IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 November 2024, accepted 8 December 2024, date of publication 13 December 2024, date of current version 30 December 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3516723

RESEARCH ARTICLE

Real-Time AI Posture Correction for Powerlifting Exercises Using YOLOv5 and MediaPipe

YEONG-MIN KO[1], **AZIZ NASRIDINOV**[2], **AND SO-HYUN PARK**[3]

[1]Department of Information Convergence Engineering (Artificial Intelligence Major), Pusan National University, Busan 46241, South Korea
[2]Department of Computer Science, Chungbuk National University, Cheongju 28644, South Korea
[3]Department of Computer Engineering, Dongguk University WISE Campus, Gyeongju 13557, South Korea

Corresponding author: So-Hyun Park (sohyunpark@dongguk.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2021R1C1C2004282).

ABSTRACT This study introduces an innovative real-time AI posture correction service for three major powerlifting exercises: bench press, squat, and deadlift, utilizing YOLOv5 and MediaPipe. Due to the rising popularity of online fitness apps post-pandemic, there is a need for accurate posture correction tools to prevent injuries associated with incorrect exercise forms. The proposed method involves a comprehensive MultiPose Exercise Dataset (MPED), which collects 3D joint coordinate data from multiple angles using a smartphone camera. This data is used to train machine learning and deep learning models for detailed posture classification and real-time feedback. The AI service offers specific corrective feedback for both concentric and eccentric contractions, improving exercise efficiency and safety. The study's results show that models built with machine learning algorithms generally outperform deep learning models for posture classification, and the proposed detailed feedback system is effective in preventing injuries and enhancing performance. This comprehensive approach ensures that users receive tailored feedback to correct their form in real-time, significantly reducing the risk of injury and promoting better overall fitness outcomes.

INDEX TERMS AI fitness applications, deep learning, exercise injury prevention, machine learning, MediaPipe, MultiPose exercise dataset (MPED), powerlifting exercises, real-time posture correction.

I. INTRODUCTION

Owing to restriction measures during the COVID-19 pandemic, many people spent more time at home and limited outside activities, making the positive effects of exercise on health more apparent [1], [2]. These changes made people realize the need to maintain health and relieve stress through physical activity, increasing interest in health and fitness. Even after the pandemic, many people continued using fitness facilities to maintain their health.

However, some people were reluctant to hire trainers due to concerns about infection risks or high costs. Online fitness apps emerged during the pandemic as an alternative, allowing users to exercise without relying on a trainer, regardless of time and place. As a result, the market for these apps grew rapidly [3], [4]. Apps like 'Peloton,' 'Perfect Posture & Healthy Back,' and 'Leg Workouts, Exercises for Men' attracted millions of users. These apps provide users with various exercise programs, allowing them to choose suitable workouts and create exercise plans while offering basic exercise posture guidance. Although online fitness apps are convenient and affordable for exercising, they lack appropriate posture correction features. Most of these apps merely demonstrate exercise postures, rarely offering real-time analysis and feedback on users' postures.

Among fitness activities, heavy lifting exercises such as powerlifting are popular. Heavy lifting generally consists of three exercises: bench press, squat, and deadlift, which involve compound multi-joint movements that use multiple joints and muscles simultaneously. In contrast, single-joint exercises are primarily used to focus on strengthening specific muscles. Therefore, compound exercises allow for lifting more weight compared to single-joint exercises, aiding overall muscle growth. However, performing these exercises

The associate editor coordinating the review of this manuscript and approving it for publication was Mouloud Denai.

with an incorrect form can damage muscles, joints, and ligaments, potentially causing chronic pain or functional disorders in severe cases [5]. Especially, compound exercises, characterized by high intensity and complex movements, pose a higher risk of injury if performed with incorrect form.

The bench press is effective for strengthening the chest, shoulders, and triceps, and is considered an upper-body exercise that uses the most muscles [6]. However, performing this exercise with incorrect form can lead to shoulder or wrist injuries [7], [8], [9]. The squat is effective for strengthening the quadriceps, hamstrings, and gluteal muscles, and is considered an excellent exercise for lower-body muscle strengthening [10]. However, performing squats with knees too close or too wide apart or with a curved spine can injure the spine or knees [7], [8], [9]. The deadlift helps strengthen the back, gluteal, and lower-body muscles, showing excellent results in muscle strengthening [11]. Similarly, performing deadlifts with improper form, such as a curved spine or with feet placed too closely together, can strain the back and spine. Thus, the three main exercises in powerlifting, despite their benefits of allowing for heavy lifting and overall muscle growth, pose a risk of muscle and joint injuries [7], [8], [9]. However, many online fitness apps only provide postures and rarely offer real-time analysis and feedback on users' postures. Additionally, research on correcting postures by providing feedback for powerlifting exercises is limited. Existing studies mostly focus on general exercises or single-joint exercises with light weights, and there is little research on classifying and correcting postures for high-intensity exercises such as powerlifting using machine learning and deep learning.

Recently, various studies have been conducted on classifying and evaluating fitness postures using computer vision and sensor technologies. These studies use pose estimation frameworks such as OpenPose or MediaPipe to extract joint landmark data and classify exercise postures based on these data [12], [13]. Furthermore, recent studies have used recurrent neural networks (RNNs) and convolutional neural networks (CNNs) to classify exercise postures [14], [15]. Various studies have also been conducted to classify exercise postures and provide feedback based on posture evaluation [13], [14], [16], [17], [18], [19], [21], [22], [29]. Early video-based studies provided feedback by classifying user postures from recorded videos, making it difficult to offer real-time feedback during actual exercise [16], [18], [19]. Studies [13], [14], [18], [20], [21], [22] have emerged to overcome these limitations, evaluating postures from real-time video data and providing feedback based on this evaluation.

However, existing studies mainly focus on exercises with low injury risk, lacking research on powerlifting exercises like bench press, squat, and deadlift, which have a higher injury risk. To supplement this, [23] investigated injury rates across various weight training activities, including weightlifting, powerlifting, bodybuilding, strongman, Highland games, and CrossFit. Bodybuilding, which primarily involves single-joint exercises, had the lowest injury rate, whereas strongman and powerlifting, which involve weights and compound joint exercises, showed higher injury rates, attributable to factors such as improper exercise techniques, inappropriate weight selection, and excessive increases in training loads. However, beginners in weight training generally have a high risk of injury due to improper posture. Many beginners rely on mirrors in the gym to check their posture during exercises, but this method cannot ensure proper posture and increases the risk of injury. Additionally, exercising under the guidance of a trainer can be costly and creates dependency, leaving potential injury risk when exercising alone [24]. Therefore, maintaining proper posture during weight training is crucial, and a real-time posture classification and feedback system can play an important role.

This study proposes a new method for classifying and evaluating exercise postures that addresses the limitations of existing studies. Its specific contributions are as follows: First, this study proposes a method for classifying and evaluating the three main powerlifting postures using MediaPipe and YOLOv5. As mentioned earlier, research on compound joint exercises with a higher risk of injury is lacking, and more studies are needed in this area. Thus, this study proposes a new posture correction service for the three main exercises in powerlifting.

Second, this study proposes the MultiPose Exercise Dataset (MPED), the first 3D coordinate dataset focused on classifying and evaluating the three main exercise postures. Given that correct posture and feedback differ during concentric and eccentric contractions for each exercise, data were collected separately for these contractions. This study also proposes a method for collecting data. In particular, the proposed MPED collects 3D joint coordinate data from multiple angles using several cameras according to the characteristics of the exercise. This approach considers the changes in angles in various exercise environments, allowing the model to accurately recognize exercise postures in diverse situations.

Finally, this study proposes a method for detailed classification of the three main exercise postures and provides feedback based on this classification. Unlike previous studies that only provide feedback on whether a posture is correct or suggest appropriate joint angle ranges for each exercise, this study offers detailed real-time feedback related to concentric and eccentric movements, enabling effective posture correction. For instance, when users perform bench presses, squats, or deadlifts, detailed feedback on various exercise movements is provided to maximize posture accuracy and prevent injuries. Moreover, this study compares the effectiveness of the proposed feedback method against the traditional comprehensive feedback approach through practical testing and surveys for the two cases of concentric and eccentric movements.

The remainder of this paper is structured as follows. Section II explains related research. Section III introduces the proposed real-time posture correction service for the three main powerlifting exercises. Section IV details the conducted experiments and analyzes their results. Section V concludes this study and discusses future research.

II. RELATED RESEARCH

This section discusses existing studies on exercise posture classification and evaluation and outlines their limitations.

A. RESEARCH ON EXERCISE POSTURE CLASSIFICATION

Various studies have been conducted to classify different exercise postures using computer vision. These studies primarily collected joint landmark data from exercisers using computer vision and then built various models to classify these data. This section reviews the notable studies.

Previous studies often used computer vision frameworks such as OpenPose or MediaPipe to extract joint point data and classify exercise postures based on this data [13], [14], [15], [16], [19], [21], [22]. First, [16], [19] proposed a system that uses the OpenPose framework to detect and classify users' exercise postures during fitness exercises. In particular, these studies recorded videos of the exercises to collect data and extracted 18 major joint landmarks, including the nose, using OpenPose. Then, to obtain vector information for the body from the extracted joint landmarks, they normalized the data based on the length from the neck joint to both hip joints. Finally, they classified the postures using geometric algorithms and machine learning techniques based on the normalized data. The geometric algorithm calculates the range of each joint vector. The machine learning technique used a binary nearest neighbor classifier based on Dynamic Time Warping (DTW) distance to predict 'correct posture' or 'incorrect posture,' achieving an average F1-score of 0.76. However, this approach can only analyze pre-recorded videos and hence cannot provide real-time analysis.

The authors in [21] proposed a system for real-time classification of six yoga-asanas. This study collected yoga motion data from 15 participants, extracting 11,097 joint landmarks using the MediaPipe framework. A classification model for six yoga-asanas was implemented using a combination of CNN and LSTM. They achieved a high accuracy of 99.53% on the test set. In another study [13], MediaPipe was used to implement an intelligent fitness trainer system that supports posture correction. This study created training sets by classifying bicep curl and shoulder press movements into two stages: 'arms up' and 'arms down.' Then, the data were used to implement a posture classification model using the Random Forest algorithm. This algorithm was chosen to train multiple decision trees on different subsets of the data, effectively reducing the bias of individual trees and increasing the overall model accuracy.

One study proposed a method to classify plank and squat postures using CNNs [22]. A dataset of 2,400 plank and holding squat images was constructed. The CNN-based MobileNetV2 architecture was used to perform efficient computations even in resource-limited environments such as smartphones. This model classified plank and holding squat postures into three classes: 'hips too low,' 'correct,' and 'hips too high.' The authors conducted experiments by varying the shooting distance, shooting angle, and lighting conditions to validate model performance in different environments. Thus, the classification performance in each situation was quantitatively evaluated. However, a limitation of this study is its inability to handle dynamic movements as it only classifies and evaluates postures based on hip position.

In [14], an autoencoder model was implemented to classify squat postures into seven types for analysis and correction. This study used Intel's depth camera to collect 1,332 pieces of video data, each 2–5 s long, from over 50 volunteers. Each video consisted of seven types of squat postures like bending forward, heels lifting, knees caving, no depth, toes lifting, Olympic squat, and powerlifting squat. To ensure posture classification regardless of the shooting direction, the study calibrated two depth cameras placed on both sides. This enabled consistent posture analysis from various shooting angles. The study implemented Bi-RNN, Bi-LSTM, Bi-GRU, Bi-RNN with Attention, Bi-LSTM with Attention, and Bi-GRU with Attention for processing sequential data and compared their accuracies. Among them, the Bi-GRU model with Attention exhibited the highest accuracy at 94%. This study proposed a method to classify squat postures into seven specific types. However, it was limited to only squat classifications. Despite this limitation, the proposed model achieved high accuracy in squat posture analysis.

The method proposed in [15] recognized yoga-asanas and provided feedback. They used OpenPose to extract joint landmarks from users and collect data and proposed a new hybrid approach combining machine learning and deep learning. In the first stage, they used support vector machines (SVM), showing a high training accuracy of 0.99 and test accuracy of 0.93. In the second stage, they used CNNs to capture the human skeletal pose and compare it to the user's target pose for similarity. Consequently, both training and test accuracy reached above 0.98. However, despite the promising results, [15] has limitations in its approach. Firstly, the reliance on a two-stage process combining SVM and CNNs might introduce unnecessary complexity, potentially increasing computation time and resource usage. This makes it less suitable for real-time applications where quick response times are essential. Secondly, while the study achieved high accuracy, the use of OpenPose limits the number of detectable joint landmarks to 18, potentially restricting the system's ability to capture finer details of complex postures. In contrast, our study leverages a more comprehensive set of 33 joint landmarks and 11 major angles, allowing for more detailed posture classification.

References [12], [17], and [18] implemented machine learning and deep learning classification models by extracting joint landmark data and calculating joint angles using these data. Reference [17] used the Yoga-82 dataset with 29,000 images, the Pilates-32 dataset with 2,473 images, and the Kungfu-7 dataset with 179 images to build a posture classification model for 82 yoga-asanas, 32 Pilates postures, and 7 Kungfu postures. This study used TransPose and HRNet to collect 18 joint landmarks and 10 joint angles. Then, they built a posture classification model using the k-nearest neighbor (KNN) and DenseNet. Use of the KNN alone yielded performances of 0.61, 0.77, and 0.81 for Yoga-82, Pilates-32, and Kungfu-7, respectively, and using DenseNet alone showed performances of 0.81, 0.79, and 0.62, respectively, for the same datasets. Combining DenseNet and KNN showed relatively higher performances of 0.79, 0.82, and 0.81, respectively, on these datasets. This study also compared the model performance with various data combinations, including joint landmarks, joint angles, segments connecting joint landmarks and vectors, and areas of joint landmarks and vectors. The combination of joint landmarks and joint angles showed the highest performance. However, the study focuses on low-intensity exercises like yoga, Pilates, and Kungfu, which involve static postures with less risk of injury. In contrast, our study focuses on high-intensity, high-risk powerlifting exercises like the bench press, squat, and deadlift, which require real-time posture correction due to their compound multi-joint movements and heavy weight involvement. The risk of injury is much higher in these exercises, making real-time feedback critical. In addition, the limited number of joint landmarks and angles used in [17] might have restricted the model's ability to capture complex posture variations, especially compared to our approach, which utilized 33 joint landmarks and 11 joint angles, allowing for more detailed posture classification.

The method proposed in [18] automatically detects incorrect postures for eight bodyweight exercises. Using MediaPipe, they collected 13 major joint landmark data, from which they extracted the joint angle data. Furthermore, they analyzed the extracted data to derive the extreme values of internal and external angles of vector pairs obtained from each exercise. Further, these values were utilized to extract rule parameters using the K-Means algorithm to quantitatively evaluate exercise postures. These rule parameters were used to specify the correct posture for each exercise. Based on the extracted rule parameters, they observed posture changes over 10 consecutive frames, classifying a posture as incorrect if the vector pair angles of those frames were not within acceptable thresholds.

In [12], the performance of various machine learning techniques and deep neural network (DNN) was compared in terms of classifying six yoga-asanas. This study used data from six broad categories among the 82 yoga-asanas included in the Yoga-82 dataset. After collecting joint landmark data using the pose estimation feature of MLKit, the authors calculated the angle data for key joints based on these data. Then, they trained and tested models using various machine learning algorithms and deep neural networks, including Logistic Regression, Extra Trees, Random Forest, Gradient Boosting, and DNN. All algorithms achieved an accuracy higher than 0.84, with the highest accuracy of 0.91 for the extra trees algorithm. Additionally, they implemented deep learning classification models using ResNet, DenseNet, and MobileNet, but these models showed lower accuracies, ranging between 0.6 and 0.7, compared with the machine learning algorithms.

B. RESEARCH ON EXERCISE POSTURE EVALUATION

Various studies have been conducted to classify exercise postures and provide feedback based on posture evaluation. Some studies [16], [19], [22] classify user postures from recorded videos, evaluate the exercise movements based on these classifications, and provide feedback through text or voice. In particular, [16], [19] proposed a method for setting joint angle ranges for exercises such as single-arm bicep curls and front raises, providing appropriate feedback when these thresholds are exceeded. For example, if the shoulder rotates during a bicep curl, the system outputs a message similar to "There is significant rotation around the shoulder during the curl. Keep the upper arm parallel to the chest and focus on rotating only around the elbow." However, these studies have limitations in providing real-time feedback because they analyze pre-recorded videos. Additionally, [22] proposed feedback based on the hip position for planks and squats. For example, if the hips are too low, the system provides voice feedback like, "Your hips are too low. Lift them up." However, this approach cannot handle dynamic exercises and is less robust in different environments.

To overcome these limitations, some studies [13], [14], [18], [20], [21] used real-time video data to classify postures, evaluate exercise movements, and provide real-time feedback to users. However, most of these studies that evaluate postures in real-time are limited to desktop environments. Reference [18] built a system that classifies postures and monitors joint ranges in real time for exercises such as overhead squats, single-leg squats, push-ups, standing overhead dumbbell presses, standing rows, upper-body abductions, upper-body rotations, and lunges, providing feedback on the screen to help users maintain correct exercise postures. Reference [20] used RNNs to classify side lateral raises and bicep dumbbell curls, providing feedback based on joint angles. However, because these studies only provided feedback on which joint angle ranges should be included, the resultant feedback was less specific, limiting their user-friendliness. Furthermore, [21] classified six yoga-asanas using SVM and CNN models, calculated the cosine similarity between the joint landmarks of the correct posture and the user's posture keypoints, and provided feedback based on a threshold. For example, if the posture is 91% similar to the correct one, feedback such as "Feedback: Keep your arms parallel to the ground" is provided as text and voice on the web screen. Reference [13] proposed a method to classify bicep dumbbell curls and dumbbell shoulder presses as either correct or incorrect postures using a Random Forest model to provide voice feedback. Reference [14] proposed a method to evaluate squat postures in detail by classifying them into seven categories. This study is significant as it provides more detailed feedback to users by evaluating a single exercise in detail. According to [23], bench press, squat, and deadlift are

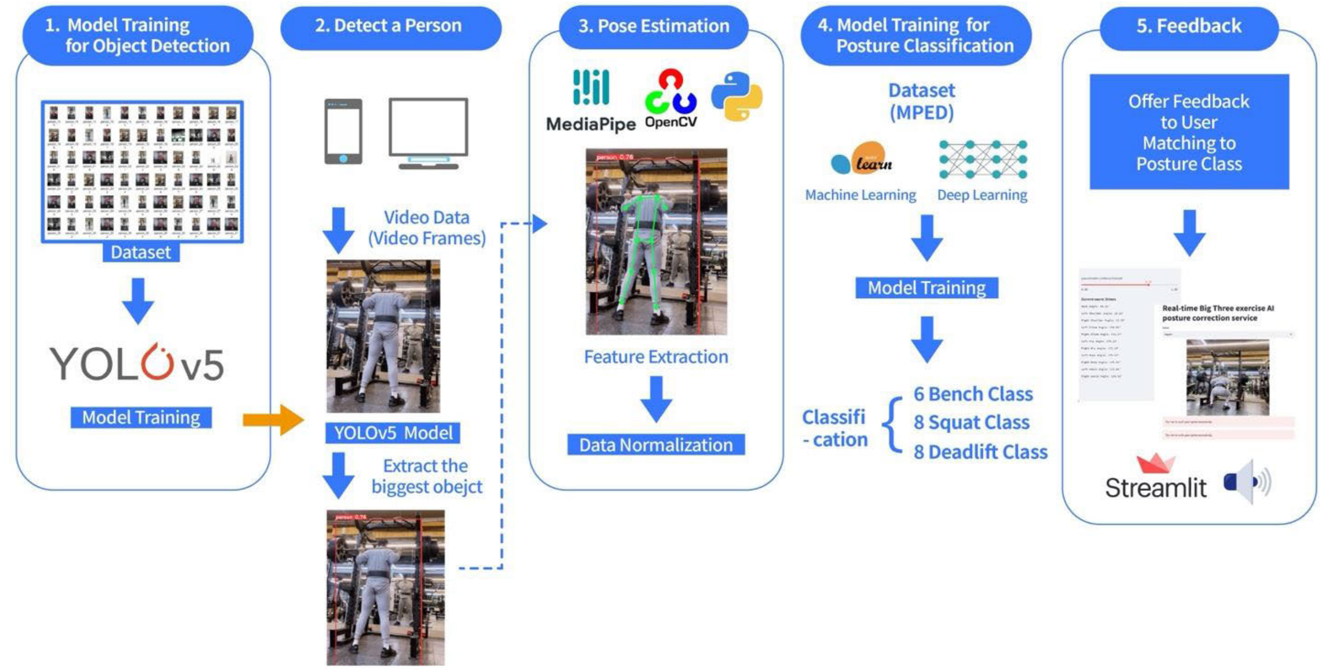


FIGURE 1. Overall process of the proposed method. The system consists of five main steps: (1) Training a YOLOv5 model for object detection using an exercise dataset, (2) detecting a person in video frames, (3) extracting and normalizing pose data using MediaPipe and OpenCV for pose estimation, (4) training machine learning and deep learning models for posture classification using the MPED dataset, and (5) providing real-time feedback to the user through a web interface to correct exercise posture.

exercises most related to injuries during workouts. Specifically, both powerlifters and bodybuilders identified squats and bench presses as major causes of injuries. However, most previous studies have focused on single-joint exercises with a low probability of injury, such as dumbbell curls, dumbbell overhead presses, and side lateral raises. In contrast, the current study is the first to focus on the three main powerlifting exercises: bench press, squat, and deadlift, and propose an a new posture correction service to improve users' exercise postures.

III. PROPOSED REAL-TIME AI POSTURE CORRECTION SERVICE FOR POWERLIFTING EXERCISES

A. SYSTEM ARCHITECTURE

The overall process of the proposed method is shown in Fig. 1. First, images and various posture datasets of people exercising were collected from Roboflow, Google Images, and Kaggle. Then, the collected data was labeled with the person object, and the YOLOv5s model was trained to derive the optimal weights (Fig. 1.1. Model Training for Object Detection). Next, the largest detected object among those identified as performing actual exercises was extracted from video frames input from webcams in PC or mobile environments using the YOLOv5-trained object detection model. This step is crucial for identifying the person exercising and designating them as the main subject for movement analysis (Fig. 1.2. Person detection). Then, MediaPipe and OpenCV were used to extract joint landmarks of the exerciser from the video data. The extracted joint landmarks were then used to calculate the angles between key joints, and these angle data of key joints was normalized using Min-Max scaler (Fig. 1.3. Pose Estimation). Next, to classify the posture of each exercise movement, the models were trained using the proposed MPED dataset with machine learning and deep learning techniques. In particular, six classes were trained for the bench press, and eight classes each were trained for squats and deadlifts (Fig. 1.4. Model Training for Posture Classification). Finally, a server was built using Streamlit to operate in a web environment. In the web environment, Steps 2 and 3 were repeated to collect joint landmark postures and key joint angle data of the user, classifying the posture based on the most frequently appearing class during the transition from eccentric to concentric contraction. Finally, based on the classified class, text and voice feedback are provided, and the exercise screen, exercise count, and key joint angle information are displayed to the user on the web interface. This helps users improve their movements and maintain correct postures (Fig. 1.5. Feedback).

B. MODEL TRAINING FOR OBJECT DETECTION

This study used datasets provided by Roboflow to build a model for detecting people performing exercises [26], [27], [28], [29]. Additional data was collected by crawling Google Images to supplement insufficient data. Additionally, various posture image datasets provided by Kaggle were compiled and used [30].

Because this study is focused on exercise-related postures, image datasets of people performing bench press, squats, and deadlift were collected. To prevent overfitting due to training solely on exercise images, a total of 805 images of general people, construction workers, and people lying down were collected from Roboflow and Google Images. Subsequently, 4,800 additional images (1,200 each of leaning forward, lying down, sitting, and standing postures) were collected from 'Silhouettes of human posture' dataset provided by Kaggle, making a total of 5,605 images.

The collected data was annotated with the person object, and the YOLOv5s (You Only Look Once version 5) model was trained to obtain the optimal weights. YOLOv5 is a deep learning-based object detection algorithm that plays a crucial role in computer vision and object detection [31], [32]. It offers a balance of high speed and accuracy, making it widely used for various applications. Although more recent versions such as YOLOv7 and YOLOv8 have been developed, YOLOv5 was chosen for this study due to its efficiency in real-time processing. According to recent studies [33], [34], YOLOv5 has fewer model parameters than its newer counterparts like YOLOv8, leading to faster inference times while maintaining similar accuracy levels. This makes YOLOv5 particularly advantageous for real-time applications, such as exercise posture estimation, where prompt feedback is essential. Furthermore, YOLOv5 utilizes the DarkNet backbone, allowing very fast computation speeds of up to 140 frames per second.

Among the different versions of YOLOv5, including YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, this study chose the lightweight YOLOv5s model for its suitability in real-time video processing. Although model accuracy increases from 's' to 'x,' processing speed decreases with more complex models. Thus, YOLOv5s offers an optimal balance between accuracy and speed, ensuring efficient real-time posture estimation. For model training in this study, the batch size was set to 16, and the number of epochs was set to 200. To prevent overfitting, early stopping was applied if the patience exceeded 100 during the training process, and the best weights were saved.

C. DETECTING A PERSON USING YOLOv5

This study uses the MediaPipe framework to extract the joint landmarks of the exerciser. However, MediaPipe has the limitation of extracting joint landmarks for only a single object. In real exercise environments, multiple people may be recorded simultaneously. To overcome this, the study proposes performing object detection first to detect the exercising person and then extracting joint landmarks for that object. This approach allows accurate identification of the exercising person, even in videos with multiple people.

The algorithm for detecting exercising people involves training the YOLOv5 model on relevant datasets. The trained weights are then used to detect people performing the three main exercises in real-time. The object with the largest bounding box is identified as the person exercising, and posture estimation is performed for that object.

D. POSE ESTIMATION

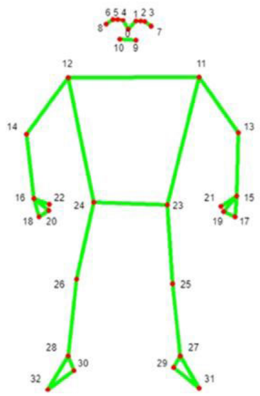
1) MEDIAPIPE

MediaPipe, an open-source framework developed by Google, is used for developing computer vision applications based on machine learning, and is specifically designed to operate on real-time data streams. MediaPipe supports various vision tasks including object detection, pose estimation, face detection, and hand tracking [35]. This study utilizes its pose estimation feature for posture estimation. The reasons for choosing MediaPipe over other frameworks such as OpenPose and PoseNet are as follows: First, MediaPipe is designed to operate on real-time data streams, making it suitable for this study. This is essential for analyzing the posture of people exercising in real-time and providing feedback. In contrast, OpenPose and PoseNet are relatively slower and more suitable for processing uploaded videos. Second, MediaPipe is implemented based on the latest research findings, yielding accurate results even under complex postures and various lighting conditions. Lastly, MediaPipe supports multiple platforms, making it usable in various environments such as mobile devices.

The goal of this study is to use MediaPipe to identify and estimate the joint landmarks of people from video data in real time, accurately analyze the postures of people exercising, and provide precise feedback to enhance exercise efficiency.

2) FEATURE EXTRACTION

This study extracts two main features for exercise posture analysis: joint coordinates and joint angles. Joint coordinates are 3D vectors (x, y, z) representing the position information of each joint landmark. This study uses the Pose model of MediaPipe to extract 33 key joint coordinates from the entire body of an exercising person from webcam footage, constructing the feature vector. However, this feature vector only contains the position information ($l = (x, y, z)$) of each joint landmark. In addition, this study calculates joint angles from the joint landmarks and uses this information to perform exercise posture analysis. Studies have calculated joint angles from joint landmarks, set threshold values for specific joint angles, and compared the angles of key joints to determine if the posture is correct [16], [17]. However, using only joint angles to determine exercise posture often results in low accuracy. Recent studies combining pose estimation technologies with machine learning models to provide feedback on yoga-asanas have shown that combining joint landmarks and joint angles in the machine learning models yields the highest accuracy [17]. Accordingly, this study builds a machine learning model using not only the feature vector composed of joint coordinates but also the 11 key joint angles inferred from the joint landmarks and the 33 joint landmarks extracted by MediaPipe.



0. nose
1. left eye inner
2. left eye
3. left eye outer
4. right eye inner
5. right eye
6. right eye outer
7. left ear
8. right ear
9. mouth left
10. mouth right
11. left shoulder
12. right shoulder
13. left elbow
14. right elbow
15. left wrist
16. right wrist
17. left pinky
18. right pinky
19. left index
20. right index
21. left thumb
22. right thumb
23. left hip
24. right hip
25. left knee
26. right knee
27. left ankle
28. right ankle
29. left heel
30. right heel
31. left foot index
32. right foot index

FIGURE 2. List of the joint landmarks [36].

By using trigonometric functions, joint angles can be calculated from the position information of joint landmarks. Fig. 2 presents the 33 joint landmark numbers that are used for calculating joint angles [36]. Each number corresponds to a specific joint or body part, such as the nose, eyes, shoulders, elbows, wrists, hips, knees, ankles, and several additional points related to hand and foot joints, which is utilized in the angle calculation algorithms. For example, landmarks 11 and 12 represent the left and right shoulders, respectively, while landmarks 13 and 14 correspond to the elbows. To facilitate the accurate computation of joint angles as described in Table 1, we use the precise joint landmark numbers.

The calculation of joint angles is integral to posture classification and evaluation. Table 1 shows the joint landmarks used to calculate the angles between key joints. Here, the angles provide insight into the alignment and movement of the neck, shoulder, elbow, hip, knee, and ankle joints. The angles for the shoulder, elbow, hip, knee, and ankle joints are calculated separately for the left and right joints. For example, the left shoulder angle is derived from landmarks 13 (left elbow), 11 (left shoulder), and 23 (left hip), while the right shoulder angle is derived from landmarks 14 (right elbow), 12 (right shoulder), and 24 (right hip). These angles reflect the upper body's relative alignment. Similarly, the elbow angle is calculated using three points: the shoulder, elbow, and wrist on each side. The left elbow angle uses landmarks 11 (left shoulder), 13 (left elbow), and 15 (left wrist), while the right elbow angle uses landmarks 12, 14, and 16. This method enables the analysis of arm bending and extension.

Each joint in the body is represented by a coordinate vector $p_i = (x_i, y_i, z_i)$, where i denotes the joint number (e.g., shoulders, elbows, knees). Fig. 3 shows the positions and formula used to calculate the joint angles during a bench press exercise. To calculate the joint angles that determines the correctness of exercise form, the proposed method utilizes three specific joint positions A(x_1, y_1, z_1), B(x_2, y_2, z_2), C(x_3, y_3, z_3). The angle θ_{deg} between these points is calculate using Equation (1).

$$\theta_{deg} = \left| cos^{-1}\left(\frac{\overrightarrow{BA}\cdot\overrightarrow{BC}}{\left|\overrightarrow{BA}\right|\left|\overrightarrow{BC}\right|}\right) \times \frac{180.0^{\circ}}{\pi} \right|, \tag{1}$$

where $\overrightarrow{BA}$ and $\overrightarrow{BC}$ are vectors calculated from the joint coordinates, and θ_{deg} is measured in degrees. This calculation allows the system to evaluate joint movement and detect any deviations that may indicate incorrect posture.

FIGURE 3. Joint Angle Calculation: Illustration of the method used to calculate joint angles between landmarks A, B, and C for evaluating posture correctness during exercises.

The calculating the joint angles is illustrated in Algorithm 1. To compute the angle at a joint formed by three landmarks, such as the left shoulder, left elbow, and left wrist, the 3D coordinates of these points are assigned to the variables A, B, and C, respectively. The vectors $\overrightarrow{BA}$ (from B to A) and $\overrightarrow{BC}$ (from B to C) are computed. The cosine of the angle between these vectors is calculated using the dot product, normalized by the product of the magnitudes (norms) of the two vectors. The inverse cosine function is then applied to determine the angle in radians, and the result is converted to degrees by multiplying by $\frac{180.0^{\circ}}{\pi}$. If the calculated angle exceeds 180.0°, it is adjusted by subtracting the angle from 360.0°, ensuring the returned angle is in the range of [0°, 180°].

3) DATA NORMALIZATION

The coordinates of each joint landmark extracted from the video using MediaPipe are represented as 3D vectors $p_i = (x_i, y_i, z_i)$, with each coordinate also having a visibility value. The visibility value indicates how well each joint is visible in the video. Additionally, the Pose Estimation module of MediaPipe provides all joint position vectors as normalized values in the range of $[-1, 1]$, and the visibility values are provided in the range of $[0, 1]$. In this study, the collected videos were loaded using OpenCV, and joint coordinates were manually labeled for each posture. The joint coordinates captured at specific moments may have missing values (i.e., $p_i = NULL$) due to video quality or movement execution. In such cases, joint landmarks could not be extracted.

TABLE 1. The joint landmarks used to calculate the angles between key joints.

Angle	Position	Joint landmark numbers	Explanation
Neck	Left	11, 0, 23 (left shoulder, nose, left hip)	The neck angle is computed as the arithmetic mean of two angles: one formed by the landmarks of the left shoulder, nose, and left hip, and the other by the right shoulder, nose, and right hip. This method ensures symmetric evaluation of the neck joint.
	Right	12, 0, 24 (right shoulder, nose, right hip)	
Shoulder	Left	13, 11, 23 (left elbow, left shoulder, left hip)	The shoulder angle is calculated separately for the left and right sides using three key landmarks: the elbow, shoulder, and hip. These angles represent the relative alignment of the upper body.
	Right	14, 12, 24 (right elbow, right shoulder, right hip)	
Elbow	Left	11, 13, 15 (left shoulder, left elbow, left wrist)	The elbow angle is computed by using three points: the shoulder, elbow, and wrist on each side. This reflects the bending of the arm.
	Right	12, 14, 16 (right shoulder, right elbow, right wrist)	
Hip	Left	11, 23, 25 (left shoulder, left hip, left knee)	The hip angle is measured by the relative positions of the shoulder, hip, and knee, providing insight into lower body alignment.
	Right	12, 24, 26 (right shoulder, right hip, right knee)	
Knee	Left	23, 25, 27 (left hip, left knee, left ankle)	The knee angle is calculated using the landmarks of the hip, knee, and ankle to assess knee joint movement.
	Right	24, 26, 28 (right hip, right knee, right ankle)	
Ankle	Left	25, 27, 29 (left knee, left ankle, left heel)	The ankle angle is computed by considering the knee, ankle, and heel, which captures foot positioning and lower leg alignment.
	Right	26, 28, 30 (right knee, right ankle, right heel)	

Algorithm 1 Calculating joint angle between three joint points

Input:
Coordinates A(x_1, y_1, z_1), B(x_2, y_2, z_2), C(x_3, y_3, z_3)
Output:
Angle between three joint points at B (θ_{deg})

$\overrightarrow{BA} = \overrightarrow{A}$ - $\overrightarrow{B}$ = (x_1 - x_2, y_1 - y_2, z_1 - z_2)
$\overrightarrow{BC} = \overrightarrow{C}$ - $\overrightarrow{B}$ = (x_3 - x_2, y_3 - y_2, z_3 - z_2)
$\left\|\overrightarrow{BA}\right\| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$
$\left\|\overrightarrow{BC}\right\| = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2 + (z_3 - z_2)^2}$
$cos\theta = \frac{\overrightarrow{BA} \cdot \overrightarrow{BC}}{\left\|\overrightarrow{BA}\right\| \left\|\overrightarrow{BC}\right\|}$
$\theta_{rad} = cos^{-1}(\cos(\theta))$
$\theta_{deg} = \left|\theta_{rad} \times \frac{180.0^\circ}{\pi}\right|$
if $\theta_{deg} > 180.0^\circ$ **then**
  $\theta_{deg} = 360.0^\circ - \theta_{deg}$
end if
return θ_{deg}

To address this issue, only data points with a visibility value of $v_i \geq 0.6$ are considered:

$$\{p_i \mid v_i \geq 0.6\}, \tag{2}$$

and data with visibility values $v_i \leq 0.6$ were discarded. Furthermore, videos that contained severe noise or were blurry were excluded from the analysis to ensure the quality of the data used for model training.

Next, joint angles were calculated using trigonometric functions. Considering that the angle ranges can vary significantly, especially for exercise posture data, adjusting the angle data to a consistent range is necessary to ensure that the model can learn uniformly. Thus, this study used Min-Max scaling to normalize the extracted angle data to the range of [0, 1]. This ensures that the joint angle data are on the same scale, allowing the model to learn without being affected by the magnitude of the angles. The min-max normalization formula is given by Equation (3):

$$\theta' = \frac{\theta - \theta_{min}}{\theta_{max} - \theta_{min}}, \tag{3}$$

where θ represents the original joint angle, θ_{min} and θ_{max} are the minimum and maximum values observed in the dataset for that joint angle and θ' is the normalized value in the range of [0, 1].

E. MODEL TRAINING FOR POSTURE CLASSIFICATION

1) DATA COLLECTION

Previous research, as summarized in Table 2, employed different methods to collect data for various exercises [14], [22], [37], [38]. For example, studies [37], [38] used wearable devices like the surface electromyography (sEMG) and gyroscopes to collect detailed positional data. However, these sensors can be expensive and may limit natural movement due to equipment attached on the body causing certain level of discomfort. On the other hand, [22] utilized static images to classify postures in exercises such as plank and squat, categorizing them as 'low hips,' 'correct,' or 'high hips.' While this approach enabled data collection using a smartphone,

TABLE 2. Comparison of data collection methods used in previous studies.

Study	Number of Classes	Total Data Samples	Method of Data Collection	Type of Exercise	Eccentric/ Concentric Classification
Hsiao et al. [37]	12	Not disclosed	Wearable devices	Bench press	X
Zhou et al. [38]	9	1,700	Wearable devices	Dumbbell curl Bent-over row Bench press Lying triceps extension Front raise Lateral raise Shoulder press Overhead triceps extension Cable crossover	X
Militaru et al. [22]	6	2,400	Static images	Plank Squat	X
Chariar et al. [14]	7	1,332	Joint information collected through MediaPipe from exercise videos	Squat	X
Ours (MPED)	22	3,023	Joint information collected through MediaPipe from exercise videos	Bench press Squat Deadlift	O

it could not process dynamic movements, which are essential for multi-phase exercises. In contrast, [14] used MediaPipe to collect joint data from exercise videos, eliminating the need for wearable sensors. However, this method did not differentiate between eccentric (lengthening) and concentric (shortening) phases of muscle contractions, which are crucial for understanding the biomechanics of exercises like squats and deadlifts [39]. Differentiating between these phases is important because they involve different movement patterns and levels of stress on the body. Without this distinction, critical variations in posture across different phases may be overlooked.

Inspired by the study conducted by [14], which used MediaPipe to collect joint data from exercise videos, we chose MediaPipe as our data collection tool. One of the key advantages of MediaPipe is that it is free software and does not require any equipment attachment, allowing exercisers to move naturally during training. This flexibility makes it especially suitable for exercises like the bench press, squat, and deadlift, where natural movement is critical for accurate posture evaluation. By leveraging MediaPipe, our study introduces a more detailed posture collection method that considers the distinct characteristics of both eccentric and concentric contractions. This dynamic approach not only improves posture classification but also enables the detection of fatigue or form deterioration over multiple repetitions, providing valuable insights into the real-world dynamics of weight training. Furthermore, this method enhances the of repetition counting. By recognizing the start and end of each contraction (eccentric and concentric), the method can reliably track the number of completed repetitions, minimizing errors caused by incomplete or improperly performed movements.

Another important factor is the size of the dataset used in each study. A larger dataset generally improves the accuracy of machine learning and deep learning models. As shown in Table 2, our study collected 3,023 data samples, making it the most extensive dataset among the compared studies. Additionally, our study includes a greater number of posture classes than the other studies. This enhances the robustness of our posture classification model, ensuring it can handle a wider range of exercisers and movement variations.

Studies have considered the issue of reduced joint landmark recognition during data collection due to environmental factors [14], [22], [40], [41]. These studies considered the issues like brightness, shooting distance, and shooting angle, which can affect joint landmark recognition rates. They collected data from various angles and distances. Specifically, [14] increased joint landmark recognition rates and improved the accuracy of exercise posture classification by filming from multiple directions for postures such as squats, where joint landmarks could be obscured from certain positions.

Referring to previous studies, this paper ensured that filming was done such that joint landmarks were not obscured by environmental factors like camera position and direction or surrounding brightness. If the camera is too close or too far, the size of the joint landmarks becomes too small to be accurately inferred [22], [41]. Therefore, to estimate joint landmarks with high accuracy, filming was done at 200 cm from the front and 180 cm from the left and right diagonal directions of the subject performing the exercise for all exercise types. Additionally, cameras were positioned not only considering distances but also considering angles. Thus, we positioned cameras at 45° angles to the left and right diagonals to ensure accurate and balanced joint estimation during exercise movements [14], [22], [40].

Fig. 4 illustrates the camera positions used for collecting data during bench press, squat, and deadlift exercises, showing both top and side views for clarity. Although the back of the subject is not captured, the multi-view camera setup compensates for this by providing enough visual coverage from the different angles to analyze the exercise properly. This ensures that joints occluded in one view can be captured in another, making multi-view camera placement a rational and effective approach for collecting exercise data, as shown in Fig. 4.

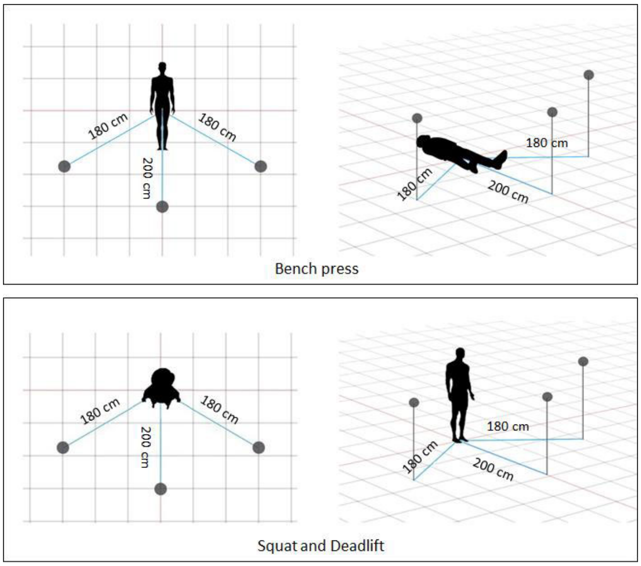


FIGURE 4. Camera Setup for Data Collection: Camera positions used for capturing exercise movements during bench press, squat, and deadlift. Cameras are placed 200 cm in front and 180 cm at diagonal angles to ensure comprehensive joint landmark detection.

To further enhance pose estimation accuracy, this study theoretically considers multi-view geometry techniques, which provide a basis for the potential 3D reconstruction of joint landmarks. Multi-view geometry offers a framework for combining multiple 2D projections from different viewpoints, allowing for a consistent 3D representation in theory. By using epipolar geometry, correspondences between points captured by different cameras can be established, enabling accurate localization of joints that may be occluded in one view but visible in another [42]. Utilizing these principles can perform the pose estimation of high performance of a multi-view setup that minimizes occlusions and maximizes visibility.

For the bench press, since the exercise is performed in a lying-down position, capturing the movement only from the front can result in occlusions, particularly where the elbows or shoulders may be blocked by the legs. By placing cameras 200 cm in front of the subject and 180 cm at 45° angles to the left and right, the symmetrical nature of the exercise can be more accurately captured and analyzed, allowing for visibility of joints that might be obscured from a single viewpoint. For squat and deadlift exercises, where the focus is on the lower body and back, multiple camera angles are equally important to capture critical joints in various body regions. The same camera placement strategy—200 cm in front and 180 cm at 45° diagonals—was used to capture the full range of movement and prevent joint occlusion. This multi-view setup enables us to observe all key landmarks, especially in the legs and torso, from different perspectives, ensuring that no joint is hidden due to a single camera angle. By integrating data from multiple viewpoints, this approach effectively entangles joint information across views, creating a comprehensive representation of the subject's posture and movement. Fig. 5 shows the shooting stand position for collecting the bench press dataset. This multi-view strategy minimizes the risk of missing critical joint landmarks due to occlusions and ensures a more reliable and detailed posture analysis across all exercises.

FIGURE 5. Shooting Stand Position for Bench Press Data Collection: Example of the shooting stand setup positioned at left diagonal, front, and right diagonal locations, 200 cm and 180 cm away, respectively, for capturing bench press exercise movements.

In this study, we propose MPED, a multi-angle dataset for the three main exercises (bench press, squat, deadlift) proposed. MPED data was collected by one individual with 5 years of weight training experience. The data collection took place in a bright and spacious gym. Data was collected using an iPhone 12 Pro camera, and OpenCV was used to manually label the posture data by reviewing the recorded videos according to the classification criteria for each posture. MPED can be used for the development of accurate posture inference and correction models or for injury prevention research by analyzing the relationship between incorrect postures and injury risks. MPED includes posture data for concentric and eccentric contractions of the deadlift. Table 3 summarizes the features of the proposed MPED.

Exercise postures can be classified into eccentric and concentric contractions based on muscle contraction methods. Eccentric contraction occurs when the muscle lengthens while exerting force, whereas concentric contraction occurs when the muscle shortens while exerting force. In exercises like the bench press, squats, and deadlifts, these contractions alternate during the lifting and lowering phases. Correct posture varies according to the contraction type, making it crucial to distinguish between eccentric and concentric phases when collecting exercise posture data.

TABLE 3. MPED dataset summary.

Number of landmarks	Bench Press	955
	Squat	928
	Deadlift	1,140
Number of times performed	10 times each	
Mean clip length	34.12 s	
Min clip length	22 s	
Max clip length	64 s	
Frame rate	30 fps	

Existing studies have often collected exercise posture data without differentiating between these contraction types, which may not accurately reflect real exercise conditions. In contrast, our study specifically collected data considering both eccentric and concentric contractions for the three main exercises (bench press, squat, and deadlift).

This approach allows for a more accurate representation of real exercise conditions and provides tailored posture feedback. Table 4 presents the number of classes and data points for each exercise posture, including details for posture class. A key advantage of the proposed MPED dataset is the even distribution of data across classes, enhancing the reliability of posture classification.

2) MODEL TRAINING

This study built models using machine learning and deep learning. For the machine learning approach, the collected data was preprocessed, and four algorithms—logistic regression, ridge classification, random forest, and gradient boosting—were used to construct posture classification models and their performances were compared to obtain optimized model weights. Each model was chosen based on the characteristics of posture classification and the complexity of the data. For instance, logistic regression was chosen as a simple baseline for performance comparison with models. Ridge classification was selected for its ability to prevent overfitting and enhance model stability, making it effective for learning complex data patterns, as required in this study. Random forest, an ensemble learning model based on decision trees, was chosen for its suitability in learning data patterns and handling complex classification problems. Gradient boosting, another ensemble learning technique, was selected for its ability to improve performance by iteratively adding models to correct the errors of previous ones. This study evaluated the performance of each model and selected the random forest model as the machine learning algorithm due to its highest F1-score.

For the deep learning approach, we initially employed a fully connected neural network (FCNN) to build the posture classification model. FCNN represents the most basic structure of an artificial neural network, where every neuron in each layer is connected to every neuron in the previous layer. This structure is suitable for learning complex data patterns. During the deep learning model training process, the model architecture was first defined, hyperparameters were tuned, and the FCNN model was trained using the preprocessed data. The model performance was evaluated based on the F1-score, which measured the accuracy across all three exercise types.

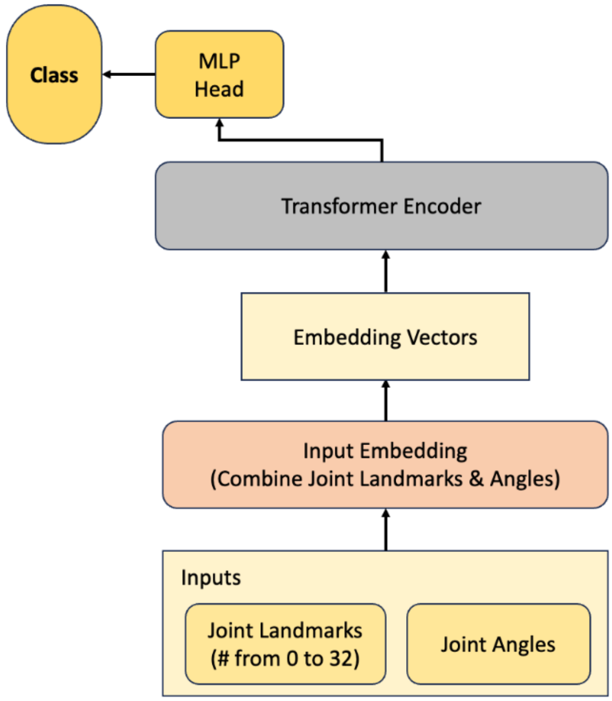


FIGURE 6. Transformer-Based Posture Classifier: Flow diagram of the Transformer-based posture classifier, showing the process from input embedding of joint landmarks and angles to the final classification output using MLP.

In addition to FCNN, this study incorporated MLP-Mixer and Transformer architectures to further explore the potential of deep learning models in posture classification. The MLP-Mixer model was chosen due to its ability to effectively mix both spatial and channel-wise information, making it well-suited for learning the relationships between the 3D joint landmarks and joint angles [43]. Unlike the FCNN, which connects all neurons in each layer, the MLP-Mixer utilizes a more structured approach that allows it to handle complex dependencies across different joints and angles more effectively. This structure enables it to better capture intricate patterns in the 3D coordinate values and joint angles without relying on a fully connected structure.

The Transformer model, on the other hand, was selected for its ability to model long-range dependencies using the self-attention mechanism [44]. Given the structured and non-image nature of our data, consisting of 3D coordinates and joint angles extracted into CSV files, the Transformer is particularly advantageous in capturing complex interactions

TABLE 4. Classification of postures across muscle contraction phases for powerlifting exercises.

Type of Exercise	Muscle Contraction Phase	Posture Classification	Exercise Posture Classes	Number of Data Samples
Bench press	Concentric Contraction (up)	Correct Posture	b_correct_up	134
		Excessive Arch in Lower Back	b_excessive_arch_up	173
		Arms Spread Too Wide	b_arms_spread_up	153
	Eccentric Contraction (down)	Correct Posture	b_correct_down	147
		Excessive Arch in Lower Back	b_excessive_arch_down	167
		Arms Spread Too Wide	b_arms_spread_down	181
Squat	Concentric Contraction (up)	Correct Posture	s_correct_up	118
		Non-neutral Spine	s_spine_neutral_up	113
		Knees Caved In	s_caved_in_knees_up	114
		Feet Spread Too Wide	s_feet_spread_up	120
	Eccentric Contraction (down)	Correct Posture	s_correct_down	115
		Non-neutral Spine	s_spine_neutral_down	117
		Knees Caved In	s_caved_in_knees_down	113
		Feet Spread Too Wide	s_feet_spread_down	118
Deadlift	Concentric Contraction (up)	Correct Posture	d_correct_up	138
		Non-neutral Spine	d_spine_neutral_up	139
		Arms Spread Too Wide	d_arms_spread_up	141
		Arms Too Narrow	d_arms_narrow_up	151
	Eccentric Contraction (down)	Correct Posture	d_correct_down	132
		Non-neutral Spine	d_spine_neutral_down	152
		Arms Spread Too Wide	d_arms_spread_down	139
		Arms Too Narrow	d_arms_narrow_down	148

between these features. This makes it more suitable than the FCNN, which may struggle with learning such complex relationships. During training, both the MLP-Mixer and Transformer models were optimized using the same preprocessed data and hyperparameter tuning strategies applied to the FCNN. Their performances were evaluated and compared against each other, with the F1-score used as the primary metric for determining model effectiveness.

F. POSTURE INFERENCE AND CORRECTION ALGORITHM FOR EACH EXERCISE

This study does not simply distinguish between correct and incorrect postures but provides detailed feedback tailored to the characteristics of each exercise. Most existing studies have focused on distinguishing correct and incorrect postures or providing simple feedback that joint angles must be within a certain range, or merely indicating incorrect postures. However, as the three main exercises involve handling heavy weights, identifying various injury-prone postures and providing customized feedback for each posture is crucial to prevent injuries.

To achieve this, the proposed system provides the actual service monitoring the joint landmarks of the object performing the bench press, squats, and deadlift in real time, providing feedback on the performance postures. Users select the posture and perform the exercise. Feedback includes whether the posture is correct, detailed textual feedback, and a video with skeleton feature to understand their posture. Additionally, each joint angle is provided for the users to accurately assess their posture. Moreover, the users can adjust the pose estimation confidence threshold.

Once the model classifies the posture, corrective feedback is generated based on the deviation of the actual posture from the reference. For each joint, the deviation is quantified as Equation (4):

$$\Delta\theta = \theta_{actual} - \theta_{reference}, \tag{4}$$

where $\theta_{reference}$ is the optimal joint angle for the given exercise. If $\Delta\theta > 0$, indicating an overextension, the proposed method recommends reducing the joint angle. On the other hand, if $\Delta\theta < 0$, a recommendation to increase the angle may be given.

Our posture evaluation criteria was developed in consultation with three experienced gym trainers, each with over three years of professional experience. These trainers provided detailed feedback on common postural errors observed in gym members, particularly those leading to injuries.

For the bench press, maintaining correct posture is essential to avoid issues like lower back strain and shoulder stress. Trainers emphasized that one of the most common causes

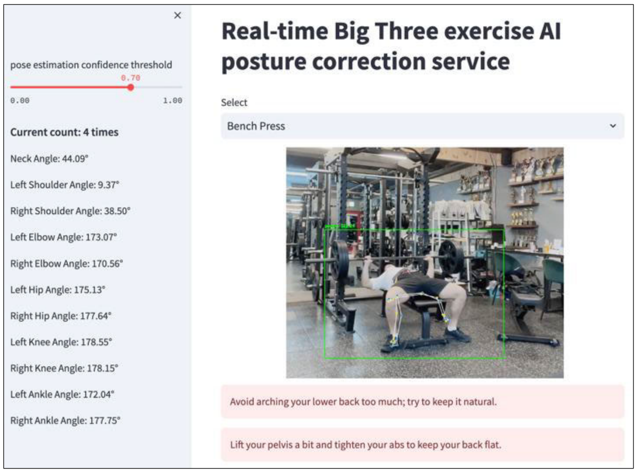


FIGURE 7. **Real-Time Bench Press Posture Feedback. Example of real-time AI feedback for bench press posture correction, demonstrating the detected issues such as excessive lower back arching and providing corrective suggestions.**

of injury is hyperextension of the lower back and improper barbell grip width. Correct posture requires a slight arch in the lower back, which provides stability and allows for better engagement of the chest muscles. Furthermore, lifting the hips off the bench, a frequent error among beginners, should be avoided to prevent lower back strain. Other frequent issues include arms spreading too wide or too narrow, both of which can lead to shoulder strain or reduce control over the barbell path. Therefore, this study classifies bench press postures into "correct posture," "excessive arch in lower back," and "arms spread too wide," providing customized feedback for each posture. For example, Fig. 7 shows a trainee performing the bench press exercise under the monitoring of the proposed system. This example detects the following issue: "Excessive Arch in Lower Back." Consequently, the system provides targeted feedback for each issue: "Avoid arching your lower back too much; keep it natural" based on the deviation $\Delta\theta$ to address the excessive arching.

For the squat, the primary issues identified were knee valgus (knees collapsing inward) and insufficient squat depth. Correct posture involves achieving thigh-parallel depth while keeping the knees aligned with the toes. Trainers also noted that excessive forward lean, often caused by limited ankle mobility, should be corrected to minimize lower back stress. Non-neutral spine positions and feet positioned too wide are additional errors that compromise stability and can lead to injury. More specifically, incorrect postures can result in knee ligament injuries or back strain. Therefore, the system classifies squats into "correct posture," "non-neutral spine," "knees caved in," and "feet spread too wide," providing appropriate feedback. Fig. 8 illustrates a trainee performing the squat exercise while being monitored by the proposed AI posture correction system. This model detects two key issues with the trainee's posture: "Non-neutral Spine" and "Knees Caved In." For the "Non-neutral Spine" issue, the model advises: "Push your hips back to keep your knees and toes in a straight line." This helps ensure proper alignment of the lower body during the movement, preventing potential back strain. For the "Knees Caved In" issue, the feedback given is: "Be cautious not to let your knees cave in during the squat." This aims to protect the knee joints and ensure stability throughout the exercise.

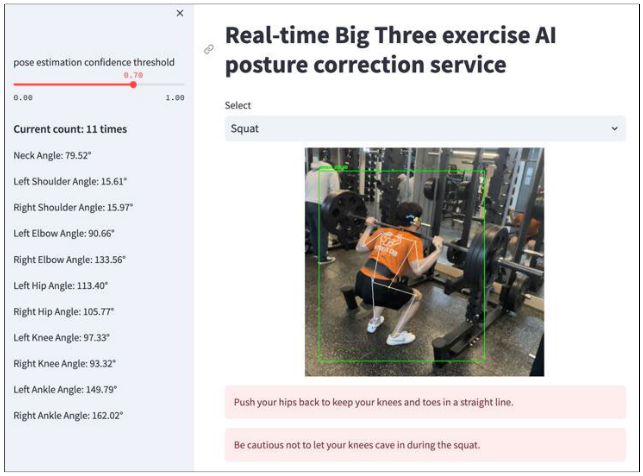


FIGURE 8. **Real-Time Squat Posture Feedback: Example of real-time AI feedback for squat posture correction, showing detected issues such as non-neutral spine and knees caving in, along with corrective suggestions.**

For the deadlift, trainers identified rounding of the lower back and improper barbell positioning as frequent errors. Correct execution requires maintaining a neutral spine throughout the lift and keeping the barbell close to the shins and thighs. Trainers further emphasized that locking out the knees too early disrupts proper lifting mechanics, increasing the risk of strain. Additionally, spinal misalignment and feet spreading too wide or too narrow can negatively impact balance and lifting efficiency. More specifically, incorrect posture, such as a curved spine or arms spread too wide, can lead to back strain. Therefore, deadlift postures are classified into "correct posture," "non-neutral spine," "arms spread too wide," and "arms too narrow." Fig. 9 shows a trainee performing the deadlift exercise, with real-time posture monitoring provided by the system. In this instance, the example identifies that the trainee is performing the exercise correctly for most aspects. Consequently, it provides positive feedback: "You are performing the exercise with the correct posture." However, one issue is detected: "Arms Spread Too Wide": The system outputs the feedback: "Your grip is too wide. Hold the bar a bit narrower." This correction is intended to help the trainee maintain better control of the bar and prevent undue strain on the shoulders and back.

By providing detailed feedback tailored to the characteristics of each exercise, exercisers can clearly identify the causes of incorrect postures and receive specific guidance on how to improve their posture, maximizing exercise effectiveness and minimizing injury risk.

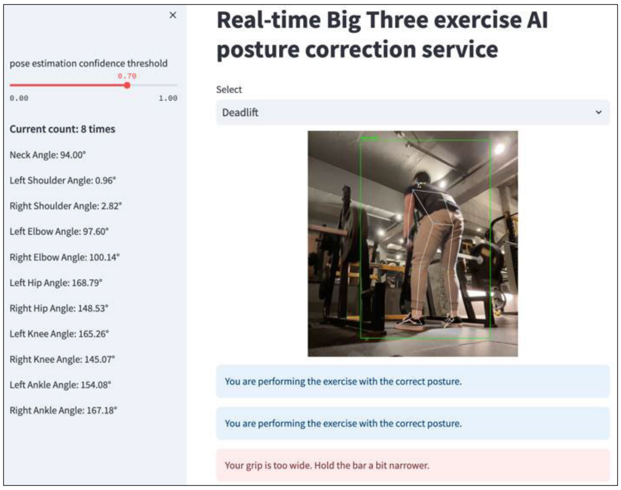


FIGURE 9. Real-Time Deadlift Posture Feedback: Example of real-time AI feedback for deadlift posture correction, indicating correct posture and providing suggestions for improving grip width.

IV. PERFORMANCE EVALUATION

This section is structured as follows: Section IV-A summarizes the models implemented in this study. Section IV-B introduces the experimental environment. Section IV-C evaluates the performance of each model implemented using machine learning and deep learning methods. Finally, Section IV-D analyzes the results of a survey on the feedback method for users of the AI posture correction service.

A. SUMMARY OF IMPLEMENTED MODELS

This study implements and evaluates models for detecting exercise objects using YOLOv5 object detection models, as well as machine learning and deep learning models for classifying postures. Table 5 summarizes the experimental information implemented in this study.

In the first experiment, different labeling methods for the training dataset were used to improve the single-person estimation accuracy of MediaPipe. Then, the performance of exercise object detection models using YOLOv5 was compared. The hyperparameters used include batch size, epochs, and weights. Batch size refers to the number of images processed by the network at once. In this study, the batch size was set to 16. This means that 16 images are simultaneously input to the model. Epochs refer to the number of times the entire dataset passes through the network. In this study, the network was trained over 200 epochs. Finally, weights refer to the initial weights of the model. YOLOv5 contains pre-trained weights, allowing the model to be tuned for new datasets by using pre-trained information. This study used the default weights file stored in yolov5s.pt to train the model.

Next, in the second experiment, the performances of various machine learning and deep learning algorithms were compared. The performances of exercise posture classification models using machine learning algorithms such as logistic regression, ridge classification, random forest, and gradient boosting were compared and analyzed. In logistic regression, C is a regularization parameter that adjusts the penalty on the model complexity. The maximum number of iterations was set to 100. In Ridge classification, Alpha (α) determines the strength of the regularization. In random forest, the number of trees and the maximum depth were set to 100 and ‘none’, respectively. Finally, for the gradient boosting, the number of trees, maximum depth, and the learning rate were set to 100, 3, and 0.1, respectively.

TABLE 5. Summarized information of the implemented models.

Ex		Parameters	
Ex. 1	Exercise Object Detection Model Using YOLOv5	batch size	16
		epochs	200
		weights	yolov5s.pt
Ex. 2	Exercise Posture Classification Model Using Machine Learning Algorithms	Logistic Regression	C = 1.0
			max_iter = 100
		Ridge Classifier	alpha = 1.0
		Random Forest	n_estimators = 100
			max_depth = None
		Gradient Boosting	n_estimators = 100
			max_depth = 3
			learning_rate = 0.1
Ex. 3	Exercise Posture Classification Model Using Deep Learning	criterion	Cross Entropy Loss
		optimizer	Adam
		learning rate	0.001
		epochs	500

In the third experiment, deep learning-based exercise posture classification models were implemented using a fully connected neural network (FCNN), MLP-Mixer, and Transformer. The FCNN model consisted of four layers, each designed to transform the input data and generate probability values for the posture classes. Initially, the 143-dimensional joint landmark and joint angle data were transformed into 128 dimensions in the first layer. This was followed by further dimensionality reduction to 64 and then 32 in the subsequent layers. The final layer produced the probability values for each posture class. To enhance model performance, a ReLU activation function was applied after each layer, and dropout was incorporated to prevent overfitting.

For the MLP-Mixer model, a series of fully connected layers were used to process the input data. The model consisted of multiple blocks, each containing three fully connected layers. The input data, initially 143-dimensional, was transformed into a 128-dimensional hidden representation. Each block successively applied transformations that captured the complex relationships between the joint landmarks and angles. Specifically, each block included a linear layer to transform the data into the hidden dimension, followed by another linear layer to mix this information, and a final linear layer to restore the original input dimension. The model then used two fully connected layers to map the transformed features into 64 dimensions and finally to the posture class outputs, utilizing ReLU activation and dropout to prevent overfitting.

The Transformer model utilized a different approach, leveraging a self-attention mechanism to capture dependencies across the input features. Initially, the 143-dimensional input was projected into a 128-dimensional embedding space using a linear layer. This embedding was then passed through a Transformer Encoder consisting of two layers, each with four attention heads, allowing the model to focus on different parts of the input data simultaneously. After passing through the Transformer layers, a global average pooling operation was applied to aggregate the information, which was then fed into two fully connected layers to reduce the dimensionality to 64 and finally to the posture class outputs. This model effectively captured long-range dependencies between joints, making it well-suited for the posture classification task.

The hyperparameters for this experiment are as follows. The Cross-Entropy Loss function was used, as this is a multi-class classification problem. The model was optimized using the Adam optimizer, with a learning rate of 0.001, and it was trained over 500 epochs. Furthermore, deep learning model used the proposed MPED dataset, which were divided into training and test sets using a 7:3 ratio. Specifically, the bench press dataset contained 668 training samples and 287 test samples; the squat dataset had 649 training samples and 279 test samples; and the deadlift dataset had 798 training samples and 342 test samples.

The server for this experiment used a PC with Windows 11 Pro OS, a 13th Gen Intel(R) Core(TM) i5-13600KF 3.50 GHz processor, a GeForce RTX 4080 graphics card, and 32 GB RAM. The client used a MacBook with macOS 14 and an M1 chip, and an iPhone 12 Pro for the experiment. Moreover, Python 3.10 was used on the server.

B. EVALUATION METRICS

The evaluation of the posture classification models in this study was conducted using the following performance metrics: accuracy, precision, recall, and F1-score [25], [45]. Specifically, Accuracy (A) is defined as the ratio of correctly predicted observations to the total observations. It is calculated using the Equation (5):

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}, \tag{5}$$

where T_p is true positives, T_n is true negatives, F_p is false positives and F_n is false negatives.

Precision (P) is the ratio of correctly predictive positive observations to the total predicted positive observations. It is an indicator of the reliability of the model's positive predictions, and it is calculated as Equation (6):

$$P = \frac{T_p}{T_p + F_p}. \tag{6}$$

Recall (R) also known as sensitivity or true positive rate, measures the ratio of correctly predicted positive observations to all the observations in the actual class. It is calculated as Equation (7):

$$R = \frac{T_p}{T_p + F_n}. \tag{7}$$

The F1-score (F_1) is the harmonic mean of precision and recall, providing a balance between the two, especially useful when dealing with imbalanced datasets. It is calculated as Equation (8):

$$F_1 = 2 \cdot \frac{P \times R}{P + R}. \tag{8}$$

Using these metrics, the performance of the posture classification models was evaluated for the bench press, squat, and deadlift exercises.

The performance metric used to quantitatively assess the implemented models was the mean Average Precision (mAP), a commonly used metric in computer vision. mAP is obtained from precision and recall. mAP is calculated by averaging the precision and recall across all classes given in Equation (9).

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i, \tag{9}$$

where N is the number of classes and AP_i is average precision for class i.

Therefore, mAP can comprehensively measure model performance in multi-class classification problems, and is a crucial metric reflecting the performance differences among classes.

C. EXPERIMENTAL RESULTS

1) EXERCISE OBJECT DETECTION MODEL USING YOLOv5

To implement the algorithm proposed in this paper for estimating the joints of a person performing the exercise, two object detection models for detecting exercisers were implemented. The first model was trained on data labeled with the lowest possible margin from head to toe of the person object, while the second model was also labeled from head to toe but with a sufficient margin on all sides.

Fig. 10 shows the labeling screen of the data used to train the first model, and details the object detection performance on the validation set of the model trained with YOLOv5s. The experimental results showed a high mAP of 0.994 for an IoU threshold of 0.5 and an mAP of 0.843 for an IoU threshold of 0.5–0.95. However, estimating the joint landmarks only within the bounding boxes detected by this model proved difficult, reducing performance.

To address this issue, we hypothesized that providing sufficient margins around the target object, as shown in Fig. 11, would improve performance, and the model was retrained

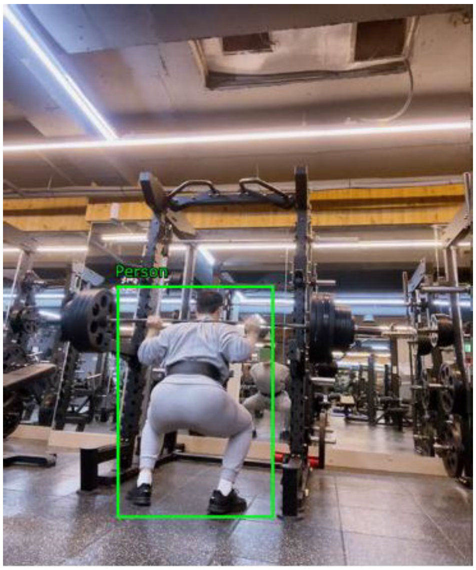


FIGURE 10. Data Labeling with Narrow Margins. Example of data labeling for exercise posture detection, showing a narrow bounding box around the subject performing a squat.

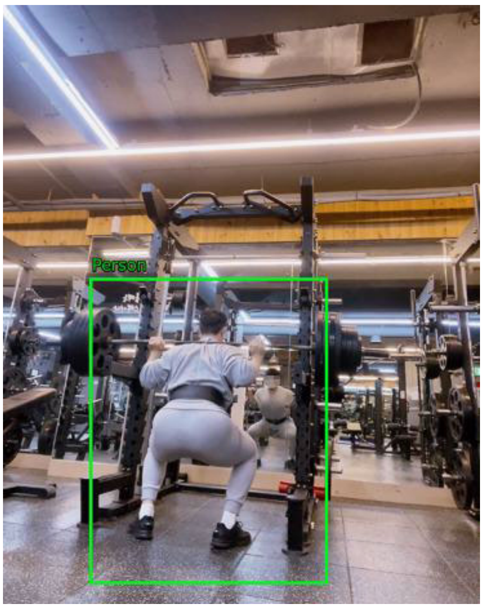


FIGURE 11. Data Labeling with Sufficient Margins. Example of data labeling for exercise posture detection, showing a sufficient bounding box around the subject performing a squat.

with this new labeling approach. Fig. 11 shows the inference screen for the validation set of the second model. The object detection performance for the validation set of the exercise object detection model trained with YOLOv5s, using datasets labeled with sufficient margins, was confirmed as shown in Table 15. The experimental results showed that the mAP value was 0.994 for an IoU threshold of 0.5, the same as the first model. For an IoU threshold of 0.5–0.95, the mAP was 0.705, a decrease of 0.138 compared to the original model. Nevertheless, higher accuracy was achieved for joint landmark estimation within the bounding boxes of detected objects.

2) MACHINE LEARNING AND DEEP LEARNING-BASED EXERCISE POSTURE CLASSIFICATION MODELS

In this experiment, exercise posture classification models for bench press, squat, and deadlift were implemented and compared using machine learning and deep learning methods. Figs. 12–14 show the training loss and test loss graphs. Typically, loss graphs are used either to understand the progress of deep learning model training or to detect overfitting. Training loss indicates how well the model has learned from the training data, while test loss indicates how well the model generalizes to new data that it has not seen during training. As shown in Figs. 12–14, the training loss decreases as the number of epochs increases, indicating that the model is

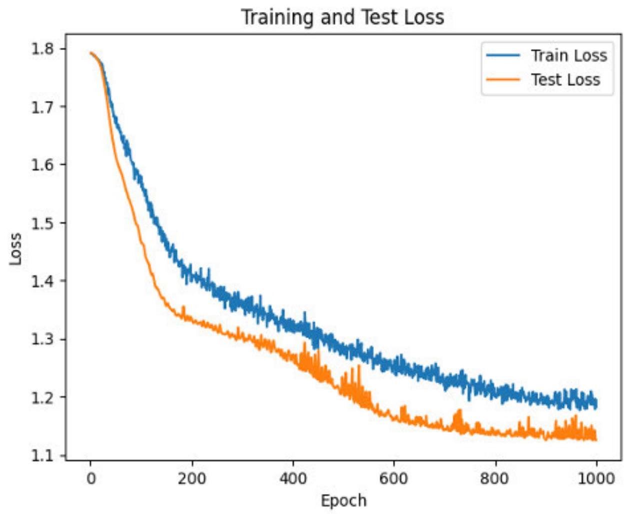


FIGURE 12. Training loss and test loss graphs for the Transformer-based bench press classification model.

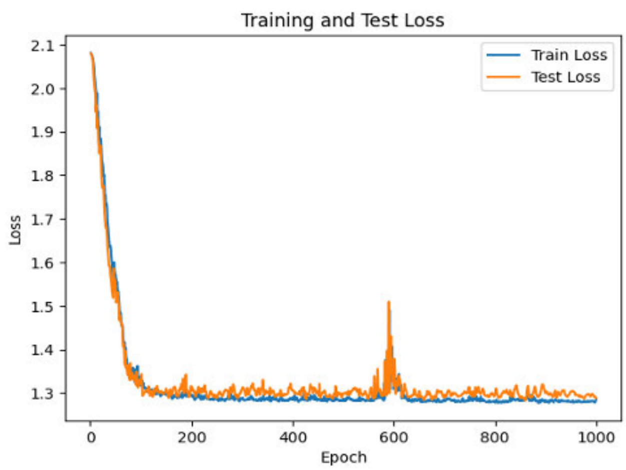


FIGURE 13. Training loss and test loss graphs for the Transformer-based squat classification model.

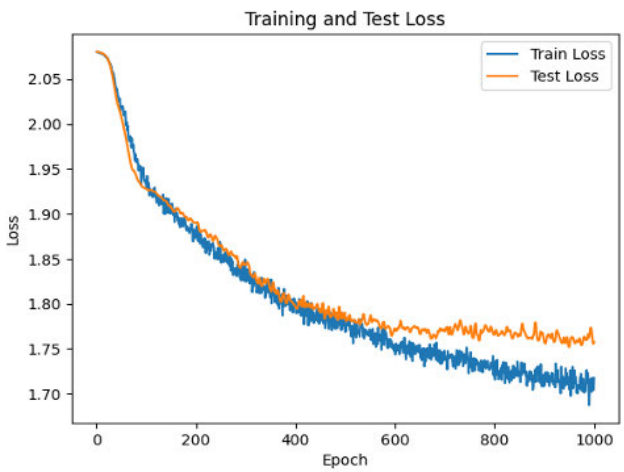


FIGURE 14. Training loss and test loss graphs for the Transformer-based deadlift classification model.

learning well from the training data. To accurately verify the performance of the exercise posture classification model, this study evaluated the performance using the accuracy, precision, recall, and F1-score metrics for each exercise posture classification model. The results of the machine learning and deep learning methods were compared based on these metrics.

Table 6 demonstrates the performance of the bench press posture classification model for each algorithm. Among the machine learning algorithms, the model built with the random forest algorithm showed the highest performance with an F1-score of 0.970. In contrast, the deep learning models yielded varying results. The Transformer model achieved the highest performance among the deep learning models with an F1-score of 0.958, which closely matched the performance of some machine learning models. The FCNN model followed with an F1-score of 0.921, while the MLP-Mixer model for bench press underperformed with a significantly lower F1-score of 0.688.

The Transformer model's competitive performance highlights its ability to effectively capture long-range dependencies in posture data, while the MLP-Mixer struggled to match its performance, possibly due to the model's architecture being less suited for the complexity of the dataset. Although the FCNN model demonstrated moderate success, it was still outperformed by the machine learning algorithms. The lower performance of deep learning models in comparison to machine learning models may be attributed to the relatively small size of the dataset, as deep learning models typically require a large amount of data to perform optimally.

Table 7 details the performance evaluation metrics of the model trained with the random forest algorithm, which exhibited the highest F1-score, for classifying correct and improper postures for the up and down phases of the bench press. Using 287 test data for classification, an accuracy of 0.97 was achieved. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the same value of 0.97.

TABLE 6. Comparison of bench press posture classification model performance by different algorithms (The best performance is marked in bold).

Bench press	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.958	0.959	0.958	0.958
Ridge Classifier	0.948	0.949	0.948	0.948
Random Forest	**0.965**	**0.966**	**0.965**	**0.970**
Gradient Boosting	0.948	0.949	0.948	0.948
FCNN	0.904	0.939	0.904	0.921
MLP-Mixer	0.686	0.689	0.686	0.688
Transformer	**0.958**	**0.958**	**0.958**	**0.958**

TABLE 7. Performance metric results of the bench press posture classification model implemented using the random forest algorithm.

Bench press	Precision	Recall	F1-Score	Support
b_correct_up	0.95	1.00	0.97	37
b_correct_down	1.00	0.94	0.97	48
b_excessive_arch_up	0.95	1.00	0.98	40
b_excessive_arch_down	0.98	0.96	0.97	51
b_arms_spread_up	0.94	0.96	0.95	51
b_arms_spread_down	0.97	0.95	0.96	60
Accuracy			**0.97**	
Macro avg	0.96	0.97	0.97	**287**
Weighted avg	**0.97**	**0.97**	**0.97**	

Table 8 shows the performance of each algorithm for the squat posture classification model. Among the machine learning algorithms, similar to the bench press, the model built with the random forest algorithm achieved the highest performance with an F1-score of 0.989. The logistic regression model performed similarly, also achieving an F1-score of 0.989, highlighting the consistency of simpler machine learning models in this classification task.

For deep learning algorithms, the Transformer model achieved the highest performance with an F1-score of 0.985, closely matching the performance of the best machine learning models. In contrast, the FCNN and MLP-Mixer models showed lower performance, with F1-scores of 0.851 and 0.838, respectively. As observed previously, the superior

TABLE 8. Comparison of squat posture classification model performance by different algorithms (The best performance is marked in bold).

Squat	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.989	0.990	0.989	0.989
Ridge Classifier	0.986	0.986	0.986	0.986
Random Forest	**0.989**	**0.989**	**0.989**	**0.989**
Gradient Boosting	0.986	0.986	0.986	0.986
FCNN	0.828	0.877	0.828	0.851
MLP-Mixer	0.836	0.840	0.836	0.838
Transformer	**0.983**	**0.987**	**0.983**	**0.985**

performance of the Transformer model underscores its ability to handle the complexity of spatial relationships in squat posture data.

Table 9 presents the results of the performance evaluation metrics for the model trained with the random forest algorithm, which had the highest F1-score, for classifying correct and improper postures for the up and down phases of the squats. Using 279 test data for classification, an accuracy of 0.99 was achieved. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the same value of 0.99.

TABLE 9. Performance metric results of the squat posture classification model implemented using the random forest algorithm.

Squat	Precision	Recall	F1-Score	Support
s_correct_up	1.00	1.00	1.00	42
s_correct_down	1.00	1.00	1.00	30
s_spine_neutral_up	0.97	0.94	0.95	33
s_spine_neutral_down	0.94	0.97	0.95	30
s_caved_in_knees_up	1.00	1.00	1.00	41
s_caved_in_knees_down	1.00	1.00	1.00	38
s_feet_spread_up	1.00	1.00	1.00	36
s_feet_spread_down	1.00	1.00	1.00	29
Accuracy			**0.99**	
Macro avg	0.99	0.99	0.99	**279**
Weighted avg	**0.99**	**0.99**	**0.99**	

Table 10 presents the performance of each algorithm for the deadlift posture classification model. Among the machine learning algorithms, the random forest model achieved the highest performance with an F1-score of 0.962, similar to its strong performance in the bench press and squat tasks. On the other hand, the Transformer model continued to outperform all other models, achieving the highest overall F1-score of 0.965. This aligns with previous observations, where the Transformer effectively handles the spatial complexity of posture data, especially in deadlift classification. Meanwhile, the MLP-Mixer model achieved a moderate F1-score of 0.610, and the FCNN model showed the lowest performance, with an F1-score of only 0.493. As observed previously, the Transformer model's superior performance further reinforces its suitability for posture classification tasks, outperforming both machine learning and other models across various exercises.

TABLE 10. Comparison of deadlift posture classification model performance by different algorithms (The best performance is marked in bold).

Deadlift	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.912	0.914	0.912	0.912
Ridge Classifier	0.880	0.882	0.880	0.879
Random Forest	**0.962**	**0.964**	**0.962**	**0.962**
Gradient Boosting	0.947	0.950	0.947	0.947
FCNN	0.439	0.563	0.439	0.493
MLP-Mixer	0.600	0.620	0.600	0.610
Transformer	**0.961**	**0.968**	**0.961**	**0.965**

Table 11 presents the results of the performance evaluation metrics for the model trained with the random forest algorithm, which had the highest F1-score, for classifying correct and improper postures for the up and down phases of the deadlift. Using 342 test data for classification, an accuracy of 0.96 was achieved. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the same value of 0.96.

These experiment quantitatively verified the model performance through the above evaluation metrics and compared the performance of models based on machine learning and deep learning methods. In all experiments, models implemented with machine learning algorithms generally outperformed deep learning models.

3) COMBINED ECCENTRIC AND CONCENTRIC CONTRACTIONS

In this experiment, a separate test was conducted to compare the accuracy of machine learning algorithms and deep

TABLE 11. Performance metric results of the deadlift posture classification model implemented using the random forest algorithm.

Deadlift	Precision	Recall	F1-Score	Support
d_correct_up	1.00	0.95	0.97	57
d_correct_down	0.97	0.94	0.96	35
d_spine_neutral_up	0.96	0.90	0.93	52
d_spine_neutral_down	0.90	1.00	0.95	44
d_arms_spread_up	0.97	1.00	0.98	32
d_arms_spread_down	0.95	1.00	0.97	38
d_arms_narrow_up	0.96	0.98	0.97	44
d_arms_narrow_down	1.00	0.95	0.97	40
Accuracy			**0.96**	
Macro avg	0.96	0.97	0.96	**342**
Weighted avg	**0.96**	**0.96**	**0.96**	

learning models when combining eccentric (down) and concentric (up) contractions into a single class for each posture. Each model was implemented with the same hyperparameters used in previous experiments and its performance was evaluated. The machine learning algorithms used were logistic regression, ridge classification, random forest, and gradient boosting algorithms, while the deep learning models used were the bench press, squat, and deadlift posture classification models. This study evaluated the performance of each exercise posture classification model using the performance evaluation metrics—accuracy, precision, recall, and F1-score.

TABLE 12. Performance comparison of bench press posture classification models implemented using machine learning algorithms (The best performance is marked in bold).

Bench press	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.993	0.993	0.993	0.993
Ridge Classifier	0.979	0.979	0.979	0.979
Random Forest	**0.997**	**0.997**	**0.997**	**0.997**
Gradient Boosting	**0.997**	**0.997**	**0.997**	**0.997**

Table 12 details the performance evaluation metric results of the model classifying correct and improper postures when performing the bench press. The algorithms with the highest F1-score were the random forest and gradient boosting algorithms. The model trained with the random forest algorithm was analyzed. Using 287 test data for classification, an accuracy of 1.00 was achieved as shown in Table 13. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the same value of 1.00. An F1-score of 1.00 implies perfect prediction for all classes, which raises suspicion of post-hoc bias.

TABLE 13. Performance metric results of the bench press posture classification model implemented using the random forest algorithm.

Bench press	Precision	Recall	F1-Score	Support
b_correct	1.00	0.99	0.99	85
b_excessive_arch	0.99	1.00	0.99	91
b_arms_spread	1.00	1.00	1.00	111
Accuracy			**1.00**	
Macro avg	1.00	1.00	1.00	**287**
Weighted avg	**1.00**	**1.00**	**1.00**	

Table 14 presents the results of the model classifying correct and improper postures when performing squat. The algorithms with the highest F1-score were the random forest and gradient boosting algorithms. For consistency, the model trained with the random forest algorithm was analyzed. Using 279 test data for classification, an accuracy of 1.00 was achieved as shown in Table 15. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the value of 1.00. An F1-score of 1.00 implies perfect prediction for all classes, raising suspicion of post-hoc bias, similar to that in the bench press case.

TABLE 14. Performance comparison of squat posture classification models implemented using machine learning algorithms The best performance is marked in bold.

Squat	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.993	0.993	0.993	0.993
Ridge Classifier	0.996	0.997	0.996	0.996
Random Forest	**1.000**	**1.000**	**1.000**	**1.000**
Gradient Boosting	**1.000**	**1.000**	**1.000**	**1.000**

Table 16 details the results of the performance evaluation metrics for the model trained using the random forest algorithm, which had the highest F1-score, for classifying correct and improper postures for each phase of the deadlift. Using 342 test data for classification, an accuracy of 0.98 was achieved as shown in Table 17. Interpreting the classification metrics using the weighted average, aligning with the purpose of this study, the achieved precision, recall, and F1-score had the same value of 0.98.

Figs. 15–17 show the training loss and test loss graphs. As shown, the training loss decreases as the number of epochs

TABLE 15. Performance metric results of the squat posture classification model implemented using the random forest algorithm.

Squat	Precision	Recall	F1-Score	Support
s_correct	1.00	1.00	1.00	72
s_spine_neutral	1.00	1.00	1.00	63
s_caved_in_knees	1.00	1.00	1.00	79
s_feet_spread	1.00	1.00	1.00	65
Accuracy			**1.00**	
Macro avg	1.00	1.00	1.00	**279**
Weighted avg	**1.00**	**1.00**	**1.00**	

TABLE 16. Performance comparison of deadlift posture classification models implemented using machine learning algorithms The best performance is marked in bold.

Deadlift	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.883	0.884	0.883	0.883
Ridge Classifier	0.874	0.876	0.874	0.874
Random Forest	**0.983**	**0.983**	**0.983**	**0.983**
Gradient Boosting	0.968	0.968	0.968	0.968

TABLE 17. Performance metric results of the deadlift posture classification model implemented using the random forest algorithm (merged classes).

Deadlift	Precision	Recall	F1-Score	Support
d_correct	1.00	0.98	0.99	92
d_spine_neutral	0.97	0.98	0.97	96
d_arms_spread	1.00	1.00	1.00	70
d_arms_narrow	0.96	0.98	0.97	84
Accuracy			**0.98**	
Macro avg	0.98	0.98	0.98	**342**
Weighted avg	**0.98**	**0.98**	**0.98**	

increases, indicating that the model is learning well from the training data.

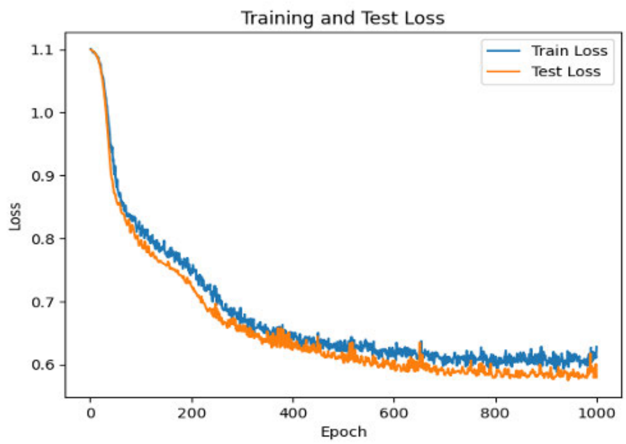


FIGURE 15. Training loss and test loss graphs for the Transformer-based bench press classification model (merged classes).

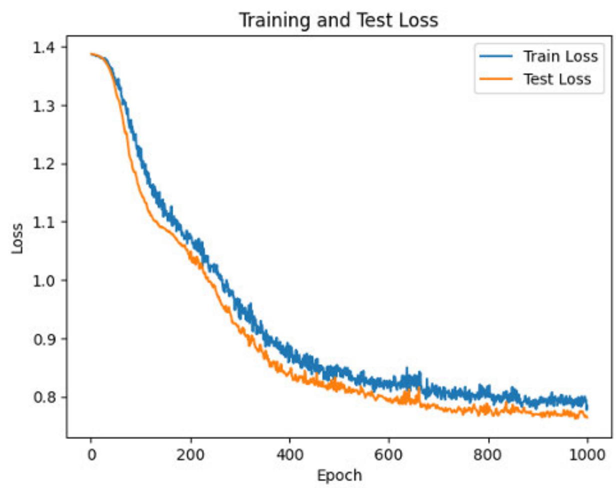


FIGURE 16. Training loss and test loss graphs for the Transformer-based squat classification model (merged classes).

Table 18 provides the performance evaluation metrics for classifying postures when combining eccentric (down) and concentric (up) contractions into a single class for each posture of bench press, squat, and deadlift. Using 239 test data for bench press data classification, an accuracy of 0.97 was achieved. Interpreting the classification metrics according to the purpose of this study, the achieved precision, recall, and F1-score values were 0.99, 0.97, and 0.98, respectively. Using 232 test data for squat data classification, an accuracy of 0.98 was achieved. Interpreting the classification metrics according to the purpose of this study, the achieved precision, recall, and F1-score values were 0.99, 0.98, and 0.98, respectively. Finally, using 285 test data for deadlift data classification, the achieved accuracy was 0.84. Interpreting the classification metrics according to the purpose of this study, the achieved values of precision, recall, and F1-score were 0.89, 0.84, and 0.86, respectively. When combining the classes, the deep learning models showed improved F1-scores

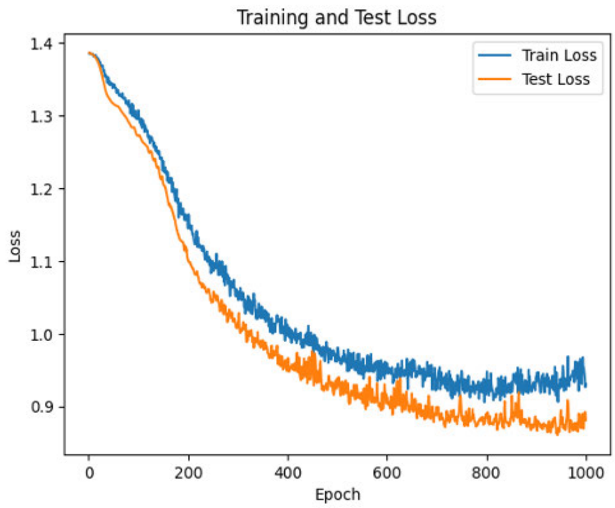


FIGURE 17. Training loss and test loss graphs for the Transformer-based deadlift classification model (merged classes).

TABLE 18. Performance comparison of transformer-based posture classification models for each exercise with combined classes.

	Accuracy	Precision	Recall	F1-Score
Bench press	0.97	0.99	0.97	**0.98**
Squat	0.98	0.99	0.98	**0.98**
Deadlift	0.84	0.89	0.84	**0.86**

for bench press, squats, and deadlift, especially showing high performance for bench press and squats. However, the performance was relatively lower for the deadlift, suggesting the need for further optimization for this exercise.

D. ANALYSIS OF SURVEY RESULTS ON FEEDBACK PROVISION FOR CONTRACTION PHASES

Existing studies generally provide feedback on the overall posture rather than on specific contraction phases. However, this study proposed a system that provides phase-specific feedback by distinguishing between concentric and eccentric contraction phases. To verify the effectiveness of phase-specific feedback proposed in this study, a survey was conducted on 35 users of the AI posture correction service from April 5 to April 8, 2024. The information about the survey participants is summarized in Table 19. The 35 participants included 21 men and 14 women. The age distribution was highest in the 20s (65.7%), followed by 50s, 60s and above, and 30s, with no participants in their teens or 40s.

TABLE 19. Summary of survey participant information.

Item	Response	Response Rate
Gender	1. Male 2. Female	1. 60% (21 people) 2. 40% (14 people)
Age	1. 19 or younger 2. 20–29 3. 30–39 4. 40–49 5. 50–59 6. 60 and older	1. 0% (0 people) 2. 65.7% (23 people) 3. 5.7% (2 people) 4. 0% (0 people) 5. 17.1% (6 people) 6. 11.4% (4 people)
Exercise Experience	1. Less than 1 year 2. 1 to less than 3 years 3. 3 to less than 5 years 4. 5 to less than 7 years 5. 7 to less than 10 years 6. 10 years or more	1. 34.3% (12 people) 2. 20% (7 people) 3. 28.6% (10 people) 4. 2.9% (1 person) 5. 8.6% (3 people) 6. 5.7% (2 people)
Major in Exercise	1. Yes 2. No	1. 14.3% (5 people) 2. 85.7% (30 people)

The participants' exercise experience values ranged from less than 1 year to over 10 years. Among these, 5 were exercise majors, and 30 were non-majors engaged in recreational sports.

This survey aimed to investigate the effectiveness of providing feedback for each contraction phase. Google Forms was used to distribute the questionnaire. The survey consisted of five main items as shown in Table 20, and for items 1–4, participants were asked to describe their reasons.

Analyzing the survey results, for the first question, 65.7% (23 people) thought 'overall posture feedback' was the most effective, while 28.6% (10 people) thought 'phase-specific posture feedback' was the most effective, and 5.8% (2 people) thought both combined were the most effective. Participants who chose 'overall posture feedback' felt that it helped improve overall exercise posture, reflecting improved overall efficiency and reduced injury risk. Participants who preferred 'phase-specific posture feedback' liked feedback that pointed out specific deficiencies during exercise, helping to closely analyze and improve individual movements. Participants who answered 'both combined' believed that both types of feedback are necessary, indicating that various feedback types are needed depending on individual exercise methods and goals.

TABLE 20. Survey questions and responses.

Question	Response
1. Do you think it is more effective to improve overall posture through a correction service or receive feedback on specific contraction phases?	1. Overall posture feedback 2. Phase-specific posture feedback (eccentric, concentric) 3. Other
2. Do you expect phase-specific posture feedback to have a greater impact on future exercise posture correction compared to overall posture feedback?	1. Strongly agree 2. Agree 3. Neutral 4. Disagree 5. Strongly disagree
3. This AI posture correction service is a web service that identifies and corrects postures for exercises with the highest injury rates in weight training: bench press, squat, and deadlift. Do you think using this service helps prevent injuries more than not using it? Please answer whether you think using this service helps prevent injuries more than not using it.	1. Strongly agree 2. Agree 3. Neutral 4. Disagree 5. Strongly disagree
4. Which method do you think is more user-friendly: overall posture improvement or contraction phase-specific posture feedback?	1. Overall posture feedback 2. Phase-specific posture feedback (eccentric, concentric)
5. Please write down any additional features you think should be added to this service.	Open-ended response

For the second question, 31.4% (11 people) chose 'strongly agree,' 22.9% (8 people) chose 'agree,' 31.4% (11 people) chose 'neutral,' 14.3% (5 people) chose 'disagree,' and 0% (0 people) chose 'strongly disagree.' The 31.4% who chose 'strongly agree' expect phase-specific posture feedback to have a very significant impact on future posture correction compared to overall posture feedback. The 22.9% who chose 'agree' expect phase-specific posture feedback to have a greater impact on future posture correction compared to overall posture feedback. The 31.4% who chose 'neutral' expect similar impacts from both types of feedback. The 14.3% who chose 'disagree' expect phase-specific posture feedback to have less impact on future posture correction compared to overall posture feedback. These responses indicate that the expectations of the impact of phase-specific posture feedback on future posture correction vary, indicating differences based on individual experiences and perceptions.

For the third question, 62.9% (22 people) chose 'strongly agree,' 31.4% (11 people) chose 'agree,' 5.7% (2 people) chose 'neutral,' and 0% (0 people) chose 'disagree' or 'strongly disagree.' Analyzing these responses, the positive opinion that using this service would help prevent injuries was generally predominant.

For the fourth question, 80% (28 people) chose 'overall posture feedback,' while 20% (7 people) chose 'phase-specific posture feedback.' Most of the participants selected 'overall posture feedback' as more user-friendly for improving overall exercise posture. This is because they felt that overall posture feedback helps maintain better posture by providing comprehensive feedback during exercise.

For the last question, participants requested various additional features. For example, the variety of responses suggested automatic diet recommendation, exercise time count, muscle activation measurement, calorie expenditure feedback, muscle fatigue measurement, and options to choose between overall posture or phase-specific feedback. Incorporating some of the additional features requested by participants in future studies is expected to further enhance the service.

V. CONCLUSION

This study used the pose estimation feature of MediaPipe to extract joint landmarks, which were utilized with the key joint angles to build machine-learning and deep-learning models. These models were deployed in a web browser environment using OpenCV and Streamlit to track the user's posture for the three main exercises in real time, classify the posture based on the trained models, and provide voice feedback for incorrect postures.

MediaPipe enables real-time pose estimation in environments with a CPU and a camera, such as mobile devices or PCs with webcams, even without a GPU or with low specifications. However, this framework is limited to pose estimation for a single individual. To overcome this issue, this study used YOLOv5 to detect only the person performing the exercise, thereby improving the accuracy of pose estimation. Additionally, the proposed real-time AI posture correction service for the three main exercises, using YOLOv5 and MediaPipe, provides feedback on static images and performs framewise analysis of the dynamic movements of exercise to ensure that the user is performing the exercises correctly and uses text-to-speech (TTS) technology to provide appropriate feedback on incorrect postures. This service is expected to contribute to safer weight training.

For future research, while our study successfully distinguishes between eccentric and concentric contractions, it does not evaluate muscle engagement or activation patterns. Future work could explore integrating muscle activation analysis, potentially through the surface EMG (sEMG), a non-invasive surface electromyography or similar technologies, to provide a more comprehensive understanding of movement quality. Moreover, expanding the dataset to include more diverse people could further enhance the generalizability of the model. Lastly, future work could involve people with physical disabilities or physical immobility to expand the applicability of our research.

REFERENCES

[1] H. Kaur, T. Singh, Y. K. Arya, and S. Mittal, "Physical fitness and exercise during the COVID-19 pandemic: A qualitative enquiry," *Frontiers Psychol.*, vol. 11, Oct. 2020, Art. no. 590172.

[2] J. M. de Abreu, R. A. de Souza, L. G. Viana-Meireles, J. Landeira-Fernandez, and A. Filgueiras, "Effects of physical activity and exercise on well-being in the context of the covid-19 pandemic," *PLoS ONE*, vol. 17, no. 1, Jan. 2022, Art. no. e0260465.

[3] H. Shaban. (Jan. 2021). *The Pandemic's Home-Workout Revolution May Be Here To Stay*. The Washington Post. Accessed: Aug. 4, 2023. [Online]. Available: https://www.washingtonpost.com/road-to-recovery/2021/01/07/home-fitness-boom

[4] R. Liu, R. Menhas, J. Dai, Z. A. Saqib, and X. Peng, "Fitness apps, live streaming workout classes, and virtual reality fitness for physical activity during the COVID-19 lockdown: An empirical study," *Frontiers Public Health*, vol. 10, Jun. 2022, Art. no. 852311.

[5] V. Bengtsson, L. Berglund, and U. Aasa, "Narrative review of injuries in powerlifting with special reference to their association to the squat, bench press and deadlift," *BMJ Open Sport Exercise Med.*, vol. 4, no. 1, Jul. 2018, Art. no. e000382.

[6] E. E. Schick, J. W. Coburn, L. E. Brown, D. A. Judelson, A. V. Khamoui, T. T. Tran, and B. P. Uribe, "A comparison of muscle activation between a Smith machine and free weight bench press," *J. Strength Conditioning Res.*, vol. 24, no. 3, pp. 779–784, 2010.

[7] E. Dudagoitia, A. García-De-Alcaraz, and L. L. Andersen, "Safety of powerlifting: A literature review," *Sci. Sports*, vol. 36, no. 3, pp. 59–68, Jun. 2021.

[8] J. Siewe, J. Rudat, M. Röllinghoff, U. J. Schlegel, P. Eysel, and J. W.-P. Michael, "Injuries and overuse syndromes in powerlifting," *Int. J. Sports Med.*, vol. 32, no. 9, pp. 703–711, Sep. 2011.

[9] E. Strömbäck, U. Aasa, K. Gilenstam, and L. Berglund, "Prevalence and consequences of injuries in powerlifting: A cross-sectional study," *Orthopaedic J. Sports Med.*, vol. 6, no. 5, May 2018, Art. no. 2325967118771016.

[10] B. J. Schoenfeld, "Squatting kinematics and kinetics and their application to exercise performance," *J. Strength Conditioning Res.*, vol. 24, no. 12, pp. 3497–3506, 2010.

[11] M. R. M. McGuigan and B. D. Wilson, "Biomechanical analysis of the deadlift," *J. Strength Conditioning Res.*, vol. 10, no. 4, pp. 250–255, 1996.

[12] D. Borthakur, A. Paul, D. Kapil, and M. J. Saikia, "Yoga pose estimation using angle-based feature extraction," *Healthcare*, vol. 11, no. 24, p. 3133, Dec. 2023.

[13] V. Singh, A. Patade, G. Pawar, and D. Hadsul, "TrAIner—An AI fitness coach solution," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (I2CT)*, Mumbai, India, Apr. 2022, pp. 1–4.

[14] M. Chariar, S. Rao, A. Irani, S. Suresh, and C. S. Asha, "AI trainer: Autoencoder based approach for squat analysis and correction," *IEEE Access*, vol. 11, pp. 107135–107149, 2023.

[15] R. Gajbhiye, V. Chawla, and A. Singh, "AI human pose estimation: Yoga pose detection and correction," *Int. J. Innov. Sci. Res. Technol. (IJISRT)*, vol. 7, no. 5, pp. 1649–1658, 2022.

[16] S. Chen and R. R. Yang, "Pose trainer: Correcting exercise posture using pose estimation," 2020, *arXiv:2006.11718*.

[17] B. Dittakavi, D. Bavikadi, S. V. Desai, S. Chakraborty, N. Reddy, V. N. Balasubramanian, B. Callepalli, and A. Sharma, "Pose tutor: An explainable system for pose correction in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3539–3548.

[18] A. Pardos, M. Tziomaka, A. Menychtas, and I. Maglogiannis, "Automated posture analysis for the assessment of sports exercises," in *Proc. 12th Hellenic Conf. Artif. Intell.*, Sep. 2022, pp. 1–9.

[19] R. R. Kanase, R. Ravikant, and P. Singh, "Pose estimation and correcting exercise posture," in *Proc. ITM Web Conf.*, vol. 33, 2021, p. 03031.

[20] A.-L. Liu and W.-T. Chu, "A posture evaluation system for fitness videos based on recurrent neural network," in *Proc. Int. Symp. Comput., Consum. Control (ISC)*, Nov. 2020, pp. 185–188.

[21] D. Swain, P. Roy, and S. Mukherjee, "Deep learning models for yoga pose monitoring," *Algorithms*, vol. 15, no. 11, p. 403, 2022.

[22] C. Militaru, M.-D. Militaru, and K.-I. Benta, "Physical exercise form correction using neural networks," in *Proc. Companion Publication Int. Conf. Multimodal Interact.*, Oct. 2020, pp. 240–244.

[23] J. W. L. Keogh and P. W. Winwood, "The epidemiology of injuries across the weight-training sports," *Sports Med.*, vol. 47, no. 3, pp. 479–501, Mar. 2017.

[24] D. A. Bonilla, L. A. Cardozo, J. M. Vélez-Gutiérrez, A. Arévalo-Rodríguez, S. Vargas-Molina, J. R. Stout, R. B. Kreider, and J. L. Petro, "Exercise selection and common injuries in fitness centers: A systematic integrative review and practical recommendations," *Int. J. Environ. Res. Public Health*, vol. 19, no. 19, p. 12710, Oct. 2022.

[25] X. Zhao and G. Liang, "Optimizing electric vehicle charging schedules and energy management in smart grids using an integrated GA-GRU-RL approach," *Frontiers Energy Res.*, vol. 11, Sep. 2023, Art. no. 1268513.

[26] *Squat-Depth Image Dataset*. Roboflow. Accessed: Sep. 10, 2023. [Online]. Available: https://universe.roboflow.com/nejc-graj-1na9e/squat-depth/dataset/14

[27] *SDT Image Dataset*. Roboflow. Accessed: Sep. 10, 2023. [Online]. Available: https://universe.roboflow.com/isbg/sdt/dataset/4

[28] *HumonBody1 Computer Vision Project*. Roboflow. Accessed: Sep. 10, 2023. [Online]. Available: https://universe.roboflow.com/multilabel-ppe/humonbody1

[29] *Faller Image Dataset*. Roboflow. Accessed: Sep. 10, 2023. [Online]. Available: https://universe.roboflow.com/jiangsu-ocean-universit/faller

[30] *Silhouettes of Human Posture*. Kaggle. Accessed: Sep. 14, 2023. [Online]. Available: https://www.kaggle.com/datasets/deepshah16/silhouettes-of-human-posture

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[32] *Ultralytics, YOLOv5*. Accessed: Jul. 1, 2024. [Online]. Available: https://github.com/ultralytics/yolov5

[33] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverria, "YOLOv5 vs. YOLOv8: Performance benchmarking in wildfire and smoke detection scenarios," *J. Image Graph.*, vol. 12, no. 2, pp. 127–136, 2024.

[34] A. S. Geetha and M. Hussain, "A comparative analysis of YOLOv5, YOLOv8, and YOLOv10 in kitchen safety," 2024, *arXiv:2407.20872*.

[35] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C. L. Chang, M. Yong, J. Lee, and W. T. Chang, "MediaPipe: A framework for perceiving and processing reality," in *Proc. 2rd Workshop Comput. Vis. AR/VR IEEE Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1–4. [Online]. Available: https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff310a69061a71cbd7c/1578307584840/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf

[36] S. Garg, A. Saxena, and R. Gupta, "Yoga pose classification: A CNN and MediaPipe inspired deep learning approach for real-world application," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 12, pp. 16551–16562, Dec. 2023.

[37] C.-C. Hsiao, P.-C. Yu, R.-G. Lee, and H. Jiang, "Deep-learning LSTM mechanism and wearable devices based virtual fitness-coach information system for barbell bench press," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Toronto, ON, Canada, Oct. 2020, pp. 3880–3885.

[38] H. Zhou, Y. Gao, W. Liu, Y. Jiang, and W. Dong, "Posture tracking meets fitness coaching: A two-phase optimization approach with wearable devices," in *Proc. IEEE 17th Int. Conf. Mobile Ad Hoc Sensor Syst. (MASS)*, Delhi, India, Dec. 2020, pp. 524–532.

[39] J. Padulo, P. J. Ardigò, and A. G. Leal, "Concentric and eccentric: Muscle contraction or exercise?" *J. Hum. Kinetics*, vol. 37, no. 1, pp. 5–6, 2013.

[40] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, Colorado Springs, CO, USA, Jun. 2011, pp. 1297–1304.

[41] L. Büker, M. Mertens, and T. Röder, "Camera-based tracking and evaluation of the performance of a fitness exercise," in *Proc. Int. Conf. Health Informat. (HEALTHINF)*, 2023, pp. 489–496.

[42] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge, U.K.: Cambridge Univ. Press, 2004.

[43] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, and M. Lucic, ''MLP-mixer: An all-MLP architecture for vision,'' in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, Jan. 2021, pp. 24261–24272.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, Jun. 2017, pp. 5998–6008.

[45] M. J. A. Rasool, S. Ahmad, U. Sabina, and T. K. Whangbo, ''KONet: Toward a weighted ensemble learning model for knee osteoporosis classification,'' *IEEE Access*, vol. 12, pp. 5731–5742, 2024.

YEONG-MIN KO received the B.S. degree in computer engineering from Dongguk University WISE Campus, in 2024. He is currently pursuing the master's degree with the Visual Intelligence and Perception Laboratory, Pusan National University. His research focuses on adversarial attacks to disrupt malicious deepfake generation, advanced exercise posture evaluation methods, and innovative techniques for piano performance assessment. He actively works on projects that integrate AI with practical applications, bridging theory, and measurable outcomes.

AZIZ NASRIDINOV received the B.Sc. degree from Tashkent University of Information Technologies, Uzbekistan, in 2006, and the M.Sc. and Ph.D. degrees in computer engineering from Dongguk University, in 2009 and 2012, respectively. He is currently a Professor with the Department of Computer Science, Chungbuk National University, South Korea. He has published over 50 papers in various high-ranked international journals and conferences. His research interests include traditional databases, big data analytics with machine learning, and computer vision. He has also served as a Program Committee Member and a Co-Organizer for numerous top-tier conferences, including ACM SAC, IEEE Big Data, AAAI, and ACM CHI, and also served on the editorial board for several international journals.

SO-HYUN PARK received the B.S. degree in piano, the M.S. degree in multimedia science, and the Ph.D. degree in IT engineering from Sookmyung University, in 2014, 2016, and 2020, respectively. She was a Senior Researcher with the Big Data Using Research Center, Sookmyung Women's University, from 2020 to 2022. She is currently an Assistant Professor with the Department of Computer Engineering, Dongguk University WISE Campus, South Korea. Her research interests include data analytics, machine learning, pose classification, pose estimation, pose evaluation, object detection, and speech recognition.