FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 10, October, 2025, pp 174 – 180
DOI: https://doi.org/10.33003/fjs-2025-0910-3908

INTUITIVE GESTURE-DRIVEN AUGMENTED REALITY FOR USER INTERACTION IN LOW-COST AR SYSTEMS

***Bilkisu Larai Muhammad-Bello, Awoke Victor Ndubuisi, Muhammad Aliyu Suleiman and Ibrahim Anka Salihu**

Department of Software Engineering, Nile University of Nigeria, Abuja, Nigeria.

*Corresponding author: bilkisu.muhammad-bello@nileuniversity.edu.ng

ABSTRACT

Augmented Reality (AR) offers promising applications in education, healthcare, and industry, but its adoption in low-resource settings is hindered by expensive hardware and complex interaction methods. This study presents a lightweight, gesture-driven AR interface designed to enable intuitive user interaction on low-cost devices. Using standard webcams and open-source tools, MediaPipe for hand landmark detection and OpenCV for visual overlays, a rule-based system was developed to recognize three core gestures (open palm, fist, point) and trigger real-time AR overlays. The system was optimized for latency and frame rate, achieving 81.8% accuracy in gesture recognition during user testing with 20 participants. Real-time performance averaged 44.8 ms latency and 24.6 FPS, demonstrating responsive feedback on entry-level hardware. Despite limitations such as variable lighting and lack of depth sensing, the system proved intuitive and effective for immersive interaction. This work contributes a scalable AR interaction model that enhances accessibility in educational and low-cost environments.

Keywords: Augmented Reality, Gesture Recognition, Human-Computer Interaction, Low-Cost AR, MediaPipe, OpenCV

INTRODUCTION

Augmented Reality (AR) is a rapidly evolving technology that blends digital content with the physical world, enabling users to interact with virtual objects while maintaining awareness of their real environment (Vertucci et al., 2023). Unlike Virtual Reality (VR), which fully immerses the user in a simulated setting, AR overlays interactive elements onto the physical space, enhancing perception and interaction. Its unique capability to integrate virtual components into real-world tasks has driven adoption across sectors such as education, healthcare, industrial training, manufacturing, and entertainment (Cao et al., 2022).

Effective AR experiences depend heavily on intuitive user interaction. While traditional input methods such as touchscreens, handheld controllers, and voice commands have enabled basic engagement, they often fall short in delivering truly immersive, hands-free, and context-aware interaction. Gesture-driven interaction, which relies on natural human hand movements, offers a compelling alternative (Ghazwani and Smith, 2020). It reduces cognitive load, fosters immersion, and aligns closely with natural human behaviours (Syed et al., 2023). This is particularly valuable in mobile or head-mounted AR systems where maintaining freedom of movement is essential.

However, implementing gesture-driven interfaces in low-cost AR systems presents significant challenges. High-end AR devices, such as Microsoft HoloLens and Magic Leap, leverage depth sensors and advanced processing capabilities to achieve accurate, real-time gesture recognition. In contrast, affordable platforms built on smartphones, laptops, or entry-level AR headsets lack such hardware, making it difficult to achieve precise tracking and responsive overlays (Majdoub Bhiri et al., 2023). Additionally, variations in lighting, limited computational resources, and dependence on proprietary software further restrict the usability of low-cost solutions in resource-constrained environments.

The growing need for accessible AR technologies in education, training, and developing regions highlights the importance of research into lightweight, hardware-agnostic interaction models (Shen et al., 2022). This study addresses these challenges by designing and evaluating a gesture-driven AR system that uses only standard webcams and open-source frameworks, Google's MediaPipe for hand landmark detection and OpenCV for rendering visual overlays. The system is optimized for real-time performance on modest hardware, aiming to deliver an intuitive, scalable, and affordable interaction model that can democratize access to immersive AR experiences.

Applied Techniques for Gesture-Driven Augmented Reality

The design of the proposed gesture-driven AR system relied on a combination of well-established computer vision frameworks and lightweight rule-based classification strategies, chosen to balance performance, accessibility, and computational efficiency. Three major techniques underpinned the implementation: hand landmark detection using MediaPipe, image processing with OpenCV, and rule-based gesture classification.

Hand Landmark Detection with MediaPipe

MediaPipe (Lugaresi et al., 2019) was employed as the core framework for real-time hand tracking. Its hand detection model is based on BlazePalm, which performs palm localization in images with high efficiency, followed by regression models that estimate 21 three-dimensional hand landmarks. MediaPipe's design ensures robustness to variations in lighting, orientation, and background, making it well-suited for real-world deployments. Compared to sensor-based hardware like Leap Motion (Guna et al., 2014) and Kinect (Sanna et al., 2013), MediaPipe provides markerless, camera-only tracking, reducing cost and complexity while retaining acceptable accuracy.

Image Processing and Augmented Overlays with OpenCV

The visualization and augmentation pipeline was implemented using the OpenCV library. OpenCV was responsible for drawing bounding boxes, lines, and overlays to represent detected gestures within the AR interface. The use of OpenCV ensured cross-platform compatibility, with

optimized performance on commodity hardware such as standard webcams and mid-range CPUs. This choice aligns with the study's low-cost design philosophy, avoiding reliance on dedicated GPUs or specialized vision hardware.

Rule-Based Gesture Classification

Once hand landmarks were detected, gestures were classified using a rule-based approach (Zhou and Deng, 2017). The algorithm compared relative positions of key landmarks such as fingertip and joint coordinates to predefined thresholds representing specific gestures (e.g., open palm, closed fist, pointing). While deep learning-based classifiers (Zhang and Zhu, 2019) can offer higher accuracy, the rule-based approach was intentionally chosen for its transparency, interpretability, and computational efficiency. This decision ensured that the system remained deployable on devices with limited resources, without sacrificing real-time responsiveness.

Rationale for Technique Selection

The integration of these techniques reflects a deliberate trade-off between accuracy, efficiency, and accessibility. Unlike hardware-dependent systems, the adopted pipeline requires only a standard webcam and mid-tier computing device, making it viable in educational, training, and low-resource environments. Furthermore, the reliance on open-source frameworks enhances reproducibility and reduces barriers for researchers and practitioners aiming to replicate or extend this work.

Related Works

Gesture recognition has been extensively explored as a natural mode of human–computer interaction, with applications spanning gaming, virtual environments, and assistive technologies. Early surveys such as Pavlovic et al. (1997) and Rautaray and Agrawal (2015) classify approaches into vision-based and sensor-based methods. Vision-based systems leverage cameras to detect and track hands in RGB or depth data, while sensor-based systems rely on wearable devices or inertial measurement units (IMUs). Sensor-based approaches such as the Leap Motion Controller (Guna et al., 2014) and Kinect (Sanna et al., 2013) offer high precision but often require specialized hardware, limiting portability and increasing deployment costs.

In vision-based gesture recognition, advances in machine learning and computer vision have enabled real-time, markerless tracking of hand positions and gestures. Frameworks like MediaPipe Hands (Lugaresi et al., 2019; Tati et al., 2019) provide efficient 2D and 3D landmark detection from monocular RGB video, eliminating the need for depth sensors. Rule-based classification methods (Zhou and Deng, 2017) remain attractive for lightweight systems due to their low computational overhead, although deep learning approaches (Zhang and Zhu, 2019) can offer higher accuracy when sufficient computational resources are available.

Augmented Reality (AR) interaction research spans diverse application domains, from industrial maintenance to education. Foundational surveys by Azuma (1997), Carmigniani et al. (2011), and Billinghurst et al. (2015) highlight the importance of seamless, intuitive input modalities in enhancing immersion. Comparative studies on AR interfaces (Lee et al., 2018; Wang et al., 2013) show that gesture-based interaction increases user engagement and reduces the need for additional controllers, making it particularly appealing for training and educational contexts. However, much of the high-performance AR work relies on costly hardware configurations, leaving a gap in accessible solutions for resource-constrained settings.

The present study builds on this body of work by designing a purely vision-based, gesture-driven AR system optimized for low-cost deployment. By combining MediaPipe's robust hand landmark detection with a rule-based classifier and lightweight OpenCV overlays, the system achieves real-time performance on standard webcams without dedicated GPUs or depth sensors. This approach directly addresses the limitations of both sensor-dependent solutions and computationally heavy deep learning methods, contributing a replicable framework for affordable, accessible AR interaction.

MATERIALS AND METHODS

System Overview

The proposed gesture-driven AR system was designed to function without specialized hardware, relying instead on standard webcams and open-source software frameworks. The development goal was to create a lightweight, real-time pipeline capable of recognizing intuitive hand gestures and translating them into responsive AR overlays, while maintaining compatibility with low-specification devices such as entry-level laptops and mobile platforms. The system architecture comprised three primary modules: hand landmark detection, gesture classification, and AR overlay rendering as depicted in

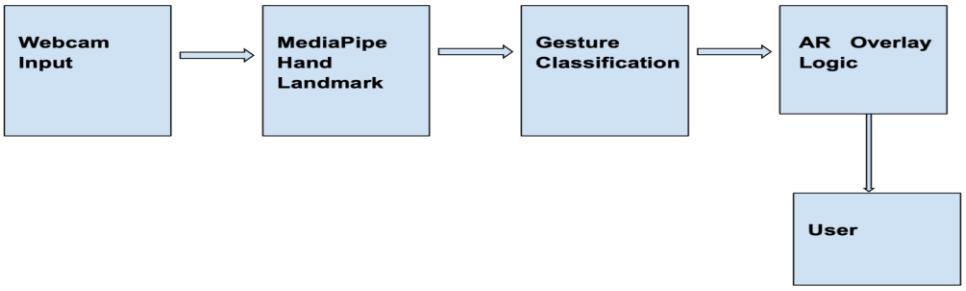

Figure 1: System Architecture

Gesture Detection and Tracking

Hand landmark detection was implemented using MediaPipe Hands, an open-source framework developed by Google. MediaPipe detects 21 distinct 3D hand landmarks from monocular RGB input, allowing robust tracking even under moderate variations in hand orientation and lighting. The choice of MediaPipe was motivated by its low computational overhead and proven real-time performance on CPU-based systems, making it ideal for environments without discrete GPUs or depth sensors (Meng et al., 2024).

Gesture Classification Approach

A rule-based classification method was employed to ensure fast decision-making without the need for training machine

learning models. This approach interprets the spatial relationships between landmarks to classify gestures into three categories*: Open Palm* – all fingers extended; *Fist* – all fingers curled; *Point* – index finger extended with remaining fingers curled. Relative Euclidean distances between fingertip coordinates and the palm base (Landmark 9) formed the basis of classification logic. The simplicity of this approach ensured high interpretability and minimized processing delays.

AR Overlay Logic

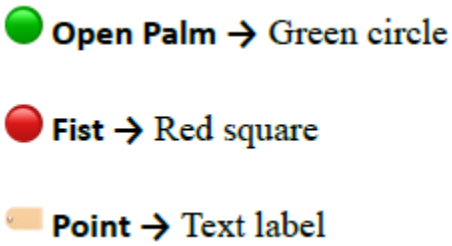

Figure 2: AR Overlay Logic

Recognized gestures triggered dynamic visual overlays rendered with OpenCV. Overlays were anchored to the palm base and updated frame-by-frame, maintaining spatial coherence with the user's hand movement. For example, an Open Palm displayed a green circle, a Fist showed a red circle, and a Point gesture rendered a textual label as shown in Figure 2. This visual feedback reinforced gesture recognition accuracy and provided a responsive user experience.

Simulated Data Pipeline

A synthetic data stream was implemented to simulate gesture inputs and system behavior during offline development or testing. This simulation mimicked real-world metrics as shown in Table 1.

Table 1: Simulated Data Pipeline

Simulated Gesture Inputs	Values/Description
Simulated Accuracy	~85%
Latency	~45ms ± 10ms
Frames per second (FPS)	~25 ± 2
Gestures	Randomized mix of Open Palm, Fist, Point, and Unknown

This environment enabled early debugging, visualization (latency, FPS, frequency charts), and classifier refinement before live deployment. The synthetic gesture data stream was also used to: generate confusion matrices and gesture transition matrices.

Performance Evaluation Metrics

Three core technical metrics were used to assess system performance: (1) *Latency per frame (ms)* – the time from video capture to overlay display; (2) *Frames per second (FPS)* – the average processing throughput; (3) *Classification Accuracy* – this is measured using Precision, Recall, and F1 scores from confusion matrix analysis. These metrics were evaluated under both simulated and real-world conditions to verify responsiveness and stability.

User Study Protocol

A usability evaluation was conducted with 20 participants representing diverse age ranges, handedness, and levels of technical familiarity. Each participant performed repeated instances of the three target gestures, along with a non-gesture "Unknown" state, yielding a dataset of 2,000 labeled gesture samples. Testing took place in a standard indoor setting with ambient lighting and a 720p webcam. Participants received minimal instruction to reflect realistic first-time usage scenarios.

Data Analysis Tools

Table 2 summarizes the tools and libraries used with justification for each. ***NumPy***, ***Matplotlib***, and ***Seaborn*** were used for numerical analysis and visualization of latency, FPS trends, and gesture frequency distributions. Scikit-learn was employed for classification performance evaluation, generating precision-recall metrics and confusion matrices.

Table 2: Tools and Libraries used

Component	Tool/Library	Justification
Gesture Detection	MediaPipe(Google)	Offers real-time, lightweight hand tracking with 21 key landmarks per hand; ideal for low-latency interaction on mobile/embedded devices.
Video Capture & AR Overlays	OpenCV	Enables real-time camera feed capture and dynamic overlay of AR elements, supporting fast prototyping and custom visualization.
Performance Analysis	NumPy, Matplotlib, Seaborn	These Python libraries are used for numerical computation and visualization of performance data, such as gesture recognition timing and user interaction metrics.
Classification Evaluation	scikit-learn	Provides tools to assess gesture classification performance, including confusion matrix, precision, recall, and F1 score metrics.

RESULTS AND DISCUSSION

Technical Performance

The gesture-driven AR system demonstrated strong real-time performance on low-specification hardware. Under real-world testing with a laptop (Intel i5, 8GB RAM, integrated webcam), the system achieved an *average latency of 44.8 ms* and an *average frame rate of 24.6 FPS* as depicted in Figures 3 and 4. Latency remained within an acceptable range (23–78 ms), indicating consistent responsiveness for interactive applications. Although simulation-based testing yielded extremely high FPS values, these reflected ideal, non-rendered conditions and were not considered representative of real-world performance.

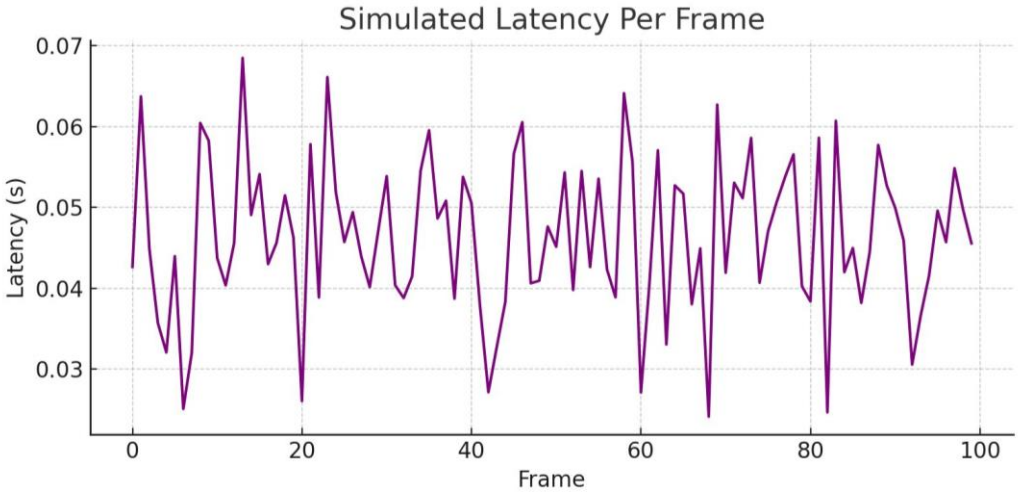

Figure 3: Simulated Latency per Frame for Real World Testing

Table 3 summarizes the results. Gesture recognition accuracy was evaluated using the dataset of 2,000 labeled samples from the user study. The overall classification accuracy was *81.8%,* with the *Fist* gesture achieving the highest reliability (Precision: 90%, Recall: 85%, F1-score: 87.4%). The *Open Palm* gesture, while the most frequently performed, exhibited slightly lower recall (76%), suggesting occasional misclassification when hand visibility or finger extension was partial. The *Point* gesture was recognized with reasonable accuracy (Precision: 72%, Recall: 82%) but showed susceptibility to misclassification as *Unknown*, particularly under varied lighting.

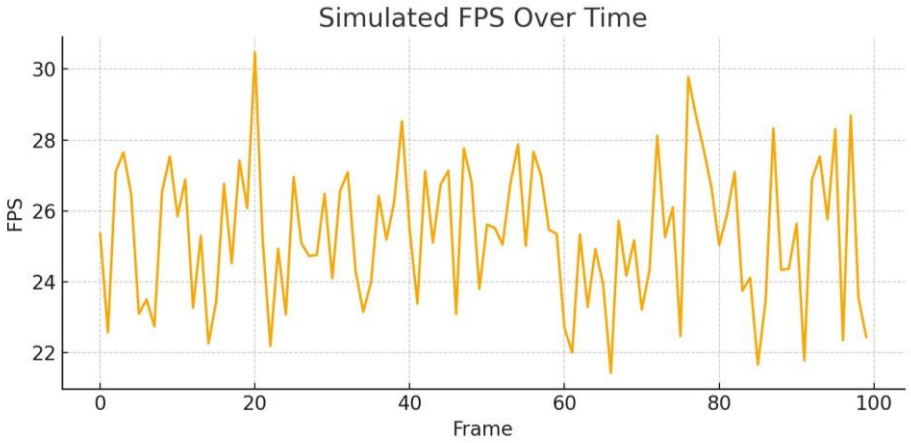

Figure 4: Simulated FPS Over Time for Real World Testing

Table 3: Gesture Classification Metrics (N=2000 samples)

Gesture	Precision	Recall	F1 Score	Support
Open Palm	88%	76%	81.6%	720
Fist	90%	85%	87.4%	600
Point	72%	82%	76.7%	400
Unknown	65%	80%	71.7%	280
Overall Accuracy	—	—	**81.8%**	2000

Gesture frequency analysis revealed that *Open Palm* dominated usage patterns, functioning as a natural default hand state during interaction. This suggests it could serve as an activation or standby gesture in interface design. The *Fist* gesture's high classification performance supports its inclusion for tasks requiring precise activation, while *Point* gestures, though slightly less reliable offer valuable directional input capability. The gesture classification confusion matrix is depicted in Figure 5, while Figure 6 shows the gesture frequency over 100 Frames.

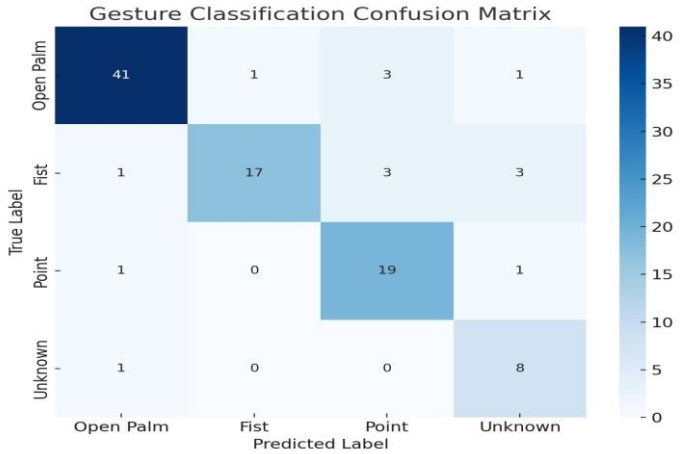

Figure 5: Confusion Matrix Visualization

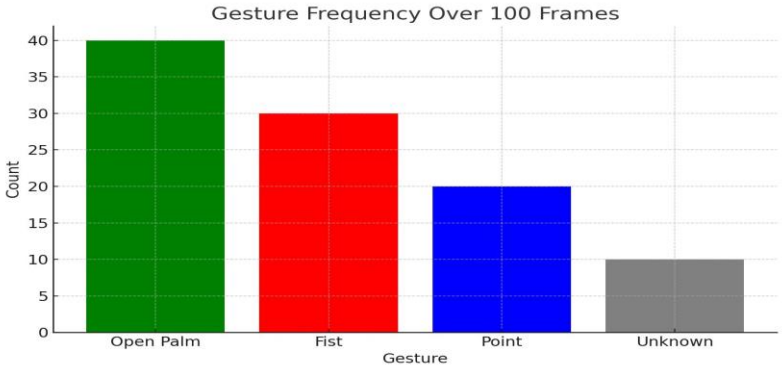

Figure 6: Gesture Frequency Histogram

Usability Outcomes

To better understand gesture dynamics during actual use, a gesture transition matrix was generated as depicted in Figure 7. This shows how often users shifted from gesture to another. It reveals patterns like "Fist →Open Palm → Point" which informs design of gesture-based menus and interaction logic. Frequent transitions between distinct gestures suggest the system's potential for multi-stage interaction workflows. For example, activating a menu with one gesture and selecting with another.

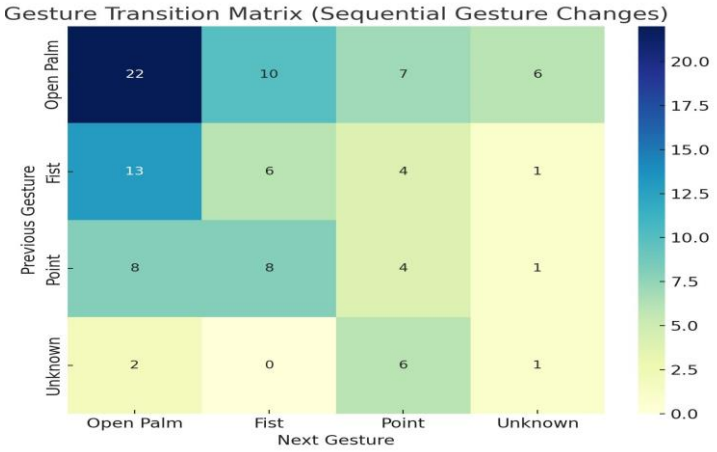

Figure 7: Gesture Transition Matrix

Participants reported that gestures were easy to learn and did not require significant training. Most users adapted quickly to the system, with interaction becoming natural within minutes of engagement. Minimal cognitive load and the lack of external controllers contributed to an immersive experience. Visual feedback in the form of overlays directly tied to gesture recognition reinforced user confidence in system responses.

Discussion

The results confirm that a lightweight, rule-based gesture recognition system can enable effective and intuitive AR interaction on low-cost platforms without specialized hardware. Achieving *sub-50 ms latency* and maintaining *over 80% classification accuracy* demonstrates that such systems

are viable for real-time applications, including educational simulations, training environments, and assistive technology. From a human-computer interaction perspective, the *Open Palm* gesture's frequency underscores the importance of aligning gesture sets with natural resting states to minimize user fatigue. Its slightly lower recall rate, however, points to the need for adaptive thresholding or calibration features to accommodate individual differences in finger positioning. The *Fist* gesture's high precision indicates that closed-hand configurations produce distinctive landmark patterns that are more reliably detected under varied conditions.

The performance of the *Point* gesture highlights a key limitation of rule-based classifiers: their sensitivity to small variations in hand posture and environmental factors such as lighting. In practical terms, this means that while the system is suitable for well-lit indoor environments, deployment in outdoor or highly dynamic lighting conditions may require more robust recognition strategies potentially integrating machine learning for adaptability.

The absence of depth sensing is another constraint, as it limits interaction to 2D overlays. While sufficient for the tested use cases, applications requiring spatial depth interaction such as object manipulation in 3D space, would require depth-aware inputs or stereo camera setups (Torres et al., 2024). This, however, must be balanced against cost and accessibility objectives.

Importantly, the study demonstrates that accessibility and scalability need not be sacrificed for interactivity. The use of open-source tools like MediaPipe and OpenCV ensures that the system can be reproduced and customized without licensing costs, supporting equitable access to AR technologies in resource-constrained contexts. This aligns with broader digital inclusion goals and positions the system as a potential platform for education in developing regions.

CONCLUSION

This study set out to address the challenge of delivering intuitive, hands-free user interaction in low-cost Augmented Reality (AR) environments by developing a lightweight, gesture-driven AR system using only standard webcams and open-source tools. The system's design rooted in rule-based gesture classification and optimized for minimal latency demonstrated that responsive, real-time AR experiences can be achieved without specialized hardware or high computational power.

Technical evaluation confirmed that the system maintained an average latency of under 50 ms and a frame rate exceeding 24 FPS in real-world testing, while delivering an overall gesture recognition accuracy of 81.8%. These performance levels meet the requirements for smooth AR interaction in educational, training, and general-purpose applications, even in resource-constrained settings. The user study validated the intuitiveness and usability of the selected gestures (Open Palm, Fist, and Point), with participants adapting quickly and interacting confidently without extensive instruction.

The findings underscore that accessibility and affordability need not come at the expense of interaction quality. By relying entirely on open-source frameworks such as MediaPipe and OpenCV, the system eliminates licensing costs and supports reproducibility, making it a practical option for deployment in classrooms, community training programs, and emerging markets where high-end AR systems are financially and logistically inaccessible.

Beyond its immediate technical success, the research contributes to the broader discourse on democratizing immersive technologies. It offers empirical performance benchmarks for gesture-based AR in low-cost environments and presents a replicable methodology that can serve as a foundation for further innovations. The system's modular architecture allows for future enhancements, including the integration of adaptive machine learning classifiers, depth-aware interaction, and multimodal input combining gesture with voice or gaze tracking.

In conclusion, this work demonstrates that carefully engineered, resource-efficient solutions can bridge the gap between the aspirations of AR technology and the realities of limited-resource contexts. It not only advances the technical feasibility of low-cost gesture-driven AR but also reinforces its potential as an inclusive, scalable, and impactful interaction model for the next generation of accessible immersive systems.

ACKNOWLEDGEMENT

The authors thank the participants of the user study and the Department of Software Engineering, Nile University of Nigeria, for technical and institutional support.

REFERENCES

Azuma, R. T. (1997). A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(4), 355–385. https://doi.org/10.1162/pres.1997.6.4.355

Billinghurst, M., Clark, A., and Lee, G. (2015). A survey of augmented reality. Foundations and Trends® in Human–Computer Interaction, 8(2–3), 73–272. https://doi.org/10.1561/1100000049

Cao, J., Lam, K.-Y., Lee, L.-H., Liu, X., Hui, P., and Su, X. (2022). Mobile augmented reality: User interfaces, frameworks, and intelligence. ACM Computing Surveys, 55(9). https://doi.org/10.1145/3557999

Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., and Ivkovic, M. (2011). Augmented reality technologies, systems and applications. Multimedia Tools and Applications, 51(1), 341–377. https://doi.org/10.1007/s11042-010-0660-6

Ghazwani, Y., and Smith, S. (2020). Interaction in augmented reality. In Proceedings of the 2020 4th International Conference on Virtual and Augmented Reality Simulations*. https://doi.org/10.1145/3385378.3385384

Guna, J., Jakus, G., Pogačnik, M., Tomažič, S., and Sodnik, J. (2014). An analysis of the precision and reliability of the Leap Motion sensor and its suitability for static and dynamic tracking. Sensors, 14(2), 3702–3720. https://doi.org/10.3390/s140203702

Lee, K., Lee, J., and Lee, J. Y. (2018). Comparative study on augmented reality approaches for interactive learning. Interactive Learning Environments, 26(4), 516–529. https://doi.org/10.1080/10494820.2017.1337038

Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., ... and Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. arXiv Preprint arXiv:1906.08172. https://arxiv.org/abs/1906.08172

Majdoub Bhiri, N., Ameur, S., Alouani, I., Mahjoub, M. A., and Ben Khalifa, A. (2023). Hand gesture recognition with focus on leap motion: An overview, real-world challenges,

and future directions. Expert Systems with Applications, 226, 120125. https://doi.org/10.1016/j.eswa.2023.120125

Pavlovic, V. I., Sharma, R., and Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7), 677–695. https://doi.org/10.1109/34.598226

Rautaray, S. S., and Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: A survey. Artificial Intelligence Review, 43(1), 1–54. https://doi.org/10.1007/s10462-012-9356-9

Sanna, A., Lamberti, F., Paravati, G., and Manuri, F. (2013). A Kinect-based natural interface for quadrotor control. Entertainment Computing, 4(3), 179–186. https://doi.org/10.1016/j.entcom.2013.01.001

Shen, J., Dudley, J., Mo, G., and Kristensson, P. O. (2022). Gesture spotter: A rapid prototyping tool for key gesture spotting in virtual and augmented reality applications. IEEE Transactions on Visualization and Computer Graphics, 28(11), 3618–3628. https://doi.org/10.1109/tvcg.2022.3203004

Syed, T. A., and et al. (2023). In-depth review of augmented reality: Tracking technologies, development tools, AR displays, collaborative AR, and security concerns. Sensors, 23(1), 146. https://www.mdpi.com/1424-8220/23/1/146

Tati, S., Sahu, N. K., and Rath, S. K. (2019). A robust hand gesture recognition system using MediaPipe framework. International Journal of Interactive Multimedia and Artificial Intelligence, 5(7), 52–59. https://doi.org/10.9781/ijimai.2019.07.003

Torres, W., and et al. (2024). A framework for real-time gestural recognition and augmented reality for industrial applications. Sensors, 24(8), 2407. https://doi.org/10.3390/s24082407

Vertucci, R., D'Onofrio, S., Ricciardi, S., and De Nino, M. (2023). History of augmented reality. In Springer Handbooks (pp. 35–50). https://doi.org/10.1007/978-3-030-67822-7_2

Wang, X., Kim, M. J., Love, P. E., and Kang, S. C. (2013). Augmented reality in built environment: Classification and implications for future research. Automation in Construction, 32, 1–13. https://doi.org/10.1016/j.autcon.2012.11.021

Zhang, X., and Zhu, S. (2019). Real-time hand gesture recognition based on deep features and support vector machines. IEEE Access, 7, 103138–103145. https://doi.org/10.1109/ACCESS.2019.2931064

Zhou, Z., and Deng, Z. (2017). Rule-based hand gesture recognition for human–computer interaction. Multimedia Tools and Applications, 76(19), 20067–20083. https://doi.org/10.1007/s11042-016-4028-5