ISSN (Online): 3139-7689

Available online: www.ijsrmes.com

International Journal for Scientific Research in Modern Engineering and Science

Published in: Vol. 1, Issue 3, pp. 15-20, July-August 2026

AI-BASED INTELLIGENT FITNESS TRAINER USING HUMAN POSE ESTIMATION FOR REAL-TIME EXERCISE MONITORING AND POSTURE EVALUATION

Nidhi Chandravanshi[1]*, Samta Gajbhiye[2], Megha Mishra[3]

[1]M. Tech. Scholar, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, C.G., India.
[2]Professor, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, C.G., India.
[3]Associate Professor, Department of Computer Science and Engineering, Shri Shankaracharya Technical Campus, Bhilai, C.G., India.

Abstract

Mobile fitness applications have become very popular because they allow people to work out at home without needing a professional trainer. However, most apps only offer instructional videos and workout timers. They do not monitor user movement in real time. This lack of instant feedback can lead to bad posture, incomplete exercises, and even injuries. This project introduces AI Fitness Trainer, a mobile app built with Flutter and Dart. It uses the smartphone's camera to track exercises and provide real-time feedback on posture, count of repetitions, and form accuracy. The app uses Google ML Kit Pose Detection to recognize 33 body landmarks from live camera feeds. It calculates joint angles using the dot-product formula based on three key landmarks. A threshold-based state machine counts exercise repetitions. A rule-based engine checks posture for each exercise and gives corrective feedback both visually and through text-to-speech.The system supports definitions for 51 exercises across various muscle groups and difficulty levels. Firebase Authentication and Cloud Firestore ensure secure user management, persistent workout storage, daily analytics, streak tracking, and leaderboard features. The design follows a client-cloud model with Provider-based state management to keep the user interface responsive during real-time pose processing. Experimental results show that the prototype successfully combines all major functional parts—pose detection, angle calculation, repetition counting, posture feedback, and workout analytics—into a user-friendly app. This method offers an affordable, sensor-free way to guide home workouts on any modern Android or iOS device.

Keywords: *AI Fitness Trainer, Human Pose Estimation, Computer Vision, Deep Learning, Exercise Recognition, Pose Detection, Flutter, Google ML Kit, Firebase.*

* *Corresponding author*

1. Introduction

Physical fitness is really important for our health and well-being in the run. We have a lot of smartphones now. That is why we have so many fitness apps that let people work out at home by themselves. Every year we see a jump in the number of people using these health and fitness apps on their phones. This shows that people really want to use technology to help them stay healthy. Even, with all this growth most fitness apps are pretty basic. They just play videos that show you how to do exercises have timers that count down and let you enter your workout information by hand. They do not observe the user's body and therefore cannot determine whether an exercise is being performed correctly. Incorrect form is a primary cause of exercise-related musculoskeletal injuries. Common errors such as knees caving inward during squats, excessive forward lean during deadlifts, or elbow flare during bicep curls often go undetected when users exercise alone. Recent advances in on-device computer vision — particularly the availability of lightweight pose estimation models that run in real time on consumer smartphones — make it feasible to build applications that passively observe the user through the camera and provide corrective feedback without requiring any additional sensor hardware. Google ML Kit Pose Detection provides a high-level developer API that detects 33 body landmarks at interactive frame rates on modern Android and iOS devices.

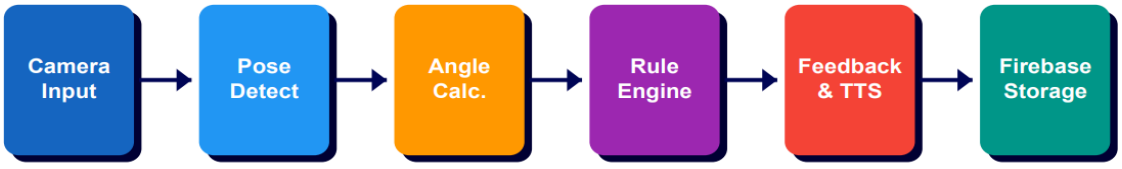

Figure 1. End to end processing pipeline of AI Fitness Trainer

1.3 Objectives

The primary objectives of this project are:

- Design and implement a cross-platform mobile application (Android and iOS) forrealtime exercise monitoring using a smartphone camera.
- Integrate Google ML Kit Pose Detection to extract 33 body landmarks from livecamera frames.
- Develop a geometric angle computation module to calculate relevant joint angles(knee, elbow, hip, shoulder, torso) using the vector dot-product formula.
- Implement a threshold-based state machine for accurate repetition counting across51 exercise definitions.
- Build a rule-based posture feedback engine that provides corrective textual andvoice feedback during workouts.
- Integrate Firebase Authentication and Cloud Firestore for user management,workout session persistence, daily analytics, and leaderboard features.
- Deliver a usable prototype and evaluate its functional completeness.

2. Methodology

AI Fitness Trainer is a cross-platform mobile application that uses the smartphone's built-in camera to monitor exercise sessions in real time. The application detects body landmarks using Google ML Kit, cal-culates joint angles geometrically, counts exercise repetitions using a state machine, evaluates posture us-ing a rule engine, and stores workout data to the cloud for analytics and progress tracking.

The AI Fitness Trainer is based on two ideas:

- Configurability: Every exercise is described in a list that has the main joint angle for counting how many times it is done, the angles that show when someone is going up or down and a list of rules for good posture. New exercises can be added by making a list without changing any of the code that does the work.
- On-device inference: All the work that finds poses and measures angles happens on the phone. The video, from the camera is not sent to a computer else. This keeps peoples information private. Lets the app work even when there is no internet connection.

The table below shows the technology that is used in parts of the application:

Table 1. Technology Stack

Layer	Technology	Purpose
UI Framework	Flutter 3.x / Dart	Cross-platform mobile UI
Pose Detection	Google ML Kit Pose Detection 0.10.0	33-landmark body detection
ML / TFLite	tflite_flutter 0.10.4	Auxiliary model support
State Management	Provider 6.1.2	App-wide state management
Authentication	Firebase Authentication 4.x	Email/password login
Database	Cloud Firestore 4.x	Workout & analytics storage
Storage	Firebase Storage 11.x	Profile images
Charts	fl_chart 0.67	Analytics visualisation
TTS	flutter_tts 4.0	Voice feedback
Animation	flutter_animate 4.5	UI transitions

3. Results and Discussion

During a workout session the app starts the smartphone camera using the Flutter camera plugin. Starts sending frames at the devices normal preview resolution. Each frame comes as a Camera Image object that has separate YUV or NV21 image planes based on the device. A frame-skip method is used to make sure the pose detector does not get called before the last check is done. If there is a check happening when a new frame comes in that frame is not used. This makes sure the detection process does not keep frames waiting and the user interface stays fast during the session.

Joint angles are calculated using the vector dot-product formula on three landmarks that're next to each other in the body. A point that is closer to the center called point A the joint point called point B and a point that is farther away called point C. Vectors BA and BC are made by taking the coordinates of B and subtracting them from the coordinates of A and C. The angle, at joint B is called theta and is calculated using the formula:

$\theta = \arccos((BA \cdot BC) / (|BA| \times |BC|))$

The result is converted from radians to degrees and clamped to [0°, 180°]. The application computes the following angles:

Table 2. Joint Angle Definitions

Angle Name	Point A	Point B (Joint)	Point C
knee_angle	Hip	Knee	Ankle
elbow_angle	Shoulder	Elbow	Wrist
hip_angle	Shoulder	Hip	Knee
shoulder_angle	Elbow	Shoulder	Hip
torso_angle	Shoulder	Hip	Ankle

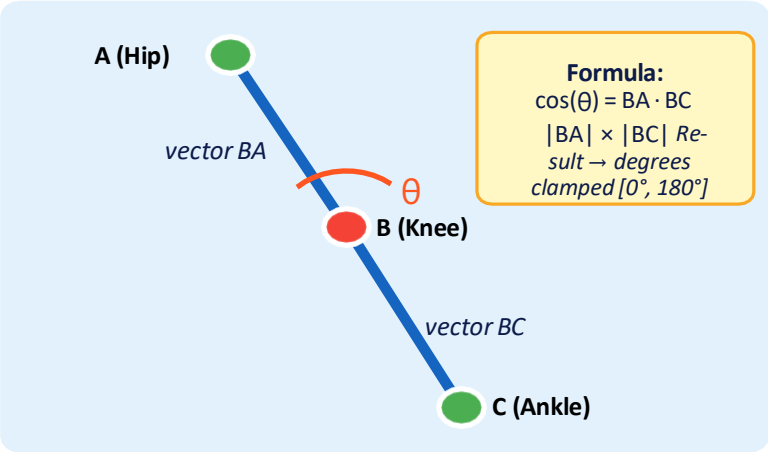

Figure 2. Joint Angle Calculation using Vector Dot-Product (Points A=Hip, B=Knee, C=Ankle)

A rolling average buffer of the last five angle readings is maintained for each angle to suppress high-frequency jitter inherent in landmark coordinates, particularly when the user is in a static hold position.

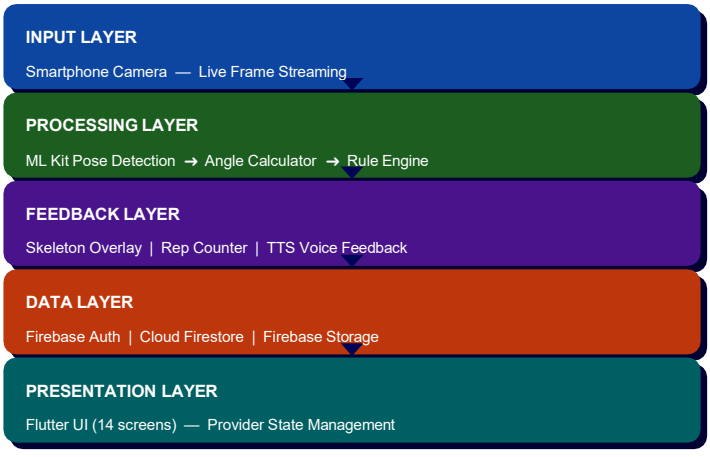

Figure 3. Five-Layer System Architecture of AI Fitness Trainer

On a -range Android device with a Snapdragon 695 and 6 GB RAM the ML Kit Pose Detector runs at about 15 to 20 frames per second in stream mode at 720p preview resolution. This frame rate is enough for skeleton overlay rendering and accurate repetition counting for exercises done at a moderate speed. For exercises like squats, lunges, bicep curls and push-ups the state machine counted repetitions very accurately when the user stayed inside the camera frame and did the movements with full range of motion. Errors happened when the user left the frame during a repetition or did fast movements that made the rolling average fall behind the real angle. The rule engine correctly found form problems like not going low enough on a squat (knee angle not reaching the down-threshold) leaning too far forward (torso angle violation) and not fully extending the bicep curl. TTS announcements were given during the set time period stopping feedback fatigue. Session documents were usually saved to Firestore, between 500 and 800 ms after the workout ended on a 4G connection. Analytics queries for the dashboard took than 200 ms because of the good document setup and Firestores client-side caching.

4. Conclusion

The paper introduces AI Fitness Trainer, a real-time mobile application for fitness monitoring that was developed using Flutter and driven by Google ML Kit's Pose Detection. The system makes use of the smartphone's camera to identify body landmarks, calculates the joint angles by applying the vector dot product formula, determines the number of exercise repetitions with the help of a threshold-based state machine, and assesses posture via a rule-based engine in order to provide visual and verbal corrective feedback.

The prototype has successfully combined all the intended functional features—namely, pose detection, angle calculation, repetition counting, form feedback, workout storage, and analytics—into a unified and usable application. Because of the 51-exercise configurable library, the Firebase cloud backend, and the cross-platform Flutter implementation, the system is more capable and extensible than the single-exercise demonstrations found in previous related work. The results show that it is feasible to develop a low-cost, sensor-free, guided home workout application which is able to monitor exercise form in real time by making use of the camera that is built into any modern smartphone. The rule-based design is transparent and easy to extend, so it is appropriate for further development.

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. Comput. Vis. Pattern Recognit. (CVPR), 2017, pp. 7291–7299.

[2] C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.

[3] Google Developers, "ML Kit Pose Detection," Google for Developers. [Online]. Available: https://developers.google.com/ml-kit/vision/pose-detection. [Accessed: Jul. 2026].

[4] TensorFlow, "MoveNet: Fast and Accurate Pose Detection Model," TensorFlow Hub. [Online]. Available: https://www.tensorflow.org/hub/tutorials/movenet. [Accessed: Jul. 2026].

[5] Flutter, "Build Apps for Any Screen," Flutter Documentation. [Online]. Available: https://docs.flutter.dev/. [Accessed: Jul. 2026].

[6] Google Firebase, "Firebase Authentication and Cloud Firestore," Firebase Documentation. [Online]. Available: https://firebase.google.com/docs. [Accessed: Jul. 2026].

[7] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 483–499.

[8] Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 1653–1660.

[9] V. Bazarevsky et al., "BlazePose: On-device Real-time Body Pose Tracking," arXiv preprint arXiv:2006.10204, 2020.

[10] Y. Liao, J. Vakanski, and M. Xian, "A Deep Learning Framework for Assessing Physical Rehabilitation Exercises," IEEE Trans. Neural Syst. Rehabil. Eng., vol. 28, no. 2, pp. 343–353, 2020.

[11] T. Chen, R. Bhaskara, and S. Narayanan, "Real-Time Squat Form Feedback Using Mobile Pose Estimation," in Proc. IEEE Int. Conf. Healthcare Informatics (ICHI), 2020.

[12] Aditya08q, "AI Fitness Trainer," GitHub Repository. [Online]. Available: https://github.com/Aditya08q/AI_trainer. [Accessed: Jul. 2026].