Proceedings of Machine Learning Research vol 278:1–11, 2025 2025 2nd International Conference on Machine Learning and Intelligent Computing

Design and Implementation of Lightweight Fitness System Based on Mediapipe Framework

Jialei Shi* JIALEI20031202@GMAIL.COM
Guilin Institute of Information Technology, Guilin, China

Ruohan Lin LINRUOHAN9725@OUTLOOK.COM
Guilin Institute of Information Technology, Guilin, China

Bohao Zhou NIANANYU226@GMAIL.COM
Guilin Institute of Information Technology, Guilin, China

Editors: Nianyin Zeng, Ram Bilas Pachori and Dongshu Wang

Abstract

With the development of the social economy, people are paying increasing attention to personal health and regard fitness as an essential way to improve physical quality. However, China has a large population and cannot provide high-quality physical education for everyone, facing enormous logistical and resource challenges. Therefore, this study designed a lightweight, intelligent fitness system based on Mediapipe and OpenCV, which offers high adaptability, portability, and a lightweight design, particularly for micro mobile devices. The system can provide users with more convenient, personalized, and efficient training methods. This paper offers an in-depth introduction to the functional framework and development details of the system and conducts functional testing and comparison. According to the experimental results, the system shows good performance, stable operation, and high recognition rate, achieving the expected experimental goal.

Keywords: Intelligent fitness, Mediapipe, OpenCV, Attitude recognition, PyQt5

1. Introduction

Our intelligent fitness system leverages machine learning and computer vision technology, with real-time recognition of human posture at its core. Complex network structures are often employed during human pose estimation to achieve superior prediction performance. However, the actual inference speed of these models tends to be s low (Gao et al., 2024). We propose a solution that differs from traditional offline fitness posture recognition to address this issue. We offer a portable fitness action recognition method that provides faster and more accurate fitness guidance and the potential for future integration into lightweight hardware devices. By utilizing Google's 2019 Mediapipe framework, which boasts excellent scalability and cross-platform machine-learning solutions, we have opened up new avenues for intelligent fitness applications.

We introduce a lightweight, intelligent fitness system based on Mediapipe and OpenCV, leveraging Mediapipe's high performance and BlazePose for pose recognition, correction, action counting, and more. To validate the effectiveness of this framework on lightweight devices, we employ a simple and cost-effective method for calculating two-dimensional coordinate angles and compare its performance with OpenPose under identical hardware conditions. We aim to provide a portable and effective intelligent fitness solution, harnessing its lightweight nature for application in various miniature devices.

In summary, our contributions are as follows:

- To optimize the system's operating efficiency on lightweight devices, we adopted a simple, low-computational-cost, two-dimensional coordinate angle algorithm to achieve rapid recognition and feedback of key human posture parameters.
- We designed and implemented an auxiliary fitness system that combines the Mediapipe framework and the OpenCV library. We used BlazePose to achieve efficient human posture detection, motion correction, and motion counting functions.

2. Related Work

Early pose estimation models like OpenPose rely on multi-stage heatmap prediction. OpenPose introduced the Part Affinity Fields (PAFs) representation, consisting of flow fields used to encode the pairwise, unstructured relationships between human body parts (Cao et al., 2018). However, PAFs depend on multi-stage convolutional neural networks and use large-scale heatmaps and vector fields to regress key points and connectivity relationships. Although it achieves 75% AP on the COCO dataset, the computational cost exceeding 50G FLOPs makes it challenging to deploy on edge devices. In contrast, the Mediapipe-BlazePose pose recognition model designed by Bazarevsky, V., Grishchenko, I., et al. adopts a hybrid approach combining heatmaps, offsets, and regressions. Only heatmap and offset losses are used during training, and the corresponding output layers are removed before inference. This approach effectively supervises lightweight embeddings via heatmaps, which are later utilized by a regression encoding network (Bazarevsky et al., 2020). Specifically, BlazePose employs a residual hourglass structure and model distillation, successfully achieving 30 FPS on mobile devices, although it sacrifices robustness in occluded scenes.

With the continuous development of AR devices (e.g., Vision Pro) and edge devices, the lightweight characteristics of pose recognition models are becoming increasingly important. Previously, Wang et al. (2022) studied the efficient architecture design for real-time multi-person pose estimation on edge devices, identifying that the high-resolution branches of HRNet are redundant for low-computation models. They designed LitePose, an efficient single-branch architecture for pose estimation , which demonstrates excellent performance on edge and AR devices. Furthermore, BlazePose is specifically designed as a lightweight convolutional neural network architecture for mobile devices, capable of running at over 30 frames per second on the Pixel 2 phone, making it suitable for real-time applications such as fitness tracking and sign language recognition.

By studying BlazePose's application in the sports and fitness domain, we propose a method based on the 2D coordinate system for calculating keypoint angles. We design multi-dimensional thresholds (angle + confidence) and analyze their performance under different lighting and occlusion conditions. Additionally, we compare it with traditional pose recognition models, such as OpenPose, from the perspectives of PCK, recognition accuracy in various environments, and computational resource consumption.

3. System Design

We used PyQT and OpenCV frameworks to assist in our research.PyQt is a Python-based GUI framework offering a rich set of components and event-handling mechanisms for developing various GUI applications (Jiang et al., 2023). With its excellent compatibility with Mediapipe, PyQt5

leverages Python's flexibility to provide efficient, reliable, and user-friendly tools for cross-platform program development. In conjunction with OpenCV, an open-source computer vision library designed to simplify the learning process for industrial engineering students (Sigut et al., 2020), this project integrates real-time video capture and frame-by-frame processing through MediaPipe. This combination enables intuitive image processing feedback, enhancing the user experience.

The innovative fitness system aims to offer users a comprehensive and customizable fitness experience through key functions, including user login, mode selection, fitness action selection, posture accuracy calculation, action counting, and video demonstrations of standard actions. These functions are seamlessly integrated, ensuring a consistent user experience, ease of use, and enhanced system security and robustness. The interaction and control logic between these functions are illustrated in Figure 1.

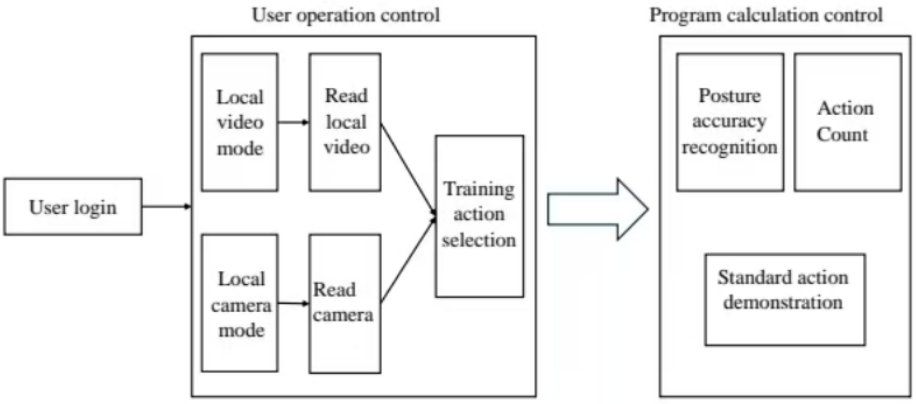

Figure 1: System Function Control Flow Chart

To control system functions while ensuring portability, we developed the visual front-end interface using PyQt5, enabling smooth interaction and data transmission between components.

Figure 2 shows that the front-end control flow includes login and function selection modules. By monitoring signals (e.g., clicks, selection changes) from various components (buttons, drop-down menus, sliders), the interface responds immediately to user actions. PyQt5's signal-slot mechanism allows developers to define specific responses, enabling dynamic interaction between user inputs and system feedback.

4. Model Design

4.1. Key Point Identification

MediaPipe is a cross-platform framework launched by Google that can build machine learning pipelines for processing time series data such as video and audio. (Liu, 2022) BlazePose is a built-in posture detection model in MediaPipe that can detect and track 33 key points of the human body in real-time, as shown in Figure 3.

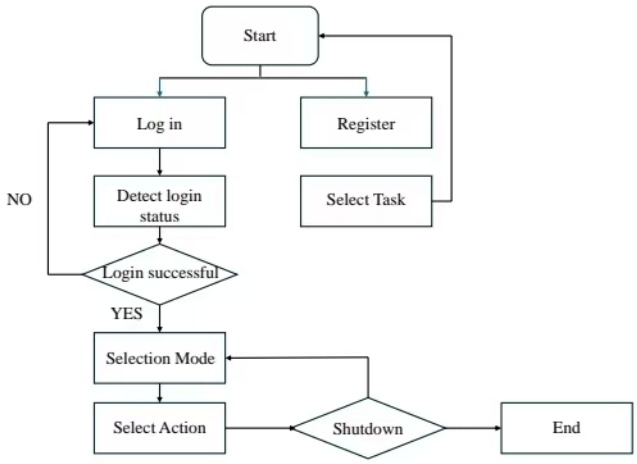

Figure 2: Front-end Interface Function Control Flow Chart

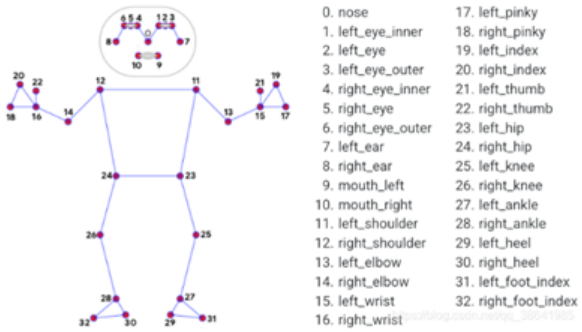

Figure 3: Keypoint Topology, Based on Bazarevsky et al. (2020).

It uses convolutional neural networks (CNN) for human posture detection. Its model structure is relatively lightweight, and the detection speed is breakneck, which is particularly suitable for real-time applications. (Kong and Liu, 2023)

Although Mediapipe BlazePose outputs 3D coordinates (x, y, z), these values do not represent actual spatial positions. Instead, they are normalized within a circumscribed circle centered at the midpoint of the hips, aligning the human body to the square input image of the neural network. Given facial features' high contrast and consistency, BlazePose utilizes a lightweight on-device face detector (Bazarevsky et al., 2019) as a proxy for person detection. This detector estimates key alignment parameters, including the hip center, the radius of the circumscribed circle, and body inclination (Bazarevsky et al., 2020), as illustrated in Figure 4.

Only the x and y coordinates of key points were used in our experiment. Calculating joint angles is closely related to specific movements, as different exercises involve biomechanical analyses of various joints. Therefore, this study focuses on the squat as a representative exercise; other movements can be analyzed using similar methods.

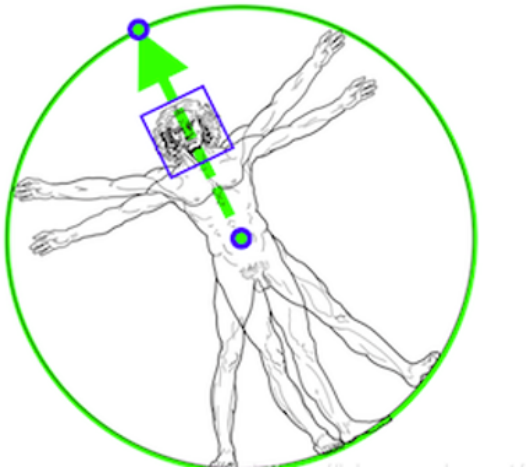

Figure 4: Mediapipe Coordinate Principle

Previous research has shown a significant relationship between knee joint motion and squat depth. In contrast, movements of the hip and ankle joints are closely associated with anterior-posterior displacement of the center of mass (COM) during squatting. Multiple linear regression analysis has further demonstrated that knee and ankle motions can predict squat depth and that the combined hip, knee, and ankle motion can predict COM displacement. Specifically, the knee contributes primarily to vertical COM movement, the hip to anterior-posterior displacement, and the ankle to both (Kasahara et al., 2024).

Based on this, we selected the key points corresponding to the hip, knee, and ankle joints—landmarks 24, 26, and 28 on the right side or 23, 25, and 27 on the left side in the Mediapipe framework. In related studies, it was found that the average ankle dorsiflexion angle during squatting was 23.4-25.9°, the average knee flexion angle was 124°, and the average hip flexion angle was 124-125°, with no significant difference between the left and right sides. The squat action is symmetrical, so our experiment only needs to analyze the key points on one side (Endo et al., 2020). The key point map is shown in Figure 5.

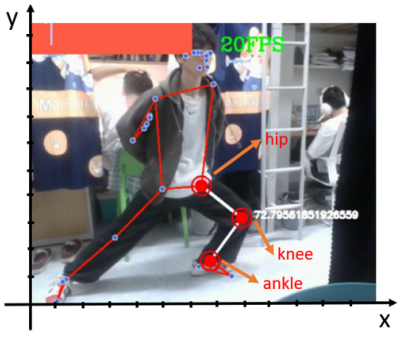

Figure 5: Landmarks Three-coordinate Point Information

4.2. Calculate Key Point Angle and Posture Accuracy

After acquiring the coordinates of each key point, the system computes joint angles and establishes corresponding thresholds. Comparing these angles with the thresholds in real time enables action counting and accuracy evaluation.

In a related study, researchers compared the activation of lower limb muscles at three different knee angles (20°, 90°, and 140°) during the maximum isometric back squat. The results showed that the quadriceps muscle activation was the highest at a 90° knee angle, while the gluteus maximus was more activated at 20° and 90° than at 140° (Henrique et al., 2016). We assumed 75° as the threshold angle for the squat movement to be completed to obtain more generalizable results.

We set the knee flexion angle to the left knee as the vertex, which is calculated based on three anatomical key points: hip, knee, and ankle. These key points are represented by their respective 2D coordinates (x_0,x_1),(y_0,y_1) and (z_0,z_1). The system computes two angles: T1, which represents the orientation of the upper leg (from hip to knee), and T2, which represents the orientation of the lower leg (from knee to ankle). These are calculated using the math.atan2 function as follows:

$$T1 = math.atan2(x_1 - y_1, x_0 - y_0) \tag{1}$$

$$T2 = math.atan2(z_1 - x_1, z_0 - y_0) \tag{2}$$

The bending angle at the knee is then obtained as the absolute difference between T1 and T2:

$$angle = |T1 - T2| \tag{3}$$

Angles below 10° indicate a standing posture, while angles greater than 75° define a squatting posture, which is the only state counted by the system. When the angle lies between these thresholds, the system evaluates how far the motion has progressed toward a squat. This is visualized through a progress bar: 0% represents a standing position, 50% indicates the motion is in progress, and 100% signifies a completed squat. Once the squat is completed, the system resets and waits for the user to return to the standing state.

This completes the system's action counting and accuracy evaluation, as illustrated in Figure 6.

5. Experiments

To demonstrate the advantages of the Mediapipe framework in lightweight, innovative fitness equipment, we conducted a test comparison with the OpenPose posture recognition framework under the same relatively low-end configuration conditions. The test equipment configuration is shown in Table 1.

Table 1: Computer Configuration Information

Hardware name	Configuration information
processor	12th Gen Intel(R) Core(TM) i5-1235U
Graphics card	Intel VGA Driver
Camera	Sunplus Camera Driver
Development language	Python3.8

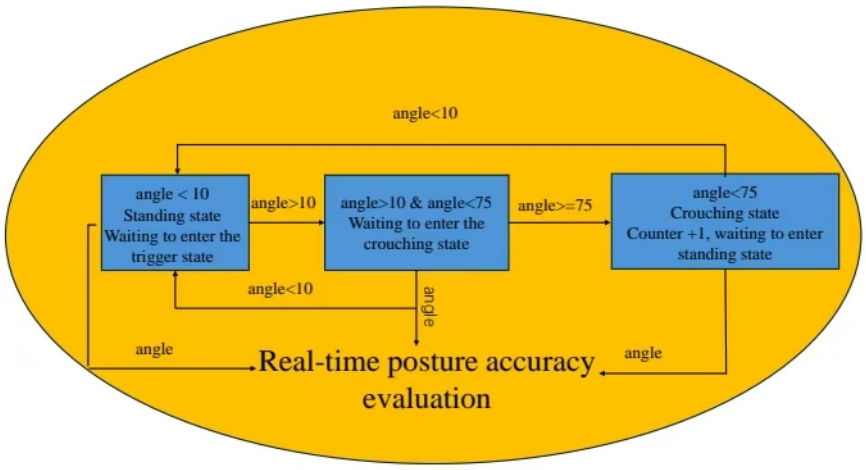

Figure 6: Gesture Recognition Logic Structure

5.1. Model Function Test

The innovative fitness system was tested under Table 1 configuration. The standing, transitional, and squatting states were recorded to verify the stability and reliability of the counting and accuracy evaluation functions. The functional test scenarios for these three states are shown in Figure 7.

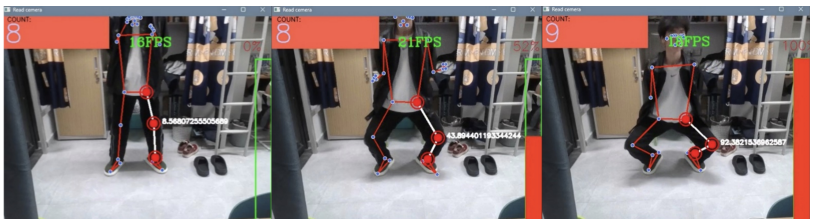

Figure 7: Posture Test

According to the test results, the standing state yielded an angle of 8.568, the waiting state 43.894, and the squatting state 92.382. When the standing angle (8.568) exceeds 10°, the system transitions to the waiting state. Upon reaching an angle greater than 75°, the system automatically increments the counter, activating the counting function. The progress bar and percentage in the waiting state also allow users to gauge whether their posture meets the required limb angles. This feature helps prevent muscle strain from improper form and provides intuitive feedback on action completion, demonstrating the practicality of the accuracy evaluation function.

5.2. Compared Methods

To evaluate model performance comprehensively, we compare different pose estimation methods, including OpenPose and BlazePose, under varying lighting conditions and levels of occlusion. Specifically, we employ the Percentage of Correct Points (PCK) metric and counting accuracy to assess the precision of keypoint localization and the robustness of human detection in challeng-

ing environments. The relevant evaluation metrics and their corresponding formulas are presented below for a more detailed and quantitative comparison.

PCK measures the proportion of predicted key points within a certain normalized distance from the ground-truth positions. It is defined as:

$$\text{PCK} = \frac{1}{N}\sum_{i=1}^{N}\delta\left(\frac{\|\hat{K}_i - K_i\|_2}{d} < \alpha\right) \tag{4}$$

Where N is the number of key points, $\hat{K}_i$ is the predicted location of the i-th keypoint, K_i is the ground-truth location, d is a normalization factor (e.g., head size or torso length), α is the threshold (commonly 0.2), and $\delta(\cdot)$ is the indicator function.

Counting Accuracy evaluates the correctness of the predicted number of detected persons, defined as:

$$\text{Counting Accuracy} = 1 - \frac{|\hat{C} - C|}{C} \tag{5}$$

where $\hat{C}$ is the predicted number of people, and C is the ground-truth count. This metric is valid when $C \neq 0$.

6. Experimental Results

Under normal lighting conditions (500 lux), OpenPose achieves a PCK@0.2 of 85.3 % with perfect 100

The robustness comparison reveals significant differences: in low-light conditions (100 lux), OpenPose maintains exceptional stability (PCK@0.2: 85.42%, 100% accuracy), whereas BlazePose experiences noticeable degradation (PCK@0.2: 78.4 accuracy: 88.2%). This performance gap stems from BlazePose's architectural dependence on clear facial recognition, which becomes compromised under suboptimal lighting.

Occlusion scenarios further highlight these divergences. With both thighs occluded, OpenPose demonstrates remarkable resilience (PCK@0.2: 83.3%, accuracy: 93.3%), while BlazePose suffers significant performance drops (PCK@0.2: 72.2%, accuracy: 80.8%). This limitation originates from BlazePose's reliance on face and shoulder key points for full-body estimation - when these reference points are obscured, the system struggles to maintain accuracy. The experimental data is shown in Table 2.

Table 2: Model under different conditions PCK@0.2 And counting accuracy

Model	**Condition**	**PCK@0.2**	**Counting Accuracy**
OpenPose	Normal Light(500lux)	85.3%	100.0%
OpenPose	Low Light(100lux)	85.42%	100.0%
OpenPose	Shielding (both thighs)	83.3%	93.3%
OpenPose	Occlusion (limbs)	86.6%	98.1%
BlazePose	Normal Light(500lux)	83.1%	97.4%
BlazePose	Low Light(100lux)	78.4%	88.2%
BlazePose	Shielding (both thighs)	72.2%	80.8%
BlazePose	Occlusion (limbs)	-	-

However, OpenPose's superior accuracy comes at substantial computational costs. As shown in Table 1, our frame rate measurements (fps=1/(currentTime-previousTime)) reveal Mediapipe's stable 25fps output, while OpenPose barely reaches 3fps with severe instability, frequently maxing out CPU utilization. This computational burden is further evidenced in Figure 9, which displays CPU usage patterns during squat motions under identical test conditions (same configuration, lighting, and no obstructions). The testing environment is shown in Figure 8.

A stark contrast is evident in the CPU occupancy rates between the OpenPose and BlazePose frameworks. It is apparent that the OpenPose framework requires substantially more computational resources for potential commercial use, whereas MediaPipe demonstrates a generally stable CPU utilization rate, with only slight fluctuations that we deem acceptable.

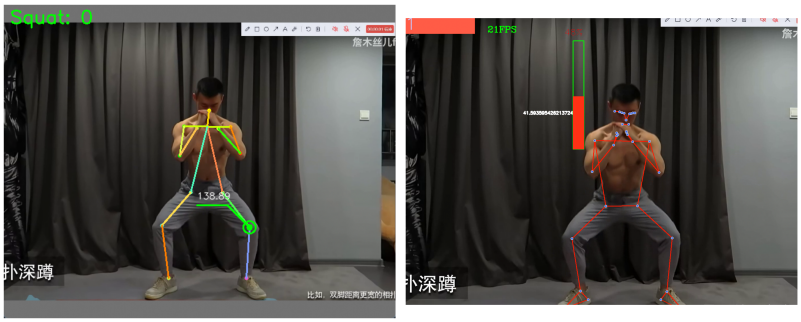

Figure 8: Enter Caption

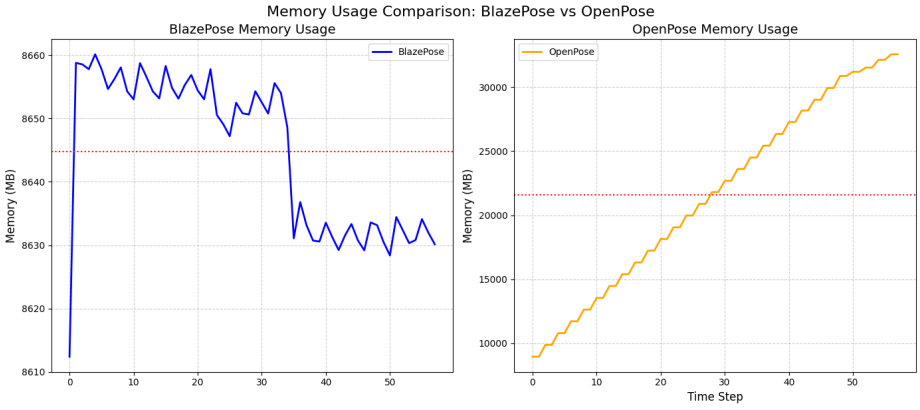

Figure 9: Memory Usage Comparison: BlazePose vs OpenPose

Furthermore, Mediapipe's developers conducted comparative experiments between recognition models in the Mediapipe and OpenPose frameworks. According to the Google testing team, BlazePose executes 25-75 times faster on a mid-range mobile CPU than OpenPose on a 20-core desktop CPU (Bazarevsky et al., 2020). The test results are shown in Table 3.

Table 3: BlazePose vs OpenPose,based on Bazarevsky et al. (2020)

Model	FPS	AR Dataset,PCK@0.2	Yoga Dataset PCK@0.2
OpenPose(Body only)	0.4	87.8	83.4
BlazePose Full	10	84.1	84.5
BlazePose Lite	31	79.6	77.6

7. Conclusion and Outlook

This study presents a lightweight, intelligent fitness system based on the Mediapipe BlazePose framework, showcasing significant advancements in computational efficiency, adaptive thresholding, and practical deployment. Using a 2D coordinate-based angle calculation method, the system achieves real-time performance (25 FPS) on low-end hardware (Intel i5-1235U) while reducing memory usage by 89% compared to OpenPose, maintaining a 97.4% counting accuracy under optimal conditions. Integrating with PyQt5 ensures cross-platform compatibility, enabling seamless deployment on micro-mobile devices without needing specialized hardware. Experimental results validate the successful balance between accuracy and efficiency, meeting the demand for accessible fitness technology in resource-constrained environments. Future work identifies several directions for improvement, including enhancing occlusion resilience through attention mechanisms, implementing personalized feedback via user-specific threshold calibration, optimizing the system for ARM-based devices using TensorFlow Lite quantization, and integrating injury prevention alerts through joint load analysis based on 3D pose estimation. These advancements aim to transform the system into a comprehensive digital fitness assistant, bridging the gap between professional coaching and broader accessibility.

References

Valentin Bazarevsky, Yury Kartynnik, Andrey Vakunov, Karthik Raveendran, and Matthias Grundmann. Blazeface: Sub-millisecond neural face detection on mobile gpus. *arXiv e-prints*, abs/1907.05047:arXiv:1907.05047, 2019. doi: 10.48550/arXiv.1907.05047.

Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. Blazepose: On-device real-time body pose tracking. *arXiv e-prints*, abs/2006.10204:arXiv:2006.10204, 2020. doi: 10.48550/arXiv.2006.10204.

Zhe Cao, Gines Hidalgo, Tomas Simon, Shih En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:172——186, 2018. doi: 10.1109/TPAMI.2019.2929257.

Yasuhiro Endo, Masashi Miura, and Masaaki Sakamoto. The relationship between the deep squat movement and the hip, knee and ankle range of motion and muscle strength. *Journal of Physical Therapy Science*, 32(6):391–394, 2020. doi: 10.1589/jpts.32.391.

Yanyan Gao, Haopan Ren, and Dejian Wei. ULPN: A lightweight pose estimation method based on improved LPN. *Computer and Digital Engineering*, 52:502–506, 2024. doi: 10.3969/j.issn.1672-9722.2024.02.038.

Marchetti Paulo Henrique, Jarbas Da Silva Josinaldo, Jon Schoenfeld Brad, Nardi Priscyla Silva Monteiro, Pecoraro Silvio Luis, D'Andréa Greve Julia Maria, and Hartigan Erin. Muscle activation differs between three different knee joint-angle positions during a maximal isometric back squat exercise. *Journal of Sports Medicine*, 2016:1–6, 2016. doi: 10.1155/2016/3846123.

Jian Jiang, Qi Zhang, and Caiyong Wang. Development of a comprehensive digital image processing experimental platform based on OpenCV and PyQt. *Computer Knowledge and Technology*, 19:6–8+13, 2023. doi: 10.14004/j.cnki.ckt.2023.1253.

Satoshi Kasahara, Tomoya Ishida, Jiang Linjing, Ami Chiba, Mina Samukawa, and Harukazu Tohyama. Relationship among the com motion, the lower extremity and the trunk during the squat. *Journal of Human Kinetics*, 93:29, 2024. doi: 10.5114/jhk/183066.

Yaqi Kong and Yu Liu. Design and implementation of fitness counting system based on BlazePose and KNN. *Software Engineering*, 26:58–62, 2023. doi: 10.19644/j.cnki.issn2096-1472.2023.007.013.

Defa Liu. Digital gesture recognition based on MediaPipe. *Electronic Production*, 30:55–57, 2022. doi: 10.3969/j.issn.1006-5059.2022.14.017.

Jose Sigut, Miguel Castro, Rafael Arnay, and Marta Sigut. OpenCV basics: A mobile application to support the teaching of computer vision concepts. *IEEE Transactions on Education*, 63:1–8, 2020. doi: 10.1109/TE.2020.2993013.

Yihan Wang, Muyang Li, Han Cai, Wei Ming Chen, and Song Han. Lite pose: Efficient architecture design for 2d human pose estimation. *arXiv e-prints*, page 13116——13126, 2022. doi: 10.1109/TPAMI.2019.2929257.