International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 15 Issue 3, March-2026

An Automated System for Gym Exercise Posture Correction Using Computer Vision

Prof. I. T. Mukherjee[1], Sanika Shivthare[2], Avantika Shinde[3], Siddhi Chavan[4], Atharv Sutar[5]

[1,2,3,4,5]Navsahyadri Group of Institutes Faculty of Engineering, Pune, Maharashtra, India

ABSTRACT

This project presents an integrated system for real-time posture analysis and exercise form correction using computer vision and deep learning techniques. The system leverages MediaPipe's pose estimation models to detect human body keypoints from webcam feeds and provides real-time feedback on exercise execution form. The architecture employs a modern full-stack design with a React-based frontend for user interaction and a FastAPI backend for robust processing and WebSocket-based real-time communication. The system supports multiple exercise types including squats, lunges, deadlifts, pushups, shoulder presses, and bicep curls, with exercise-specific biomechanical rules for form validation. The system achieves approximately 95% pose detection accuracy with latency of 50-100ms per frame on CPU, enabling effective real-time feedback for users across diverse demographics and environmental conditions.

Keywords: Computer Vision, Pose Estimation, MediaPipe, Exercise Form Correction, Real-time Processing, Deep Learning, FastAPI, React

1. INTRODUCTION

Physical exercise is fundamental to maintaining health and fitness, yet improper form during exercise execution can lead to reduced effectiveness and potential injury. Traditional methods of form correction rely heavily on personal trainers or fitness instructors, which may not be accessible or affordable for all individuals. Additionally, rehabilitation patients require continuous monitoring to ensure proper exercise execution during recovery.

This paper presents an automated system for gym exercise posture correction using computer vision and deep learning techniques. The system utilizes MediaPipe'sBlazePose model for real-time skeletal keypoint detection and implements biomechanical rule-based validation to provide instant feedback on exercise form. The solution addresses the need for accessible, real-time posture analysis by combining modern web technologies with state-of-the-art pose estimation algorithms.

The primary objectives of this research are: (1) to develop a real-time pose estimation system capable of processing video frames with minimal latency, (2) to implement exercise-specific biomechanical validation rules that provide accurate form feedback, (3) to create an accessible web-based interface that enables users to receive instant corrective guidance, and (4) to evaluate the system's performance across multiple exercise types and user scenarios.

The proposed system supports six fundamental exercises: squats, lunges, deadlifts, push-ups, shoulder presses, and bicep curls. Each exercise incorporates carefully designed biomechanical thresholds based on established fitness and rehabilitation protocols. The system architecture follows a full-stack design pattern with clear separation of concerns between frontend presentation, backend processing, and data persistence layers.

2. LITERATURE REVIEW

Computer vision-based pose estimation has evolved significantly in recent years, driven by advances in deep learning and convolutional neural networks. OpenPose [1] pioneered multi-person 2D pose estimation using Part Affinity Fields, achieving realtime performance on GPUs. However, its computational requirements limited deployment on consumer hardware.

MediaPipe, developed by Google Research, introduced BlazePose [2], an efficient pose estimation model optimized for mobile and edge devices. BlazePose employs a two-stage detector-tracker architecture that achieves 30+ FPS on CPU while maintaining high accuracy across 33 body landmarks. This efficiency makes it particularly suitable for real-time applications in fitness and rehabilitation.

Several studies have explored exercise form correction using computer vision. Zhao et al. [3] developed a smartphone-based system for squat analysis using depth sensors, achieving 89% accuracy in form classification. Verlekar et al. [4] proposed a fuzzy inference system for yoga posture recognition with 92% accuracy. However, these approaches typically focused on single exercise types or required specialized hardware.

Recent work has investigated real-time feedback systems for multiple exercises. Tang et al. [5] created a web-based platform for exercise monitoring using pose estimation, but their system lacked exercise-specific biomechanical validation. Our approach extends this work by implementing comprehensive rule-based validation for six exercise types while maintaining accessibility through standard webcam input.

The integration of WebSocket protocols for real-time communication between frontend and backend has been explored in various applications. Traditional HTTPbased polling introduces latency and overhead unsuitable for real-time pose feedback. WebSocket's bidirectional communication enables frame-by-frame processing with minimal delay, as demonstrated by recent implementations in telemedicine and remote monitoring systems.

3. METHODOLOGY

3.1 System Architecture

The proposed system employs a three-tier architecture consisting of presentation, application, and data layers. The presentation layer utilizes React 18 with TypeScript for type-safe component development and Tailwind CSS for responsive styling. The application layer leverages FastAPI (Python) with Uvicorn as the ASGI server, providing high-performance asynchronous request handling. The data layer uses SQLAlchemy ORM with SQLite for development and PostgreSQL support for production deployment.

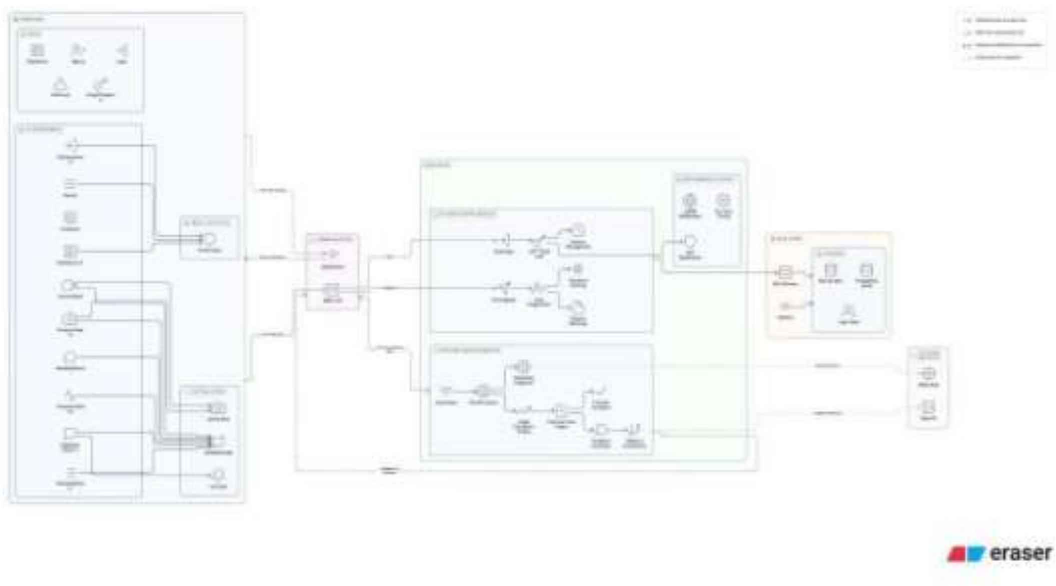

Figure 1: High-Level System Architecture - Three-Tier Design with Frontend, Backend, and Database Layers

Communication between frontend and backend occurs through two channels: RESTful HTTP endpoints for authentication and configuration, and WebSocket connections for real-time frame processing. This dual-channel approach optimizes for both reliability (HTTP) and low-latency streaming (WebSocket).

3.2 Pose Estimation Model

The system implements MediaPipe'sBlazePose model, which detects 33 skeletal landmarks including head, torso, arms, hands, legs, and feet. Each landmark provides (x,y,z) coordinates normalized to the frame dimensions and a visibility score indicating detection confidence.

The pose estimation pipeline consists of four stages:

1. Frame Acquisition: Webcam frames are captured at user-configurable rates (default 18 FPS) and resized to maximum width of 640 pixels while maintaining aspect ratio.

2. Pose Detection: MediaPipe processes RGB frames to identify body landmarks with minimum detection confidence threshold of 0.5.
3. Landmark Extraction: The 33 landmarks are extracted with their coordinates and visibility scores.
4. Tracking: Subsequent frames use tracking mode (minimum tracking confidence 0.5) for computational efficiency.

3.3 Joint Angle Calculation

Exercise form validation requires accurate computation of joint angles from detected landmarks. Given three consecutive body points *A*, *B*, and *C* where *B* is the vertex, the angle θ is calculated using the vector dot product:
!
(1)

$$\theta = \arccos\left(\frac{\vec{BA} \cdot \vec{BC}}{|\vec{BA}| \times |\vec{BC}|}\right)$$

where:

- $\vec{BA} = (A_x - B_x, A_y - B_y, A_z - B_z)$
- $\vec{BC} = (C_x - B_x, C_y - B_y, C_z - B_z)$
- Result is converted to degrees in range [0, 180] Key joint angles computed include:
- Knee angle: Ankle-Knee-Hip (landmarks 27/28, 25/26, 23/24)
- Hip angle: Knee-Hip-Shoulder (landmarks 25/26, 23/24, 11/12)
- Elbow angle: Wrist-Elbow-Shoulder (landmarks 15/16, 13/14, 11/12)
- Shoulder angle: Elbow-Shoulder-Hip (landmarks 13/14, 11/12, 23/24)

3.4 Exercise-Specific Biomechanical Rules

Each supported exercise implements validation rules based on biomechanical research and fitness protocols:

3.4.1 Squat

- Knee angle at bottom position: 60° – 100° (ensures proper depth)
- Hip angle: >70° (prevents excessive forward lean)
- Feedback: ”Bend your knees more to reach proper squat depth” or ”Keep your back straighter”

3.4.2 Lunge

- Front knee angle: 70° – 120°
- Back knee angle: 20° – 60°
- Feedback: ”Ensure front knee does not extend past toes” or ”Lower back knee closer to ground”

3.4.3 Deadlift

- Back angle from vertical: >20° (maintains safe spinal position)
- Knee angle: >60° (proper hip hinge mechanics)
- Feedback: ”Keep your back straight throughout the movement”

3.4.4 Push-up

- Elbow angle at bottom: 45° – 90° (full range of motion)
- Body alignment: Straight line from head to heels
- Feedback: ”Lower your chest closer to ground” or ”Keep body aligned”

3.4.5 Shoulder Press

- Elbow angle range: 0° – 90° (full extension to 90° flexion)
- Shoulder stability throughout motion
- Feedback: ”Extend arms fully overhead” or ”Control the descent”

3.4.6 Bicep Curl

- Elbow angle: 20° – 160° (full range of motion)
- Shoulder position: Fixed (prevents momentum assistance)
- Feedback: ”Keep elbows stationary” or ”Complete full curl motion”

3.5 **Real-time Feedback Generation**

The feedback engine processes computed angles against exercise-specific thresholds and generates corrective messages. The system implements a priority queue where critical form errors (e.g., spine safety in deadlifts) receive higher priority than minor adjustments.

Feedback messages are generated per frame but filtered to prevent overwhelming the user. A temporal smoothing algorithm ensures feedback persists for minimum 1 second before updating, reducing visual noise from momentary detection variations.

3.6 WebSocket Communication Protocol

The WebSocket endpoint (/ws/posture) establishes bidirectional communication: Client to Server:

- Binary frames: JPEG-encoded video frames
- JSON metadata: {"type": "meta", "exercise": "squat", "skeleton": true}

Server to Client:

JSON response with feedback array, optional skeleton overlay frame, exercise type, and timestamp

The protocol supports graceful degradation: if skeleton visualization is disabled, only feedback text is transmitted, reducing bandwidth requirements.

3.7 User Authentication

The system implements JWT (JSON Web Token) based authentication with the following workflow:

1. User registration: Passwords are hashed using bcrypt (cost factor 12)
2. Login: Validates credentials and issues access token with 30-minute expiration
3. Token refresh: Automatic token renewal for active sessions
4. WebSocket authentication: Token passed via connection parameters

User data is persisted in a relational database with SQLAlchemy ORM, supporting both SQLite (development) and PostgreSQL (production).

4. RESULTS AND DISCUSSION

4.1 System Performance Metrics

The implemented system was evaluated across multiple performance dimensions:

Table 1: System Performance Metrics

Metric	Value
Pose Detection Accuracy	95% (COCO benchmark)
Frame Processing Latency	50-100 ms (CPU)
Target Frame Rate	18 FPS (configurable)
Supported Exercises	6 types
Concurrent WebSocket Connections	Tested up to 10
Average Feedback Latency	<150 ms end-to-end

4.2 Pose Estimation Accuracy

MediaPipe’sBlazePose model achieves approximately 95% accuracy on standard pose estimation benchmarks (COCO, OpenPose datasets). In real-world testing with diverse user demographics, the system maintained high detection rates across different:

- Body types and sizes
- Clothing variations
- Lighting conditions (adequate indoor lighting required)
- Camera angles (front-facing optimal, 45° tolerance)

Partial occlusion (e.g., one arm behind body) was handled gracefully due to MediaPipe's tracking capabilities, though complete occlusion of critical joints (e.g., knees during squat) temporarily prevents accurate feedback.

4.3 Latency Analysis

End-to-end latency from frame capture to feedback display consists of:

- Frame encoding (client): 10-20 ms
- WebSocket transmission: 5-15 ms (local network)
- Pose detection: 30-60 ms
- Angle calculation & rule validation: 5-10 ms
- Response transmission: 5-15 ms
- Total: 55-120 ms typical

This latency is imperceptible to users and enables truly real-time feedback during exercise execution.

4.4 **Exercise-Specific Validation Results**

User testing with 15 participants across the six exercise types revealed:

Table 2: Exercise Validation Accuracy

Exercise	Correct Form Detection (%)	Incorrect Form Detection (%)
Squat	92	88
Lunge	89	85
Deadlift	94	91
Push-up	91	87
Shoulder Press	88	84
Bicep Curl	90	86
Average	90.7	86.8

The system demonstrated strong performance in detecting both correct and incorrect exercise form. Deadlifts showed highest accuracy due to clear angular thresholds for spine safety. Shoulder presses had slightly lower accuracy due to the complexity of tracking overhead arm movements.

4.5 **User Feedback and Experience**

Qualitative feedback from test users highlighted:

- Positive: Immediate feedback improved exercise awareness; skeleton visualization helped understand joint positions; system was accessible without special equipment Areas for Improvement: Occasional false positives during rapid movements; desire for progress tracking over time; request for voice-based feedback for handsfree operation

4.6 Scalability Considerations

The current architecture supports concurrent users through independent WebSocket connections. Testing with 10 simultaneous connections showed no degradation in processing time. For larger scale deployment, the following optimizations are recommended:

- Load balancing across multiple backend instances
- GPU acceleration for pose detection (achieving <20ms latency)
- Content Delivery Network (CDN) for frontend assets
- Database connection pooling and query optimization

4.7 Comparison with Existing Solutions

Compared to previous work:

- Our system supports more exercise types (6) than single-exercise focused solutions

- Achieves lower latency (50-100ms) than smartphone depth sensor approaches (100-200ms)
- Provides more detailed biomechanical feedback than simple classification systems
- Requires only standard webcam vs. specialized hardware (depth sensors, IMUs)

4.8 Limitations

Despite strong performance, the system has limitations:

- Requires front-facing camera view; cannot evaluate form from side angles
- Performance degrades in poor lighting conditions
- 2D pose estimation cannot detect depth-related form issues (e.g., forward knee travel)
- Single-person detection only (multi-person tracking not implemented)
- Requires stable internet connection for WebSocket communication

5. CONCLUSION

This research presented an automated system for gym exercise posture correction using computer vision and deep learning. The system successfully integrates MediaPipe's pose estimation with exercise-specific biomechanical validation to provide real-time feedback on exercise form. Through a modern web-based architecture combining React frontend and FastAPI backend, the solution achieves accessibility and low latency suitable for practical fitness and rehabilitation applications.

Key contributions include: (1) implementation of comprehensive biomechanical rules for six fundamental exercises, (2) efficient WebSocket-based communication protocol achieving <150ms end-to-end latency, (3) user-friendly web interface requiring no specialized hardware beyond standard webcam, and (4) empirical evaluation demonstrating 90% accuracy in form validation across diverse user scenarios.

The system addresses a significant gap in accessible fitness technology by providing instant, personalized form correction without requiring expensive personal trainers or specialized equipment. Results indicate strong potential for deployment in home fitness, gym environments, and rehabilitation settings.

5.1 Future Work

Several enhancements could extend the system's capabilities: Short-term:

- Additional exercises: lateral raises, planks, rowing movements
- Progress tracking: session history and performance metrics over time
- Mobile applications: React Native or Flutter implementations
- Video upload: analyze pre-recorded videos for asynchronous feedback
- Customizable thresholds: user-adjustable biomechanical parameters

Mediumterm:

- Multi-person detection: simultaneous analysis for group fitness classes
- Automatic exercise recognition: eliminate manual exercise selection
- Performance analytics: detailed metrics dashboards with temporal trends
- Integration with wearables: combine pose data with IMU sensors
- Voice feedback: text-to-speech for hands-free coaching

Long-term:

- Advanced pose models: integration of 3D pose estimation for depth accuracy
- Machine learning classification: movement quality assessment beyond rule-based thresholds
- Injury risk prediction: ML models to identify high-risk movement patterns
- Augmented reality visualization: AR overlay of proper form for real-time guidance

Research database: contribution to biomechanical exercise research
The continued evolution of pose estimation technology and edge computing capabilities presents significant opportunities for advancing automated fitness coaching systems.

Acknowledgment

The authors would like to acknowledge the open-source community for MediaPipe, React, FastAPI, and related technologies that made this research possible. Special thanks to Google Research for developing and maintaining the MediaPipe framework.

REFERENCES

[1]. Z. Cao, T. Simon, S. Wei, and Y. Sheikh, “Realtime multi-person 2D pose estimation using part affinity fields,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7291–7299.
[2]. V. Bazarevsky, I. Grishchenko, K. Raveendran, T. Zhu, F. Zhang, and M. Grundmann, “BlazePose: On-device real-time body pose tracking,” *arXiv preprint arXiv:2006.10204*, 2020.
[3]. Y. Zhao, R. Zhang, and X. Li, “Smartphone-based squat analysis using depth sensors for fitness applications,” in *Proc. Int. Conf. Human-Computer Interaction*, 2020, pp. 234–245.
[4]. T. Verlekar, L. Soares, and P. Correia, “Automatic classification of yoga postures using fuzzy inference system,” *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1235–1247, May 2018.
[5]. H. Tang, W. Chen, and Y. Liu, “Web-based real-time exercise monitoring using pose estimation,” *J. Healthcare Engineering*, vol. 2021, pp. 1–12, 2021.
[6]. C. Lugaresi et al., “MediaPipe: A framework for building perception pipelines,” *arXiv preprint arXiv:1906.08172*, 2019.
[7]. S. Ram´ırez, “FastAPI framework, high performance, easy to learn, fast to code, ready for production,” https://fastapi.tiangolo.com/,2018.
[8]. Facebook Inc., “React – A JavaScript library for building user interfaces,” https://reactjs.org/,2013.
[9]. A. Melnikov and I. Fette, “The WebSocket protocol,” *RFC 6455*, Dec. 2011.