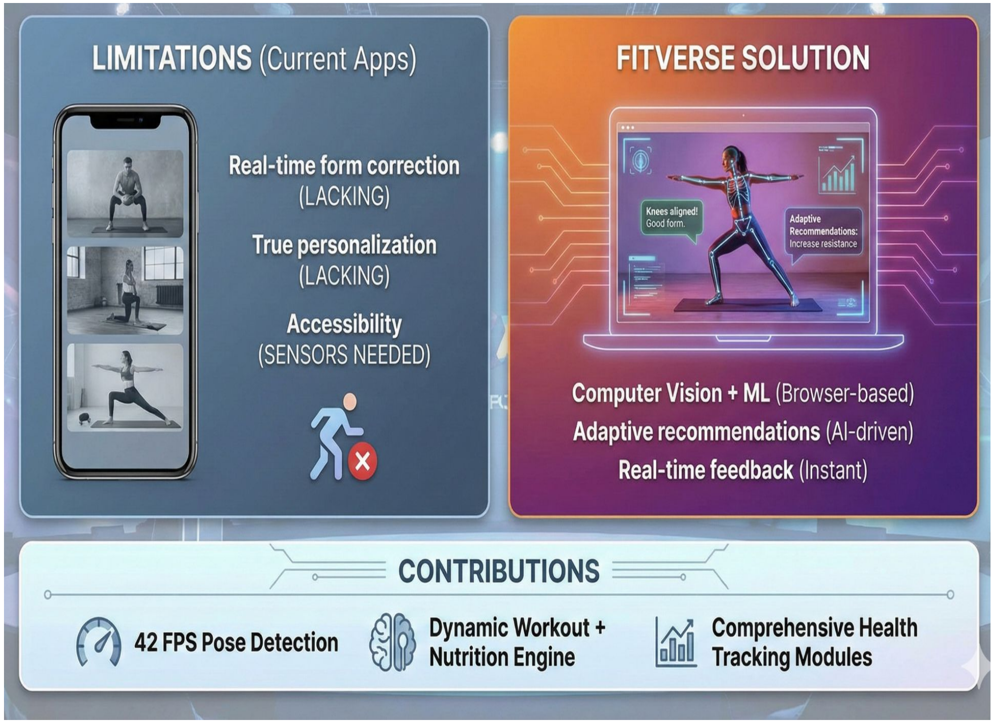
- Real-time form correction
- True personalization
- Accessibility (need sensors/hardware)

Fitverse solves these using:

- Computer Vision + ML in the browser
- Adaptive recommendations
- Real-time feedback

Contributions:

- 42 FPS pose detection
- Dynamic workout + nutrition engine
- Comprehensive health tracking modules



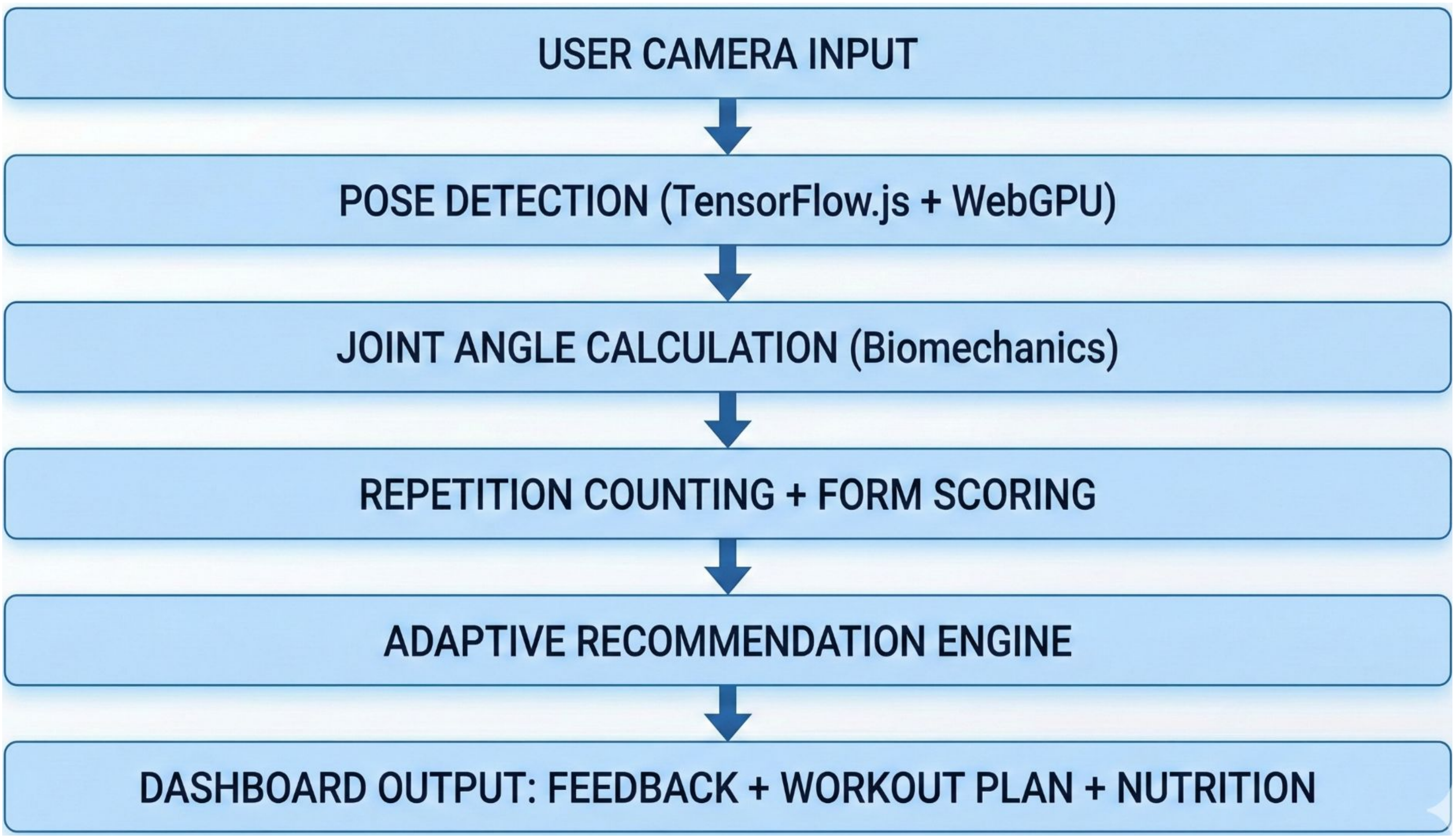
LITERATURE REVIEW :

Study Area	Existing Research	Limitations in Past Work	What Fitverse Uses
Computer Vision in Fitness	Openpose	Needed High-End hardwares for good performance.	WebGPU accepelerated in browser pose detection.
Adaptive Fitness Systems	Deep Learning Recommender Models	Manual inputs required not real time.	Fully Automatic, Computer vision driven adaptability
Biomechanical Analysis	Traditional form correction Models.	Required Sensors; limited accessibility	Real time sensor free biomedical analysis
Real Time Feedback systems.	Existing Fitness apps with basic tasking.	No live corrections, only tracks reps	Instant form correction + angle based rep counting

RESEARCH GAPS:

Existing Gap	Description	Our Solution
Lack of real-time feedback	Apps do not correct form live	Real-time pose estimation
Hardware dependency	Depth sensors/cameras needed	Pure browser-based CV
Limited personalization	Generic exercises	Adaptive ML-based workout plans
Poor accuracy in counting reps	Threshold-based errors	Mathematical + biomechanical angle tracking

PROPOSED METHODOLOGY :



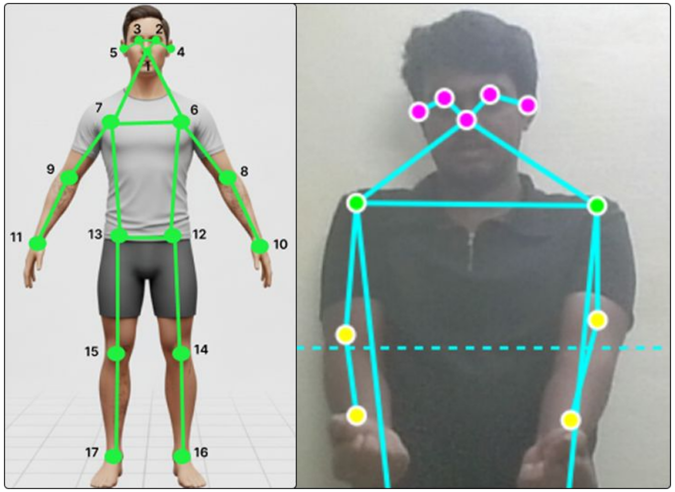


Fig. 1: Multi-model approach to pose estimation using TensorFlow.js with WebGPU acceleration for human pose estimation. provides higher performance with real-time capabilities.

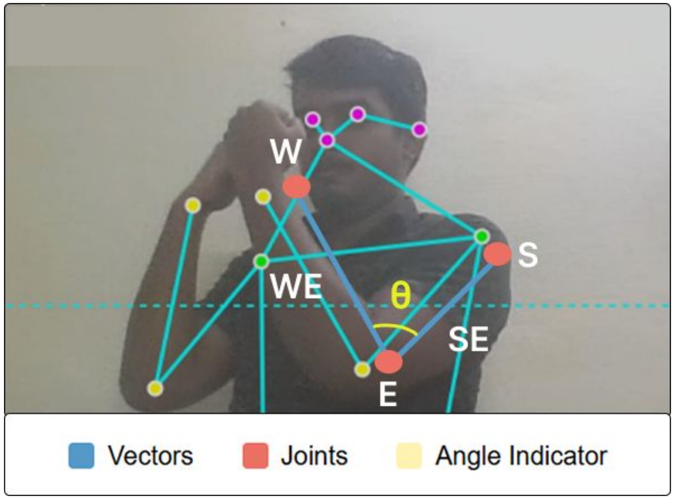


Fig. 2: Visualization of joint angle calculation for bicep curl exercise. Vectors SE is he vector from shoulder to elbow. EW is the vector from elbow to wrist.

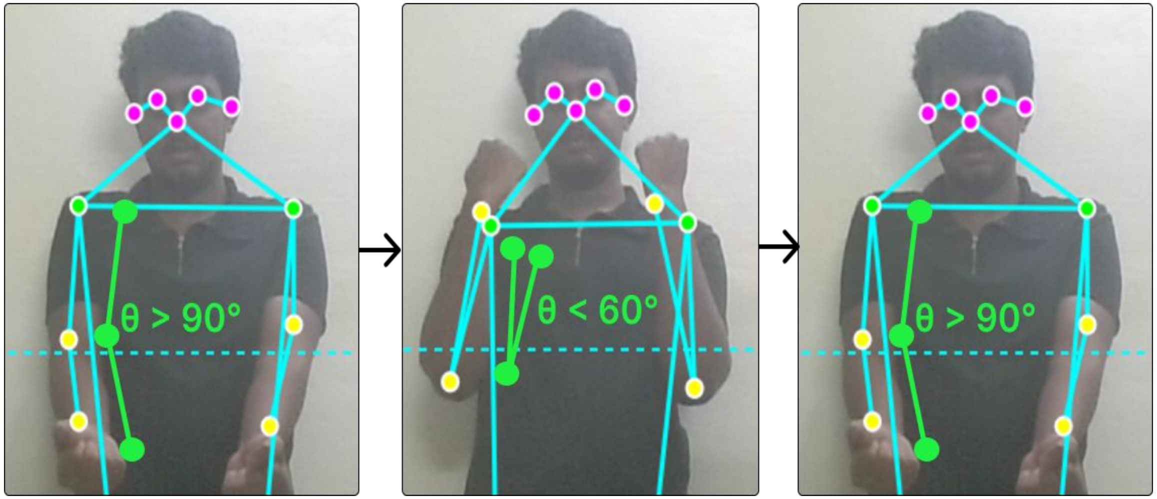


Fig. 3: State machine diagram for bicep curl repetition counting. A repetition is counted when the elbow angle transitions below 60° and then returns above 90°.

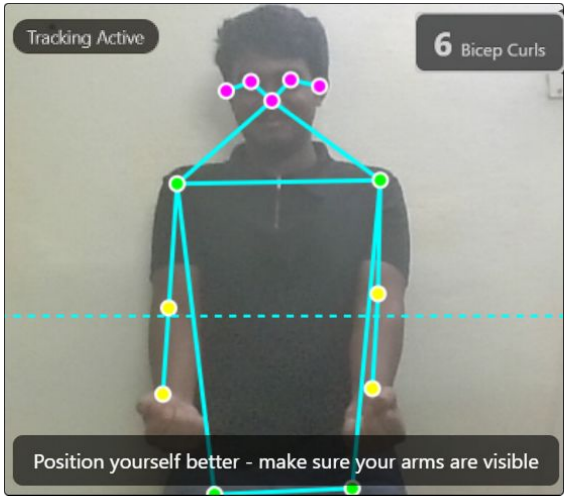


Fig. 4: Final output providing real-time feedback to the user, including repetition count and form analysis.

Angle Calculation for Bicep Curl Tracking

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

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

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

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

Mathematical Validation

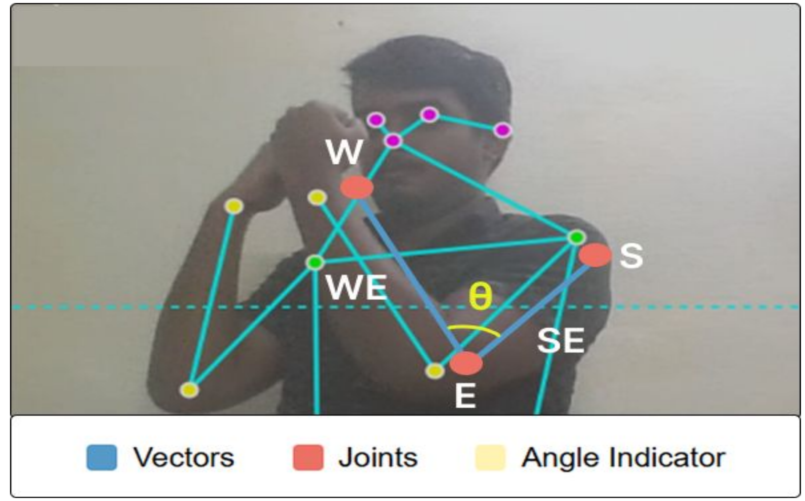
$$\cos\theta = \frac{|\vec{SE}|^2 + |\vec{EW}|^2 - |\vec{SW}|^2}{2 \cdot |\vec{SE}| \cdot |\vec{EW}|}$$

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

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

Repetition Counting Algorithm

foreach *frame in videoStream* **do**
  $\theta \leftarrow \text{calculateElbowAngle}(S, E, W)$
  if $\theta < \textit{thresholdAngle AND NOT isInCurl}$ **then**
    $\text{repCount} \leftarrow \text{repCount} + 1$
    $\text{isInCurl} \leftarrow \text{true}$
  else if $\theta > \textit{thresholdAngle} + 30^\circ$ *AND isInCurl* **then**
    $\text{isInCurl} \leftarrow \text{false}$



Threshold Optimization

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

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%

Form Correction Metrics

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

w_i = importance weight for each joint

RESULTS & DISCUSSION

Performance Output

Metric	Fitverse	Traditional apps	Improvements
FPS	42	19	+121%
Model Local Time	1.8 s	3.1s	42% Faster
Form Accuracy	92.3%	70.1%	+32%
User Retention	78%	55%	41%

Key Observations:

- WebGPU significantly boosts performance
- Accurate rep counting (96.7% optimal threshold at 60°)
- Form deviations reduced with real-time feedback

Performance Output (Visualization of Results)

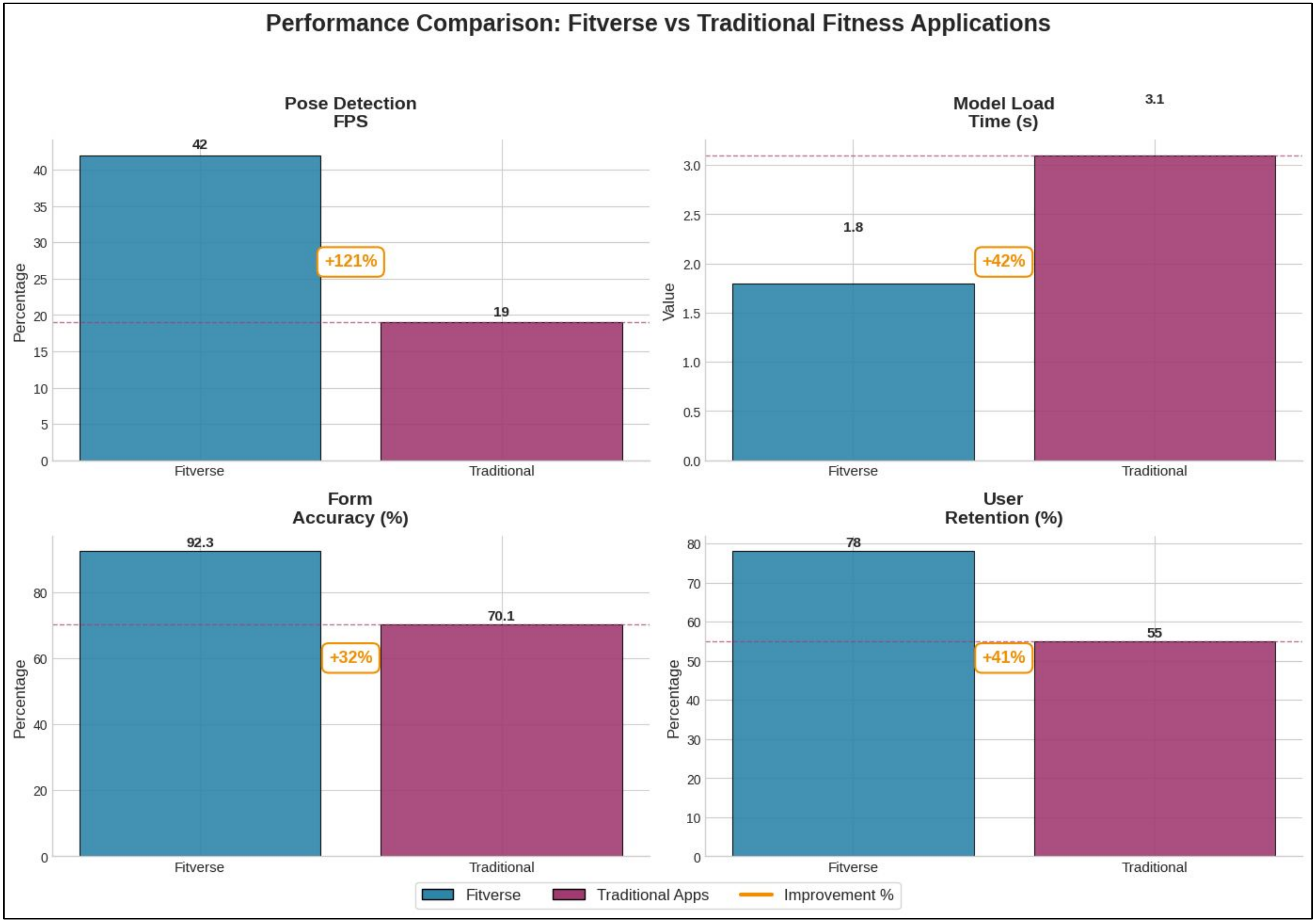


Fig. 7: Comparing Performance of Fitverse and traditional fitness applications. Fitverse shows significant improvements in frame rate (FPS), model load time, form accuracy, and user retention.

CONCLUSION & FUTURE WORK

CONCLUSION

- Fitverse proves advanced fitness analysis is possible inside a browser
- Provides accurate form correction, high FPS, and personalized workouts
- Outperforms traditional apps in accuracy, retention, and engagement
- Creates an accessible and scalable fitness ecosystem

FUTURE WORK

- 3D pose estimation using multi-camera
- Wearable integration for richer data
- Medical-grade monitoring system
- Social fitness challenges & community features

10. 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.

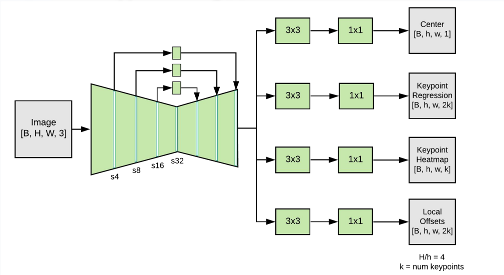
Models we used for pose classification

MoveNet and **PoseNet** are deep learning models designed for **human pose estimation**, which involves detecting the position of key body joints (like shoulders, elbows, knees) in images or video frames.

MoveNet

MoveNet is a **high-performance, real-time pose estimation model** developed by Google.

- **Architecture:**
  MoveNet is based on a **centered single-person pose estimation** approach. Internally, it uses a **Convolutional Neural Network (CNN)** backbone for feature extraction, followed by **regression heads** that output keypoint coordinates directly.

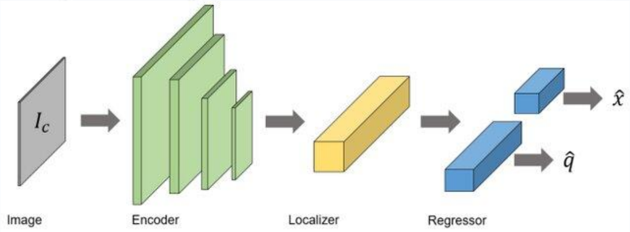


- **Advantages:**
  - Real-time performance (can run at >30fps on mobile).
  - High accuracy.
  - Robust to occlusions and fast movement.

PoseNet

PoseNet is an earlier pose estimation model developed by Google, also using a CNN-based architecture.

- **Architecture**:
  - Based on **MobileNet** or **ResNet** as the backbone.
  - Outputs a **heatmap** for each keypoint indicating the likelihood of its location.
  - Also outputs **offset vectors** to refine the predicted positions.



- **Advantages**:
  - Lightweight and runs in the browser (e.g., with TensorFlow.js).
  - Good for mobile and web-based applications.

Exercises we have on our platform

Upper Body

- Pushup
- Pullup
- Shoulder Press
- Bicep Curls
- Front Raises

Lower Body

- Squats
- Lunges
- HighKnees

Desk Exercises

- Knee Raises
- Curls
- Hand Raises