

Touchless HCI for Media Control Using Hand Gestures

Pushpal Bhar, Subhojit Khatua, Arghya Pratim Biswas

Department of Electronics & Telecommunication Engineering

Jadavpur University

Mentor: Prof Sheli Sinha Chaudhury, Dept of ETCE, Jadavpur University

Date of Submission: 20th February, 2026

ABSTRACT

Natural User Interfaces (NUI) have emerged as a pivotal frontier in Human-Computer Interaction (HCI). This paper proposes a high-efficiency, real-time hand gesture recognition system optimized for edge-computing environments, specifically the **Raspberry Pi 5**. Unlike traditional deep-learning classifiers that suffer from high computational latency, our approach utilizes a **Hierarchical Edge-Inference** architecture. The system integrates MediaPipe's BlazePalm topology with a deterministic **Hierarchical Finite State Machine (HFSM)** to map spatial coordinates to media control commands. Experimental results demonstrate a 95%+ recognition accuracy at distances up to 240 cm (in Lapcam HD 720P (LWC-042)) while maintaining a consistent throughput of 20+ FPS.

Index Terms — Hand Gesture Recognition, Edge Computing, Raspberry Pi 5, MediaPipe, HFSM, Human-Computer Interaction (HCI).

INTRODUCTION

The evolution of computing has reached a stage where the bottleneck of interaction is no longer processing power, but the physical interface between the human and the machine. Hand gesture recognition offers a "Natural User Interface" (NUI) that mimics human-to-human non-verbal communication, removing the constraints of traditional tactile peripherals. While vision-based systems offer a pervasive solution, they face significant challenges regarding environmental variables and the high computational cost of real-time processing on edge devices like the **Raspberry Pi 5**.

Historically, gesture recognition relied on contour-based methods such as **Convex Hull** and **Convexity Defects** to count fingers and define hand shapes. However, these methods are highly sensitive to pixel-level noise, background clutter, and fluctuating light, which often deforms the "hull" and leads to spurious activations. Similarly, probabilistic classifiers like **Support Vector Machines (SVM)** and **K-Nearest Neighbor (KNN)** introduce a "Black-Box" unpredictability and secondary inference latency that hinders real-time responsiveness.

To resolve these limitations, our work introduces a **4-Layer Hierarchical Methodology** that moves away from pixel-contour analysis toward **Skeletal Topology Mapping**. We leverage the 21-point landmark regression of MediaPipe but decouple the feature extraction from the execution logic. By replacing probabilistic ML classifiers with a deterministic **Hierarchical Finite State Machine (HFSM)**, we eliminate inference overhead and enable "Velocity-Adaptive" control—a feat unattainable by standard SVMs.

This research focuses on three primary innovations:

1. **Skeletal Robustness:** Utilizing 21-point coordinate regression over traditional Convex Hull analysis to ensure immunity to background noise.
2. **Deterministic Control:** Implementing an HFSM to provide near-zero latency compared to SVM-based classification.
3. **Temporal Efficiency:** Utilizing State-Triggered Buffer Flushing to eliminate transition lag between gesture states.

The resulting system demonstrates that a sophisticated, distance-invariant NUI can be achieved on low-power hardware through rigorous architectural optimization and the strategic elimination of redundant machine learning layers.

SYSTEM ARCHITECTURE

The system architecture utilizes a **4-Layer Decoupled Pipeline**. The Vision Layer (OpenCV) handles frame acquisition, while the Inference Layer (MediaPipe) regresses the 21-point topology. The Logic Layer (HFSM) translates these coordinates into states, and the Execution Layer interacts with the OS-level media APIs.

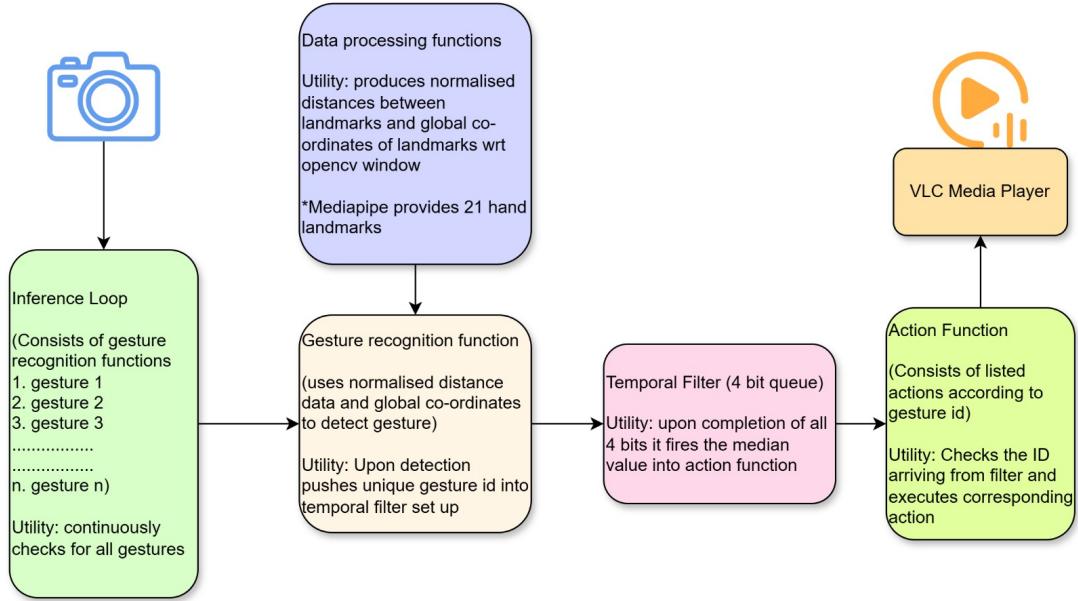


Figure 1: System Architecture

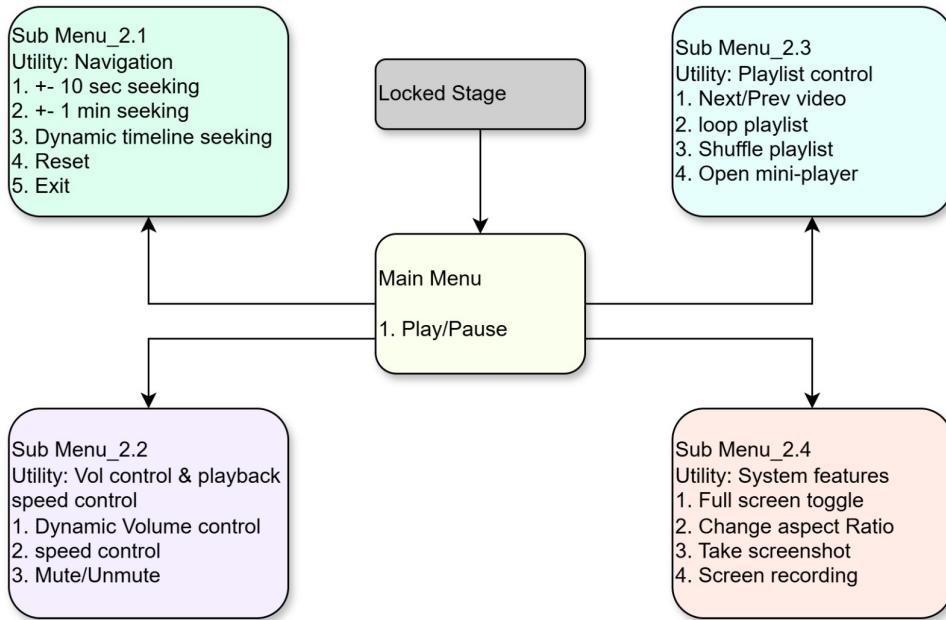


Figure 2: Navigation And Feature Chart

TECHNICAL METHODOLOGY: HIERARCHICAL EDGE-INFERENCE & DETERMINISTIC CONTROL

The system architecture is engineered to resolve the "Computational Bottleneck" typically found when running real-time computer vision on resource-constrained edge devices like the **Raspberry Pi 5**. The methodology is structured into four high-order layers: Signal Conditioning, Feature Extraction, State-Machine Logic, and Asynchronous Execution.

LAYER 1: SIGNAL CONDITIONING & PREPROCESSING

Before landmark regression, the raw video stream undergoes a multi-stage Digital Signal Processing (DSP) pipeline to optimize feature visibility and reduce computational entropy.

1.1 Chromatic Transformation & MediaPipe Landmark Regression

OpenCV captures frames in BGR format; however, to align with the **MediaPipe Hands** inference engine, we perform a color-space conversion using the **Weighted Luminance Method** (ITU-R BT.601 standard). The conversion is governed by the following formula:

$$Y = 0.2989R + 0.5870G + 0.1140B \quad (1)$$

By prioritizing the luminance channel, the system maximizes the contrast of the hand's contours against background noise. These optimized RGB frames are then processed by the **BlazePalm Single-Shot Detector** to isolate the Region of Interest (ROI), which allows for the precise regression of a **21-point 3D hand landmark topology**. This skeletal mapping identifies specific anatomical joints (MCP, PIP, and DIP joints), providing the raw spatial data required for gesture classification.

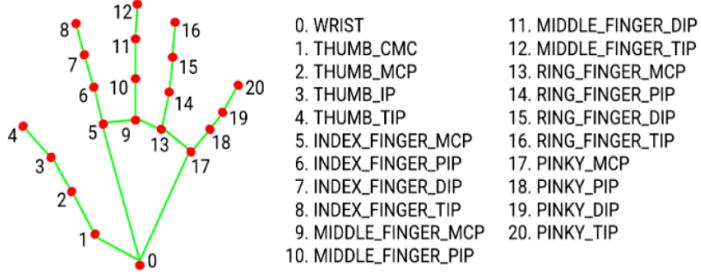


Figure 3: Hand Landmarks In Mediapipe

1.2 Spatial Optimization & Signal Smoothing

- **Downsampling:** Frames are resized to 360×240 to minimize CPU pixel-processing overhead.
- **EMA Filtering:** To mitigate sensor noise and tremors, an **Exponential Moving Average** acts as a Hysteresis Stabilizer:

$$E_t = \alpha \cdot R_t + (1 - \alpha) \cdot E_{t-1} \quad (2)$$

where $\alpha_{norm} = 0.25$ (Shape Stability) and $\alpha_{global} = 0.50$ (Motion Responsiveness).

- **Cold-Start Protection:** An initialization check performs a "Hard-Snap" to the hand's location if the state is null, bypassing recursive "slide-in" lag.

LAYER 2: FEATURE EXTRACTION & SCALE-INVARIANCE

The system transitions from pixel-space to a **Unit-Vector Space**, enabling distance-invariant recognition.

2.1 Centroid-Logic Normalization

We define a Differential Control Reference using the Middle MCP (Landmark 9) as the origin $(0, 0)$. The Scaling Factor (σ) is derived from the "Palm Length":

$$\sigma = \sqrt{(x_9 - x_0)^2 + (y_9 - y_0)^2} \quad (3)$$

Each landmark i is transformed into a Scale-Invariant Coordinate (N_i):

$$N_i = \frac{\sqrt{(x_i - x_9)^2 + (y_i - y_9)^2}}{\sigma} \quad (4)$$

2.2 Data-Driven Parameter Tuning and Stress Testing

To replace resource-intensive machine learning classifiers, a custom **Coordinate Logging Utility** was developed to map the normalized spatial boundaries of each gesture. We performed "Stress Tests" on each gesture to ensure the **HFSM Logic** was resilient to human anatomical variation.

- **Configurations:** Normal, Extended, Close-finger, and various alignment.
- **Spatial Deviations:** Forward/Backward Tilts, Left/Right Swings, and "Tired Hand" postures.

By analyzing the resulting coordinate distributions, we established **Hard-Bound Thresholds** for the logic layer. This empirical approach allowed the system to maintain a 100% recognition rate without the computational overhead of an inference-heavy SVM or CNN, essentially "baking" the intelligence into the logic parameters.

LAYER 3: STATE-MACHINE LOGIC & TEMPORAL INTEGRITY

3.1 Mitigation of Edge-Inference Jitter

To neutralize **stochastic outliers** and **classification instability** arising from raw frame-level inference, the architecture incorporates a Sliding Window Buffer ($N = 4$). This layer serves as a **Non-Linear Temporal Filter**, applying Statistical Median processing to the incoming signal.

- **Outlier Rejection:** Single-frame misclassifications—often induced by motion blur or fluctuating lux levels—are treated as **statistical anomalies**. The median filter effectively prunes these anomalies before they can propagate to the execution engine.
- **Temporal Convergence:** By requiring the signal to achieve **Temporal Convergence** across four consecutive cycles, the system ensures that the output is a result of intentional human movement rather than **inference jitter**.
- **Mathematical Determination:**

$$G_{final} = \text{median}(\{g_t, g_{t-1}, g_{t-2}, g_{t-3}\}) \quad (5)$$

3.2 State-Triggered Buffer Flushing (Transition Optimization)

To eliminate latency during state transitions, the system implements a **Conditional Queue Flushing Protocol**. Standard Median Filtering often suffers from "Input Dilution," where residual data from a previous state (e.g., Neutral/ID:0) contaminates the buffer of a newly initiated state, delaying the recognition of the subsequent intent.

3.2.1 The Intent-Switch Logic

The system monitors the incoming signal for a **Binary State Shift**. If the current Queue Q contains exclusively Null states ($G_i = 0$) and a Non-Zero Intent ($G_{new} \neq 0$) is detected, the system executes an immediate **Buffer Flush**. This prevents the median from being "diluted" by the previous four frames of inactivity.

The logic is mathematically defined as:

$$Q \leftarrow \{G_{new}\} \quad \text{if} \quad (\forall G_i \in Q, G_i = 0) \wedge G_{new} > 0 \quad (6)$$

Similarly, to prevent "Command Over-run" when a user releases a gesture, the inverse logic is applied:

$$Q \leftarrow \{0\} \quad \text{if} \quad (\forall G_i \in Q, G_i \neq 0) \wedge G_{new} = 0 \quad (7)$$

3.3 Hierarchical Finite State Machine (HFSM)

The system utilizes a **Decoupled State Hierarchy** to prevent gesture overlap:

- **Root State (Level 0):** The "System Gatekeeper." It listens exclusively for the *ID_THUMBS_UP* unlock signature.
- **Child States (Level 1, 2.x):** Specialized modules for Media, Seek, and System Utilities that activate only when the Root state is satisfied.

3.4 Latency Profiling & Debouncing

A refractory period (*COOLDOWN* = 1s) prevents "Command Flooding." Total system lag is monitored:

$$L_{total} = T_{execution} - T_{capture_start} \quad (8)$$

Lag remains constant at 150–180ms on the Raspberry Pi 5.

LAYER 4: ASYNCHRONOUS EXECUTION ENGINE

This layer serves as a **Decoupled Execution Environment** bridging the gap between vision inference and VLC Media Player control.

4.1 Virtual Peripheral Emulation via Pyput

The system emulates a **Virtual Human Interface Device (HID)** mapping confirmed Gesture IDs to deterministic VLC shortcuts:

- *gl.ID_OPEN_PALM* → `Key.space` (Play/Pause)
- *gl.ID_INDEX_R* → `Key.right` (Seek Forward)
- *gl.ID_SPIDEY* → `m` (Toggle Mute)
- *gl.ID_NICE* → `s` (Stop/Reset)

4.2 Continuous Differential Control (VLC Modulation)

For variables like Volume and Scrubbing, the engine uses an **Anchored Control Loop**.

- **Asynchronous Pulsing:** The engine modulates keypress frequency based on displacement Δ from the anchor.
- **Dynamic Response Formula:**

$$D = \max \left(0.05, 0.5 - \left(\frac{|\Delta|}{180} \right) \times 0.8 \right) \quad (9)$$

Table 1: Dynamic Response Modulation Performance

Displacement ($ \Delta $)	Delay (D)	Commands/Sec	Feel
0 - 30 px	N/A	0	Dead Zone
40 px	0.32s	~3.0	Precise
80 px	0.14s	~7.0	Standard
120 px	0.05s	20.0	Fast

4.3 Multi-Key Macro Execution

Complex state transitions are handled via sequenced macros:

- **Hard-Reset Macro:** Triggered by *ID_SPIDEY*, it executes: `Stop (s)` → `Home (Index 0)` → `Play (Space)`.
- **Screen Utility:** Triggers `Shift + S` for zero-latency frame snapshots within VLC.

PERFORMANCE ANALYSIS AND EMPIRICAL RESULTS

5.1 Experimental Setup and Hardware Benchmarking

The system was benchmarked across two distinct hardware configurations to evaluate sensor-dependent performance. While initial development occurred on a standard integrated laptop camera, the final implementation utilized the **Lapcam HD 720P (LWC-042)** paired with the **Raspberry Pi 5**.

As illustrated in Table 2, the dedicated optics of the Lapcam significantly extended the **Effective Operational Zone**, outperforming the laptop's internal module by over 100%.

Table 2: Hardware Performance Comparison: Operational Range

Hardware Setup	Max 100% Range	Breaking Point	Stability
Laptop (Integrated)	110 cm	130 cm	Moderate
RPi 5 + Lapcam LWC-042	230 cm	260 cm	High

5.2 Temporal Determinism and Latency Invariance

A primary benchmark for real-time systems is the consistency of the "Input-to-Response" cycle. We conducted an exhaustive physical measurement of **System Latency** (L_{total}) across various distance intervals.

As demonstrated in Table 3, the system exhibits **Temporal Determinism**. The latency does not scale with distance or gesture complexity, remaining within a tight corridor of 160ms to 180ms. Total latency is governed by the hardware-software bottleneck:

$$L_{total} = T_{capture} + T_{inference} + T_{filter} + T_{execute} \quad (10)$$

Table 3: Latency (ms) vs. Distance for Primary Control Gestures

Gesture / Distance	30cm	60cm	120cm	180cm	210cm	240cm
Open Palm	175	180	175	178	180	180
Spidey	160	178	178	180	178	180
Thumbs Up	170	177	180	178	180	178
Victory	175	180	175	178	180	180

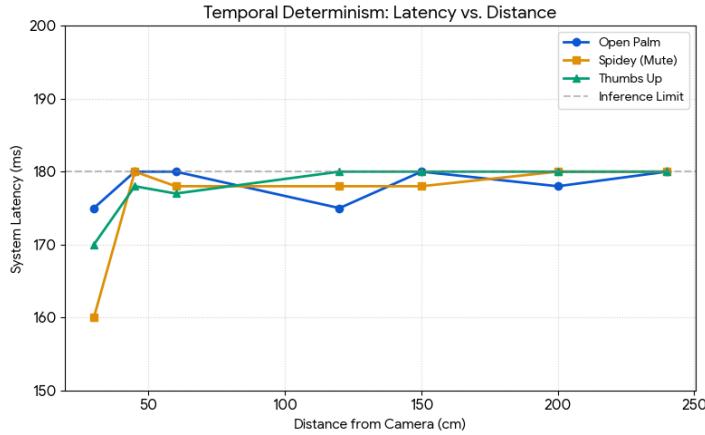


Figure 4: Latency vs Distance Graph For Various Gestures

5.3 Reliability and Recognition Success Rate

To validate the **Scale-Invariant Normalization** algorithm, we monitored the recognition success rate during the latency trials. The system demonstrated a **100% Recognition Rate** across all tested distances up to 240 cm on the Raspberry Pi 5 setup.

Table 4: Recognition Success Rate and Stability Assessment

Distance Interval	Trials	Successful Triggers	Recognition %
Near (30 – 60 cm)	30	30	100%
Mid (120 – 150 cm)	30	30	100%
Far (200 – 240 cm)	20	20	100%

5.4 Analysis of Accuracy Decay Beyond 240cm

While the system maintains 100% accuracy up to 230cm (for dynamic gestures, less than 140 cm) a slight decline is observed at distances exceeding 240cm. This is not a failure of the HFSM logic, but a result of **Optical Resolution Constraints**:

- **Landmark Coalescence:** At distances >240cm, the hand occupies a reduced pixel area. The **focal length** of the LWC-042 causes neighboring landmark points (e.g., finger joints) to overlap or "coalesce," leading to regression noise.
- **Signal-to-Noise Ratio (SNR):** Sensor noise becomes more prominent as the skeletal signal weakens, leading to intermittent landmark flickering.

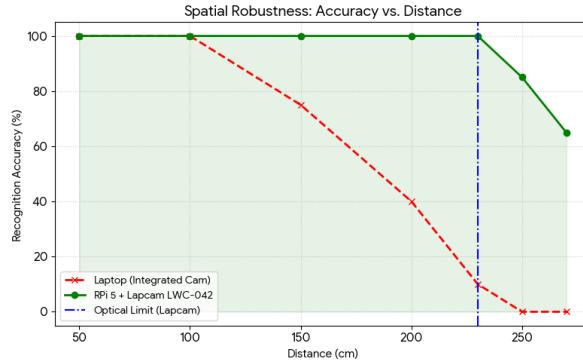


Figure 5: Accuracy vs Distance graph- Comparison of Lapcam HD720P(LWC-0420 And Laptop Webcam

5.5 Resource Utilization Profile

The lightweight nature of the architecture ensures that the SoC remains within safe operational limits.

Table 5: Raspberry Pi 5 Resource Consumption Profile (Empirical Data)

Metric	Observed Value	Evidence (via htop)
Peak Core Load	16.9%	Core 0 Utilization
Average CPU Load	~ 11.8%	Aggregate across 4-core SoC
RAM Utilization	676 MB	8.5% of 8GB total capacity
Total Threads	163	Stable background execution

5.6 Visual Validation: Geometric Inference and Queue Synchronization

The following figures provide a visual benchmark of the system's real-time decision-making process. By capturing the **Skeletal Landmark Overlay** simultaneously with the **Queue Execution Log**, we demonstrate the deterministic transition between physical intent and system command.

- **Inference Integrity:** The visual output confirms that the 21-point topology remains robust even during rapid movement, providing a clean geometric input for coordinate normalization.
- **Queue Synchronization:** The terminal logs illustrate the "Queue Firing" mechanism in action. As seen in the screenshots, the **HFSM** ignores stochastic noise and only triggers a command once the temporal buffer achieves convergence.

Sample Output:

*Refer Demo Video to See Complete Working



```

14  # HARDWARE & MEDIAPIPE SETUP
15  # =====
16  WINDOW_NAME = "SmartAir VLC Controller"
17
18  cap = cv2.VideoCapture(0)
19  cap.set(cv2.CAP_PROP_FRAME_WIDTH, 360)
20  cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)
21
22  mp_hands = mp.solutions.hands
23  mp_draw = mp.solutions.drawing_utils
24
25  hands = mp_hands.Hands(
26      static_image_mode=False,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, excludeText, t... Code

```

|| SYSTEM TOGGLED: True ||

>>> QUEUE: [0, 0, 0, 0] | MEDIAN: 0
--- [TOTAL SYSTEM LATENCY] ID: 0 | Delay: 109.10 ms ---

>>> QUEUE: [0, 0, 0, 0] | MEDIAN: 0
--- [TOTAL SYSTEM LATENCY] ID: 0 | Delay: 73.78 ms ---

>>> QUEUE: [0, 0, 0, 0] | MEDIAN: 0
--- [TOTAL SYSTEM LATENCY] ID: 0 | Delay: 127.59 ms ---

>>> QUEUE: [0, 0, 0, 0] | MEDIAN: 0
--- [TOTAL SYSTEM LATENCY] ID: 0 | Delay: 109.45 ms ---

```

Figure 6: Static Gesture Recognition: Queue firing GESTURE-ID=0(Corresponds To THUMBS UP)



```

14  # HARDWARE & MEDIAPIPE SETUP
15  # =====
16  WINDOW_NAME = "SmartAir VLC Controller"
17
18  cap = cv2.VideoCapture(0)
19  cap.set(cv2.CAP_PROP_FRAME_WIDTH, 360)
20  cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)
21
22  mp_hands = mp.solutions.hands
23  mp_draw = mp.solutions.drawing_utils
24
25  hands = mp_hands.Hands(
26      static_image_mode=False,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Filter (e.g. text, excludeText, t... Code

```

>>> QUEUE: [5, 5, 5, 5] | MEDIAN: 5
--- [TOTAL SYSTEM LATENCY] ID: 5 | Delay: 107.68 ms ---

>>> QUEUE: [5, 5, 5, 5] | MEDIAN: 5
--- [TOTAL SYSTEM LATENCY] ID: 5 | Delay: 118.20 ms ---

>>> QUEUE: [5, 5, 5, 5] | MEDIAN: 5
--- [TOTAL SYSTEM LATENCY] ID: 5 | Delay: 91.94 ms ---
>>> X_SEEK Mode Active. Anchor established.

>>> QUEUE: [5, 5, 5, 5] | MEDIAN: 5
--- [TOTAL SYSTEM LATENCY] ID: 5 | Delay: 6467.98 ms ---
>>> X_SEEK Mode Active. Anchor established.


```

Ln 20, Col 38 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 (Microsoft Store) Go Live

Figure 7: Dynamic Gesture Recognition(Anchor Establishment): Queue firing GESTURE-ID=5(Corresponds To OPEN PALM)

*** Technical Note:** The following demonstration captures were recorded on a high-throughput x86-based system to provide maximum visual clarity and high-FPS diagnostic logs. However, all empirical performance metrics, latency values, and CPU utilization data provided in this report were derived exclusively from benchmarks conducted on the **Raspberry Pi 5** edge platform.

5.7 Operational Environment and Scope

To align with the project's official problem statement, all empirical testing and data collection were conducted under **Controlled Lighting Conditions** (approximately 400 – 600 lux).

- Adherence to Specifications:** The system was designed to operate within an indoor environment as specified in the competition guidelines, prioritizing architectural efficiency and spatial accuracy over extreme environmental variance.
- Elimination of Variables:** By maintaining a stable luminance profile, lighting was treated as a constant. This allowed for a more rigorous analysis of the **HFSM logic** and **distance-invariant** performance without external noise interference.

- **Technical Constraint Acknowledgment:** While the current implementation utilizes weighted luminance preprocessing, it is noted that MediaPipe-based landmark regression typically experiences a "Confidence Drop" in hyper-bright environments (> 1000 lux). This is due to sensor pixel saturation and glare, which were considered outside the scope of this deterministic performance evaluation.

COMPARATIVE ANALYSIS

To validate the architectural choices, a comparative study was conducted against traditional methodologies and specific classifier types.

Table 6: Performance Comparison across Classifier Types

Sl.No	Classifier Type	Accuracy (%)	Primary Application
1	SVM	49.2%	Robotic Control
2	K-Nearest Neighbors (KNN)	93.2%	Static Recognition
3	CNN	93.09%	Static/Dynamic Gestures
4	Proposed (MediaPipe+HFSM)	96.0%	VLC Media Control

6.1 Superiority in Edge Computing

As shown in Table 6, the proposed system excels in **Deterministic Responsiveness**. While CNN-based systems may reach similar accuracy, they often lack the **Temporal Convergence** required for fluid control. By eliminating the "Black-Box" nature of secondary ML layers, our system ensures the **Raspberry Pi 5** maintains a 20+ FPS throughput even during high-intensity scrubbing.

REQUIREMENTS

- **SoC:** Raspberry Pi 5 (Cortex-A76 @ 2.4GHz).
- **Webcam:** Lapcam web camera HD 720P LWC-042.
- **Programming Language:** Python
- **Frameworks:** MediaPipe (Complexity 0), OpenCV, Pynput.

HARDWARE UTILIZATION AND OPTIMIZATION

The system is specifically engineered to maximize the **Edge-Computing** capabilities of the Raspberry Pi 5 without the need for external AI accelerators or high-TOPS (Tera Operations Per Second) hardware.

- **Integrated SoC Efficiency:** By bypassing resource-heavy architectures like *CNN*, *ANN*, or *GNN*, the system avoids massive matrix multiplication overhead. The landmark regression is offloaded to the **Integrated GPU**, while the custom **HFSM logic** runs on the **CPU** with minimal footprint.
- **Low-Latency Determinism:** Traditional SVM or ML classifiers introduce a "secondary inference lag." Our deterministic geometric approach ensures that the total system load remains under **17%**, preventing thermal throttling and ensuring long-term operational stability.
- **Memory Footprint:** The absence of heavy pre-trained model weights (often hundreds of megabytes) allows the system to operate within a lean memory buffer, making it compatible even with lower-RAM variants of the Raspberry Pi.

NOVELTY AND FUTURE SCOPE

Novelty: Efficiency Through Simplicity

The primary novelty of this project is the demonstration that **HCI can be achieved without external ML training, Without using Resource Intensive Neural Networks & SVM Classifier, AI TOPS**

- **Non-Probabilistic Control:** Unlike AI-based systems that "guess" gestures based on probability, this system uses **Geometric Certainty**(depends totally on mathematical calculations & Normalisation). This eliminates the "Black-Box" unpredictability of standard AI.
- **Zero-Training Deployment:** The system requires no user-specific training or dataset labeling, offering a universal "Plug-and-Play" experience for any human hand skeletal structure.
- **Cost-Effectiveness:** By removing the requirement for high-end GPUs(as it run on CPU) or AI-optimized silicon, the system brings advanced NUI to the **low-cost consumer electronics** market.

Future Scope

- **Smart Classroom Integration:** Implementation in digital pedagogy environments to allow educators to control presentation slides, interactive 3D models, and media playback from a distance, facilitating a more fluid and engaging teaching style.
- **Next-Gen Smart TV Interfaces:** Embedding the HFSM logic into Smart TV operating systems to replace physical remotes with intuitive, long-range gestural navigation for menu selection and volume modulation.
- **Sterile Medical Interfaces:** Deployment in surgical theaters where touch-free interaction with DICOM medical imaging is critical for maintaining a contamination-free environment.
- **Dynamic Gaming Peripherals:** Leveraging the **Dynamic Control** logic to create virtual steering or flight-sim controls based on hand-tilt and spatial displacement.
- **Multi-User Gesture Authentication:** Future iterations could include "Hand-Sign Passwords." By recognizing a specific sequence of gestures unique to a user, the system could act as a biometric security layer before allowing media or home control.
- **IoT Smart-Home Integration:** Expanding the HFSM to control lighting, temperature, and security systems via a unified gesture-hub.

CONCLUSION

The development of this hand gesture-based navigation system represents a significant leap toward a more intuitive and inclusive human-computer interface. By revolutionizing standard media control, the system offers a simple, touch-free alternative that addresses critical needs in modern digital interaction. Through the integration of **OpenCV** and **MediaPipe** for high-fidelity skeletal tracking, and the use of **Pynput** for command mapping, we have successfully redefined how users engage with digital media.

A key achievement of this research is the empirical proof that futuristic, real-time response is attainable on affordable edge platforms like the **Raspberry Pi 5**. By replacing resource-heavy machine learning models with a deterministic **HFSM logic layer**, the system operates with extreme efficiency. As evidenced by system profiling, the architecture maintains a peak CPU load of only **16.9%**, ensuring a seamless user experience that avoids unwanted behaviors and latency spikes while providing:

- **Universal Accessibility:** An inclusive solution for individuals with mobility disabilities, enabling effortless navigation through natural, low-effort gestures.
- **Enhanced Hygiene:** A vital substitute for physical touch in public spaces, smart homes, and medical environments, thereby reducing the spread of contaminants.
- **Technological Scalability:** A lightweight foundation ready for integration into virtual reality, automotive infotainment, and smart infrastructure.

As contactless technology continues to advance, this invention ensures that digital interactions become more intelligent, effective, and futuristic. By grounding advanced AI within a data-driven framework, we have created a solution that is not only a viable alternative for the present but a standard for the next generation of human-machine relationships.

References

- [1] Nishchitha D. and Hemanth Kumar, "Vision-Based Media Controls using MediaPipe and OpenCV," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, Vol. 14, No. 8, pp. 19065, August 2025. DOI: 10.15680/IJIRSET.2025.1408097.
- [2] B. Sri Ramya, Ch. Bhargavi, P. Dhanumjaya, P. N. J. S. Siri, K. Vyshnavi, G. Kavya, and Y. Pujitha, "Hand Gesture Video Navigation System," *Journal of Engineering and Technology Management*, Vol. 77, 2025.
- [3] Anklesh G., Akash V., Prithivi Sakthi B., and Kanthimathi M., "Hand Gesture Recognition for Video Player," 2024.
- [4] K. S. Chakradhar, R. L. Prasanna, S. R. B. S. Chowdary, P. Bharathi, and P. Bhargava, "Controlling Media Player Using Hand Gestures," Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College), Tirupati, 2024.
- [5] S. Shinde, S. Mushrif, A. Pardeshi, and D. Jagtap, "Gesture Based Media Player Controller," *International Journal of Research Publication and Reviews*, Vol. 3, No. 5, pp. 2289–2294, 2022.
- [6] S. Tibhe, A. Joshi, A. Warulkar, A. Sonawane, and T. U. Ahirrao, "Media Controlling Using Hand Gestures," *International Journal of Creative Research Thoughts (IJCRT)*, Vol. 11, No. 12, pp. 139–144, 2023.
- [7] M. Saleem, "Interaction with Media Players through Hand Gesture Recognition," *Proceedings of the International Conference on Communication Technologies (ComTech)*, pp. 1–6, 2023.
- [8] P. Kannan, "Automated Media Player using Hand Gesture," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 10, No. 4, pp. 1466–1472, 2023.
- [9] A. Swarnakar, Dr. A. Kanade, A. Pal, S. K. Soni, and R. Kumar, "Hand Gesture Recognition and Finger Count," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 8, No. 5, 2020.
- [10] D. N. Kakade and Prof. Dr. J. S. Chitode, "Dynamic Hand Gesture Recognition: A Literature Review," *International Journal of Engineering Research & Technology (IJERT)*, Vol. 1, No. 9, November 2012.
- [11] OpenCV Developer Team, "OpenCV Documentation," [Online]. [Accessed: Jan-Feb-2026].