Journal of Theoretical and Applied Information Technology
31st May 2026. Vol.104. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OPTIMIZING FEEDBACK IN DIGITAL FITNESS ENVIRONMENTS: CAN THE INTEGRATION OF REAL-TIME POSE ESTIMATION AND RETRIEVAL-AUGMENTED GENERATION ARCHITECTURES IMPROVE USER ACCURACY?

LEONARDO CARLOS VELIZ ARCE[1], JAIDER PARAGUAY JUNCO [2], BRAD JHOMERS ROSALES TAPIA [3], ROSARIO DELIA OSORIO CONTRERAS [4]

[1]Department of engineering, Perú
[2]Department of engineering, Perú
[3]Department of engineering, Perú
[4]Department of engineering, Perú
E-mail: [1]73057970@continental.edu.pe, [2]76369393@continental.edu.pe, [3]73218792@continental.edu.pe

ABSTRACT

Physical inactivity and the lack of technical supervision in home-based training lead to increased injury risks and reduced exercise effectiveness. This study addresses this problem by developing FitPoseTracker, an intelligent platform that integrates computer vision and conversational artificial intelligence to optimize physical training. The proposed system, FitPoseTracker, performs real-time posture analysis and automatic repetition counting using human pose estimation and vectorial geometry, while a specialized conversational assistant based on a Retrieval-Augmented Generation (RAG) architecture provides contextualized technical guidance in natural language. The posture analysis module leverages MediaPipe to detect anatomical keypoints and compute joint angles, enabling accurate movement state classification. System performance was evaluated using a two-factor ANOVA to analyze the effects of exercise type and user experience level on repetition-counting accuracy. Results indicate that exercise type significantly influences accuracy ($F(2,141)=166.98$, $p<2e{-}16$), whereas user experience level shows no significant effect, demonstrating consistent performance across different skill levels. Squats achieved the highest accuracy, followed by sit-ups and push-ups. Additionally, the conversational assistant was evaluated through automated testing on 50 representative queries, achieving high accuracy (0.92), recall (0.78), and F1-score (0.84), with strong contextual coherence. Overall, the results confirm that integrating real-time pose analysis with Large Language Models (LLMs) effectively bridges the gap between movement quantification and specialized technical guidance.

Keywords: *Artificial Intelligence; Human Pose Estimation; Computer Vision; Fitness Training; Retrieval-Augmented Generation; Conversational Assistant*

1. INTRODUCTION

The promotion of physical activity has become imperative due to its direct impact on reducing non-communicable diseases and improving global health **[1].** In this context, the digital revolution has transformed sports, allowing technology to optimize both performance and access to training **[2]**. A fundamental pillar in this evolution is real-time human pose estimation, where tools such as OpenPose have proven crucial for machines to understand human movement non-parametrically **[3].** These technologies enable the development of tracking applications in gyms that identify the user's posture to offer personalized corrections **[4].**

Nonetheless, the expansion of solutions like tele-exercise reveals a gap in the evaluation of training parameters for healthy individuals **[5].** Furthermore, the adoption of Artificial Intelligence (AI) in sports practice raises ethical challenges related to athlete autonomy and the preservation of natural talent **[6].** Therefore, the design of fitness applications must consider cultural factors to improve the persuasion and effectiveness of interventions **[7].** Likewise, rigorous metrological validations of 3D tracking systems are necessary to guarantee their precision in both laboratory and field conditions **[8].**

The significance of this issue lies in the global rise of non-communicable diseases. While tele-exercise has expanded, the lack of an 'expert eye' in remote settings is a real problem that leads to poor biomechanical execution. This research contributes to the international field of digital health by proposing a scalable solution that does not require expensive 3D hardware.

Beyond computer vision, physical exercise recognition has also been addressed using chest accelerometers and LSTM-type neural networks **[9].** These innovations are especially valuable for home training, facilitating access to structured routines for diverse populations **[10].** Modern training analysis systems now integrate posture feedback and repetition grading through frameworks such as MediaPipe [**11].** This approach allows for the quantification of exercise form to prevent muscle injuries, acting as an automated personal trainer **[12].**

Technically, gesture recognition requires skeletonization processes to encode movements before interpretation **[13].** Improvements in multi-person detection algorithms have allowed these systems to adapt to complex environments and occlusions **[14].** Machine learning applied to health currently offers a practical guide for integrating sensors and the Internet of Things (IoT) into officiating and performance enhancement **[15].** Recent applications use keypoint estimation to measure joint angles and predict exercise stages **[16].** Specifically, AI trainers have been developed for shoulder exercises that classify movement and calculate execution speed **[17],** as well as activity recognition systems based on multi-view videos for a detailed description of joint **configuration [18].**

In high-complexity exercises, such as the deadlift, skeletal analysis through deep learning allows for technique monitoring and risk reduction **[19].** The efficiency of these systems has improved with single-network approaches for full-body pose estimation, optimizing real-time performance **[20].** Tools like AlphaPose have advanced multi-person joint tracking with high precision **[21].** In parallel, smartphone sensors have proven effective for daily activity recognition using convolutional neural networks **[22].** Mobile safety applications utilize fuzzy inference systems to classify sports performance and alert users to potential injuries **[23].** Complementarily, sensors like Kinect offer non-invasive capture for motor rehabilitation in home settings **[24].** Finally, the combination of detection and temporal tracking in video sequences allows for error correction in scenarios with overlapping individuals **[25].**

Despite these advances, there is a need to integrate biomechanical analysis with intelligent assistance systems that facilitate access to specialized technical information. The present work introduces FitPoseTracker, an integrated system that combines posture analysis via vector geometry with a conversational assistant based on the Retrieval-Augmented Generation (RAG) architecture. The study validates the system's precision through a two-way ANOVA analysis, evaluating the impact of exercise type and user experience. The structure of the article continues with the methodology in Section II, followed by the results and technical discussion.

The central problem this paper strives to resolve is the disconnect between automated repetition counting and the provision of contextualized, expert-level feedback in real-time.

2. METHODOLOGY

The fitness web platform was developed using the MERN technology stack (MongoDB, Express.js, React.js, and Node.js), selected for its capability to create scalable, real-time web applications **[26].** The SCRUM agile methodology was adopted for project management, implementing development iterations in 2–3-week sprints, with continuous reviews of the product backlog and user stories prioritized based on value for the end user **[27].** Two specialized artificial intelligence systems were developed and integrated to enrich the user experience:

2.1 FitPoseTracker System for Real-Time Exercise Analysis

FitPoseTracker constitutes an integrated web system for the automatic analysis of physical exercises using computer vision techniques and real-time video processing. The system implements a distributed client-server architecture that integrates the MediaPipe pose estimation model from Google Research **[28]** for the detection and tracking of anatomical keypoints, allowing for automated exercise repetition counting and the generation of instantaneous visual feedback.

Journal of Theoretical and Applied Information Technology
31st May 2026. Vol.104. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The system architecture relies on three main components. First, a React.js web interface manages user interaction and real-time data visualization. Second, a Flask processing server executes the motion analysis algorithms. Finally, a bidirectional communication module synchronizes the state between these components using HTTP protocols and Server-Sent Events (SSE).

2.1.1 Frontend Design

The frontend was developed using React.js, implementing the functional component paradigm with hooks for reactive state management **[29].** The main component, BodyTrackAnalyzerComponent, encapsulates all user interaction logic and exercise session control.

2.1.2 Backend Communication

Frontend-backend communication is established through a system of RESTful endpoints optimized for real-time processing **[30].** The frontend consumes backend endpoints to display processed video in real-time, integrating SSE events that continuously transmit the movement state **[31]** ("up" or "down"). It utilizes the /toggle_landmarks endpoint to dynamically switch the visualization of detected joints, allowing the user to customize the interface according to their training preferences.

2.1.3 Backend Development

The backend implements a robust Flask server that processes frontend requests in real-time. CORS (Cross-Origin Resource Sharing) was enabled to allow secure communication with the React frontend. OpenCV is used for efficient camera frame capture with optimized processing, while MediaPipe is responsible for detecting body poses and calculating relevant joint angles specific to each exercise (e.g., shoulder, elbow, and wrist for push-ups; or hip, knee, and ankle for squats).

The angle calculation function implements **vector geometry** to precisely determine the state of body movement. **Figure 1** illustrates the applied vector algorithm. The system calculates vectors between three consecutive anatomical points and applies inverse trigonometric functions to determine the precise joint angle. This mathematical implementation guarantees precise measurements regardless of the user's orientation relative to the camera.

```
def calculate_angle(a, b, c):
    a, b, c = np.array([a.x, a.y]), np.array([b.x, b.y]), np.array([c.x, c.y])
    ba, bc = a - b, c - b
    cosine_angle = np.dot(ba, bc) / (np.linalg.norm(ba) * np.linalg.norm(bc) + 1e-6)
    angle = np.arccos(np.clip(cosine_angle, -1.0, 1.0))
    return np.degrees(angle)
```

Figure 1: Function to determine the user's movement angle through vector calculation

2.1.4 Repetition Counting Logic

The backend determines the movement state ("up" or "down") based on the detected angle. When a transition from "down" to "up" is detected, the repetition counter is incremented. The backend can toggle the visualization of the joints according to the user's preference.

- **Down**: Initial or descent position (angle > upper_threshold).
- **Up**: Extension or elevation position (angle < lower_threshold).
- **Transition**: Intermediate state during the phase change.

Figure 2 illustrates the complete real-time processing flow, showing video capture, pose detection through MediaPipe, joint angle calculation, and the transmission of results to the frontend for immediate visualization.

Figure 2: Real-time processing for postural analysis

2.2 Specialized Conversational AI System

The conversational AI system was designed as a specialized fitness virtual assistant, implementing a Retrieval-Augmented Generation (RAG) architecture that combines contextual information retrieval with natural language generation. The system integrates three main components: a specialized document processing module, a semantic retrieval engine, and a natural language generative model **[32]**. **Figure 3** presents the conversational assistant interface, designed with usability and user experience principles, featuring

Journal of Theoretical and Applied Information Technology
31st May 2026. Vol.104. No.10

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

an intuitive chat that allows for natural language queries and structured responses with specialized technical information.

Figure 3: Interface of the specialized conversational assistant.

2.2.1 Base Model Selection

A systematic comparative evaluation was conducted among three Large Language Models (LLMs): mistral-nemo-instruct (NVIDIA), llama-3.1-8b-instruct (Meta), and gpt-3.5-turbo (OpenAI). The evaluation was based on a weighted metrics framework including four critical dimensions for specialized fitness applications, as detailed in Table **1**.

Table 1: Comparative evaluation criteria for specialized LLM selection.

Criterion	Weight	Description
Coherence	30%	Logical consistency and narrative fluency in responses
Support	30%	Ability to base responses on specific knowledge
Assurance	20%	Reliability and accuracy in critical information
Functionality	20%	Response speed and system stability

Technology Stack

Backend: Python 3.9+ with Flask as the web framework

- RAG Framework: LangChain v0.0.330 for component orchestration
- Vector Database: ChromaDB for semantic storage and retrieval
- Embeddings Model: sentence-transformers/all-MiniLM-L6-v2
- LLM Model: mistral-nemo-instruct via NVIDIA NIM API

2.2.2 RAG Pipeline Architecture

Figure 4 illustrates the complete RAG pipeline architecture, showing the flow from the user query to the generation of the final response **[33].** The process begins with query processing using semantic embeddings, continues with the retrieval of relevant documents from the ChromaDB vector database, and culminates with the generation of contextualized responses using the selected LLM model.

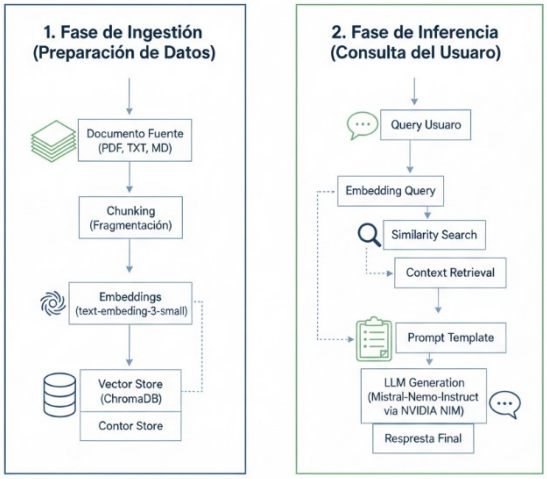

Figure 4: RAG system pipeline

2.2.3 Specialized Conversational AI System Flow

Figure 5 illustrates the complete system flow: from the input of the query, through the processing of the questions, to the parameterization and delivery of the final response to the user.

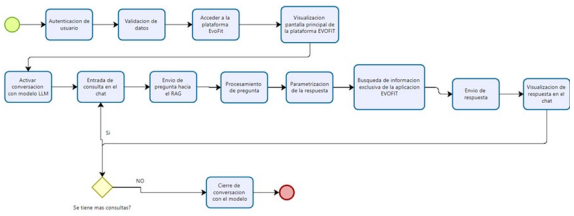

Figure 5: Integrated Process Flow: Specialized Conversational AI System

2.2.4 Specialized Conversational AI System Flow

The system was trained using a specialized corpus of 15 structured documents (Exercise routines, Exercise descriptions, Nutritional plans).

2.2.5 Document Processing

- Semantic Chunking: Division of documents into 512-token fragments with a 50-token overlap to preserve context.

- Vectorization: Generation of 384-dimensional embeddings using a sentence-transformer model.
- Indexing: Storage in ChromaDB with structured metadata for contextual filtering.

2.2.6 API Query Module

The module implements a robust communication mechanism with the RAG system, incorporating error handling, automatic retries, and load balancing to ensure high availability. **Figure 6** details the communication architecture, showing the request flow from the user interface, processing through the RAG pipeline, and the delivery of structured responses with full source traceability.

```
def hacer_consulta(self, pregunta, max_retries=3, timeout=30):

    # Mecanismo de reintentos con backoff exponencial
    for attempt in range(max_retries):
        try:
            response = requests.post(
                self.ngrok_url,
                json={"message": pregunta},
                timeout=timeout
            )
            # Manejo de códigos HTTP
            if response.status_code == 200:
                return response.json().get("response", "")

        # Manejo específico de errores
        except requests.Timeout:
            print(f"Timeout en intento {attempt + 1}")
        except requests.ConnectionError:
            print("Error de conexión")

        # Backoff exponencial
        if attempt < max_retries - 1:
            wait_time = 2 ** (attempt + 1)
            time.sleep(wait_time)

    return None
```

Figure 6: Communication mechanism with the RAG system

3. VALIDATION AND EVALUATION

3.1 FitPoseTracker System Evaluation

To determine whether the user's experience level and the type of exercise influence the accuracy of the repetition counting model, a two-way analysis of variance (ANOVA) with interaction was applied **[34].** The independent variables were:

- **Factor A (Experience Level):** Categorical variable with three levels (1 = basic, 2 = intermediate, 3 = advanced).
- **Factor B (Exercise Type):** Categorical variable with three levels (1 = sit-ups, 2 = squats, 3 = push-ups).
- **Dependent Variable:** Model accuracy.

3.1.1 Statistical Model

The two-way ANOVA model with interaction is mathematically expressed through the decomposition of the total variance into its specific components. The fundamental formula for calculating the total sum of squares constitutes the basis for the subsequent decomposition of variance among the different factors analyzed.

$$ss_{tot} = \sum\sum\sum(x_{mij} - \bar{x})^2 \quad [34]$$

Where:

ss_{tot} = Total sum of squares

x_{mij} = Observed value in group **ij** for observation **m**

$\bar{x}$ = Grand mean value

The decomposition of variance was performed by identifying the specific contributions of each factor and their interaction. The mathematical formulas for each component of variation allow for the quantification of the effect of the user's experience level, the type of exercise, their interaction, and the residual error, respectively.

$$ss_A = n * q\sum(A_i - \bar{x})^2 \quad [34]$$

$$ss_B = n * p\sum(B_i - \bar{x})^2 \quad [34]$$

$$ss_{AB} = ss_{btw} - ss_A - ss_B \quad [34]$$

$$ss_{err} = \sum\sum\sum(X_{mij} - AB_{ij})^2 \quad [34]$$

This mathematical decomposition allows for the identification of what proportion of the total variance can be attributed to each specific factor, facilitating the interpretation of the experimental results.

3.1.2 Statistical Hypotheses

For the exercise type factor:

- **H_0:** There are no significant differences in accuracy among the different types of exercises
- **H_1:** There are significant differences in accuracy between at least two types of exercises.

For the experience level factor:

- **H_0:** There are no significant differences in accuracy among the different user experience levels.
- **H_1:** There are significant differences in accuracy between at least two user experience levels.

3.1.3 Post-hoc Analysis

Tukey's HSD test (Tukey's Honestly Significant Difference test) was applied to specifically identify which groups exhibit statistically significant differences from each other when the ANOVA indicates significant main

effects. This test controls the family-wise error rate, maintaining statistical reliability across multiple comparisons **[34].**

3.2 Evaluation of the Specialized Conversational AI System

3.2.1 Response Quality Verification

The implemented evaluation system consists of an automated testing framework that validates response quality through 50 simultaneous test cases. The system architecture is composed of:

- **Lexical Validation:** Search for specific keywords using basic NLP processing with text normalization and stopword removal.
- **Semantic Validation:** Implementation of fuzzy matching with a similarity threshold ≥ 0.8 using the Levenshtein algorithm to capture acceptable variations in technical terminology.
- **Structural Validation:** Automatic differentiation between simple responses (requirement: 100% of keywords) and structured lists (minimum requirement: 30% of keywords).

3.2.2 Response Quality Verification

The system calculates four fundamental metrics for a comprehensive performance evaluation:

Accuracy: The system measures the proportion of completely correct responses relative to the total number of evaluated queries. This metric is crucial for specialized information systems where precision is fundamental.

$$Accuracy = \frac{\text{correct classifications}}{\text{all classifications}} \quad [35]$$

Recall: This metric evaluates the system's ability to retrieve relevant information from the knowledge corpus. In RAG contexts, it indicates what proportion of pertinent information is effectively utilized in the responses.

$$Recall = \frac{\text{Relevant Results Retrieved}}{\text{Total Relevant Results}} \quad [36]$$

F1-Score: To obtain a balanced metric that combines both precision and recall, the F1-Score is calculated using the harmonic mean of both metrics.

$$F1\ Score = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad [37]$$

Contextual Coherence: The contextual coherence evaluation analyzes the grammatical structure and logical flow through syntactic pattern analysis and discursive connectors. This metric is especially relevant for conversational assistants where dialogue naturalness is crucial **[38].**

$$Context\ Precision = \frac{|\text{Relevant Sentences in Retrieved Context}|}{|\text{Total Retrieved Sentences}|} \quad [38]$$

Level of Detail: To evaluate the informational density of the generated responses, a calculation system was developed to quantify the level of technical detail present in each LLM model response.

3.2.3 Calculation Methodology

- **Identification of technical elements**: Specialized terms, specific numerical data, references to methodologies, and procedural descriptions are tallied.
- **Length normalization:** The count is normalized by the total length of the response to avoid bias toward longer responses.
- **Categorical weighting:** Differentiated weights are applied according to the type of information:
  - Specialized technical terms: weight 0.4
  - Specific quantitative data: weight 0.3
  - Methodological references: weight 0.2
  - Detailed practical examples: weight 0.1

3.2.4 Evaluation Scale

The result is scaled within a range from 0.0 to 1.0, where:

- **0.0-0.3:** Basic/general response
- **0.4-0.6:** Moderately detailed response
- **0.7-1.0:** Highly specialized response

3.2.5 Testing Protocol

Fifty simultaneous queries were executed covering different categories:

- **Basic queries (20):** General information about exercises.

- **Specific queries (15):** Personalized routines by objective.
- **Complex queries (10):** Combination of exercise and nutrition.
- **Edge-case queries (5):** Boundary situations and special cases.

3.2.6 Operational Definitions for Metrics:

- **True Positives (TP):** Responses that contain the requested information and are factually correct.
- **False Positives (FP):** Responses that appear relevant but contain incorrect or unrequested information.
- **False Negatives (FN):** Cases where the system failed to provide available and relevant information from the corpus.

4. RESULTS

4.1 FitPoseTracker

The effect of the user's experience level and the type of exercise on the accuracy of the repetition counting model was evaluated using a two-way ANOVA. The results of the statistical analysis are presented in **Table 2**, which displays the degrees of freedom, sums of squares, mean squares, F-values, and statistical significance for each source of variation.

Table 2: Results of the Two-Way ANOVA Analysis

Source of Variation	Df	Sum Sq	Mean Sq	F-value	(Pr (>F))	Significance
Experience Level	2	20	10.2	1.00	0.370	-
Exercise Type	2	3412	1706.0	166.98	<2e-16	***
Experience Level * Exercise Type	4	33	8.3	0.81	0.521	-
Residuals	141	1441	10.2	-	-	-

We found statistically significant differences in the average precision of the model only by exercise type **(F(2,141)=166.98, p<2e-16);** the experience level factor did not show a significant effect **(F(2,141)=1.00, p=0.370),** and the interaction between these factors was also not significant **(F (4,141) = 0.81, p = 0.521).** Since only the exercise type showed significant effects, a Tukey post-hoc test was performed to identify which types of exercise differ in the model's precision.

The comparisons revealed statistically significant differences among all exercise types:

- **Squats vs. Sit-ups:** difference = 2.58%, $p < 0.001$, IC 95% = [1.07, 4.10]
- **Push-ups vs. Sit-ups:** difference = -8.58%, $p < 0.001$, IC 95% = [-10.09, -7.06]
- **Push-ups vs. Squats:** difference = -11.16%, $p < 0.001$, IC 95% = [-12.67, -9.64]

Squats achieved the highest model precision, followed by sit-ups, while push-ups showed the lowest precision. All differences are statistically significant. No post-hoc comparisons were performed for the experience level since it did not show significant effects ($p = 0.370$). **Figure 7** graphically illustrates the precision distribution of the FitPoseTracker system by exercise type, where the significant differences between categories and the variability within each exercise group are clearly observed.

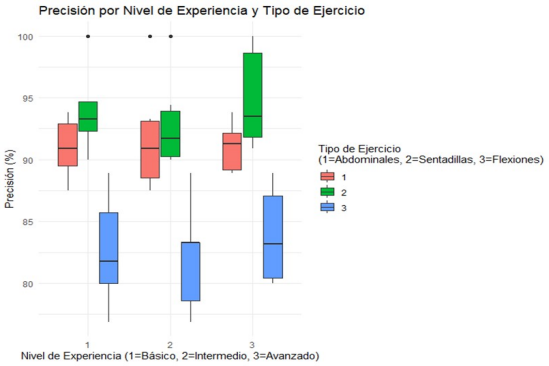

Figure 7: Precision distribution of the FitPoseTracker system by exercise type showing significant differences between categories (n=150).

Figure 8 presents the ANOVA diagnostic plots that validate compliance with the statistical assumptions necessary for the analysis's validity. The plots include: (a) normality of residuals via Q-Q plot, (b) homoscedasticity of variances, (c) independence of residuals, and (d) detection of outliers. Compliance with these assumptions confirms the statistical robustness of the results obtained.

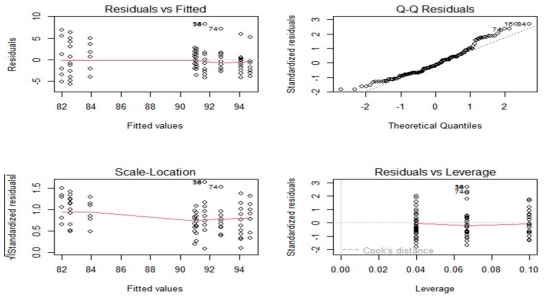

Figure 8: ANOVA diagnostic plots showing normality of residuals, homoscedasticity, and validation of statistical assumptions.

In applied terms, the results obtained demonstrate that the FitPoseTracker system significantly favors the automation of physical training monitoring by showing high precision in exercises with lower biomechanical complexity, such as squats and sit-ups. This behavior is especially beneficial for unsupervised training environments where users lack immediate technical feedback. Furthermore, the user's experience level does not significantly influence the system's precision. This suggests that the platform is effective for both beginners and advanced users, thereby reducing the reliance on prior technical knowledge. Taken together, these results support the system's utility as an accessible, scalable tool oriented toward improving the quality of physical training, contributing to the prevention of postural errors and the efficient use of artificial intelligence technologies in the digital fitness field.

4.2 Results of the Specialized Conversational AI System

Table 3 presents the results of the comparative evaluation between the three candidate LLM models, including weighted scores according to the established criteria and the resulting total score.

Table 3: Results of the Comparative Evaluation Between Specialized LLM Models

LLM Model	mistral-nemo-instruct	llama-3.1-8b-instruct	gpt-3.5-turbo
Coherence (30%)	7.0 (2.1)	7.0 (2.1)	6.0 (1.8)
Support (30%)	8.0 (2.4)	6.0 (1.8)	7.0 (2.1)
Assurance (20%)	9.0 (1.8)	8.0 (1.6)	8.0 (1.6)
Functionality (20%)	8.0 (1.6)	6.0 (1.2)	6.0 (1.2)
Total Score	7.9	6.7	6.9

The mistral-nemo-instruct model was selected for its superior performance, particularly in the dimensions of technical support (8.0/10) and information assurance (9.0/10), which are critical aspects for a specialized fitness assistant.

The automated evaluation of the RAG system yielded highly satisfactory results across 50 representative test queries. **Table 4** details the performance metrics obtained, along with their interpretation in the context of specialized RAG applications.

Table 4: Performance metrics of the RAG system in automated evaluation.

Metric	Value	Interpretation
Accuracy	0.92 (92%)	46 out of 50 queries met the established quality criteria
Recall	0.78 (78%)	78% of relevant information was successfully retrieved
F1-Score	0.84 (84%)	Optimal balance between accuracy and coverage: (2×0.92×0.78)/(0.92+0.78)
Coherence	0.87 (87%)	High consistency in grammatical structure and logical flow
Level of Detail	0.76 (76%)	High density of specialized technical information

The results demonstrate that the developed system achieves competitive performance in accuracy and F1-score, maintaining an appropriate balance between precision and completeness in the generated responses. The slight reduction in accuracy compared to general systems is compensated by the high level of specialization and technical detail achieved.

5. DISCUSSION

The work of Cao et al. (2021) with OpenPose introduced an innovative approach to real-time human pose estimation through the use of Part Affinity Fields (PAFs), which allow for the simultaneous association of body joints of multiple individuals within the same scene **[3].** This model demonstrated that a bottom-up architecture can achieve high levels of precision without depending on prior person detection, thus optimizing system scalability in complex scenarios. In the present study, this principle of structured detection is revisited in the development of FitPoseTracker, which utilizes Google's MediaPipe model as a lighter and more efficient alternative for real-time

web applications. While OpenPose prioritizes precision in multi-person environments, FitPoseTracker directs its design toward the individual physical training sphere, where reduced latency and immediate feedback are determining factors.

The findings of this study highlight significant advancements but also specific discrepancies when compared to the current state-of-the-art (SOTA) in digital fitness. Cao et al. (2021) demonstrated that OpenPose achieves high precision using Part Affinity Fields for multi-person environments, but its heavy computational cost limits its web accessibility. Conversely, our implementation of MediaPipe prioritizes low-latency processing for individual web users, explicitly accepting a trade-off in precision during partial occlusions.

Furthermore, while recent studies like Dong and Du (2024) optimized YOLOv8 architectures for dynamic scenarios, they primarily address the visual dimension of tracking. Similarly, Karmakar et al. (2024) developed the TrainERAI system for postural correction but lacked an advanced natural language interaction mechanism. A major shortcoming in the current literature is the disconnect between kinematic data and contextualized user feedback. FitPoseTracker bridges this gap through its RAG-based conversational assistant, achieving a 0.92 accuracy in specialized technical guidance. However, a notable drawback of our system, as evidenced by the lower accuracy in push-ups compared to squats, is its reliance on 2D vector geometry. This geometric approach struggles with complex biomechanical movements that are not perpendicular to the camera plane, identifying an area where 3D SOTA models outperform our current implementation.

6. CONCLUSIONS

In conclusion, this research successfully addresses the disconnect between automated repetition counting and contextualized technical guidance by integrating real-time pose estimation with a RAG-based conversational assistant. The core argument of this work is that physical training optimization in digital environments requires both accurate kinematic quantification and specialized, accessible educational feedback.

The two-way ANOVA results demonstrated that while exercise type significantly impacts tracking accuracy ($F(2,141)=166.98$, $p<2e-16$), the system remains robust across all user experience levels, effectively democratizing access to expert-level training. Squats achieved the highest accuracy, validating the model's efficiency for exercises with clear vertical displacement. Furthermore, the conversational assistant validated this synergy, achieving an F1-score of 0.84 and providing highly coherent technical support (0.87 coherence score).

Despite these contributions, the shortcomings of 2D vector geometry in complex exercises highlight open issues for the scientific community. Based on the limitations of this work, future research directions and pending issues include:

Multi-perspective and 3D Tracking: Developing lightweight algorithms to infer 3D depth from single-camera views to mitigate parallax errors and improve accuracy in complex biomechanical exercises, such as push-ups.

Multi-modal Integration: Incorporating wearable IoT devices and biometric sensors to correlate kinematic tracking data with physiological responses, providing a holistic view of user health.

Latency Optimization in RAG Systems: Investigating advanced semantic caching and edge computing techniques to further reduce the inference time of Large Language Models on low-end mobile devices."

REFERENCES:

[1] Organización Panamericana de la Salud, "Actividad Física", PAHO, Washington DC (USA), 2023, https://www.paho.org/es/temas/actividad-fisica [Accessed: 15/12/2024].

[2] M.A. García-López and L. Rodríguez-Sánchez, "Impacto de la tecnología sobre el deporte: una revolución digital", Revista Tecnología y Deporte (Vol. 15, No. 3), 2023, pp. 45-62.

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S.E. Wei, and Y.A. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", IEEE Transactions on Pattern Analysis and Machine Intelligence (Vol. 43, No. 1), 2021, pp. 172-186.

[4] A.H. Babu, S. Shanthakumar, and G. Malarselvi, "Live gym tracker using artificial intelligence", 4th International Conference on Internet of Things 2023 (ICIoT2023), AIP Publishing, 2024, p. 020255.

[5] A. Fucarino et al., "Emerging Technologies and Open-Source Platforms for Remote Physical Exercise: Innovations and Opportunities for Healthy Population—A Narrative Review", Healthcare (Vol. 12, No. 15), 2024, p. 1466.

[6] I. López-Fernández, J. García-Unanue, and J. Sánchez-Sánchez, "La Inteligencia Artificial en el deporte: Problemas y principios para su adopción", Revista Española de Educación Física y Deportes (No. 429), 2022, pp. 67-78.

[7] K. Oyibo, A.H. Olagunju, B. Olabenjo, I. Adaji, R. Deters, and J. Vassileva, "BEN'FIT: Design, Implementation and Evaluation of a Culture-Tailored Fitness App", Proc. 27th ACM Conf. User Model Adapt Pers., 2019, pp. 161-166.

[8] M. Zago, M. Luzzago, T. Marangoni, M. De Cecco, M. Tarabini, and M. Galli, "3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision", Front Bioeng Biotechnol (Vol. 8), 2020, p. 181.

[9] A. Hussain, K. Zafar, A.R. Baig, R. Almakki, L. Alsuwaidan, and S. Khan, "Sensor-Based Gym Physical Exercise Recognition: Data Acquisition and Experiments", Sensors (Vol. 22, No. 7), 2022, p. 2489.

[10] I. Khaghani Far, S. Nikitina, M. Baez, F. Casati, and A. Sarigiannis, "Fitness Applications for Home-Based Training", IEEE Pervasive Comput (Vol. 15, No. 4), 2016.

[11] T. Prateek, A. T, and R. B, "AI-Powered Workout Analysis Application for Posture Feedback and Repitition Grading", Proc. Second Int. Conf. Inventive Comput. Inform. (ICICI), IEEE, 2024.

[12] K.R. Sowmia, T. Jayaganeshan, F.M. Abraar Khan, S. Madhesh, and S. Kabilesh, "An Artificial Intelligence Approach to Quantifying Exercise Form for Optimal Performance and Injury Prevention", Proc. Third Int. Conf. Comput. Commun. Netw., Springer (Singapore), 2024, pp. 639-647.

[13] S. Kuganesan, A. Thusyanthan, C.N. Joseph, and S. Kokulakumaran, "Skeletonization in a Real-Time Gesture Recognition System", Proc. 2010 Int. Conf. Inform. Autom. Sustain. (ICIAfS), IEEE, 2010, pp. 213-218.

[14] Z. Shu, P. Wang, and W. Zhan, "The Research and Implementation of Human Posture Recognition Algorithm via OpenPose", Proc. 2020 2nd Int. Conf. Artif. Intell. Adv. Manuf. (AIAM), 2020, pp. 90-94.

[15] K. Ashley, "Applied Machine Learning for Health and Fitness: A Practical Guide to Machine Learning with Deep Vision, Sensors and IoT", Apress (USA), 2020.

[16] N. Faujdar, S. Saraswat, and S. Sharma, "Human Pose Estimation using Artificial Intelligence with Virtual Gym Tracker", Proc. 6th Int. Conf. Inform. Syst. Comput. Netw. (ISCON), IEEE, 2023, pp. 1-5.

[17] S.E.H. Mahmoud and Z.A.E. Taha, "AI Personal Trainer for Lateral Raises and Shoulder Presses Exercises", Proc. 2023 Intell. Methods Syst. App. (IMSA), IEEE, 2023.

[18] M.B. Holte, C. Tran, M.M. Trivedi, and T.B. Moeslund, "Human Pose Estimation and Activity Recognition From Multi-View Videos", IEEE J Sel Top Signal Process (Vol. 6, No. 5), 2012, pp. 538-552.

[19] B. Kaldarova, A. Toktarova, and R. Abdrakhmanov, "Enhancing Deadlift Training Through an Artificial Intelligence-Driven Personal Coaching System Using Skeletal Analysis", Retos (No. 60), 2024, pp. 439-448.

[20] G. Hidalgo, Y. Raaj, H. Idrees, D. Xiang, H. Joo, and T. Simon, "Single-Network Whole-Body Pose Estimation", Proc. 2019 IEEE/CVF Int. Conf. Comput. Vis. (ICCV), IEEE, 2019, pp. 6982-6991.

[21] H.S. Fang, J. Li, H. Tang, C. Xu, H. Zhu, and C. Lu, "AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time", IEEE Trans. Pattern Anal. Mach. Intell., 2022.

[22] C.A. Ronao and S.B. Cho, "Human Activity Recognition with Smartphone Sensors Using Deep Learning Neural Networks", Expert Syst. Appl. (Vol. 59), 2016, pp. 235-244.

[23] A. Flores, B. Hall, L. Carter, M. Lanum, R. Narahari, and G. Goodman, "Verum Fitness: An AI Powered Mobile Fitness Safety and Improvement Application", Proc. 33rd IEEE Int. Conf. Tools Artif. Intell. (ICTAI), IEEE, 2021, pp. 980-984.

[24] C. Guerrero and A. Uribe, "Kinect-Based Posture Tracking for Correcting Positions During Exercise", Med Meets Virtual Real (Vol. 20), 2013, pp. 158-163.

[25] M. Wang, J. Tighe, and D. Modolo, "Combining Detection and Tracking for Human Pose Estimation in Videos", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), IEEE, 2020, pp. 11085-11093.

[26] Oracle, "What Is the MERN Stack?", https://www.oracle.com/ae/database/mern-stack/ [Accessed: 15/02/2026].

[27] Scrum.org, "What is Scrum?", https://www.scrum.org/resources/what-scrum-module [Accessed: 15/02/2026].

[28] V. Bazarevsky et al., "BlazePose: On-device Real-time Body Pose tracking", Computer Vision and Pattern Recognition, 2020, arXiv:2006.10204.

[29] React, "Quick Start – React", https://react.dev/learn [Accessed: 15/02/2026].

[30] Amazon Web Services, "¿Qué es una API de RESTful?", https://aws.amazon.com/es/what-is/restful-api/ [Accessed: 18/02/2026].

[31] I. Hickson, "Server-Sent Events: W3C Recommendation", World Wide Web Consortium, 2015, https://www.pubnub.com/guides/server-sent-events/.

[32] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks", Neural Information Processing Systems (Vol. 33), 2020, pp. 9459-9474.

[33] Meilisearch, "How to Build a RAG Pipeline: A Step-by-Step Guide", https://www.meilisearch.com/blog/how-to-build-a-rag-pipepline [Accessed: 15/02/2026].

[34] A. Field, "Discovering Statistics Using IBM SPSS Statistics", 5th ed., SAGE Publications, London (UK), 2018.

[35] A. Radojević, "Fine-tuning BERT with Masked Language Modelling", Medium, March 24, 2025, https://medium.com/@a.radojevic01/fine-tuning-bert-with-masked-language-modelling-7777f441db7d [Accessed: 08/10/2025].

[36] Qdrant, "Reranking in Semantic Search", https://qdrant.tech/documentation/search-precision/reranking-semantic-search/ [Accessed: 24/08/2025].

[37] IJFMR, "International Journal For Multidisciplinary Research", 2025, https://www.ijfmr.com/papers/2025/1/36867.pdf [Accessed: 08/10/2025].

[38] S. Ragas and A. Krishnamurthy, "Evaluating LLM and RAG systems: Comprehensive metrics and methodologies", Journal of Machine Learning Applications (Vol. 12, No. 3), 2024, pp. 145-162.

[39] S. Saleem, J. Nunes, and Aruna M., "TrainERAI - Live Gym Tracker using Artificial Intelligence", SSRN Electronic Journal, 2023, doi:10.2139/ssrn.4383182.

[40] C. Dong and G. Du, "An enhanced real-time human pose estimation method based on modified YOLOv8 framework", Scientific Reports (Vol. 14, No. 1), 2024, doi:10.1038/s41598-024-58146-z.