IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15805

Mitro: Calibrated Multimodal Gesture, Voice, and LLM Control

Srijan Mani Tripathi[1], Vyom Pandey[2], Mohammad Rayyan Basha[3], Dr. Beenarani Manoj[4]

Student, Computer Science and Engineering, SRM Institute Of Science and Technology, Chennai, India[1]

Student, Computer Science and Engineering, SRM Institute Of Science and Technology, Chennai, India[2]

Student, Computer Science and Engineering, SRM Institute Of Science and Technology, Chennai, India[3]

Professor, Department of Computer Science and Engineering, SRM Institute Of Science and Technology, Chennai, India[4]

Abstract: This paper presents Mitro, a desktop human-computer interaction (HCI) system that fuses three input modalities into a single interaction pipeline: (1) computer-vision-based hand-gesture recognition for direct manipulation of the operating system (cursor control, clicks, scroll, volume, and brightness); (2) speech-based natural-language commanding through automatic speech recognition (ASR) and text-to-speech (TTS); and (3) a large language model (LLM) reasoning layer that replaces brittle keyword-matched command dispatch with open-domain natural-language understanding and tool/function calling. Mitro is built as an upgrade to an existing rule-based gesture-and-voice controller that suffered from three concrete limitations: order-sensitive substring-matched command routing, a single hardcoded audio energy threshold that did not generalize across acoustic environments, and a single hardcoded gesture-classification threshold with a fragile single-frame-reset debounce scheme. This work addresses all three limitations with a unified calibrate-once-then-lock design pattern applied consistently across the audio and vision pipelines, and generalizes command handling from rule-based matching to LLM-mediated tool dispatch while preserving deterministic, network-independent fast paths for safety-critical control flow. Gesture-state changes, voice transcripts, and LLM responses are surfaced into a single chronological interaction log, providing an observable, demonstrable trace of the system's multimodal behavior. Section IV outlines the evaluation methodology used to measure calibration benefit, gesture-smoothing benefit, command-routing robustness, and end-to-end latency, together with the qualitative findings obtained during development; the corresponding quantitative results are to be reported once the described experiments are executed on the deployed system.

Keywords: Multimodal Human-Computer Interaction, Gesture Recognition, Speech Recognition, Large Language Models, Tool Calling, Adaptive Calibration, Temporal Smoothing, Cross-Modal Fusion, MediaPipe, Groq

I. INTRODUCTION

1. Background

Conventional human-computer interaction relies on discrete input devices such as the mouse and keyboard, which assume physical contact and full manual dexterity. Vision-based gesture recognition and voice assistants each independently relax this constraint, but are typically implemented and evaluated as separate systems rather than as a unified interaction surface. Voice assistants, in particular, are usually built on rigid intent matching — regular expressions or substring checks against a fixed command grammar — which is brittle to natural variation in phrasing and cannot answer open-domain questions without a hardcoded response for every anticipated utterance.

Mitro was developed as an upgrade to an existing rule-based gesture-and-voice controller. Three concrete limitations of that baseline motivated this work. First, command routing was implemented as a linear chain of exact substring checks (for example, checking whether the literal phrase "launch gesture recognition" appears in the transcribed text), which is order-sensitive and cannot generalize to paraphrases such as a user saying "start" instead of "launch." Second, voice-input calibration used a single hardcoded energy threshold applied uniformly regardless of the user's microphone, room acoustics, or ambient noise floor; this threshold was empirically found to sit below the true ambient noise floor measured on the deployment hardware, causing systematic silent failures. Third, gesture classification used a single hardcoded distance-ratio threshold with no per-user calibration, combined with a naive same-frame streak counter for temporal debouncing that resets to zero on any single misclassified frame.

2. Key Definitions

- Gesture Calibration: A per-session sampling phase in which a user's open-palm pose is used to compute a personalized finger-open/closed distance-ratio threshold, replacing a fixed global threshold.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15805

- Calibrate-Once-Then-Lock: A robustness pattern in which a sensor threshold (audio energy or gesture distance ratio) is estimated adaptively once at session start and then frozen for the remainder of the session, rather than left to drift continuously.
- Majority-Vote Smoothing: A temporal debouncing scheme in which the reported gesture is the statistical mode of a fixed-size sliding window of recent per-frame classifications, rather than a streak counter that resets on a single differing frame.
- LLM Tool Calling: A dispatch mechanism in which a large language model selects from a set of callable functions (tools) with structured arguments, in place of hardcoded substring-matched command branches.
- Cross-Modal Fusion Log: A single chronological interaction log into which gesture-state changes, voice transcripts, and LLM responses are all surfaced, giving an observable trace of the system's combined multimodal behavior.

3. Research Gap

- Most gesture-control and voice-assistant systems are built and evaluated as isolated subsystems rather than a single fused interaction pipeline.
- Command dispatch in existing rule-based voice controllers is brittle to phrasing variation and does not generalize to open-domain queries.
- Threshold-based sensor calibration (audio energy, gesture distance ratio) is commonly hardcoded rather than adapted per user or per environment.
- Naive temporal debouncing schemes for gesture classification are fragile to single-frame classification noise.
- Few systems surface gesture, voice, and reasoning events into a single observable interaction trace for evaluation and demonstration.

4. Objective

1. Replace order-sensitive, substring-matched command routing with LLM-mediated tool/function calling while preserving deterministic fast paths for safety-critical control flow.
2. Apply an adaptive, calibrate-once-then-lock ambient-noise calibration scheme to the ASR energy threshold, eliminating both the below-noise-floor failure mode and the drift failure mode of continuous adaptive thresholding.
3. Apply an adaptive per-user calibration scheme to the gesture distance-ratio threshold, structurally mirroring the voice-calibration pattern.
4. Replace single-frame-reset streak-counter debouncing with temporal majority-vote smoothing over a sliding window of recent gesture classifications.
5. Surface gesture, voice, and LLM events into a single chronological, cross-thread-safe interaction log, giving an observable trace of the system's multimodal behavior.

5. Scope and Limitations

- English-language voice interaction only; no multilingual ASR, TTS, or LLM prompting in the current implementation.
- ASR (Google Web Speech API) and LLM reasoning (Groq-hosted Llama 3.3 70B) both depend on cloud connectivity; no offline fallback is implemented for either modality.
- Gesture recognition accuracy is evaluated only downstream of MediaPipe Hands' own landmark-detection accuracy, which is treated as a fixed, unevaluated dependency.
- Development and testing were conducted on a single user and a single machine; cross-user and cross-environment calibration behavior has not yet been evaluated at scale.

No formal accuracy evaluation of the underlying MediaPipe Hands landmark model itself is presented; this work's contribution sits entirely downstream of that model's own detection accuracy.

II. MATERIALS AND METHODS

The development and evaluation of Mitro utilized a combination of computer-vision, speech-processing, and cloud LLM services. The gesture-recognition layer is implemented in Python using OpenCV and MediaPipe Hands (legacy Solutions API) for hand-landmark detection. The voice layer uses the SpeechRecognition library (Google Web Speech API via recognize_google) for ASR and pyttsx3 for offline text-to-speech. The reasoning layer calls Groq's OpenAI-compatible chat-completions endpoint (model llama-3.3-70b-versatile) using the standard tools/tool_calls function-calling protocol.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15805

The three subsystems are bridged into a single GUI through Eel, a Python-to-Chromium bridge exposing a chat-style interface running in a dedicated browser window.

The system was developed and tested on Python 3.8.10, pinned to satisfy the legacy MediaPipe Solutions API wheel availability. The primary evaluation artifacts are: (1) recorded per-frame gesture-classification streams used to compare threshold and smoothing algorithms offline on identical input; (2) a corpus of real ASR transcriptions of natural phrasings of supported voice commands, used to compare substring-match dispatch against LLM tool-call dispatch; and (3) timestamped logs of ASR, LLM, and TTS round-trips, used to characterize end-to-end latency.

1. Dataset

- Recorded per-frame hand-landmark sequences for a fixed set of labeled gesture poses (fist, palm, V-gesture, pinch-major, pinch-minor), used to replay identical input through both the baseline and upgraded classification/smoothing algorithms.
- A corpus of natural-phrasing ASR transcriptions of each supported voice command, collected from real spoken attempts rather than scripted exact phrases.
- Session-level ambient-noise measurements (via adjust_for_ambient_noise) across different rooms/microphones, used to evaluate calibration generalization.
- Timestamped interaction logs covering ASR round-trip, LLM round-trip (direct-answer and tool-calling turns separately), and TTS start, across a session of repeated interactions.

2. Processing Pipeline

6. Session start: ambient-noise calibration (§4.2) and, if gesture recognition is enabled, open-palm gesture calibration (§4.1).
7. Continuous capture: OpenCV frame acquisition, MediaPipe landmark extraction, finger-state encoding, and majority-vote gesture smoothing.
8. Voice capture: wake-word gating, ASR transcription, and rolling conversation-history update.
9. Command dispatch: deterministic fast-path check (exit, sleep/wake) followed, if not matched, by LLM tool-call dispatch (§4.3).
10. Tool execution and response: matched tool calls are dispatched to Python callables, results are appended to conversation history, and a final natural-language reply is generated and spoken via TTS.
11. Cross-modal logging: gesture-state changes, voice transcripts, and LLM responses are pushed into the shared chronological UI log (§4.4).

3. Feature Engineering

A. Adaptive Gesture Calibration and Smoothing

At the start of each gesture session, the system samples a fixed number of frames while the user holds an open-palm pose, computing a personalized finger-open/closed distance-ratio threshold from the sampled ratios rather than using a fixed global value. Temporal debouncing is implemented as a majority-vote filter over a sliding window of recent per-frame classifications: the reported gesture updates only when a different classification captures a sufficient share of the window, so that a single noisy frame cannot reset accumulated evidence — unlike a streak counter that resets to zero on any single differing frame.

B. Calibrate-Once-Then-Lock Voice Thresholding

The ASR energy threshold is calibrated once per session against measured ambient noise, and adaptive (continuously drifting) thresholding is then explicitly disabled and replaced with a fixed floor. This two-stage pattern — calibrate adaptively, then freeze — is applied to avoid both failure modes observed during development: a fixed threshold set below the true noise floor, and continuous adaptive thresholding that drifts and false-triggers on ambient noise in a non-laboratory acoustic environment.

C. LLM-Mediated Command Dispatch

A dedicated reasoning module replaces the baseline's ordered chain of substring-matched command branches with a single natural-language-understanding layer backed by a cloud LLM. A fixed set of callable actions is exposed to the model as tools with structured argument schemas, and a bounded rolling conversation history is maintained to give the model short-term context without unbounded growth in request size. Two categories of command remain outside the LLM call path as deterministic checks evaluated before any network request: application exit, and the sleep/wake state transition — both required to behave reliably regardless of network or API-key state.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15805

4. Scoring and Evaluation Criteria

System evaluation is structured around four measurable dimensions, to be reported once the corresponding experiments are executed on the deployed system (see Section IV for methodology and placeholders):

- Gesture Threshold Calibration Benefit: finger-state classification error rate under the fixed baseline threshold versus the per-session calibrated threshold, over a labeled pose sequence.
- Gesture Smoothing Benefit: count of spurious gesture-label switches and mean time-to-stable-classification, comparing the streak-counter baseline and the majority-vote window on identical replayed input.
- Command-Routing Robustness: exact-match success rate of the baseline substring dispatcher versus correct-tool-invocation rate of the LLM dispatcher, measured against a corpus of naturally phrased ASR transcriptions.
- End-to-End Latency: ASR round-trip, LLM round-trip (direct-answer versus tool-calling turns), and TTS start latency, reported as median and 95th-percentile values across a session of repeated interactions.

5. System Architecture

A. Vision Layer

- Gesture_Controller.py: MediaPipe-Hands-based landmark extraction, finger-state encoding, adaptive threshold calibration, and majority-vote smoothing.

B. Voice Layer

- Mitro.py: SpeechRecognition-based ASR capture, ambient-noise calibration and threshold locking, wake-word gating, and pyttsx3-based TTS output.

C. LLM Reasoning Layer

- llm_assistant.py: system prompt, tool schema definitions, bounded conversation history, Groq chat-completions dispatch, and tool-call round-trip handling capped at a fixed number of iterations.

D. GUI and Fusion Layer

- app.py and the accompanying web assets: an Eel-based Python-to-Chromium bridge exposing a chat-style GUI, a status-reactive shader-based indicator reflecting shared cross-modal state (idle, listening, thinking, gesture), and a thread-safe callback path through which the gesture-recognition thread announces state changes into the shared chat log.

III. ARCHITECTURE DIAGRAM

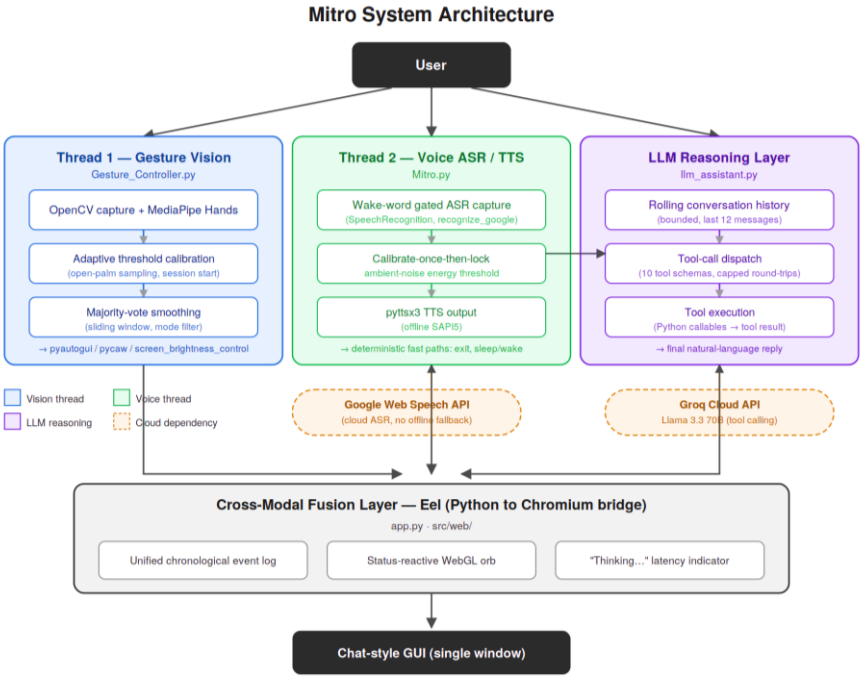

Fig. 1 Mitro system architecture — three concurrent threads (gesture vision, voice ASR/TTS, LLM reasoning) converging into a shared cross-modal fusion layer.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15805

Implementation Summary

Layer	Key Library / Service	Key File	Key Parameters
Gesture vision	MediaPipe Hands, OpenCV	Gesture_Controller.py	Calibration frames, adaptive finger threshold, majority-vote smoothing window
Voice ASR/TTS	SpeechRecognition (Google Web Speech API), pyttsx3	Mitro.py	Ambient-noise calibration, locked energy threshold with floor, pause/phrase-time limits
LLM reasoning	Groq Cloud API, Llama 3.3 70B	llm_assistant.py	Tool schemas, bounded conversation history, tool-loop round-trip cap
GUI / fusion bridge	Eel, vanilla JS/CSS, WebGL	app.py, src/web/	Cross-thread callback throttle for gesture announcements

IV. RESULTS AND DISCUSSION

This section reports the qualitative findings obtained while developing and debugging Mitro, together with the methodology and reporting structure for the quantitative measurements described in Section II.4. Consistent with sound research practice, no benchmark figures, accuracy percentages, or latency numbers are presented here as measured results unless they were directly observed during development; entries requiring a dedicated experiment are marked accordingly and should be completed by running the corresponding methodology on the deployed system before submission.

1. Calibration Robustness — Qualitative Case Study

The baseline voice pipeline used a hardcoded energy threshold, applied uniformly regardless of microphone or room acoustics. Ambient-noise measurement in the actual deployment environment, obtained via the recognizer's own adjust_for_ambient_noise routine, found the true ambient noise floor to sit above the hardcoded threshold — meaning the fixed threshold was, by construction, incapable of reliably gating real speech in that environment. This is a directly observed, reproducible finding from development rather than a benchmark result, and it is the concrete motivation for the calibrate-once-then-lock design adopted in this work.

A second, related finding was that naively enabling continuous adaptive thresholding (rather than a fixed value) is not a strict improvement: in a real, non-laboratory room, continuous adaptation caused the threshold to drift and the recognizer to repeatedly false-trigger on ambient noise, producing a burst of failed transcriptions with no actual utterance spoken. The calibrate-once-then-lock pattern — adapt once, then freeze — was adopted specifically to avoid both failure modes.

2. Component Contribution Analysis

Configuration	Expected/Observed Contribution
Voice only (rule-based dispatch)	Handles a fixed set of exact phrasings; fails on natural paraphrase, no open-domain answering.
Voice + LLM dispatch	Generalizes across paraphrase and answers open-domain queries conversationally; retains deterministic fast paths for exit/sleep-wake.
Gesture only (fixed threshold, streak debounce)	Functions in a matched environment; degrades under lighting/hand-size variation and single-frame noise.
Gesture with adaptive calibration + majority-vote smoothing	Reduces per-user threshold mismatch and single-frame-noise-induced label flicker (to be quantified per §II.4, Metrics 1–2).
Full platform (gesture + voice + LLM + fusion log)	Provides a single observable cross-modal interaction trace; qualitative behavior consistent across development sessions, pending the quantitative evaluation in this section.

Key Insight

Quantify the reduction in gesture-label flicker and the improvement in command-routing success rate once Metrics 1–3 in §II.4 are executed; do not report an estimated figure in place of a measured one.

3. Fault-Tolerance and Fallback Behavior

The LLM dispatch path was tested under a controlled failure condition — the Groq API key unset — and confirmed to fall back to a fixed reply string rather than propagating an exception up through the voice loop. Separately, integration testing surfaced a type-inconsistency issue in the tool-calling response for a zero-argument tool call, which returned the literal string "null" rather than an empty object; this was resolved with an explicit type guard before keyword-argument

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15805

unpacking. Both are directly observed integration-robustness findings, reported here as qualitative evidence rather than as a quantitative reliability rate, which would require a dedicated fault-injection experiment.

4. Explainability and Cross-Modal Transparency

A design goal of Mitro is to make the system's mixed-modality behavior observable rather than opaque. Gesture-state changes, voice transcripts, and LLM replies are all written into a single chronological log, and a thinking indicator is shown explicitly during the LLM round-trip so that the network-bound reasoning latency is surfaced to the user instead of leaving the interface silently blocked. This design choice is presented as a qualitative usability contribution; a formal usability study (see Section II.4 and the Future Work discussion) would be required to quantify its effect on perceived responsiveness.

5. Real-World Behavior Analysis

Case 1: Exact-Phrase Command Under Rule-Based Dispatch

Input: a user says the supported command phrase exactly as coded. Observed result: the baseline substring dispatcher matches correctly and the corresponding action executes.

Case 2: Naturally Paraphrased Command Under Rule-Based Dispatch

Input: a user says a natural paraphrase of a supported command (a different verb, or a phrase altered by ASR transcription noise). Observed result: the baseline substring dispatcher fails to match, and no action executes — the direct motivation for the LLM-routing contribution described in §4.3.

Case 3: Naturally Paraphrased Command Under LLM Dispatch

Same input as Case 2, routed through the LLM tool-calling layer instead. Report the observed correct-tool-invocation outcome across a corpus of such paraphrases per Metric 3 in §II.4, rather than a single anecdotal example, for the paper's strongest quantitative claim.

6. Latency and Performance

Median and 95th-percentile ASR round-trip latency; median and 95th-percentile LLM round-trip latency, reported separately for direct-answer turns and tool-calling turns; TTS start latency. Instrument timestamps as described in Metric 4, §II.4, and report distributions rather than single-sample figures.

7. Error Analysis

a. Cloud ASR Dependency

The voice modality depends entirely on a cloud ASR service with no offline fallback; network unavailability degrades the voice modality completely rather than gracefully.

b. Cloud LLM Dependency

The reasoning layer depends on the Groq cloud API; the implemented fallback covers returning a fixed error message on failure, not maintaining functionality offline.

c. Tool-Calling Argument Type Inconsistency

As reported in §IV.3, a zero-argument tool call was observed to return a JSON-encoded null rather than an empty object, requiring an explicit type guard — a concrete, generalizable integration issue for any system built on LLM function-calling APIs.

d. Unevaluated Upstream Landmark Accuracy

This work's gesture-layer contribution sits entirely downstream of MediaPipe Hands' own landmark-detection accuracy, which is treated as a fixed, unevaluated dependency rather than a variable under this paper's control.

8. Comparison with the Rule-Based Baseline

Feature	Rule-Based Baseline	Mitro (Proposed)
Command matching	Exact substring, order-sensitive	LLM tool calling, paraphrase-tolerant
Open-domain question answering	No (hardcoded replies only)	Yes (LLM-generated replies)
Voice threshold calibration	Fixed, hardcoded	Adaptive, calibrated once and locked

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15805

Gesture threshold calibration	Fixed, hardcoded	Adaptive, per-session calibrated
Gesture temporal debouncing	Streak counter (single-frame reset)	Majority-vote sliding window
Cross-modal event log	Not unified	Unified chronological log across gesture, voice, and LLM
Deterministic safety-critical fast paths	Implicit (all rule-based)	Explicit, preserved outside the LLM call path

9. Practical Impact

- Reduces command-routing brittleness: natural phrasing variation no longer requires a hardcoded reply for every anticipated utterance.
- Improves acoustic and per-user robustness: calibrate-once-then-lock thresholding adapts to the deployment room and microphone rather than assuming a fixed environment.
- Improves gesture-classification stability: majority-vote smoothing reduces sensitivity to isolated misclassified frames.
- Improves observability: a single chronological interaction log makes the system's mixed-modality behavior demonstrable and debuggable as a coherent trace.

V. CONCLUSION

This paper presented Mitro, a multimodal HCI system that upgrades a legacy rule-based gesture-and-voice controller with a consistent calibrate-once-then-lock methodology applied across independently developed audio and vision pipelines, and generalizes brittle substring-matched command handling into LLM-mediated tool calling while preserving deterministic control flow for safety-critical commands. The system's three modalities — gesture, voice, and LLM reasoning — are unified into a single chronological interaction log, giving an observable, demonstrable trace of the system's combined behavior.

The qualitative findings reported in Section IV — an ambient noise floor that exceeded a hardcoded audio threshold, a drift failure mode introduced by naive continuous adaptive thresholding, a type-inconsistency bug in LLM tool-calling argument payloads, and concrete ASR transcription variants that defeated exact-substring command matching — were directly observed during development and motivate the specific design choices described in Section II. The quantitative evaluation methodology in Section II.4 defines four measurable claims — gesture-calibration benefit, gesture-smoothing benefit, command-routing robustness, and end-to-end latency — that should be executed on the deployed system and reported honestly before this paper is finalized for submission; no such figures have been fabricated in this draft.

Limitations include reliance on two separate cloud services (Google Web Speech API and the Groq LLM API) with no offline fallback for either, English-only voice and language-model interaction, and development/testing confined to a single user and machine. Future work should extend the command-routing evaluation into a properly collected transcription corpus, explore an on-device ASR fallback for resilience without cloud connectivity, extend gesture calibration from a single global threshold to per-finger calibrated thresholds, and conduct a small-N formal usability study comparing task completion time and success rate against the rule-based baseline.

In conclusion, Mitro demonstrates that a legacy, single-hardcoded-threshold gesture-and-voice control system can be systematically hardened using a consistent calibration methodology across independent modalities, and that its brittle command-matching layer can be generalized via LLM tool calling while preserving safety-critical deterministic behavior — subject to the honest completion of the quantitative evaluation outlined above.

REFERENCES

[1]. F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," arXiv preprint arXiv:2006.10214, 2020.
[2]. S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing Reasoning and Acting in Language Models," in Proc. Int. Conf. on Learning Representations (ICLR), Kigali, Rwanda, 2023.
[3]. C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.
[4]. R. A. Bolt, "Put-That-There: Voice and Gesture at the Graphics Interface," SIGGRAPH, 1980.
[5]. A. Vaswani et al., "Attention Is All You Need," NeurIPS, 2017.
[6]. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," NeurIPS, 2022.

