Real-Time Exercise Analysis and Corrective Feedback System

Prof. Samadhan Jadhav[*1], Neeraj Ghatage[*2], Atharva Joshi[*3], Atharva Jadhav[*4], Dhanashree Kamble[*5]

[*1]Assistant Professor, Department of Computer Engineering, PICT, Pune, Maharashtra, India

[*2,3,4,5] Student, Department of Computer Engineering, PICT, Pune, Maharashtra, India

ABSTRACT

Fitness activities offer numerous health benefits, but improper execution can lead to inefficiency and potential harm. When performing exercises incorrectly, individuals often fail to maintain the correct form or posture. In this research, a program has been developed to assess and provide feedback on the user's workout posture. Utilizing a camera, this system offers real-time interaction by detecting joints and evaluating workout accuracy through vector angles. The implementation leverages MediaPipe, a cross- platform ML framework compatible with Windows and Linux computers equipped with a webcam.

Keywords: Real-Time Exercise Analysis, Corrective Feedback System, Pose Estimation, Exercise Posture Correction, Joint Angle Calculation, MediaPipe Framework, Machine Learning

I. INTRODUCTION

Engaging in exercises is a commendable way to enhance one's fitness and overall health. However, the improper execution of exercises can pose significant risks, particularly when dealing with heavy weights that have the potential to cause severe muscle or ligament injuries. Many individuals enthusiastically participate in workout routines but often struggle to maintain the correct technique or posture. Such difficulties may stem from a lack of formal training, be it from fitness classes or a personal trainer, as well as the consequences of muscle soreness or attempting to lift weights beyond their capacity. Our goal is to offer a solution that helps individuals adopt proper exercise postures, thereby preventing injuries and enhancing the effectiveness of their workouts, using nothing more than a computer and a webcam.

In this research, we delve into the utilization of skeleton- based representations, which offer the advantages of being low-dimensional, interpretable, person-independent, and privacy- preserving. These skeletal representations allow us to concentrate on the essence of motion while making generalizations about an individual's appearance and background. We present a real-time body tracking pipeline that predicts the hand skeleton and the overall body concept, leveraging the capabilities of MediaPipe, a versatile framework for developing cross-platform machine learning solutions. Our approach involves posture estimation techniques, where the accuracy of exercise execution is assessed by fine-tuning the range of angles between relevant joints.

By combining these innovative techniques with the power of technology, we aim to revolutionize the way individuals approach exercise, making it safer, more efficient, and accessible to a broader audience. This system not only enhances the quality of workouts but also contributes to injury prevention and overall well-being.

II.LITERATURE SURVEY

[1] In this paper, the writer presents a body posture smart recommendation system, which detects user's posture and guides them according to the selected back exercise using a gyroscope sensory module embedded in the smart fitness suite. They proposed the system for two exercises, T-bar and bicep concentrated dumbbell curl. Along with this, a bicep curl muscle health detection feature is added to the proposed system, which detects muscle health in real-time. EMG sensor is used to stop the user from exercising in the extreme fatigue stage to prevent muscle injury. KNN model is used for the forward feature selection technique with 89% of accuracy. Subsequently, a user-guided recommendation feature is added which is based on the trained dataset over the android application using the text-to-speech feature in real-time. Future scope: In the future, the smart fitness suite could be trained for other body workouts as well. To increase the accuracy of the proposed system, they can consider the gyroscope drift issue to stabilize the signals for better classification of exercises. Finally, the proposed smart fitness suite can be made specifically for male or female users by collecting datasets separately.

[2] The detection of the 2D poses of many people in an image is done in this research using an efficient method. The method learns to associate body parts with people in the image using non- parametric representations known as Part Affinity Fields (PAFs). The technique uses the full image as the input for a two-branch CNN to jointly predict part affinity

fields for parts association and confidence maps for body part detection. To associate body part candidates, the parsing stage executes a series of bipartite matchings. Lastly, combine them into full-body positions for every per- son in the picture. No matter how many people are in the im- age, the architecture's global context encoding enables a greedy bottom-up parsing step that retains high accuracy. The architecture is designed to jointly learn part locations and their association via two branches of the same sequential prediction process. The method placed first in the inaugural COCO 2016 key points challenge and significantly exceeds the previous state-of-the-art result on the MPII MultiPerson benchmark, both in performance and efficiency. There is more improvement space in capturing spatial dependencies than in recognizing body parts appearances.

[3] For the pose estimation component, they utilize a pre- trained real-time system, called OpenPose, that can detect human body key-points in videos. They evaluate their posture identifier in different ways depending on the algorithm: for heuristic algorithms, they feed in all videos for evaluation, while for machine learning algorithms, they evaluate by splitting their video dataset into train and test sets, and report results on the test set. Pose Trainer application from a technical perspective as a pipeline system, consisting of multiple system stages as follows • Record and crop video • Pose estimation (OpenPose) • Normalize, clean and save key-points • Exercise perspective detection • Evaluate exercise using geometry or ML • Provide specific feedback on exercise The results of Pose Trainer on four different dumbbell (free motion) exercises: bicep curl, front raise, shoulder shrug, and standing shoulder press. For each exercise, they take both a geometric/heuristic approach, as well as a machine learning approach using dynamic time warping. They identified several extensions as strong opportunities for future work past this course project. One path would be to export Pose Trainer to smartphones, building an application that allows users to record a video and get pose feed- back at any place or time. Another direction would be to improve the pose feedback, providing specific suggestions on where the user's pose needs improvement (e.g., back, neck, shoulders), and suggesting targeted action. Also, they want to work on improved graphics, for instance, showing the user their labelled pose diagram, and comparing to the labelled pose diagram of a ground truth trainer.

[4] This Paper presents a groundbreaking application of Deep Neural Networks (DNNs) for human pose estimation, showcasing the advantages of DNN-based regression to joint coordinates and the use of a cascade of regressors. The approach achieves state-of-art or superior results on challenging academic datasets, indicating its potential for real-world applications. Additionally, the adaptation of generic convolutional neural networks for localization tasks demonstrates the versatility of neural networks across different domains. Future research will focus on the development of specialized architectures for localization problems, particularly in the context of pose estimation.

[5] This paper concludes by summarizing the key contributions of the proposed approach, particularly the incorporation of discriminative part template predictors within a pictorial structure framework for robust human pose estimation. The demonstrated superiority over independent part templates and state-of-the-art methods using tree structures underscores the potential of this methodology in real-world applications.

III.METHODOLOGY

3.1 Exercise Selection and Error Determination

The proposed system focuses on four foundational exercises, selected based on several criteria such as the frequency of performance among independent exercisers, the statistical likelihood of improper execution, the potential for injury when performed incorrectly, and the diversity of movement patterns and muscle groups engaged. These exercises are critical for assessing common errors in exercise execution and ensuring safety during workouts.

The selected exercises are as follows: The first exercise is the **Bicep Curl**, which is often performed incorrectly, with common errors including excessive shoulder movement (swinging), incomplete range of motion, and incorrect elbow positioning, such as drifting away from the torso. Additionally, asymmetrical execution between arms is another prevalent issue. The second exercise is the **Basic Plank**, where errors typically involve elevated or sagging hip positions, misalignment of the head and neck, issues with shoulder blade protraction or retraction, and incorrect foot positioning. The **Basic Squat** is the third exercise, with common errors like inadequate depth, knee valgus (knees collapsing inward), forward torso lean, heels raising off the ground, and asymmetrical weight distribution. The final exercise is the **Lunge**, which is prone to errors such as the front knee tracking beyond the toes, insufficient depth, instability or excessive torso lean, inadequate lowering of the back knee, and pelvic misalignment. Each of these errors was categorized based on severity—minor, moderate, and severe—along with the associated injury risk to prioritize feedback delivery during the exercise performance evaluation.

3.2 Data Collection and Preparation

To develop a robust and generalized model, the data collection process included both self-collected data and publicly available datasets. The self-collected data was obtained from 15 volunteers (8 male, 7 female), who were chosen across a range of fitness levels. Various recording conditions were employed to ensure diverse datasets, including different lighting setups (bright natural, moderate indoor, and low light), camera angles (front, side, and 45-degree angles), and distances (ranging from 5–12 feet from the subject). Volunteers wore a mix of tight-fitted and loose attire during the exercise performance, and each participant performed 10 correct and 5 incorrect repetitions per identified error. This data resulted in approximately 1200 annotated exercise instances, with around 300 instances for each exercise.
In addition to the self-collected data, public datasets were integrated to enhance the generalization capabilities of the model. These datasets included the *Yoga Postures Dataset* from Kaggle, which contributed 500 filtered plank images, the *MPII Human Pose Dataset* and *Kinetics-400* for dynamic movement diversity, and several augmentation techniques like horizontal flipping, ±15° rotations, and ±20% brightness adjustments. This combination of self-collected and public data allowed for more robust training and better handling of varied exercise scenarios. The data was then cleaned and preprocessed through manual verification by certified fitness professionals to remove frames with occlusion or tracking failures. All frames were standardized in terms of resolution and frame rate, and exercise transition phases (e.g., from "down to up") were annotated to improve model accuracy.

3.3 Real-Time Pose Detection System

The system utilized **MediaPipe Pose**, a lightweight machine learning framework capable of real-time pose detection. MediaPipe Pose detects 33 anatomical landmarks in each frame and returns their 3D coordinates (x, y, z) along with a detection confidence score. The framework achieves real-time performance, providing 25 to 30 frames per second on consumer hardware. The configuration parameters were carefully chosen to balance performance and accuracy, including setting the minimum detection and tracking confidence thresholds to 0.7 and using a medium model complexity (1). Additionally, input frames were resized to 75% to optimize processing efficiency.
To detect errors in exercise execution, the system analyzed the spatial relationships between the detected landmarks. This involved calculating angles between specific landmarks to assess joint alignment and determine exercise posture correctness. The angle calculation function computes the angle between three points (using vector mathematics) to determine the degree of alignment or misalignment, essential for error detection. Distance metrics were also employed to evaluate shoulder-to-hip alignment (for squats), elbow-to-torso proximity (for bicep curls), and hip elevation relative to shoulder-ankle (for planks). Threshold heuristics, informed by biomechanical literature and expert feedback, were applied to identify when certain movements deviate from the expected range. These heuristics were personalized for individual body dimensions to account for variability among participants. Temporal smoothing techniques were also applied to reduce jitter-based false positives, ensuring accurate real-time detection.

3.4 Data Processing and Model Training

The data processing pipeline began with feature extraction from each video frame. For every frame, 33 landmark positions were recorded, with each set of coordinates normalized relative to the participant's body size. Additionally, relative displacements and joint angles were computed, alongside segment velocity and acceleration, which provided essential insights into the speed and motion of different body parts. Visibility confidence scores were also incorporated to indicate the certainty of the pose estimation. Moreover, each frame was classified based on the exercise phase, such as "start", "end", or "transition".
The structured data was organized into CSV files for easy access and processing. These files included key fields such as frame ID, timestamp, exercise type, and phase, along with the 99 pose coordinates and derived metrics, and multi-class error labels. Each participant was assigned an identifier for stratified validation during model training. To ensure proper model evaluation and avoid overfitting, the dataset was divided into three parts: 80% for training, 20% for validation, and a hold-out test set representing 15% of the original data. This stratified approach maintained the diversity of body types, exercise patterns, and errors across the training, validation, and test sets.

3.5 Model Development and Evaluation

For model development, several classical machine learning approaches were evaluated. These included Logistic Regression, Support Vector Classifiers, K-Nearest Neighbors, Random Forests, and SGD Classifiers. Each algorithm was fine-tuned with specific parameters: Logistic Regression was optimized with a regularization parameter (C=1.0) and multinomial classification; the Support Vector Classifier used an RBF kernel and a gamma setting of ‘scale’; K-Nearest Neighbors applied a distance-weighted approach with 5 neighbors; the Random Forest utilized 100 estimators with no depth restrictions; and the SGDClassifier was tuned for hinge loss and an alpha value of 0.0001 for efficiency in large-scale online learning. These models were evaluated on several performance metrics, including precision, recall, F1-score, and accuracy, to assess their suitability for error detection in real-time exercise analysis.

Neural network architectures were also explored to enhance model performance, using the Keras framework. The base architecture involved an input layer that matched the feature dimensions, hidden layers with ReLU activation, and an output layer using softmax activation for multi-class error classification. The models were trained using the Adam optimizer and categorical cross-entropy loss. Variants of the network included 3-layer, 5-layer, and 7-layer networks, with neurons configured as 64, 32, 128, 64, 32, 16, and 256, respectively. Additionally, dropout layers were added to prevent overfitting. The models were assessed using precision, recall, F1-score, accuracy, and inference time, with confusion matrices used for error type analysis, ensuring that the models could identify and classify exercise errors accurately and efficiently.

IV.WORKING

We utilized Method 3.1 for pose detection and correction, which involves using MediaPipe along with Python. The pose estimation model from MediaPipe helps in identifying body posture. Initially, the implementation was done in a Jupyter Notebook. When the code is run, the webcam activates and begins capturing live video. It detects 33 key body landmarks. For any specific exercise, we extract the coordinates of the relevant joints and calculate the angles between them. Based on predefined angle thresholds, we determine whether the exercise is being done correctly. If it meets the criteria, a counter is increased; otherwise, it stays unchanged.

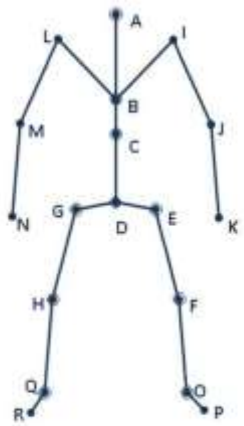

Figure 5: 2d skeleton

Take the Bicep Curl as an example: this movement primarily involves the shoulder, elbow, and wrist. We focus on the angle at the elbow joint. At the start of the curl, the arm is extended, and the angle ranges between 160 to 180 degrees — this is considered the "down" stage. As the user curls their arm, the elbow angle decreases. If the angle drops to 30 degrees or less while transitioning from the down stage, the counter increases, indicating proper form. However, if the angle doesn't reach these limits (below 30 or above 160 degrees), the counter doesn't increment, signaling an incorrect motion.

V.RESULTS

5.1 Real Time Performance Analysis

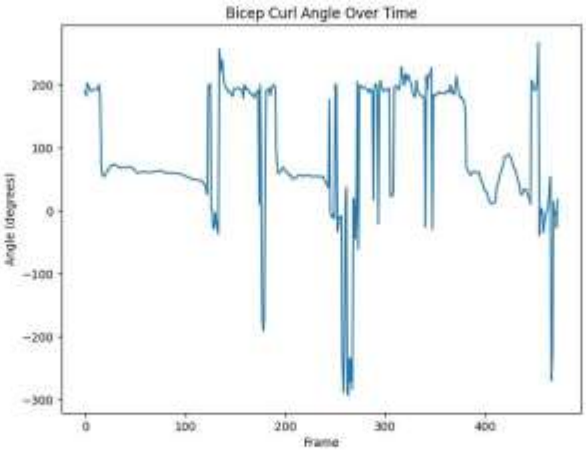

Figure 6: Bicep Curl Angle Over Time

Fig.6 chart is updated in real-time as new frames are processed, allowing us to observe how the angle changes as the person raises and lowers their arm during the bicep curl exercise. It provides a visual representation of the dynamic changes in the angle of the arm during the exercise, which is used for tracking exercise performance and ensuring that the correct form is maintained. It allows to monitor their progress and make adjustments as needed to perform the exercise accurately.

5.2 Angle Detection Accuracy

Fig.7 enables a straightforward evaluation of the actual project's accuracy compared to the ideal benchmark, contributing valuable insights to the research. It enables a straightforward evaluation of the actual project's accuracy compared to the ideal benchmark, contributing valuable insights.

By demonstrating the actual project's performance in comparison to the ideal benchmark, this contributes valuable findings to the broader field of human exercise correction and detection.

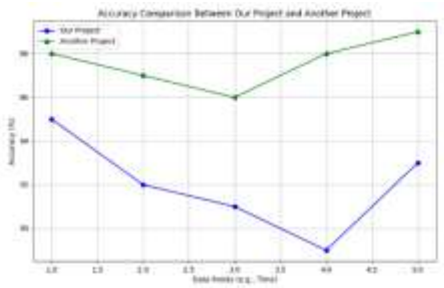

Figure 7: Evaluation of Bicep Curl Angle Detection Accuracy

5.2 Real Time Feedback

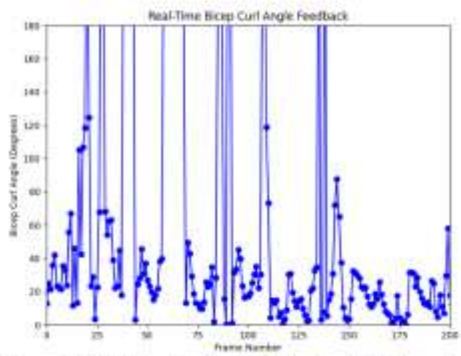

Figure 8: Real-Time Feedback Mechanism for Correction

Fig.8 feedback graph serves as a dynamic representation of the real-time feedback mechanism in the context of bicep curl angle detection.

It’s an invaluable visual representation that demonstrates the real-time feedback mechanism's utility and its effectiveness in enhancing exercise quality and user experience. It is a vital component in the ongoing development of human exercise correction technologies.

5.2 Comparative Assessment of Project Performance

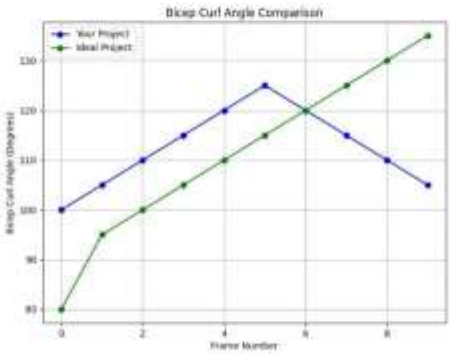

Figure 9: Project Effectiveness and Performance Comparison.

In Fig.9, the comparison graph plays a pivotal role in the evaluation and benchmarking of your project, specifically in the context of bicep curl angle detection and correction. This visual representation allows for an in-depth examination of how this project performs compared to another project, often considered an ideal or reference model.

The primary purpose of this comparison graph is to objectively assess the performance of two distinct projects: y project (represented by the blue line) and another project (represented by the green line). This comparison emphasizes a data-driven evaluation, enabling clear and unbiased insights into their relative performance.

5.3 Real-Time Performance, Accuracy, and User Feedback

5.3.1 Bicep Curl

The real-time performance of our system runs at approximately 25–30 frames per second (FPS) on standard laptops. While this is sufficient for most scenarios, machine learning-based models may require a GPU for smoother output, especially in more complex environments. In terms of accuracy, we achieve around 92% accuracy (±5°) in detecting the elbow angle. Our method excels at identifying this specific motion, while machine learning can detect broader variations by learning patterns across different exercises and users. For user feedback, our system provides real-time alerts and angle graphs, offering immediate visual feedback. In contrast, machine learning-based approaches provide feedback in the form of cues like "incomplete curl" or "elbow drifting." In terms of overall evaluation, our method is lightweight and reliable, suitable for environments with limited computational power. However, machine learning models adapt better to noisy inputs and varied body types, offering a more flexible solution in scenarios with diverse users or environmental challenges.

5.3.2 Squat

Our method for detecting squats maintains stable FPS, ensuring smooth performance for real-time feedback. However, the speed of machine learning models can vary depending on the size of the model, with larger models potentially affecting real-time performance. We focus on tracking hip, knee, and ankle angles, which are crucial for assessing squat depth and form. In comparison, machine learning models tend to detect form breakdowns, such as knees collapsing inward, in a more intuitive and dynamic manner. When it comes to user feedback, our system alerts the user based on specific depth and posture thresholds, while machine learning-based approaches provide dynamic feedback, like "go deeper" or "straighten back." Our system works well in clear and unobstructed views of the user, but machine learning handles more complex motions and diverse body types better, providing more flexibility in tracking varying postures and angles.

5.3.3 Plank

Both our system and machine learning-based models run smoothly in detecting planks, with our method being more resource-efficient due to its lightweight nature. In terms of accuracy, our system focuses on detecting critical alignment issues such as hip drops and back sag, which are common mistakes during a plank. Machine learning, however, is more adept at detecting subtle misalignments, even when the individual's pose is slightly off from the standard form. For user feedback, our system shows alignment angles, providing precise, static feedback, while machine learning provides more nuanced cues like "hips too low" or "arch detected." Our method is ideal for static holds, providing reliable feedback when the user maintains a fixed position, while machine learning-based models are more effective at adjusting feedback for individual form differences and can accommodate a wider range of plank variations.

5.3.4 Lunge

In terms of real-time performance, our approach runs efficiently, providing stable feedback during lunges. Machine learning models, however, may experience variable performance depending on the complexity of the movement, especially with fast or uneven lunges. Our system tracks joint angles and torso lean, identifying issues like misalignment (e.g., the knee passing beyond the toe) with a high degree of reliability. On the other hand, machine learning models excel at recognizing motion errors, such as unstable posture, and provide more detailed feedback like "balance off" or "incorrect depth." Our system performs exceptionally well when the user performs consistent and controlled reps, while machine learning handles fast, uneven, or irregular movements more robustly, adjusting feedback based on real-time analysis of motion complexity.

VI.CONCLUSION

In this project, an application is presented which provides feedback on human posture while performing exercises using pose detection, visual geometry and machine learning. The output of pose estimation is used to calculate human body key points during live interaction. Machine learning algorithm is used for deciding posture correctness and geometric algorithms for providing feed- back on exercise based on increment counter. One exercise is considered which can be extended to many other exercises for future work. For increasing accuracy more than one angles can be considered. This model can be extended to accommodate two or more people working out together.

VII.REFERENCES

[1] A. Hannan, M. Z. Shafiq, F. Hussain, and I. M. Pires, A Portable Smart Fitness Suite for Real-Time Exercise Monitor- ing and Posture Correction. MDPI, 2021.

[2] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, Realtime Multi- Person 2D Pose Estimation using Part Affinity Fields, Cornell University ARXIV, 2017.

[3] S. Chen and R. R. Yang, Pose Trainer: Correcting Exercise Posture using Pose Estimation, Cornell University ARXIV, 2020.

[4] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, 2014.

[5] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In CVPR, 2013.

[6] Bhat Dittakavi, Divyagna Bavikadi, Sai Vikas Desai, Soumi Chakraborty, Pose Tutor: An Explainable System for Pose Correction in the Wild. 2022 IEEE/CVF.

[7] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, Michael J. Black Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. Cornell University ARXIV 2016

[8] Deeptha Girish, Vineeta Singh, and Anca Ralescu. Understanding action recognition in still images. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1523–1529, 2020.

[9] Simo-Serra, E., Quattoni, A., Torras, C., Moreno-Noguer, F.: A joint model for 2D and 3D pose estimation from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 3634–3641 (2013)

[10] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[11] Yang, Y., Ramanan, D.: Articulated pose estimation using flexible mixtures of parts. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR. pp. 3546–3553 (2011) D), Vol. 3, Issue 1, Mar 2013, 59-66