International Metaverse Association

Metaverse Convergence Research Institute

ISSN 3059-0442

Journal of Virtual Convergence Research

Volume 1 Number 4 Nov. 2025

Received: 31 August 2025. Accepted: 01 October 2025

Journal of Virtual Convergence Research
2025, VOL. 1, NO. 4, 30~69.
https://intlmetaverse.com/home/journal?lang=en

International Metaverse Association &
Metaverse Convergence Research Institute

Personalized Fitness in the Metaverse: An AI-Based Approach

TaeHwan Lee[1,*]

[1] Ph.D. Student., Graduate School of Metaverse, Sogang Univ., Korea

[*] Corresponding author: TaeHwan Lee. Email: luisfynn1@sogang.ac.kr

Abstract

This study investigates the feasibility of deploying an AI-based fitness platform within a metaverse environment. The proposed platform integrates three core AI technologies—pose estimation, facial emotion recognition, and large language models (LLMs)—to deliver personalized feedback and immersive exercise experiences. The pose estimation component tracks user movements in real time, providing corrective guidance to ensure accurate performance. The facial emotion recognition module analyzes users' affective states to deliver motivational prompts and emotional support. Meanwhile, the LLM enables natural, conversational coaching and interaction, thereby enhancing user engagement and exercise adherence. By combining these AI capabilities, this research proposes a next-generation, metaverse-based fitness experience. Experimental results indicate that the platform can positively influence both user satisfaction and exercise efficiency

Keywords : Metaverse, AI, virtual reality, personalized fitness, Face Emotion Recognition, LLM, Pose Estimation

Personalized Fitness in the Metaverse: An AI-Based Approach

1. Introduction

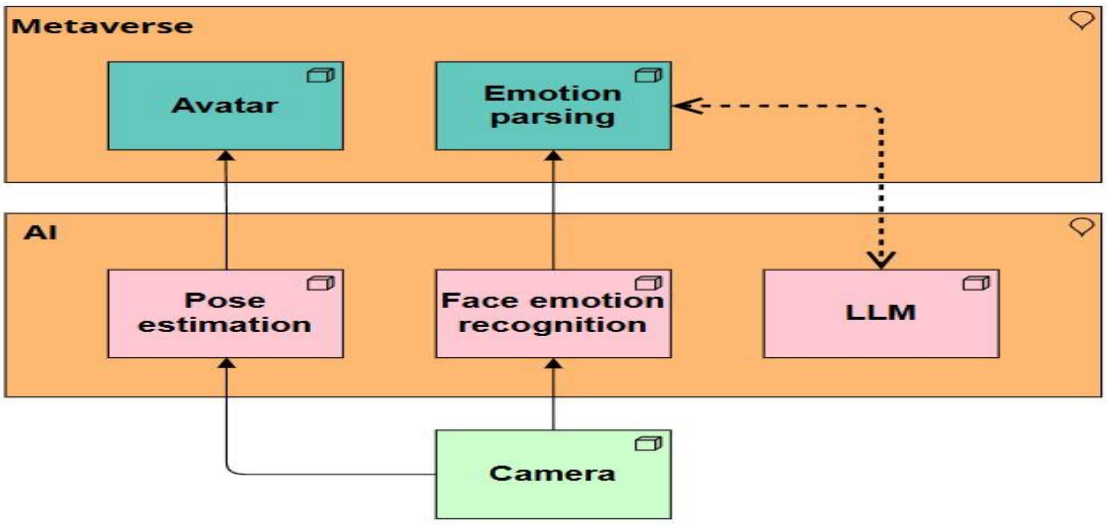
This study aims to propose a personalized fitness program by integrating the metaverse with artificial intelligence (AI)(Ha & Lee, 2020; Lee, 2023; Orlandi et al., 2023). It explores the implementation of an AI-based fitness platform within a metaverse environment to deliver real-time feedback and affective interaction, thereby realizing a next-generation exercise experience. Focusing specifically on the squat exercise, the system is designed to issue a corrective prompt—"Posture incorrect"—in real time whenever the user's knee, hip, or trunk angles deviate beyond predefined thresholds, without performing separate repetition counting, thus enhancing movement focus. While prior work has typically concentrated on single modalities (e.g., pose estimation or emotion recognition) or static applications, the present research demonstrates the potential to address the functional limitations of existing fitness systems by integrating multiple AI technologies to provide a more comprehensive user experience.

The proposed platform comprises three core AI technologies. First, the pose estimation module employs MediaPipe BlazePose to perform precise, three-dimensional (3D) joint-coordinate–based posture analysis, thereby overcoming the limitations of

conventional YOLO-based two-dimensional (2D) pose estimation(Shin & Kang, 2022). Second, the emotion recognition component analyzes users' facial expressions using the Kaggle FER-2013 dataset to infer affective states. Third, a Llama 3.2–based large language model (LLM) delivers naturalistic conversational feedback and personalized coaching.

The platform's implementation is structured around a real-time data interaction pipeline between the AI analysis modules and the metaverse virtual environment. Users' body posture and emotional state are visualized via their metaverse avatars. When a posture deviation is detected, the LLM issues corrective feedback in natural language in real time, while motivational and encouraging messages are delivered in accordance with the emotion recognition output. The integrated system architecture is illustrated Figure 1.

Figure 1. *System structure of the AI-based fitness platform*



2. Methods

2.1 System Communication Architectur

In this system, the AI analysis module (Python) is responsible for executing and inferring the AI models, while the metaverse system manages the user interface and 3D visualization. Data received from the camera (e.g., 3D pose coordinates, facial images) are processed by the AI analysis module, and the resulting outputs or control commands are transmitted to the metaverse system (Unity) to drive the avatar's movements. This bidirectional communication architecture may be deployed over wired links (e.g., TCP/IP–based sockets) or wireless channels (e.g., Wi-Fi, WebSockets), depending on the system environment.

2.1.1 Wired communication

Wired communication mechanisms typically include serial communication over USB and Ethernet-based TCP/IP socket communication. USB serial communication is commonly employed for interfacing with microcontrollers or sensor devices; in the AI analysis module, this can be implemented using libraries such as pyserial, whereas in the metaverse system, one may utilize System.IO.Ports. Although virtual COM ports enable serial communication between PCs or applications, their configuration complexity, limited throughput, and challenges with concurrent processing render them suboptimal for application-level data

exchange. By contrast, socket-based communication over TCP or UDP remains one of the most prevalent wired methods: TCP is chosen when reliability is paramount, while UDP is preferred for low-latency requirements. Both the AI analysis module and the metaverse system can reliably employ these socket protocols to facilitate bidirectional data transfer.

2.1.2 Wireless communications

Wireless communication predominantly relies on Wi-Fi–based networks, although other protocols—such as Bluetooth, WebSocket, and WebRTC—are also employed to support device connectivity and real-time data exchange.

Wi-Fi–based TCP/UDP Communication. Within the same network, Wi-Fi–based TCP/UDP communication enables seamless data exchange between the metaverse virtual environment and the AI analysis module. The AI analysis module can implement its networking functionality using libraries such as Python's socket or asyncio, while the metaverse system may leverage the .NET–based System.Net.Sockets interface.

WebSocket communication. WebSocket communication provides full-duplex, real-time data exchange between the AI analysis module and the metaverse system. In the AI analysis module, this can be implemented using libraries such as Python's websockets or FastAPI's WebSocket support. On the metaverse side, popular implementations include

WebSocketSharp or BestHTTP. WebSocket is especially advantageous for integration with WebGL–based metaverse applications, as it enables low-latency, bidirectional messaging within the browser environment.

Bluetooth Low Energy (BLE). Bluetooth Low Energy (BLE) is well suited for low-power communication with sensors and wearable devices. In the AI analysis module, BLE connectivity can be implemented using libraries such as bleak, while the metaverse system may employ dedicated BLE plugins. However, direct BLE communication between the AI analysis module and the metaverse system is typically subject to significant architectural constraints; consequently, data exchange is usually facilitated indirectly via an intermediary microcontroller (e.g., Arduino or ESP32) that relays information between the two systems.

WebRTC. WebRTC offers ultra–low–latency, real–time peer-to-peer communication. In the AI analysis module, it can be implemented using libraries such as aiortc, while the metaverse system may leverage tools like MixedReality-WebRTC. However, the complexity of configuring the required network infrastructure for initial connections—including signaling server setup, NAT traversal, and TURN server deployment—can limit its applicability in typical deployment environments.

Therefore, this study adopts TCP-based socket communication to facilitate real-time

data exchange between the two systems. This approach ensures reliable transmission in wireless environments while providing sufficient low latency and throughput for real-time performance. To standardize message structure and guarantee interoperability, all exchanged data are serialized and deserialized using the JSON (JavaScript Object Notation) format. Such a communication architecture simplifies integration between heterogeneous software components, enhancing both data interpretability and system extensibility.

2.2 Pose Estimation

Recent work has demonstrated the feasibility of integrating pose estimation with avatar animation to realize metaverse–based fitness applications(Lee, 2023). Ha and Lee (2020) implemented a smart healthcare exercise management app using PoseNet(Ha & Lee, 2020), focusing on the analysis of user exercise posture and the provision of real-time corrective feedback. Google's BlazePose, introduced in 2020, offers high-accuracy, real-time three-dimensional pose estimation (x, y, z coordinates) and has been adopted for avatar motion control in metaverse environments(Bazarevsky et al., 2020; Lugaresi et al., 2019). Unlike YOLO-based methods specialized for two-dimensional (x, y) estimation, BlazePose supports 3D coordinate inference, enabling more precise motion replication (see Table 1). For example, Gu et al. (2024) proposed an AI-driven pose reconstruction technique for remote metaverse avatar creation(Gu et al., 2024), Zhao et al. proposed an

expressive 3D human reconstruction method for multiple subjects using a single monocular camera, thereby demonstrating its potential for metaverse applications(Zhao et al., 2024).

This study extends prior work by employing Google's BlazePose (Figure 2) to develop a real-time, three-dimensional pose estimation and avatar animation system. The system is designed to capitalize on BlazePose's lightweight neural-network architecture—utilizing a single RGB camera to achieve high accuracy and real-time processing performance(Bazarevsky et al., 2020) —thereby meeting the demands of a metaverse-based fitness application. In the proposed implementation, BlazePose tracks user movements with sub-millimeter precision, and the resulting x, y, z coordinate vectors are directly used to animate the user's metaverse avatar. The three-dimensional pose data extracted via BlazePose are transmitted to the metaverse system to enable real-time synchronization between the user's physical movements and their virtual avatar. Unlike YOLO-based two-dimensional pose estimation, the inclusion of the z-coordinate permits more precise and lifelike motion replication. BlazePose outputs 33 body landmarks (Figure 3), each described by x, y, z coordinates and a visibility flag. Two coordinate systems are provided—camera coordinates and world coordinates—with the latter preferred for avatar animation because it references absolute positions in three-dimensional space, thereby preserving accurate depth and spatial relationships. Once extracted, landmark coordinates

Table 1. *Comparison of YOLO Pose Estimation and MediaPipe BlazePose*

Feature	YOLO Pose Estimation	MediaPipe BlazePose
Model Objective	Pose estimation	Pose estimation
Number of Keypoints	17 major joints	33 detailed joints
Performance Characteristics	High accuracy, suitable for complex scenes	Fast inference speed, optimized for single-person tracking
Platform Support	Optimized for desktop environments	Supported on mobile, web browsers, and desktop
Model Size	Relatively large, suited for high-performance hardware	Lightweight, runnable on low-spec devices
Output Data	2D keypoint coordinates with confidence scores	2D and 3D keypoint coordinates with confidence scores

rare converted into the avatar's reference coordinate frame(Moliner et al., 2024), serialized into JSON, and sent over socket communication to the metaverse engine. This pipeline ensures that the virtual avatar reflects user movements in real time with high fidelity. To improve data stability and reduce jitter, a moving-average filter is applied to the raw BlazePose outputs. Leveraging BlazePose's lightweight neural network, which operates on a single RGB camera to deliver high-accuracy, real-time 3D coordinate estimation, this study captures user motion and processes the resulting x, y, z data for precise avatar reproduction in the metaverse environment. Prior studies have applied various filter-based techniques—such as the Kalman filter and low-pass filters—to process data(Gu et al., 2024; Güler et al., 2018). In this study, however, we adopt a moving-average filter to simplify the data-processing pipeline while still effectively reducing error. The moving-average filter can be implemented with relatively simple arithmetic operations and

offers the following advantages.

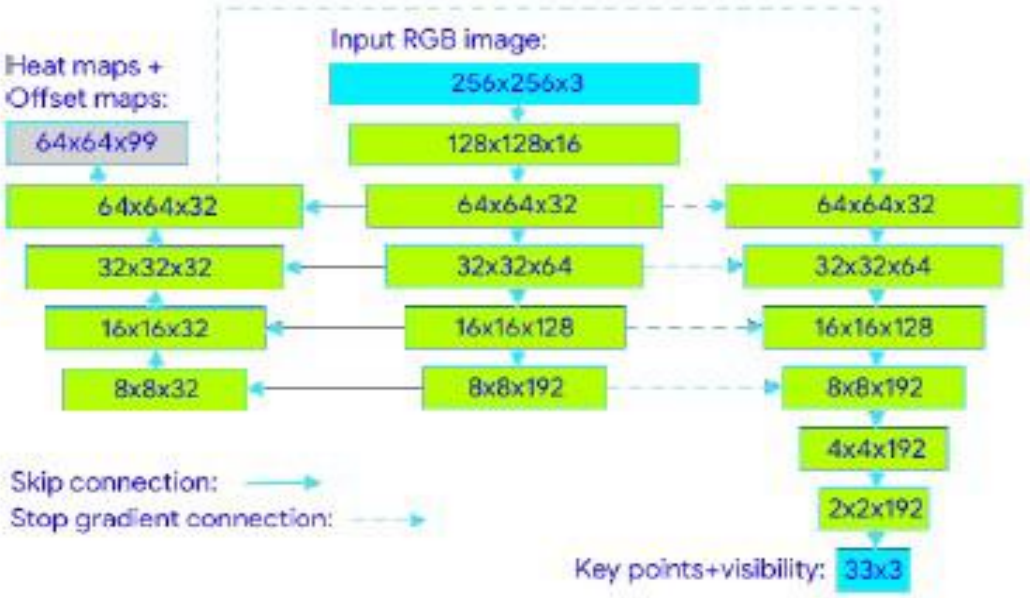
Low computational cost. The Kalman filter requires iterative prediction and update steps, and the low-pass filter necessitates design based on its frequency-response characteristics. In contrast, the moving-average filter can be implemented using only simple arithmetic averaging operations, making it well suited for deployment in resource-constrained, real-time metaverse systems.

Rapid implementation and broad applicability. The moving-average filter features an intuitive algorithmic structure and is easy to implement, enabling rapid adaptation to changing data environments. It is particularly advantageous for initial prototyping stages and lightweight system designs.

Effective noise attenuation. This method effectively mitigates outliers present in the data and reduces overall noise, thereby ensuring a stable data flow.

Real-time performance assurance. Since the computations rely solely on historical data, this method delivers smooth processing in real-time data streaming environments, thereby contributing to fluid avatar animation in the metaverse. Meanwhile, the moving-average filter has the limitation that its sensitivity and response speed vary with the chosen window size. In this study, a fixed window size was employed to conduct experiments,

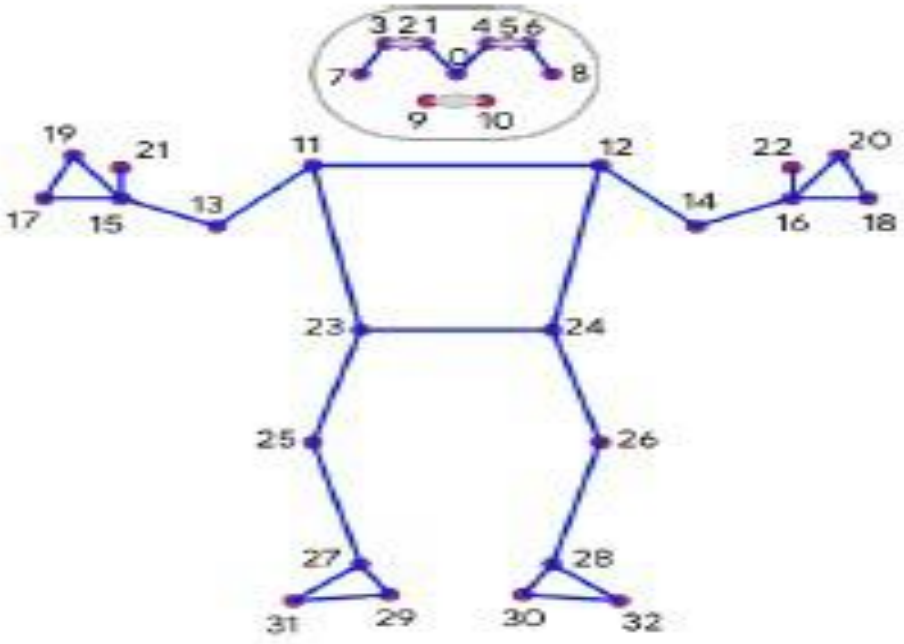
Figure 2. *BlazePose network architecture*



enabling real-time synchronization between user movements and the virtual avatar. However, an in-depth analysis of how window-size variations affect filter sensitivity and quantitative data-processing performance is left for future work.

In conclusion, by combining BlazePose–based three-dimensional pose estimation

Figure 3. *Keypoints defined by BlazePose*



with a moving-average filter, this research achieved efficient real-time data processing in a metaverse environment and contributed to smooth and stable avatar motion driven by user actions.

2.2.1 Avatar Head Movement Control

In this study, avatar head rotation was governed by the x-coordinate of the user's face as extracted by BlazePose. The continuous x values were quantized into predefined intervals, and changes in interval indices were mapped to rotations about the avatar's y-axis. This quantization process attenuated coordinate noise and enabled stable head-motion control.

Interval Partitioning and Mapping. To quantify the user's head-rotation state, the horizontal coordinate x is defined as follows. First, $\mathrm{x} = 0$ corresponds to the neutral, forward-facing position. When the face is rotated 90° to the right,$\mathrm{x} = 0.44$; when it is rotated 90° to the left,$\mathrm{x} = -0.44$. The entire rotation range $[-0.44, 0.44]$ is then divided into 18 equal-sized intervals, allowing fine-grained discrimination of rotation angles. The size of each interval Δ is computed as

$$\Delta = \frac{0.44 - (-0.44)}{18} = 0.049.$$

Therefore, in this study, the continuous xx values are discretized in increments of 0.049, enabling precise determination of the user's head-rotation angle.

Rotation Angle Calculation. After mapping the current horizontal coordinate x to one of the 18 predefined intervals, any change in the interval index between consecutive frames is used to determine the head-rotation angle proportionally. Let k_t denote the interval index at time t and k_{t-1} the index at time $t-1$. The head-rotation angle θ_t is then defined as.

$$\theta_t = (k_t - k_{t-1}) \times 10^\circ$$

That is, if the interval index advances or retreats by one between frames, the avatar's head rotates by $+10^\circ$ or -10°, respectively; if the index difference is greater than one, the rotation angle is scaled accordingly. This proportional mapping enables the system to reflect changes in the user's head-rotation smoothly and in real time.

Overall Algorithm. The overall algorithm is presented below. The detailed procedures for head-rotation control and limb-joint control are summarized in Algorithm 1 and Algorithm 2, respectively.

Conclusion. The overall algorithm is presented below. The detailed procedures for head-rotation control and limb-joint control are summarized in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1: Adjust Head Rotation

$\Delta x \leftarrow x_{current} - x_{prev}$

$s_{current} \leftarrow (\Delta x - x_{min}) \div s_{size}$

$if\, s_{current} \neq s_{prev},$

$then\ RotateHead\left(\left(s_{current} - s_{prev}\right) \times 10.0\right)$

$s_{prev} \leftarrow s_{current}$

$end\ if$

Algorithm 2: Adjust Limb Rotation

$for\ each\ joint\ j\ in\ joints\ do$

$k \leftarrow 1.0$

$r \leftarrow QFTR\left(j.d_{init}, SLERP(j.d_{init}, j.d_{curr}, k)\right)$

$j.p_{rot} \leftarrow r \cdot j.r_{init}$

$end\ for$

QFTR (Quaternion From To Rotation)

$\mathrm{QFTR(u,v)} = (\mathrm{u} \times \mathrm{v})/(\|\mathrm{u}\| \cdot \|\mathrm{v}\|)$

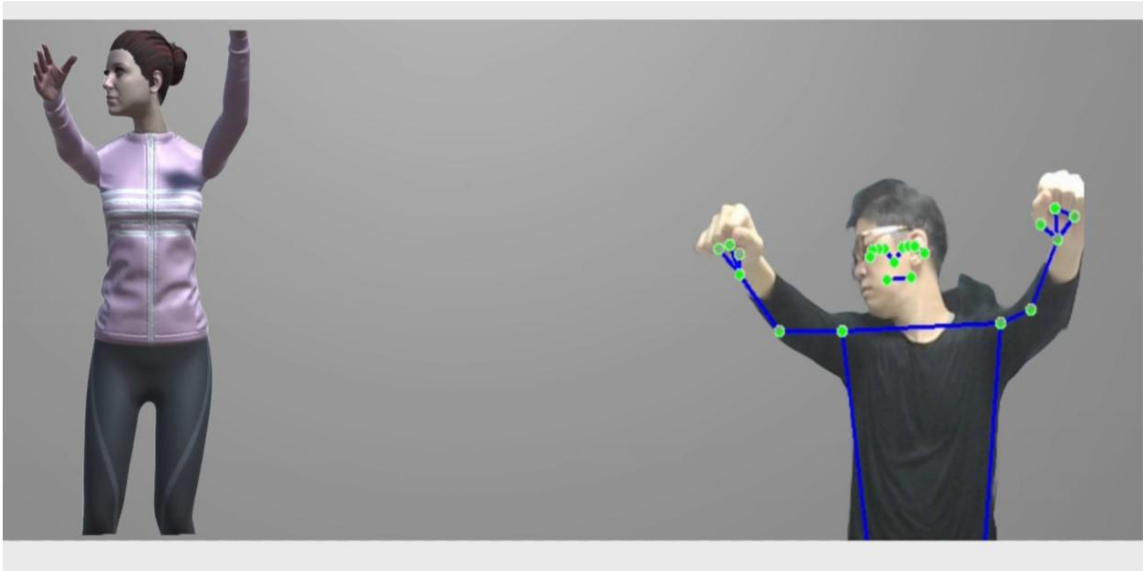
SLERP (Spherical Linear Interpolation):

$\mathrm{SLERP(u,v,t)} = \vec{\mathrm{u}} \cdot (1 - \mathrm{t}) + \vec{\mathrm{v}} \cdot \mathrm{t}$

2.2.2 Avatar Pose and Motion Control

In the metaverse environment, the avatar's skeletal movements are driven in real time by data received from the AI analysis module's pose-estimation thread. By leveraging the body-part coordinate information extracted by the AI module, the virtual avatar can accurately replicate the user's actual motions(Gu et al., 2024). This approach enables remote avatar motion reconstruction and constitutes a critical technical foundation for metaverse applications. When the user's posture deviates from the target form, the AI

Figure 4. *Avatar reflecting movements in real time through coordinate synchronization*



module computes corrective guidance that is immediately visualized by the avatar, allowing the user to intuitively perceive and adjust their exercise form. Such real-time feedback enhances the effectiveness of the fitness experience.

However, when performing pose estimation with a single-camera setup, limitations in the camera's viewpoint can compromise the consistency of posture interpretation. To address this issue, the present study proposes an adaptive interpretation framework that applies different analysis strategies according to the user's position and movement. For instance, during a squat, if the user's feet are located anterior to the hip, the system infers a lateral camera viewpoint, whereas if the hip lies between the feet, a frontal viewpoint is assumed. By selectively invoking the appropriate interpretation method based on the user's spatial relation to the camera, viewpoint-induced errors are minimized and more precise corrective feedback can be delivered. This approach enables multi-angle posture analysis

within a single-camera environment, thereby enhancing exercise efficiency and improving the accuracy of personalized feedback.

Additionally, within the metaverse environment, the avatar's arm and leg joints are driven in real time using the body-part coordinates provided by the AI analysis module's pose-estimation thread. To effect smooth rotational transitions, the rotational delta between the initial and current orientations is computed via a Quaternion-From-To-Rotation operation, and spherical linear interpolation is performed between the two orientation vectors using the Vector3.Slerp function.

This approach prevents gimbal lock[1] and efficiently handles complex rotations in three-dimensional space. By using this method, the avatar faithfully mirrors the user's actual movements. When a posture deviation is detected, the corrective information computed by the AI analysis module is visualized in real time through the avatar, enabling the user to clearly perceive and immediately improve their exercise form.

This study implemented a real-time 3D pose estimation and avatar control system based on Google's BlazePose model. Owing to its lightweight network architecture and

[1] Gimbal lock: a condition in Euler - angle rotations in which two of the three axes align, resulting in loss of one degree of freedom

real-time 3D coordinate estimation capabilities, BlazePose enables high-fidelity motion replication in metaverse environments, as illustrated in Figures 4–6.

For future work, we plan to draw on the integrated facial, body, and hand modeling approach of the MEPO (Multi-person Expressive Pose) framework proposed by Zhao et al. to achieve more precise digital-twin representations. MEPO, which builds upon the SMPL-X model, offers a unified reconstruction of facial expressions, hand articulations, and full-body pose, facilitating the generation of highly expressive 3D human avatars.

This study currently centers on a single BlazePose model; however, we anticipate that integrating multiple MediaPipe AI models—such as Face Mesh, Hands, and BlazePose—will enable even more precise and richly detailed virtual human representations.

By unifying the metaverse environment with the AI analysis module, the resulting metaverse-based fitness system affords the following key advantages: Posture Accuracy Assessment. The system continuously evaluates whether the user's body alignment conforms to predefined biomechanical criteria, providing immediate corrective feedback for any detected deviations.

Efficient Data Processing and Minimal Bandwidth. Unlike conventional AI-driven fitness platforms that require uploading video recordings of exercise sessions—resulting

Figure 5. *Avatar reflecting movements in real time through coordinate synchronization2*

in high data transfer volumes—our approach transmits only avatar-driving coordinate vectors to the metaverse engine, thereby substantially reducing network load.

User Privacy Protection. By operating solely on abstracted pose coordinates (without transmitting facial imagery), the platform preserves user anonymity and mitigates concerns over appearance exposure, enhancing accessibility and satisfaction in remote fitness contexts.

Asynchronous Feedback Delivery. Uploaded exercise data can be reviewed and annotated by trainers at their convenience, obviating the need for real-time sessions and

affording both trainers and users greater scheduling flexibility and operational efficiency.

2.2.3 Squat Posture Correction Algorithm

In this study, we implemented a real-time module that detects and issues corrective messages for two additional squat posture errors frequently observed in frontal-view video: narrow stance and knee varus.

First, the narrow-stance error is defined when the user's inter-ankle distance falls significantly below their inter-shoulder distance. Applying the tolerance determined during experimentation, the system issues the "Narrow stance" corrective message whenever the ankle span remains smaller than the shoulder width for three consecutive observations.

Figure 6. *Avatar reflecting movements in real time through coordinate synchronization3*

Algorithm 3: Narrow Stance Error Recognition

$d_{foot} \leftarrow |x_{ankle\ left} - x_{ankle\ right}|,$

$w_{shoulder} = |x_{shoulder\ left} - x_{shoulder\ right}|.$

$if d_{foot} < w_{shoulder} - \Delta w, then\ show\ message$

Algorithm 4: Excessive Knee Varus Error Recognition

$d_{knee} \leftarrow |x_{knee\ left} - x_{knee\ right}|$

$d_{foot} \leftarrow |x_{ankle\ left} - x_{ankle\ right}|$

$if d_{knee} > d_{foot} + \Delta k, then\ show\ message$

Second, the excessive-varus error is defined when the user's inter-knee distance becomes significantly greater than their inter-ankle distance. Applying the experimentally determined tolerance threshold, the system confirms the error and issues the "Knee varus" corrective message only if this condition persists for ten consecutive frames.

Thus, this module accurately detects and corrects both stance width and knee varus in frontal-view squat execution, helping users maintain proper form. It operates independently of any repetition-counting logic and provides natural-language feedback only upon error detection, thereby enhancing the user's focus on movement quality.

2.2.4 Evaluation of Correction Module Robustness under Illumination Variations

Participants and Experimental Setup. In this preliminary experiment, the sole participant was the author. The participant recorded a 16-second frontal-view squat video—including intentional narrow-stance errors—using a smartphone (720p resolution,

Table 1. *Experimental Platform Hardware and Software Specifications*

Category	Specification
Operating System	Microsoft Windows 11 Pro, Version 10.0.26100
Processor	13th Gen Intel® Core™ i9-13900H @ 2.60 GHz
Memory	64.0 GB (63.8 GB available)
Graphics	NVIDIA GeForce RTX 4060 Laptop GPU
Input resolution	720p
Output resolution	Full HD, QHD, 4K UHD
Driver	NVIDIA Studio Driver 576.52 (May 19, 2025)

30 fps), yielding approximately 479 frames of data. Hardware and software specifications are summarized in Table 2.

Experimental Procedure and Error Detection Rate Evaluation. OpenCV was used to generate five levels of illumination by multiplying the V channel of the HSV color space by brightness coefficients b∈{0.5,0.8,1.0,1.2,1.5}. Each resulting video was played back in a Unity-based metaverse environment at three resolutions (Full HD, QHD, and 4K UHD), with the display scale fixed at 1.0. During each playback session, the proposed narrow-stance error detection module was applied frame by frame, and the presence or absence of an

Figure 7. *Image by brightness*

Table 2. *Detection Performance under Varying Brightness Conditions & Resolution*

Brightness	Detected Frames	Total Frames	Detection Rate (%)
0.5	478	478	100
0.8	478	478	100
1.0	478	478	100
1.2	478	478	100
1.5	478	478	100

The original video comprised 478 frames, and the Unity environment likewise processed all 478 frames successfully, indicating that the tested resolutions (720p, 1080p, and 4K) and a fixed scale of 1.0 did not affect frame processing. Moreover, we conducted iterative experiments on the threshold parameter www of Algorithm 3, testing multiple candidate values. With the optimally chosen threshold, the module achieved 100% detection accuracy for all frames in which the inter-ankle distance remained smaller than the shoulder width. These results demonstrate that the proposed correction module delivers consistent frame processing performance and perfect error detection regardless of resolution or rendering environment.

Validity of Corrective Feedback and Statistical Testing. In this study, three participants each recorded two 15-second videos: one demonstrating the narrow-stance error and one showing correct squat form. At 30 fps, each video contained approximately 450 frames. Playback occurred in the Unity environment at the original brightness level (b=1.0b=1.0b=1.0) and a fixed display scale of 1.0, with no software-based illumination adjustments.

For each participant, the playback of the error video was analyzed to count the number of frames in which the "Narrow stance" corrective message was issued, yielding True Positive (TP) and False Negative (FN) counts. Similarly, during the playback of the correct-form video, the number of frames without a corrective message was recorded to obtain True Negative (TN) and False Positive (FP) counts. The resulting TP, FN, FP, and TN values from all three participants were then aggregated for subsequent statistical evaluation. Aggregating the data from the three participants, we calculated True Positives (TP) and False Negatives (FN) for the error videos, and False Positives (FP) and True Negatives (TN) for the correct-form videos. Sensitivity and specificity are defined as

$$\text{Sensitivity} = \frac{TP}{TP + FN}, \ \text{Specificity} = \frac{TN}{TN + FP}.$$

Evaluating sensitivity and specificity in tandem enables a more refined performance analysis than using overall detection rate alone. Sensitivity reflects the module's ability to detect actual errors without omission, whereas specificity indicates its capacity to avoid misclassifying correct movements as errors. By considering both metrics, we can independently assess the algorithm's error-miss and false-alarm rates and select an optimal threshold that maximizes true error detection while minimizing unnecessary feedback. Moreover, by verifying that sensitivity and specificity remain above predefined levels under varying conditions—such as changes in illumination, resolution, and user physique—we

can evaluate the robustness and generalizability of the proposed correction module. Ultimately, a comprehensive analysis of sensitivity and specificity is indispensable for designing a stable and trustworthy feedback system that preserves user experience.

Across the three participants, Participant A correctly detected 389 of 482 actual error frames (sensitivity ≈ 80.7 %) and missed 93, while producing zero false alarms over 402 correct-form frames (specificity = 100 %). Participants B and C each attained perfect performance, correctly identifying all 418 and 453 correct-form frames, respectively, and yielding 100 % sensitivity and specificity. When pooled, the cohort comprised 1,353 correct-form frames—of which 1,260 were correctly passed—resulting in an overall sensitivity of approximately 93.1 %, and 1,279 error frames—all of which were accurately detected—yielding an overall specificity of 100 % (see Table 4).

Table 3. *Aggregated confusion matrix and performance metrics across all subjects*

USER_A	Detection True	Detection False	Total
Actual True	389	93	482
Actual False	0	402	402
Total	389	495	884
USER_B	Detection True	Detection False	Total
Actual True	418	0	418
Actual False	0	399	399
Total	418	399	817
USER_C	Detection True	Detection False	Total
Actual True	453	0	453
Actual False	0	478	478
Total	453	478	931

An independent-samples χ^2 test of independence (df = 1) was conducted on frame-level observations treated as two independent groups—correct-form videos and narrow-stance error videos. Aggregating data across the three participants, the error videos comprised 1,279 frames with zero false negatives (FN = 0) and 1,279 true positives (TP = 1,279), whereas the correct-form videos comprised 1,353 frames with 1,260 true negatives (TN = 1,260) and 93 false positives (FP = 93). The χ^2 statistic, computed as

$$\chi^2 = \sum \frac{(O-E)^2}{E} \approx 807.5, \quad p < 0.001,$$

indicates a highly significant difference in detection outcome distributions between correct and error videos at the 0.001 significance level. These results confirm that the proposed correction module statistically differentiates between the two independent conditions—correct posture and error posture—by accurately detecting error frames without generating false alarms on correct frames.

2.3 Emotion Recognition

2.3.1 Related Work and Technological Trends

Emotion recognition serves as a key technology for enhancing affective interaction in metaverse environments, and a variety of datasets and methodologies have been developed to support this task. Among these, AffectNet stands out as one of the largest and most comprehensive facial-image datasets for emotion analysis, comprising

approximately 450,000 real-world images(Mollahosseini et al., 2019). The dataset provides eleven categorical labels: eight primary emotion classes (Neutral, Happy, Sad, Surprise, Fear, Anger, Disgust, and Contempt) and three supplementary categories (None, Uncertain, No-Face). Each image is annotated according to the facial expression it portrays, making the dataset well-suited for training and evaluating emotion-recognition models under noisy, real-world conditions. In addition to categorical labels, AffectNet offers continuous valence and arousal values, enabling fine-grained affective analysis and thereby improving the precision of emotion-recognition systems.

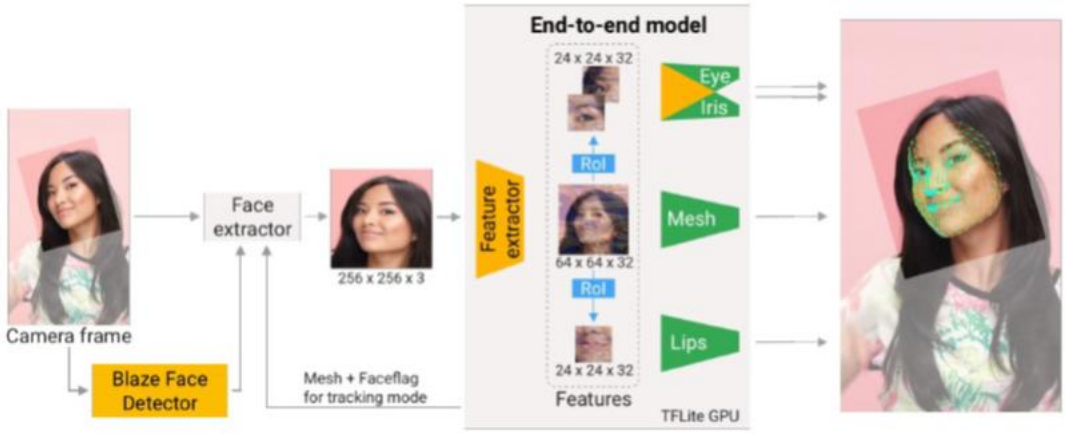
FER2013, made publicly available through a Kaggle competition, comprises approximately 35,000 grayscale facial images labeled with seven fundamental emotion categories(Goodfellow et al., 2013). It is extensively used for benchmarking lightweight, neural-network–based emotion classifiers. EmoReact contains around 11,000 video clips annotated with 26 combined emotion–behavior labels, enabling simultaneous analysis of affective states and actions and supporting real-time evaluation of emotion recognition systems(Kosti et al., 2017). The Surrey Audio-Visual Expressed Emotion (SAVEE) dataset provides multimodal recordings (audio and video) capturing seven emotions—Anger, Disgust, Fear, Happy, Neutral, Sadness, and Surprise—acted by four British males, serving as a valuable resource for audio-visual affect analysis(Haq & Jackson, 2009). Physiological-

signal datasets such as DEAP, which utilize EEG and peripheral biosignals to infer emotional states, have been instrumental in enhancing the precision and reliability of affective computing(Koelstra et al., 2012). However, the large scale and multimodal nature of these resources often necessitate high-performance computing and extensive computational power, restricting their use in personal computing environments. To address this, the present study employs the Kaggle FER-2013 dataset to develop a lightweight, machine-learning–based emotion recognition framework.

2.3.2 Implementation of Methodology and System Deployment

In this study, we developed an emotion-recognition model using the Kaggle FER-2013 dataset. The dataset comprises seven emotion categories—angry, disgust, fearful, happy, neutral, sad, and surprised—and is organized into separate training and testing directories. Each image was processed with Google's MediaPipe Face Mesh(Grishchenko et al., 2020) to extract 468 precise facial landmark coordinates (Figure 8). The extracted features were consolidated into train.csv and test.csv, with emotion labels mapped as follows: angry = 0, disgust = 1, fearful = 2, happy = 3, neutral = 4, sad = 5, and surprised = 6. To address class imbalance, we applied targeted preprocessing techniques—such as class weighting and data augmentation—to construct a balanced dataset suitable for training robust classifiers.

Figure 1. *MediaPipe Face Mesh Structure*



2.3.3 Model Evaluation Results

In this paper, various models were evaluated using accuracy, precision, recall, F1-score, AUC, and loss metrics, with the results presented in Table 5.

VGG. VGG employs a very deep convolutional neural network (CNN) architecture to maximize learning performance on large-scale image datasets(Simonyan & Zisserman, 2015). Experimental evaluation revealed that VGG achieved high precision (0.8097), demonstrating robust classification capability for certain classes. However, its recall (0.1690) and F1-score (0.2805) were markedly low, indicating a limitation under data-imbalance conditions. This performance profile suggests that the network's depth may have led to

Table 4. *Model Performance Evaluation*

Model	Accuracy	Precision	Recall	F1-Score	AUC	Loss
XGBoost	0.5283	0.5288	0.5283	0.5286	0.7111	N/A
LightGBM	0.5366	0.5339	0.5366	0.5353	0.7141	N/A
MobileNet	0.5482	0.7565	0.3663	0.4931	0.8788	1.1964
VGG	0.4205	0.8097	0.1690	0.2805	0.8015	1.4843
ResNet	0.2924	0.6032	0.0106	0.0210	0.7065	1.7242

overfitting on specific patterns during training.

ResNet. ResNet introduces residual learning to mitigate the vanishing-gradient problem in very deep networks(He et al., 2015). Despite its reputation for maximizing training stability, our experiments yielded a low recall (0.0106) and F1-score (0.0210), indicating that ResNet failed to adequately learn the underlying data patterns. This outcome suggests that the characteristics of the experimental dataset were not well aligned with ResNet's residual-learning architecture.

MobileNet. MobileNet is a lightweight neural network architecture designed for computational efficiency in mobile environments, reducing operations while maintaining high performance(Howard et al., 2017). In our experiments, MobileNet demonstrated strong classification capability with a high AUC of 0.8788; however, its F1-score was relatively low at 0.4931, indicating that the model did not fully capture complex data distributions. This suggests that the lightweight architecture may have inherent limitations in learning fine-grained patterns.

LightGBM. LightGBM achieved an F1-score of 0.5353, demonstrating stable performance. Its tree-based learning mechanism effectively captures complex data patterns, yielding balanced accuracy and overall robustness. For these reasons, LightGBM was

selected as the final model in this study.

Literature-Based Comparison. Although VGG and ResNet demonstrate strong performance on large-scale datasets, they exhibited degraded results when applied to the limited and imbalanced data of this experiment. MobileNet achieved computational efficiency and a high AUC but showed limitations in learning complex patterns. In contrast, LightGBM—leveraging its tree-based architecture—delivered superior performance under constrained data conditions and was therefore deemed the most suitable model for the objectives of this study.

Conclusion. The model performance evaluation is summarized in Table 2. Among the compared models, LightGBM achieved the highest performance, recording an F1-score of 0.5353 and an accuracy of 53.66%. This outcome suggests that LightGBM's tree-based architecture effectively captures the data's nonlinear characteristics and handles class imbalance. During testing, participants were prompted to adopt strenuous facial expressions, which the system tended to classify as "fearful." Accordingly, this study evaluates emotional states on a one-second interval basis and treats four consecutive "fearful" detections as a single salient affective event to be transmitted to the metaverse system. This strategy offers a practical means of eliciting real-time emotional feedback within the constrained emotion categories of the FER-2013 dataset, which does not include

labels such as “struggling” or “tired.”

In the camera-based test environment, real-time feedback was provided in accordance with the detected emotional state to enhance user immersion. For example, upon detection of a fearful state, an encouraging message—”Keep going!”—was displayed. Such a feedback mechanism enables users to recognize and positively modulate their emotional states in real time, thereby promoting emotional self-regulation and strengthening the overall user experience (UX).

Comparison with AffectNet and Future Research Directions. A comparative analysis with AffectNet and directions for future research are as follows. Prior studies using the AffectNet dataset with an AlexNet-based CNN model reported F1-scores ranging from 0.08 to 0.95 across emotion classes, with an overall average accuracy of approximately 58%(Mollahosseini et al., 2019). In contrast, the LightGBM and XGBoost models proposed in this study achieved accuracies of 52.83% and 54.82%, respectively, yielding comparable or slightly lower performance overall. This discrepancy arises because AffectNet is based on a large-scale collection of roughly 450,000 images and high-capacity CNN architectures, whereas the present work constructs a novel dataset using high-dimensional coordinate features extracted via MediaPipe Face Mesh. Face Mesh provides 468 facial-landmark coordinates, and the resulting high-dimensional numerical features can increase input

dimensionality and thus impact the performance of lightweight machine-learning models. Nonetheless, Face Mesh–based coordinates serve as a critical technology for implementing fine-grained affective interfaces—such as avatar facial expression control and emotion rendering—in metaverse environments. This design approach can be interpreted as offering practical advantages in long-term system scalability and expressive precision, despite potential short-term performance trade-offs. Moreover, both the AffectNet–based studies and the present experiments exhibited low classification performance for minority classes such as Disgust, indicating that data imbalance remains a significant challenge in the field of emotion recognition.

This study adopted a coordinate-based lightweight model architecture to implement a real-time emotion-recognition system in low-spec environments, and experimentally demonstrated that satisfactory performance can be achieved under constrained computational resources. In future work, we plan to leverage large-scale, structured image datasets such as AffectNet to develop refined deep-learning–based emotion-recognition models and extend the model architecture to support continuous affective analysis by incorporating valence and arousal values. Additionally, we will address class-imbalance issues through data-augmentation and class-resampling techniques, and mitigate model bias arising from varying lighting conditions and individual expression

styles. Furthermore, we intend to integrate pretrained face-detection models—such as YOLO—into the emotion-recognition pipeline to enhance facial alignment and recognition accuracy, thereby improving overall classification performance.

2.4 LLM-Based Feedback System

The proposed Llama 3.2–based feedback system is architected to enable real-time, interactive engagement with users by delivering personalized natural-language feedback informed by both emotional and postural data. Leveraging advanced natural language processing techniques, conversational feedback enhances user immersion, promotes exercise adherence, and boosts motivation. As illustrated in Figure 9, the system continuously monitors the user's state and issues context-appropriate messages, thereby extending beyond mere fitness assistance to facilitate affective interaction. Future work will explore multifaceted system extensions aimed at further refining the user experience.

Figure 2. *Emotion detection and LLM process*

Dedicated Feedback Dataset and Prompt Engineering. We plan to construct a specialized dataset of diverse encouragement and motivational phrases, then design conversational flows using prompt engineering and conditional response rules to deliver contextually appropriate natural-language feedback. This dataset will comprise curated sentences reflecting users' emotional states, exercise performance, and goal settings, thereby enabling the generation of responses tailored to real-world scenarios. To achieve this, we will extend existing emotion-recognition corpora (e.g., AffectNet(Mollahosseini et al., 2019), EmoReact(Kosti et al., 2017)) and collect new emotion-expression data specific to fitness feedback.

Voice Interaction Integration. We will incorporate voice-based interaction by leveraging state-of-the-art speech-to-text technologies—such as Google Speech-to-Text API or Whisper —to allow users to request feedback or pose questions verbally. Spoken input will be transcribed and passed to the Llama 3.2 model, which will generate appropriate natural-language responses. This design enables hands-free, screen-free interaction, significantly enhancing user convenience and immersion.

Multimodal Feedback Enhancement and Adaptive Response Strategies. By combining voice interaction with emotion-recognition outputs, we aim to diversify and naturalize system responses according to the user's state. We will apply rule-based analysis

of recurring user-interaction patterns to iteratively refine the feedback strategy. This approach is expected to evolve the platform into a comprehensive, user-centric healthcare assistance system—extending beyond basic exercise guidance to encompass motivation, goal management, and emotional support.

These research directions build on recent advances in LLM development and speech-recognition methodologies, with the ultimate goal of dramatically improving the precision and immersion of personalized feedback.

3. Conclusion

This study proposes an AI-based fitness platform within a metaverse environment, founded on real-time interaction between the metaverse system and an AI analysis module. The proposed system precisely captures users' biometric data via BlazePose-based 3D pose estimation and Face Mesh-based emotion recognition, and synchronizes these data with virtual avatars to deliver an immersive fitness experience. Furthermore, by integrating an LLM-driven natural-language feedback system, it enhances affective interaction and sustains exercise motivation. The primary contributions of this work are as follows:

- Integration of Heterogeneous AI Technologies. We implement practical interactive functionality in a metaverse fitness system by unifying pose estimation, emotion recognition, and natural-language generation.

- Real-Time Squat Correction. Leveraging BlazePose's 3D pose estimation, we correct squat form and combine posture detection with immediate feedback to deliver personalized fitness services.
- Personalized LLM Feedback. Utilizing the Llama 3.2 model, we realize accurate, personalized, real-time feedback generation.
- Real-Time Communication Architecture. We design and implement a bidirectional socket-based communication pipeline between the metaverse system and the AI module, enabling intuitive visualization of user movements and contributing to posture correction and enhanced immersion.

Experimental results demonstrate that the proposed metaverse fitness platform yields significant improvements in user satisfaction, immersion, and exercise adherence. In particular, the fusion of emotion-recognition feedback with natural-language interaction effectively supports personalized fitness experiences.

However, the current system exhibits the following technical limitations:

- Occasional unnatural avatar movements in certain exercises
- Latency issues during data transmission between the metaverse system and AI module

Need for improved accuracy in emotion recognition and pose-estimation models

These limitations will be addressed in future work through communication-pipeline optimization, adoption of parallel processing techniques, and augmentation-based

enhancement of AI models. We also plan to incorporate Text-to-Speech (TTS) functionality to diversify user interaction and further boost immersion via real-time vocal feedback.

The system presented herein has the potential to extend beyond fitness into diverse metaverse applications—such as rehabilitation therapy, remote education, virtual conferencing, and affect-driven entertainment—suggesting its role as a foundational technology for next-generation immersive platforms based on AI–metaverse convergence.

Funding: This research was supported by IITP and MSIT of Korea through the Graduate School of Metaverse Convergence Program (RS-2022-00156318), and by MCST of Korea through the KOCCA grant (RS-2023-00219237) as part of the Culture, Sports, and Tourism R&D Program.

References

Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). BlazePose: On-device Real-time Body Pose tracking (arXiv:2006.10204). arXiv. https://doi.org/10.48550/arXiv.2006.10204

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., … Bengio, Y. (2013). Challenges in Representation Learning: A report on three machine learning contests (arXiv:1307.0414). arXiv. https://doi.org/10.48550/arXiv.1307.0414

Grishchenko, I., Ablavatski, A., Kartynnik, Y., Raveendran, K., & Grundmann, M. (2020). Attention Mesh: High-fidelity Face Mesh Prediction in Real-time (arXiv:2006.10962). arXiv. https://doi.org/10.48550/arXiv.2006.10962

Gu, X., Yuan, Y., Yang, J., & Li, L. (2024). AI-empowered Pose Reconstruction for Real-time Synthesis of Remote Metaverse Avatars. 2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE), 86–93. https://doi.org/10.1109/JCSSE61278.2024.10613638

Güler, R. A., Neverova, N., & Kokkinos, I. (2018). DensePose: Dense Human Pose Estimation In The Wild (arXiv:1802.00434). arXiv. https://doi.org/10.48550/arXiv.1802.00434

Ha, T., & Lee, H. (2020). Implementation of Application for Smart Healthcare Exercise Management Based on Artificial Intelligence. Journal of the Institute of Electronics and Information Engineers, 57(6), 44–51. https://doi.org/10.5573/ieie.2020.57.6.44

Haq, S., & Jackson, P. J. B. (2009). Speaker-dependent audio-visual emotion recognition. Proceedings of the International Conference on Auditory-Visual Speech Processing, 53–58. https://doi.org/10.1109/AVSP.2009.5350835

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition (arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (arXiv:1704.04861). arXiv. https://doi.org/10.48550/arXiv.1704.04861

Koelstra, S., Muhl, C., Soleymani, M., Jong-Seok Lee, Yazdani, A., Ebrahimi, T., Pun, T., Nijholt, A., & Patras, I. (2012). DEAP: A Database for Emotion Analysis ;Using Physiological Signals. IEEE Transactions on Affective Computing, 3(1), 18–31. https://doi.org/10.1109/T-AFFC.2011.15

Kosti, R., Alvarez, J. M., Recasens, A., & Lapedriza, A. (2017). Emotion Recognition in Context. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1960–1968. https://doi.org/10.1109/CVPR.2017.212

Lee, J. (2023). Real-Time Pose Estimation and Motion Animation Generation of Avatar for Metaverse Home Training. Journal of Korea Game Society, 23(1), 25–34. https://doi.org/10.7583/JKGS.2023.23.1.25

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A Framework for Building Perception Pipelines

(arXiv:1906.08172). arXiv. https://doi.org/10.48550/arXiv.1906.08172

Moliner, O., Huang, S., & Åström, K. (2024). Geometry-Biased Transformer for Robust Multi-View 3D Human Pose Reconstruction. 2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG), 1–8. https://doi.org/10.1109/FG59268.2024.10581930

Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2019). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. IEEE Transactions on Affective Computing, 10(1), 18–31. https://doi.org/10.1109/TAFFC.2017.2740923

Orlandi, L., Martinelli, G., Laiti, F., Lobba, D., Bisagno, N., & Conci, N. (2023). Meta-Trainer: An Augmented Reality Trainer for Home Fitness with Real-Time Feedback. 2023 IEEE International Workshop on Sport, Technology and Research (STAR), 90–93. https://doi.org/10.1109/STAR58331.2023.10302670

Shin, H.-J., & Kang, T.-K. (2022). Research and Development for Artificial Intelligence-Based Fitness Posture Correction. Proceedings of the Korean Institute of Electrical Engineers Annual Conference, 65–66.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition (arXiv:1409.1556). arXiv. https://doi.org/10.48550/arXiv.1409.1556

Zhao, Z., Tang, H., Wan, J., & Yan, Y. (2024). Monocular Expressive 3D Human Reconstruction of Multiple People. Proceedings of the 2024 International Conference on Multimedia Retrieval, 423–432. https://doi.org/10.1145/3652583.3658092