Fitness Buddy: A Client-Side Web Application for Real-Time Pose Detection, Automated Repetition Counting, and Personalized Nutritional Planning

Anuj - 2202310100019
Ashish - 2302310109003
Piyush Gupta - 220231010062 Aman
Singh - 2202310100010

Abstract

Traditional fitness tracking and nutritional planning often require manual data entry, which is prone to inaccuracy and user fatigue. This paper presents 'Fitness Buddy,' a privacy-centric, serverless web application that utilizes edge computing within the browser to deliver automated workout tracking and personalized diet generation. By integrating TensorFlow.js and the MoveNet SinglePose Lightning model, the application achieves real-time, accurate kinematic analysis for automated repetition counting across multiple exercises. Furthermore, it incorporates a dynamic nutritional algorithm based on the Mifflin-St Jeor equation to generate customized, downloadable diet plans. This approach demonstrates the viability of utilizing client-side machine learning for complex health and fitness applications, ensuring zero-latency processing and strict user data privacy.

1. Introduction

The advent of computer vision and machine learning (ML) has revolutionized the fitness industry, offering new modalities for monitoring physical activity. However, many existing solutions rely on cloud-based processing, which introduces latency, requires persistent internet connectivity, and raises significant privacy concerns regarding the transmission of user video data. Fitness Buddy addresses these challenges by processing video streams entirely client-side. The primary objectives of this project are:

1. To implement real-time, accurate human pose estimation in a standard web browser without specialized hardware.
2. To automate repetition counting using heuristic-based kinematic analysis.
3. To provide a scientifically backed, dynamically generated nutritional plan customized to the user's specific biometric data and fitness goals.

2. Methodology

2.1 System Architecture

The application is designed as a Single Page Application (SPA) utilizing HTML5, Vanilla CSS3, and modern JavaScript (ES6+). The architecture is inherently serverless, relying on the user's local computing resources. Data persistence is managed via the browser's native localStorage API, maintaining workout histories and user preferences across sessions.

2.2 Real-Time Pose Estimation

At the core of the workout tracking module is the MoveNet SinglePose Lightning model, deployed via TensorFlow.js. MoveNet is an ultra-fast and accurate model that detects 17 keypoints of a body. By leveraging the WebGL backend, TensorFlow.js accelerates tensor operations using the device's GPU, enabling the model to process the getUserMedia video stream at upwards of 30 frames per second (FPS). This low latency is critical for providing immediate feedback during exercises.

2.3 Kinematic Analysis and State Machine

Repetition counting is achieved through a deterministic state machine based on joint angles. For each supported exercise (e.g., Bicep Curls, Squats, Push-ups), three specific keypoints are tracked. The angle theta between three

points A, B (vertex), and C is calculated using the arctangent function. To mitigate jitter caused by occlusions or minor model inaccuracies, a 5-frame rolling average smoothing function is applied to the calculated angle. The state machine transitions between 'up' and 'down' states based on predefined empirical thresholds.

2.4 Nutritional Planning Algorithm

The diet generation module calculates the user's Basal Metabolic Rate (BMR) using the Mifflin-St Jeor equation. For males: BMR = 10 * weight(kg) + 6.25 * height(cm) - 5 * age + 5. For females: BMR = 10 * weight(kg) + 6.25 * height(cm) - 5 * age - 161. The Total Daily Energy Expenditure (TDEE) is derived by applying an activity multiplier (1.2 to 1.9). Goal-specific adjustments determine the final daily caloric target.

3. Implementation Details

3.1 Edge Computing and Privacy

By processing the camera feed entirely within the browser via camera.js and TensorFlow.js, Fitness Buddy ensures that no visual data is ever transmitted over the network.

3.2 Automated PDF Generation

To provide actionable nutritional advice, the application utilizes jsPDF to compile the generated diet plan into a formatted, downloadable PDF document.

4. User Interface and Experience (UI/UX)

The application features a modern, responsive design with a dark theme utilizing glassmorphism effects. Real-time visual feedback is provided during workouts.

5. Conclusion and Future Work

Fitness Buddy successfully demonstrates that complex, AI-driven computer vision tasks can be executed efficiently within a standard web browser.

References

1. TensorFlow.js: Machine Learning for the Web and Beyond.
2. MoveNet: Ultra fast and accurate pose detection model.
3. Mifflin, M. D., St Jeor, S. T., et al. (1990).
4. jsPDF: Client-side JavaScript PDF generation.