AI Fitness Trainer Real-Time Pose Correction and Feedback

Ting Li
Khoury College of Computer Sciences
Northeastern University
Oakland, CA
li.ting4@northeastern.edu

Duo Xu
Khoury College of Computer Sciences
Northeastern University
San Jose, CA
xu.duo3@northeastern.edu

Alexander Leon
Khoury College of Computer Sciences
Northeastern University
Boston, MA
leon.a@northeastern.edu

Yulong Feng
Khoury College of Computer Sciences
Northeastern University
San Jose, CA
feng.yulo@northeastern.edu

***Abstract*— This paper presents a real-time AI fitness trainer that provides users with immediate feedback on exercise performance using computer vision and sequential modeling. The system captures live video via webcam, extracts pose landmarks using MediaPipe, and classifies exercise types—squat, push-up, and plank—through a pre-trained Long Short-Term Memory (LSTM) network. Based on a rolling window of 70 frames, the model identifies actions and triggers custom rule-based evaluation logic to detect form errors. Users receive voice and visual feedback in real time, including posture correction prompts, repetition counts, and form scores. The system is fully interactive, CPU-efficient, and requires no prior setup or additional training. Experimental results confirm the system's ability to deliver accurate classification and actionable feedback across various exercise scenarios. This work demonstrates the potential for lightweight, extensible, and accessible AI-powered solutions in personal fitness training.**

Keywords—Pose Estimation, Real-Time Feedback, LSTM, Mediapipe, Fitness Coaching

I. Introduction

Regular physical exercise is essential for maintaining health and fitness, yet performing movements with incorrect form can lead to reduced effectiveness and an increased risk of injury. Traditionally, personal trainers have been relied upon to monitor and correct exercise techniques; however, this approach is not always accessible or affordable for all users. In response, this paper proposes an AI-powered fitness trainer that leverages computer vision to provide real-time feedback on workout form. By analyzing live or recorded videos of users performing common exercises—such as squats, push-ups, or planks—the system offers automatic pose evaluation, form correction, and repetition counting. Through integrated visual and voice guidance, users receive immediate feedback to help them maintain proper technique, enabling safer and more effective workouts without the need for human supervision.

To facilitate accurate pose estimation without the need to collect or annotate a large dataset, this paper adopts MediaPipe Pose, a pre-trained deep learning model developed by Google.[1] MediaPipe Pose detects 33 key body landmarks in real time using only a single RGB camera, making it highly efficient and practical for deployment. The framework provides accessible Python APIs and does not require any additional training, allowing the system to focus on downstream tasks such as pose correction and repetition tracking. This integration enables the proposed solution to deliver robust, real-time feedback with minimal computational overhead and development cost.

To further enhance the system's ability to interpret movement over time, we incorporate a temporal modeling component that can capture the dynamics of full exercise sequences. While MediaPipe Pose provides accurate per-frame landmark detection, effective action recognition requires understanding how these landmarks move across consecutive frames. To this end, the system employs a Long Short-Term Memory (LSTM) network to model the temporal dependencies within sequences of selected body landmarks. Introduced by Hochreiter and Schmidhuber[2], LSTM is a variant of recurrent neural networks (RNNs) designed to capture long-range dependencies in sequential data while mitigating issues such as vanishing or exploding gradients that commonly affect standard RNNs. Unlike traditional RNNs, which often struggle to retain information beyond a few time steps, LSTMs utilize a memory cell and three gated mechanisms—forget gate, input gate, and output gate—to selectively manage the flow of information. These gates allow the network to determine which past information to keep or discard and which new input to incorporate, thus maintaining a meaningful internal state throughout the sequence. Simplified variants such as Gated Recurrent Units (GRUs) have also been proposed to reduce complexity while preserving sequential modeling performance [3]. Bidirectional LSTM (BiLSTM) models further improve sequence understanding by incorporating both past and future context during inference [4]. Prior studies have successfully applied convolutional and LSTM-based architectures to activity recognition tasks using multimodal data, demonstrating the effectiveness of temporal modeling in fitness and healthcare applications [5].

In this work, we present an AI-based fitness feedback system that combines real-time pose estimation with temporal sequence modeling to evaluate and guide bodyweight exercises. The system supports three fundamental exercises—squats, push-ups, and planks—and automatically recognizes the performed activity without requiring prior user setup. Leveraging MediaPipe Pose, it extracts frame-level body landmarks, which are then analyzed using an LSTM network to model motion dynamics and classify exercise types. Drawing on predefined biomechanical rules, the system detects common posture errors and provides audio-visual feedback to encourage proper form. It also counts valid repetitions and assigns form scores at both the repetition and set levels, offering users immediate and personalized insight into their exercise performance.

II. Related Works

Recent advancements in computer vision and deep learning have significantly contributed to the development of

intelligent fitness coaching systems. Our work builds upon and diverges from several key studies in this domain.

Fieraru et al. introduced AIFit, a system that reconstructs 3D human pose and motion to provide real-time feedback during fitness exercises.[6] By comparing user performance against standards learned from professional trainers, AIFit offers localized, quantitative feedback to reduce injury risk and promote continuous improvement. While AIFit focuses on 3D pose estimation and requires extensive training data, our system leverages 2D pose landmarks from MediaPipe and employs rule-based logic for feedback, enabling a more lightweight and accessible solution.

Lee et al. proposed a novel 3D pose estimation method using propagating LSTM networks that model joint interdependencies.[7] Their approach captures the spatial correlations of human posture by sequentially reconstructing 3D joint positions from 2D inputs. This method enhances the accuracy of 3D pose estimation, particularly at body extremities. In contrast, our work utilizes a pre-trained LSTM model to classify sequences of 3D pose landmarks into specific exercises, focusing on real-time feedback rather than precise 3D reconstruction.

Carreira et al. introduced an iterative error feedback mechanism for 2D human pose estimation, where predictions are refined through successive corrections.[8] This approach improves the accuracy of pose estimation by incorporating feedback loops into the prediction process. Our system does not modify the pose estimates produced by MediaPipe Pose, but instead uses the estimates for exercise form correction.

These studies collectively inform our approach, which combines real-time 3D pose estimation with sequential modeling and rule-based feedback to create an accessible and efficient AI fitness trainer.

III. METHODS

A. Data Preparation

To develop and validate our real-time feedback system, we constructed a small dataset consisting of pose sequences (exercise activities) from publicly available videos and self-recorded exercise clips. These videos were used both for fine-tuning the system's form correction logic and for training the action recognition model.

For initial testing and threshold calibration, we sourced a variety of workout demonstration videos from YouTube, along with webcam recordings of ourselves performing the target exercises. These samples provided diverse viewpoints and execution styles, allowing us to empirically determine angle thresholds for posture evaluation.

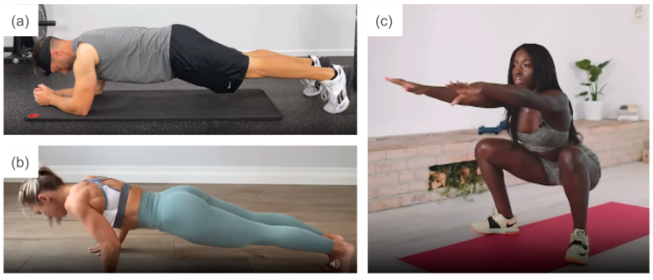


Fig. 1. Example of data sets

Our dataset includes three primary exercise categories: plank, push-up, and squat, each represented in multiple camera angles. The plank category consists of two variations—left and right side planks, both captured from a side view (see Figure 1(a)). The push-up category includes left-side, right-side, and front-facing views (see Figure 1(b)). The squat category similarly contains side (left and right) and front-facing perspectives (see Figure 1(c)). For each pose type, we collected approximately 10 video samples, providing sufficient variation to train a preliminary classifier and validate the system's generalizability across different users and camera angles.

This curated video dataset enables both pose label annotation and the extraction of sequential landmark data, serving as the foundation for the LSTM-based action recognition pipeline described in the subsequent sections.

B. Data Preprocessing

To prepare the data for model development and system validation, a structured pipeline was implemented to extract and label pose landmarks from categorized exercise videos. The pipeline begins by initializing the pose estimation module and creating an output CSV file, where each row contains the coordinates (x, y, z), visibility scores of detected landmarks (confidence), and the associated action label (plank, push-up, or squat).

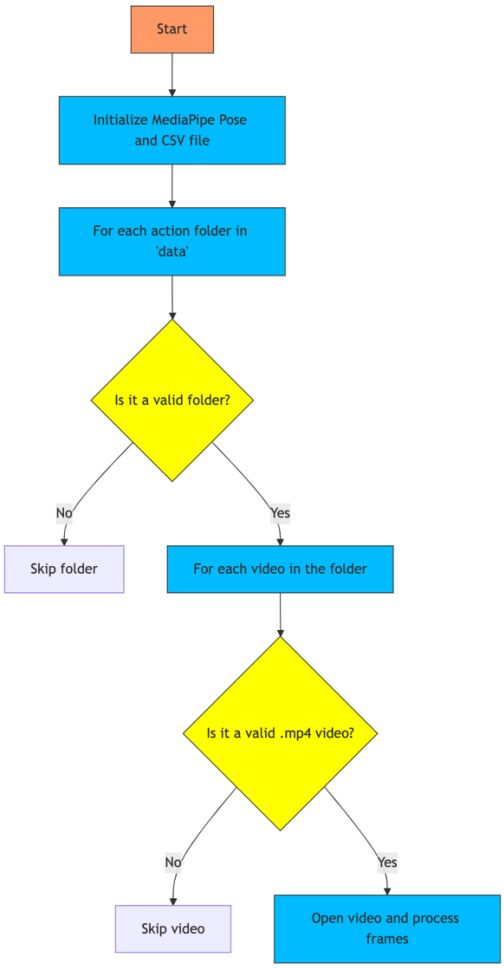


Fig. 2. Data preprocessing workflow – part 1: Initialization, folder/video validation, and frame-level processing logic.

The system processes each subdirectory within the main dataset folder, where each subdirectory corresponds to a specific exercise type. Only valid folders are included, and any non-relevant files are excluded. Within each folder, all video files are opened and read frame by frame using a standard computer vision interface (cv2 library).

As shown in Figure 2, each video frame is converted to the required format and passed to the pose estimation module. If pose landmarks are successfully detected, the joint positions and visibility values are extracted and recorded, along with the action label derived from the folder name. This process is repeated for all frames in the dataset, as outlined in Figure 3. Once processing is complete, all resources are released and the system outputs a final dataset ready for temporal modeling.

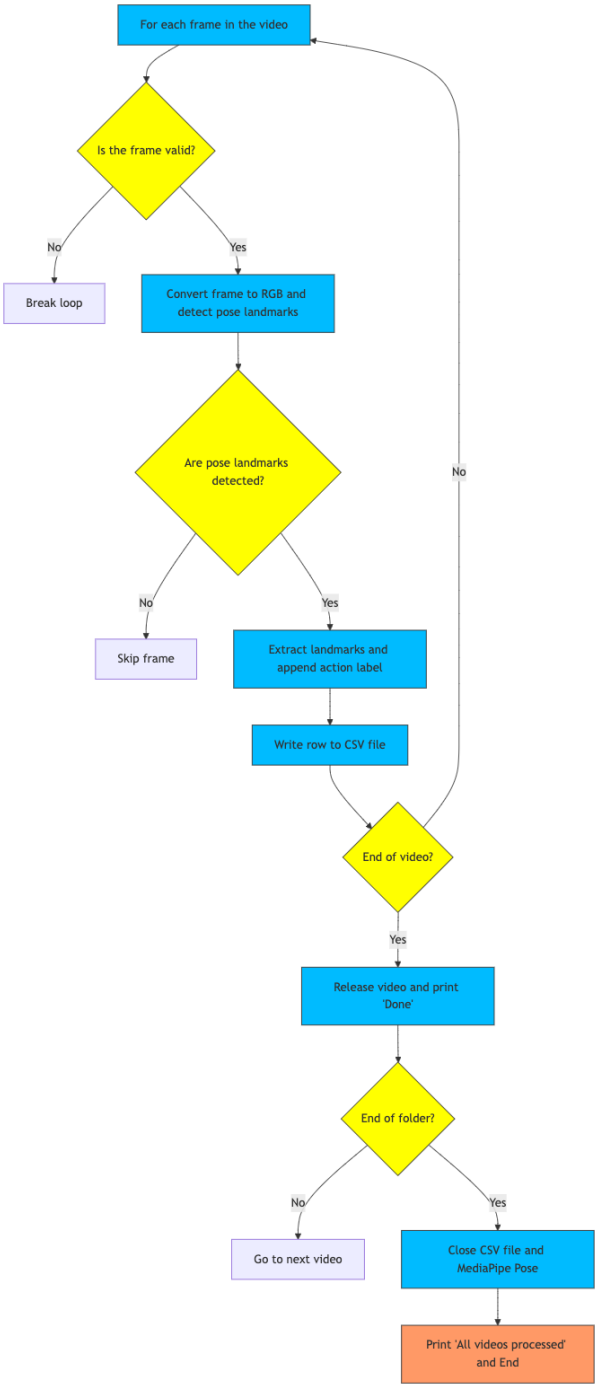


Fig. 3. Data preprocessing workflow – part 2: RGB conversion, pose landmark extraction, CSV writing, and resource cleanup.

Beyond dataset construction, this pose data serves as the foundation of our real-time feedback system. Our solution uses MediaPipe to extract and track joint positions frame by frame, enabling dynamic motion analysis. For each supported exercise—such as squats, push-ups, and planks—we define custom logic based on joint angles and relative body part positions to detect common form errors (e.g., "knees caving in" or "hips too high"). This logic is used to classify pose correctness and trigger immediate feedback through on-screen messages and voice alerts. In addition, the system supports automatic exercise recognition, real-time rep counting, and form scoring per repetition, all without requiring any prior setup from the user. Each exercise type applies a different set of evaluation rules, as summarized in a reference table included in the appendix.

Because MediaPipe is pre-trained and optimized for real-time inference, no additional model training is required to estimate pose landmarks (in subsequent sections, we train a mdoel to classify exercise types upon application launch). All computations are executed locally on standard personal laptops. The lightweight nature of the MediaPipe runtime allows the system to perform efficiently on CPUs, making real-time video analysis feasible without relying on external GPU resources.

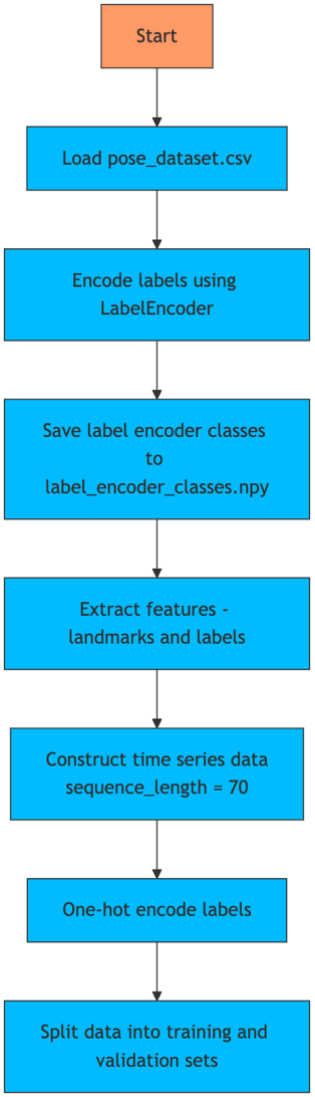


Fig. 4. Model training pipeline – part 1: Dataset loading, label encoding, sequence construction, and data splitting.

C. Model Training

To classify exercise types from sequential pose data, we trained a Long Short-Term Memory (LSTM) neural network using the processed dataset of labeled pose landmarks. The overall training workflow is illustrated in Figure 4, covering the steps from data loading and preprocessing to model construction and evaluation.

The process begins with loading a CSV file containing frame-wise pose data. Each row includes the 3D coordinates (x, y, z) and visibility scores of 33 body landmarks, along with an action label indicating the exercise type. Action labels are encoded into numeric values using a label encoding technique, and the label-to-class (reverse) mapping is saved for future inference.

Next, the dataset is divided into features and labels. Landmark coordinates and visibility values serve as the input features, while the corresponding exercise labels are extracted as targets. To enable sequential learning, the frame-wise data is restructured into fixed-length sequences of 70 consecutive frames. Each sequence is labeled based on the action associated with those frames. The resulting data has the shape (total number of frames, 70, number of features), which is compatible with LSTM input requirements.

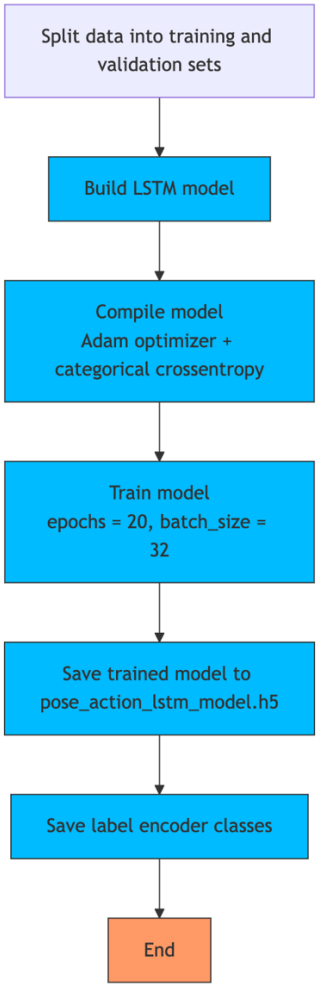


Fig. 5. Model training pipeline – part 2: LSTM model architecture, compilation, training, and saving outputs.

The labels are then converted to one-hot encoded vectors to support multi-class classification. The complete dataset is split into training and validation sets in an 80:20 ratio to allow for performance monitoring and generalization analysis during training.

The model architecture is shown in Figure 5. It consists of two LSTM layers: the first contains 64 units and returns the full sequence; the second contains 64 units and outputs only the final hidden state. A dropout layer is placed between the two LSTM layers to reduce overfitting. This is followed by a fully connected dense layer with 64 neurons and ReLU activation, and a final output layer with softmax activation to produce class probabilities for each exercise type.

The model is compiled using the Adam optimizer, with categorical crossentropy as the loss function and accuracy as the performance metric. It is trained for 20 epochs with a batch size of 32, using the validation set to monitor training progress.

Upon completion, the trained model is saved in HDF5 format for inference, along with the label encoder classes (in a .npy file format) for decoding predictions. Console outputs during training include data shape summaries, model architecture logs, and epoch-wise performance metrics, ensuring transparency in the training process.

This LSTM model provides the core classification capability of the system, enabling it to identify exercise types in real-time based on pose sequences.

IV. Experiments and Results

To validate the effectiveness of our real-time pose-based feedback system, we implemented and tested a complete application that captures live video, detects exercise actions, and provides immediate voice and visual feedback based on form correctness. The system was evaluated across multiple modules under realistic usage conditions to ensure robustness and usability.

The experiment setup simulates a home fitness environment, where a user performs supported exercises in front of a webcam. The system captures webcam video streams and processes each frame to obtain pose information for downstream activity recognition and feedback generation. A sliding window of recent frames (length 70) is maintained, forming a time-series input that is fed into the LSTM model for exercise classification.

To improve prediction reliability, the system implements a stability check mechanism. A predicted label is only confirmed if it appears consistently over multiple frames (e.g., 10). Once stabilized, the detected activity triggers a voice notification and invokes a corresponding evaluation module—such as squat, push-up, or plank—for posture-specific feedback. When push-up posture is detected without sufficient motion, the system dynamically adjusts the classification to plank, accounting for motion context.

Each exercise module applies custom biomechanical rules based on joint angles and body alignment to assess form quality. In squats, for example, the system checks for proper knee tracking and depth; in push-ups, elbow angles are measured to ensure full extension; and in planks, hip height and head angle are monitored for alignment. These evaluations inform real-time feedback delivered through visual text and synthesized voice alerts.

The system also supports interactive control: users can press ‘r’ to reset and start a countdown (default 5 seconds) or

'q' to exit the session. Real-time pose visualization is overlaid on the live video feed, including skeletal connections and status indicators. An additional motion-detection logic monitors landmark changes across frames to distinguish between active repetitions and static holds.

To summarize the scope and functionality of the system, Table 1 presents a detailed breakdown of the key modules tested during the experimental phase. It highlights the range of supported features, real-time responsiveness, and practical interaction design, all verified using a CPU-only laptop setup.

TABLE I. ERRORS DETECTION AND FEEDBACK MESSAGES

Squats:

Mistake	Detection Logic	Feedback
Not deep enough	Knee angle > 90°	"Go lower!"
Knees caving in	Distance between knees < threshold	"Keep knees aligned!"
Leaning too forward	Chest to hip angle < threshold	"Keep back straight!"

Push-ups:

Mistake	Detection Logic	Feedback
Hips too high	Hip higher than shoulders & ankles	"Lower your hips!"
Arms flaring	Elbow angle > 90°	"Keep elbows in!"
Not going low enough	Elbow < 90°	"Go deeper!"

Planks:

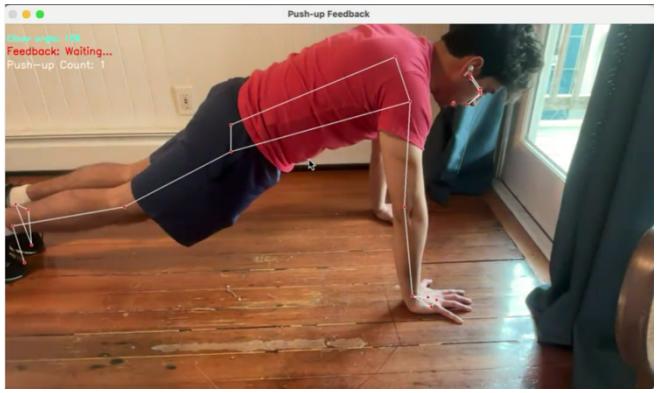
Mistake	Detection Logic	Feedback
Hips sagging	Hip lower than shoulders	"Raise your hips!"
Head position bad	Head tilt angle off from spine	"Keep head neutral!"

Overall, the experimental results demonstrate the system's ability to integrate multiple real-time components—pose tracking, sequence modeling, and rule-based feedback—into a unified user experience. The following results highlight its effectiveness in specific exercise scenarios.

A. Plank Detection and Feedback

As shown in Figure 6, when performing a standard side plank in front of the camera, the system correctly identified the pose and overlaid key feedback components. These included joint angle measurements, a visual assessment message, and a running timer for hold duration. The guidance helped the user maintain proper form throughout the position.

Fig. 6. Plank detection and feedback



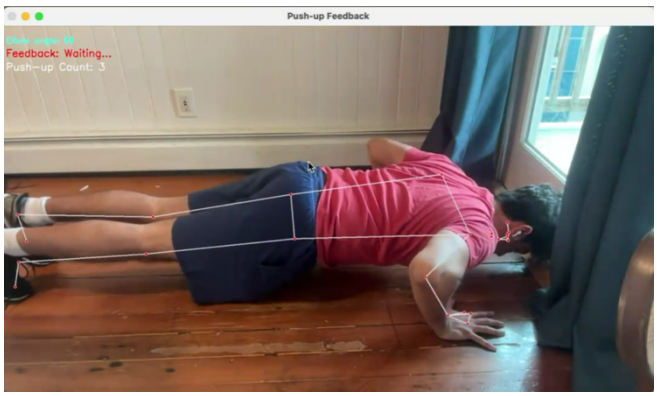


Fig. 7. Push-up detection and repetition counting

B. *Push-Up Detection and Repetition Counting*

In Figure 7, the user performs a series of standard push-ups. The system accurately recognized the movement, counted valid repetitions, and displayed real-time angle values and posture correctness feedback. Voice cues also notified the user of successful reps, enhancing engagement and awareness.

C. *Squat Detection and Scoring*

Figures 8 demonstrates the system's performance with squats. The application reliably detected the action from both frontal and lateral views, evaluated squat depth and knee position, and provided live repetition counts and form scoring per rep. Visual cues kept the user informed about posture quality on screen.

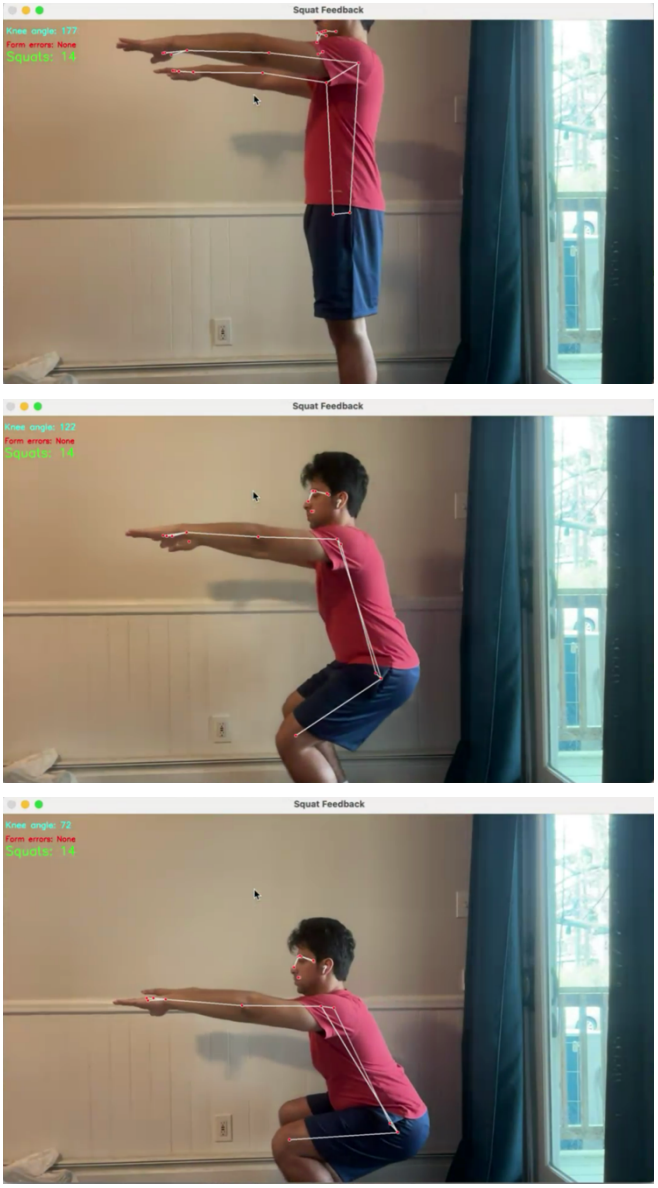


Fig. 8. Squat detection and scoring

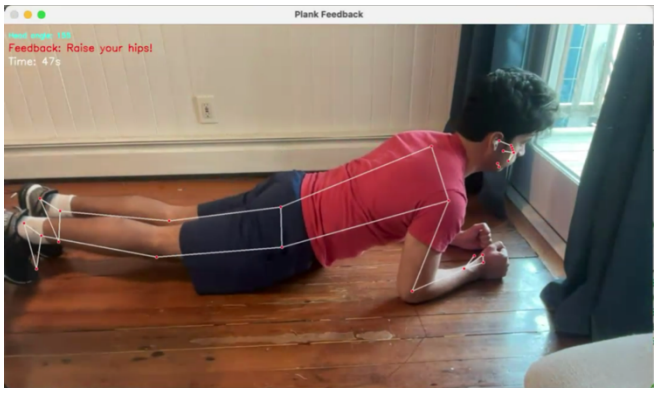


Fig. 9. Error detection and correction feedback

D. *Error Detection and Correction Feedback*

When exercises were executed incorrectly, the system successfully flagged form issues using pre-defined rules. As shown in Figure 9, a warning message appeared on screen alongside an audio alert, clearly instructing the user on how to correct posture errors, such as insufficient joint movement or misalignment.

V. Discussion and Summary

The proposed system demonstrates the feasibility of delivering intelligent, real-time exercise feedback using lightweight, accessible technologies. By integrating MediaPipe Pose for landmark detection and a pre-trained LSTM for sequence classification, the system provides meaningful guidance on both activity recognition and form correction—without the need for manual dataset annotation or GPU-based inference.

One of the core strengths of the system is its modular design, which enables extensibility to additional exercises or logic-based rules. Each action is assessed with specific biomechanical thresholds, making the system adaptable to various body types and workout styles. The audio-visual feedback loop proved highly effective in reinforcing correct technique and enhancing user engagement.

However, there are still limitations. The current rule-based feedback mechanism, while interpretable, may not generalize well across complex variations in motion. Moreover, the use of static thresholds for detecting errors (e.g., minimum angle changes) could lead to false positives in certain lighting or clothing conditions. Future improvements could include training a lightweight neural network for form scoring or integrating multi-camera input for 3D accuracy enhancement.

In summary, this work presents a functional and extensible real-time fitness coaching system. It combines pose estimation, temporal modeling, and real-time feedback into a cohesive experience that runs entirely on local CPU hardware. The system opens the door to accessible, AI-enhanced personal training tools that can be deployed at home, in gyms, or in rehabilitation contexts.

REFERENCES

[1] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, L. Ceze, and J. Shlens, "MediaPipe: A framework for building perception pipelines," arXiv preprint arXiv:1906.08172, 2019.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673–2681, 1997.

[4] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[5] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," Sensors, vol. 16, no. 1, p. 115, 2016.

[6] M. Fieraru, M. Zanfir, A. I. Zanfir, E. Marinoiu, and C. Sminchisescu, "AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, Jun. 2021, pp. 14056–14066.

[7] K. Lee, S. Roh, and S. Lee, "Propagating LSTM: 3D Pose Estimation Based on Joint Interdependency," in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 119–135.

[8] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human Pose Estimation with Iterative Error Feedback," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 4733–4742.