Cureus Journal of Computer Science
https://doi.org/10.7759/s44389-026-00265-x

Original Article

Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation

Ashish Sharma [1] , Dhiraj Rajput [1, ✉] , Sushant Kumar [1] , Yashraj Narke [1] , Dhanashri Wategaonkar [1]

1. *Computer Engineering and Technology Specialization in Cybersecurity and Forensics, Dr. Vishwanath Karad MIT World Peace University, Pune, IND*

Received: May 01, 2026 | Review began: May 24, 2026 | Review ended: September 07, 2026 | Published: September 07, 2026

Abstract

The emergence of artificial intelligence (AI) in assistive technology is a rapidly evolving field that offers a great opportunity for visually impaired people to gain greater freedom. Despite the high pace of development in AI, current solutions are still rather incoherent and expensive and lack contextual reasoning and multimodal interaction. The proposed research combines computer vision and speech processing with Internet of Things modules and large language model-based reasoning in a single hands-free device using a wearable, head-mounted design. The architecture is based on a Raspberry Pi-powered edge-computing platform and cloud-assisted AI services to support real-time perception of the environment, voice-based execution of tasks, and context-specific reactions. The experimental evaluation demonstrated a wake-word detection latency of 0.6 s and an overall response time of 3.8 s, 93% speech recognition in quiet conditions, and 88% object detection in typical lighting. A satisfaction score of 8.5/10 was obtained as a result of user testing with 10 participants. These findings indicate the practical feasibility of the proposed system for real-time assistive navigation, demonstrating low-cost implementation, contextual interaction capabilities, and the potential to address several limitations of existing wearable navigation systems.

Categories: AI/ML-based decision support systems, Computer Vision, IoT Integration with Emerging Technologies

Keywords: large language models, internet of things(iot), wearable devices, assistive technology, speech recognition, visual question answering, computer vision, edge artificial intelligence

Introduction

The rapid development of embedded artificial intelligence (AI) and edge computing has brought new opportunities for small-scale intelligent assistive devices with real-time operation. Despite these advances, most assistive technologies available today are still limited in scope and single-function in their abilities or require costly proprietary hardware, which limits accessibility [1]. Approximately 2.2 billion people worldwide are affected by some form of vision impairment, and at least 1 billion of them have a condition that could have been prevented or is unaddressed [2]. This population suffers from permanent difficulties in autonomous navigation, spatial awareness, and interaction with the environment in real time. Traditional assistive tools provide only partial solutions [3].

Smartphone-dependent applications necessitate manual device handling, which is not well-suited for genuine hands-free navigation in the real world. Ultrasonic canes and smart sticks can sense physical obstacles but do not provide any context about the surrounding environment. Optical character recognition (OCR) wearables can recognize text and objects but are limited by high prices and closed, proprietary designs. Crucially, there are no existing solutions that combine real-time scene understanding, adaptive conversational reasoning, and multifunction task execution in a single

hands-free wearable platform. A significant advancement in this space is the development of large language models (LLMs) with multimodal input processing: LLMs can parse structured scene descriptions, reason about spatial and hazard relationships, and produce contextually adaptive guidance beyond traditional rule-based object detectors.

In this paper, the target user population is the visually impaired, i.e., blind individuals, according to the World Health Organization's classification of vision impairment [2]. This work builds on and differs from prior systems, including vision-language navigation models, YOLOv8-based smart-glasses systems, and LLM-integrated commercial smart glasses, in several ways: it combines on-device wake-word detection with cloud-hosted multimodal reasoning in a single hands-free platform; it integrates e-mail, media, and Internet of Things (IoT) tool-calling rather than perception alone; it provides an explicit accounting of original versus third-party contributions; it reports statistically rigorous benchmarks with Wilson-score and t-distribution confidence intervals; it documents reproducible hardware wiring and pin assignments; and it proposes a concrete, hardware-based migration path to on-device neural processing unit (NPU)-accelerated inference, directly addressing the cloud-dependency limitation identified in the current prototype.

Objectives of the research

The overall concept of the system is to build a wearable, artificially intelligent, voice-activated assistant for visually impaired users. The system aims to provide navigation and real-time object detection by recognizing scenes through a camera, communicate simply by e-mail, search for information, and directly control physical devices. It also allows remote connection, extensibility, and modularity with IoT modules. Secondary objectives include testing the system in real-world conditions, evaluating its cost-effectiveness compared to existing commercial products, and ensuring low power usage to make it portable.

Materials And Methods

Literature review

Single-Modality and Rule-Based Assistive Systems

The development of assistive technologies for visually impaired people has progressed from basic sensor-based mobility aids to AI-driven wearable systems utilizing computer vision, IoT connectivity, and conversational intelligence. In a review, Kuriakose et al. [1] trace the progression of navigation-support tools for this population and report that the majority of deployed systems are still narrow in scope, providing obstacle detection or object recognition, but rarely both in an integrated, conversational form.

Similarly, in their extensive survey of navigation systems for visually impaired people, Abidi et al. [3] conclude that currently available tools only partially satisfy the needs for independent mobility and interaction with the environment, with few systems providing adaptive and context-aware guidance. One of the smartphone-based approaches is Microsoft's Seeing AI application, which provides object recognition, text reading, and scene description on a handheld device using cloud-based vision application programming interfaces (APIs) [4]. However, it requires physical phone interaction and permanent connectivity, limiting hands-free operation in practical navigation. OrCam MyEye gets around this hands-free limitation with an eye-level camera and on-device OCR and object recognition [5]. This comes at the cost of a high price point and a closed, proprietary architecture; the device labels text and objects but does not maintain conversational context or invoke software or physical actions beyond reading aloud what it detects.

Abdel-Jaber et al. [6] propose a representative IoT-based design based on a smartphone app and a smart stick with sensors to detect obstacles and perform basic navigation. They find the approach useful for basic mobility assistance, but it is limited to a single hardware modality and does not support voice-driven task execution. Zhang et al. [7] report, in a bibliometric narrative review of smart wearable mobility aids, that depth-sensing RGB-D wearables provide improved spatial awareness over single-camera systems, but often at the expense of higher power consumption and hardware complexity, and generally without a conversational AI layer.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Campero-Jurado et al. [8] show that head-mounted architectures can allow for the integration of multiple sensors and AI-driven feedback at the hardware level using an industrial Smart Helmet 5.0 platform. Their system is built for occupational hazard monitoring, not conversational, language-based assistance, which shows that hands-free, actuation-capable hardware already exists but has not yet been coupled with conversational reasoning for assistive applications.

LLMs as Contextual Reasoning Engines

LLMs bring contextual reasoning to what has mostly been rule-based processing. In a survey of LLMs in wearable sensor applications, Ferrara [9] demonstrates how such models process multimodal sensor data and combine it with traditional machine learning to produce adaptive outputs for activity recognition and health monitoring. Fu et al. [10], in a survey of generative AI in assistive technology, discuss current trends and limitations in the areas of care, medicine, and support, and highlight the challenges of explainability and trust that remain before such reasoning can be reliably deployed in real-world assistive settings.

This reasoning ability depends on strong multimodal fusion. Xu et al. [11] reviewed multimodal transformer architectures and described the mechanisms by which vision and language inputs are jointly encoded to support this type of contextual inference. In an independent study [12] on the integration of LLMs with IoT systems, the main challenges of deploying this reasoning directly on embedded or wearable hardware are latency, computational cost, and limited context windows, further strengthening the case for a hybrid edge-cloud design rather than a fully on-device implementation.

Zhao et al. [13] synthesize this emerging subfield and catalog a shift from single-task perception models toward general-purpose multimodal reasoning agents in the 2024 VIAML survey and benchmark of visually impaired assistance with large models, while noting that few published systems couple this reasoning with physical actuation or multi-tool task execution rather than description alone. One practical implementation of this hybrid design philosophy is an early 5G-enabled edge-vision prototype for blind and low-vision users that couples low-latency wireless connectivity with cloud-offloaded perception [14].

Vision-Language and Agentic Systems

The reasoning-centric direction is further developed in some 2025 and 2026 published systems. In a workshop paper at the International Conference on Machine Learning in 2025, Li et al. [15] fine-tune a BLIP-2 vision-language model with low-rank adaptation on a purpose-built indoor-navigation image and question-answer dataset to generate step-by-step navigation instructions. This specialized fine-tuning approach sacrifices the broader task coverage of a general-purpose commercial LLM for potentially more precise spatial language and is complementary to, rather than a substitute for, the multi-tool agentic design adopted in the present work. Liu et al. [16] propose ObjectFinder, an open-vocabulary assistive system for interactive object search that allows a blind user to describe a target object in natural language and receive guided search feedback. Their system addresses a single, well-scoped task with high fidelity but, unlike the present system, does not integrate e-mail, media, or IoT actuation.

Zhang et al. [17] propose NaviGPT, an AI-based mobile navigation system that supports multimodal interaction for visually impaired people and operates in real time. The system is tested as a smartphone-companion application. Since it is implemented on a handheld device (rather than a hands-free wearable), it carries the manual-handling limitation that originally motivated hands-free wearable designs. Pfreundschuh et al. [18] describe Sight Guide, developed for the Vision Assistance Race at Cybathlon 2024, a multi-sensor (RGB and depth camera) wearable that fuses classical robotics algorithms with learning-based perception on an embedded computer and achieves a 95.7% task-success rate in a competition testing environment. Its multi-sensor fusion and competition-grade validation represent a considerably higher-fidelity perception stack than the single-RGB-camera, cloud-LLM-reasoning design used here, at the cost of substantially greater hardware complexity and cost.

Kumari and Hammady [19] evaluate real-time object detection on Vuzix Blade 2 smart glasses with three variants of the YOLOv8 model: YOLOv8-N, YOLOv8-S, and YOLOv8-M. Their models are trained and evaluated on a dataset of 15,951 annotated images collected specifically on campus, combining detection, distance estimation, and OCR into a single

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

auditory-feedback pipeline. Their on-device detector approach demonstrates a viable alternative to the cloud-vision-API approach adopted in the present system and is a direct contributor to the edge-migration roadmap to be discussed later in this paper. This choice of detector is put into a broader context by a 2025 survey of machine-learning techniques for object detection by Trigka and Dritsas [20]. Audo-Sight [21] also considers AI-powered ambient perception partitioned across edge and cloud tiers for blind and low-vision users, further underscoring that edge-cloud partitioning appears to be an emerging consensus architecture in this domain, as opposed to an all-cloud or all-edge approach.

Commercial and Near-Commercial Smart-Glasses Platforms

The commercial platforms have coalesced around common architectural themes. In November 2024, Meta's Ray-Ban Meta glasses added an accessibility-relevant capability through an integration with Be My Eyes to stream the wearer's first-person camera view to a sighted volunteer, along with an on-device conversational AI assistant for scene description [22]. As the glasses are phone-tethered and not purpose-designed for blind users, they operate as a general-purpose consumer device with incidental accessibility value, rather than a dedicated assistive platform, according to Waisberg et al. [22].

Envision's Ally Solos Glasses, explicitly marketed as a dedicated accessible AI assistant for blind and low-vision users, began shipping in late 2025. In a clinical-style evaluation of activities-of-daily-living performance and user experience with AI-assisted devices for people with vision impairment, Seiple et al. [23] provide quantitative evidence that AI-mediated assistance meaningfully improves task performance over baseline, reinforcing the practical value of the general design direction pursued here. Zaman et al. [24], in WhatsAI, propose to turn Meta's Ray-Ban platform into an open, extensible generative-AI accessibility layer through an unofficial software bridge, explicitly motivated by the closed, non-extensible nature of the manufacturer-provided assistant. This closed-versus-open tension is precisely the gap that an open, Raspberry Pi-based platform such as the one described here is positioned to fill, at the cost of the industrial design, battery life, and manufacturing polish of a commercial consumer product.

Edge-Cloud Collaborative Inference and Agentic Tool-Calling

A known limitation of LLM-based assistive systems is the incurred latency and connectivity cost of routing every reasoning step to the cloud. A survey of collaborative mechanisms between large and small language models in 2025 discusses edge-cloud architectures, where a lightweight small language model on the wearable device handles simple commands locally, and an uncertainty-quantification mechanism escalates complex queries to a cloud-hosted LLM, with end-to-end latency reduced by up to 40% compared to an all-cloud baseline in comparable voice-assistant workloads [25].

A contemporary survey of mobile edge intelligence for LLMs by Qu et al. [26] catalogs model-compression, knowledge-distillation, and edge-cloud collaborative techniques and reports latency reductions on the order of 50% for workload-distribution strategies in general mobile-edge settings. A comprehensive 2026 survey of edge-LLM deployment further documents quantization and distillation techniques that reduce memory footprint by up to 75% while preserving accuracy [27].

Regarding reasoning, the agentic tool-use paradigm underlying the function-calling architecture of the current system has quickly matured: recent surveys formalize the JSON-schema-based function-calling interface pioneered by major LLM providers, describe retrieval-augmented and reinforcement learning-based approaches to improve tool selection accuracy, and introduce standardized interoperability protocols for multi-tool, multi-agent orchestration [28,29].

These developments directly inform both the tool-invocation-layer design described below and the edge-migration roadmap discussed in the "Results and Discussion" section. Efficient LLM-serving techniques and general tool-learning frameworks further contextualize the design trade-offs made in the cloud-hosted reasoning layer [30,31]. The surveys on AI-enabled sensing and intelligent IoT by Mukhopadhyay et al. [32] and Aouedi et al. [33], respectively, provide further background for the sensor-and-connectivity layer of the proposed architecture.

Analysis of existing systems and research gap

This review is synthesized in Table *1*. Among the reviewed systems, three structural patterns are repeated. First, systems that integrate perception, communication, and physical actuation in a single hands-free device (e.g., industrial smart helmets and the proposed system) are rare compared with perception-only or navigation-only systems. Second, systems with general-purpose, multi-turn conversational reasoning (e.g., LLM-integrated smart glasses and our proposed system) are usually closed, phone-tethered, or non-extensible, while open and extensible platforms rarely have such a reasoning layer.

Third, even when LLM reasoning is present, few published systems provide an explicit accounting of which reported capabilities are due to the authors' own integration work versus the underlying commercial foundation-model service, an omission that this paper directly addresses in Table *2*, which provides an explicit accounting of original versus third-party contributions for each system capability.

The gap the present system is designed to close is therefore not raw perceptual accuracy, where cloud vision-language models and specialized on-device detectors such as YOLOv8 are already highly capable [19], but the tight, latency-budgeted, hands-free integration of that reasoning with multi-tool task execution and synchronized physical and auditory feedback on open, low-cost, reproducible hardware.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

System	Core Technology/Year	Key Features	Limitations
Ultrasonic Smart Cane	Ultrasonic sensors, buzzer	Obstacle detection, low cost	No AI or vision capability
Seeing AI App [4]	Smartphone + cloud vision API	Scene description, document reading	Requires phone handling, continuous connectivity
OrCam MyEye [5]	Camera + on-device OCR + AI	Text reading, object recognition	High cost, closed/proprietary hardware
Mobile IoT Smart Stick [6]	Smartphone app + sensor cane	Object identification, basic navigation	Single hardware modality, no voice/LLM reasoning
RGB-D Wearable [7]	RGB-D camera	Independent-travel spatial assistance	Complex setup, higher power draw, no conversational AI
ObjectFinder [16]	Open-vocabulary object-search agent	High-fidelity single-task object search	Narrow task scope; no e-mail/media/IoT tool integration
Ray-Ban Meta + Be My Eyes [22]	Smart glasses + phone-tethered AI + volunteer call	Scene description, live volunteer video call	Phone-tethered, closed platform, not purpose-built for blindness
NaviGPT [17]	Smartphone multimodal navigation app	Real-time AI-driven route guidance	Handheld, not hands-free; no physical actuation
Sight Guide [18]	Multi-sensor (RGB + depth) wearable, embedded computer	95.7% competition task-success rate	High sensor/hardware complexity and cost; no LLM tool integration
Envision Ally Solos Glasses [23]	Dedicated assistive smart glasses + AI assistant	Purpose-built accessible AI assistant	Closed platform, premium price point
Vuzix Blade 2 + YOLOv8 [19]	On-device YOLOv8-N/S/M smart glasses	Real-time detection, distance estimation, OCR; 15,951-image dataset	No conversational reasoning or multi-tool task execution

TABLE 1: Comparative analysis of existing systems

AI, Artificial Intelligence; API, Application Programming Interface; IoT, Internet of Things; LLM, Large Language Model; OCR, Optical Character Recognition; RGB, Red Green Blue; RGB-D, Red Green Blue-Depth

Proposed system design

Five-Layer Architectural Overview

The architecture in Figure *1* is in the form of a closed signal-flow loop consisting of five layers: Input, Processing, Cloud/API, Output, and an implicit Control/State layer that controls the transitions between the other four layers. The Input Layer is fed with information from the environment through the microphone module and the camera module. The Processing Layer is hosted on the central processing unit of the Raspberry Pi 4 and performs wake-word detection and speech-to-text transcription, as well as calls to the cloud-hosted LLM agent and tool-invocation layer. The Cloud/API Layer provides access to external large language models and auxiliary cloud services over HTTPS, namely the YouTube Data API and the Gmail API. The Output Layer translates the LLM response into synchronized speech, pitch-timed LED feedback, and servo-driven haptic actuation.

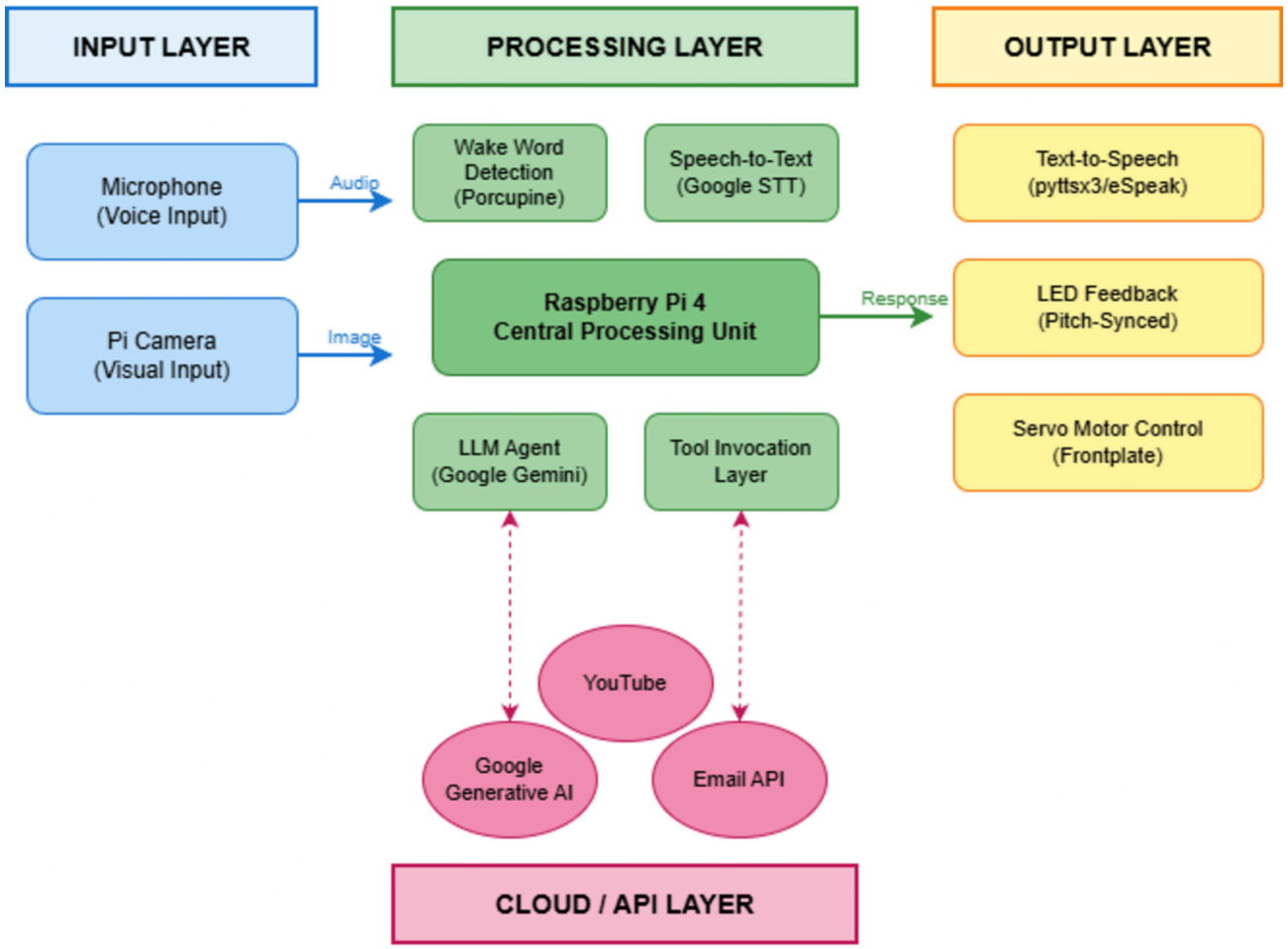

FIGURE 1: Proposed system architecture with respect to our proof of concept

API, Application Programming Interface; LLM, Large Language Model; STT, Speech-to-Text

Wake-Word Detection and Voice Activity Endpointing

The wake-word detection is performed on-device using the Porcupine keyword-spotting engine. Porcupine implements a compact deep neural network optimized for continuous, low-power inference on embedded hardware and does not require network connectivity. We deliberately designed the architecture to run wake-word detection fully on-device; this is not an artifact of our design. It enables the system to be activated and readiness to be confirmed through an audio cue even in the event of a total network outage. Furthermore, it does not add any marginal cloud latency to the response-time budget reported in the "Latency and Processing Analysis" subsection. When the wake phrase is detected, the system

plays an audio and LED readiness cue and begins recording. Recording stops when silence is detected (using voice-activity endpointing) or after a fixed maximum duration, whichever comes first, bounding the worst-case latency of the subsequent transcription stage.

Speech-to-Text Pipeline

The recorded audio is streamed over HTTPS to Google Cloud Speech-to-Text as 16 kHz mono pulse-code-modulated audio. Streaming recognition provides interim hypotheses as audio is received and a final transcript when endpointing is finished. This allows the processing layer to begin preparing the next LLM call before the very last word of the utterance has been transcribed. This introduces a minor amount of pipeline overlap, which is discussed further in the "Latency and Processing Analysis" subsection in relation to the measured latency figures.

LLM Reasoning and Agentic Tool-Invocation Layer

The system's intelligence is provided by an agentic LLM, Google Gemini, set up with a function-calling (tool-use) interface in line with the JSON-schema-based paradigm now standard among the major LLM providers [28]. Each available capability (scene description, e-mail composition and sending, media search and playback, and general information retrieval) is declared to the model as a named function with a structured parameter schema. The model does not use a hard-coded intent-classification table. Instead, at each turn, the model chooses to either respond directly in natural language or to produce a structured function call. The function call is executed by the tool-invocation layer, where it is applied to the corresponding cloud API, the Gmail API, or the YouTube Data API, or to the current camera frame for scene-description and visual-question-answering calls routed through Gemini's vision input.

A short-term conversation buffer is maintained across turns within a session and cleared on session timeout to bound memory growth on the constrained embedded platform, allowing for coherent multi-turn interaction, e.g., a follow-up question about an object mentioned in a previous scene description. This function-calling design allows new capabilities to be added by declaring an additional tool schema, rather than modifying the core control loop, directly addressing the extensibility design goal stated above.

Output Layer: Speech, Visual, and Haptic Feedback

The output layer uses pyttsx3 with the eSpeak backend to convert the LLM's natural-language response into speech. The eSpeak backend was chosen specifically because it performs text-to-speech synthesis entirely offline, so that even if a cloud call fails or times out (as described below), the system can always speak a locally generated status or error message rather than failing silently. Concurrently, a pitch-synchronized LED array provides a lightweight visual status indicator: the LED brightness is pulse-width modulated in sync with the amplitude envelope of the outgoing speech waveform, allowing sighted bystanders, e.g., a caregiver, or users with residual light perception to visually confirm that the system is actively responding. The wearable's faceplate hinge is physically and haptically actuated by two SG90 servo motors, used both as a readiness cue and in tool responses that carry directional information, e.g., an obstacle described as being to the user's left, as a coarse directional haptic signal.

System State Machine and Control Flow

The narrative description of control flow in the original prototype methodology is formalized here as an explicit state machine to make the real-time behavior of the system and its error-handling behavior unambiguous. The system is in the IDLE state and continuously listens for the wake phrase using the on-device wake-word engine; there are no cloud calls. When a wake phrase is detected, the system enters WAKE_DETECTED, emitting a readiness cue (audio tone + LED pulse) and starting recording. The system then transitions to LISTENING, where it records and sends the audio to the speech-to-text service. This continues until either voice-activity endpointing or a maximum-duration timeout occurs. The subsequent TRANSCRIBING state outputs the final transcript and appends it to the short-term conversation buffer.

In the REASONING state, the LLM agent is sent the transcript and current conversation context, and the latest camera frame, if applicable, and it returns a natural-language response or one or more structured function calls. For a function call, we enter TOOL_EXECUTION, where the requested calls are executed against the corresponding cloud API or camera

input, and the results are returned to the LLM agent for a final natural-language response. This state is optional and is skipped when the LLM agent responds directly without invoking a tool. The system then transitions to RESPONDING, during which the response is synthesized to speech, LED feedback is driven in concert with the audio envelope, and servo haptics are activated where directional information exists, before returning to IDLE. Finally, an ERROR_RECOVERY state can be reached from LISTENING, TRANSCRIBING, REASONING, or TOOL_EXECUTION on network failure, service error, or timeout, and in this state, the system speaks a locally synthesized status message using the offline text-to-speech engine and returns to IDLE without a further network round-trip.

Figure *2* summarizes this control flow in simplified diagrammatic form, distinguishing on-device states, which incur no network dependency, from cloud-dependent states, which are guarded by the bounded timeouts described below and can transition to ERROR_RECOVERY on failure.

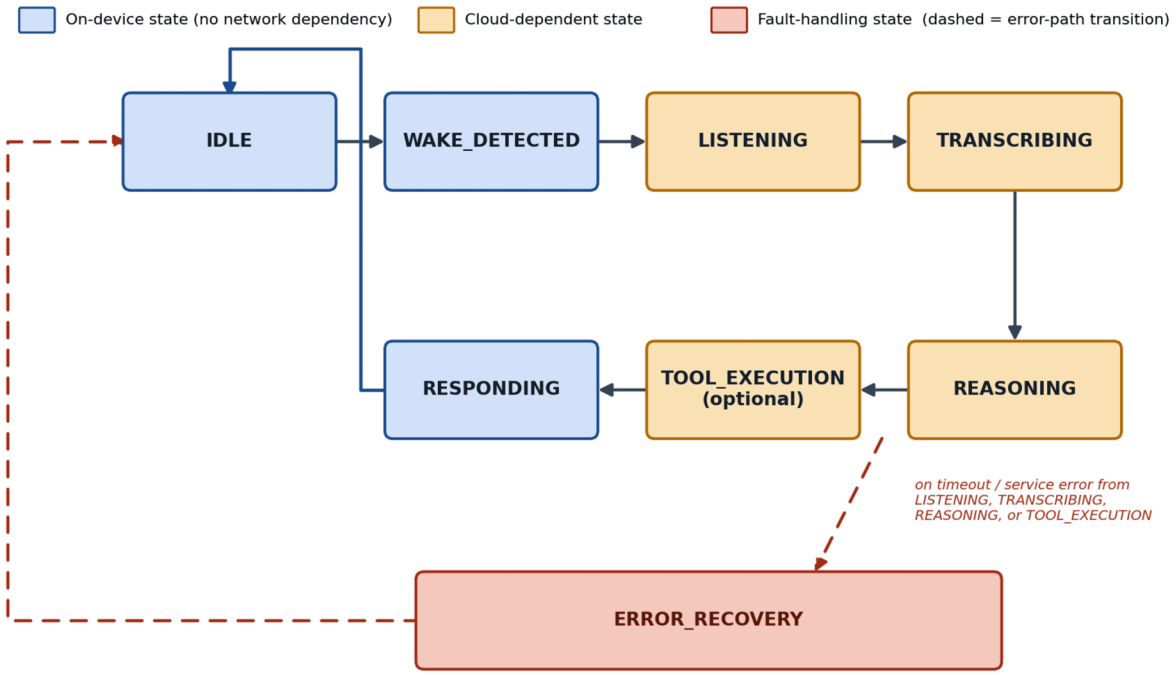

FIGURE 2: Simplified control-flow diagram of the system state machine

Latency Budget, Network Protocol, and Fault Tolerance

All-cloud communication is HTTPS/TLS. The end-to-end response-time budget reported in the "Latency and Processing Analysis" subsection is dominated by two network round-trips, speech-to-text and LLM reasoning, rather than on-device computation; the "Toward Reduced Cloud Dependency" subsection discusses how recent NPU-accelerated edge hardware could shift part of this budget on-device. Each cloud-dependent state, LISTENING, TRANSCRIBING, REASONING, and TOOL_EXECUTION, has a bounded timeout that it guards, so that the system transitions to ERROR_RECOVERY when the timeout expires, rather than waiting forever. This bounded-timeout design is a common fault-tolerance pattern for embedded voice assistants that are cloud dependent and ensures that a degraded or unavailable network connection produces a quick, locally synthesized status message, rather than an indefinite hang, which is particularly important for a device relied upon for real-time environmental awareness.

$$T_{response} = T_{wake} + T_{stt} + T_{llm} + T_{tts} \quad (1)$$

where T_{wake} is the on-device wake-word latency, and T_{stt}, T_{llm}, and T_{tts} denote speech-to-text, LLM reasoning including any tool execution, and text-to-speech latency, respectively. The arithmetic sum of the median per-stage latencies reported in the "Latency and Processing Analysis" subsection, 3.9 s, is close to, but not identical to, the independently measured end-to-end total of 3.8 s with an SD of 0.4 s. This small difference is consistent with limited pipeline overlap between the final portion of speech-to-text streaming and the start of LLM context preparation, as described above, rather than measurement error. Table *2* lists each system capability, its primary source, and description.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

System Capability	Primary Source	Description
Wake-word spotting accuracy	Third-party engine (Porcupine), on-device	Off-the-shelf keyword-spotting model; the authors selected, integrated, and threshold-tuned the engine but did not train the underlying model.
Speech-transcription accuracy	Third-party service (Google Cloud Speech-to-Text)	The 90–95% (quiet) / 75% (moderate-noise) figures reflect the provider's acoustic and language models, not original signal processing by the authors.
Scene understanding / visual question answering accuracy	Third-party service (Google Gemini Vision)	The 88% object-recognition figure reflects the provider's vision-language model; the authors' contribution is prompt design, camera-frame capture, and result routing, not the underlying detector.
Natural-language reasoning quality and tool-call selection	Third-party model (Google Gemini)	Reasoning fluency is provided by the foundation model; the function-call schema, available tool set, and system prompt constraining model behavior for this assistive use case are original design work.
Camera frame capture, preprocessing, and query-time triggering	Our contribution	Frame capture is triggered per query rather than by continuous streaming; capture timing, framing, and routing to the vision API are configured by the authors.
Session and conversation-context management	Our contribution	Design of the short-term conversation buffer, its multi-turn scope, and its timeout-based clearing policy.
Tool orchestration and multi-turn task execution	Our contribution	Design and implementation of the tool-invocation layer, conversation-context management, and mapping of tool outputs back into synchronized multimodal feedback.
Multimodal feedback synchronization (speech + pitch-synced LED + servo haptics)	Our contribution	Original integration linking the audio amplitude envelope to LED pulse-width modulation and mapping response semantics to haptic actuation; not provided by any third-party service.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Wearable hardware integration, pin-level wiring, and ergonomic design	Our contribution	Component selection, wiring, power budgeting, and physical wearable design.
Latency budgeting, timeout thresholds, and offline fallback behavior	Our contribution	State-machine design and fault-tolerant degradation to locally synthesized speech on network or service failure.
Reproducibility documentation (wiring diagram, pin table, tool-schema pattern)	Our contribution	Provided explicitly in this manuscript to support independent replication; not typically supplied by closed commercial competitor systems.
End-to-end system integration and empirical evaluation	Our contribution	Functional, latency, and usability evaluation, including the statistical treatment applied to the results.

TABLE 2: Contribution and third-party cloud services

API, Application Programming Interface

Hardware specifications

The prototype is implemented on a Raspberry Pi 4 Model B (4 GB RAM) with a 1.5 GHz quad-core ARM Cortex-A72 processor, onboard dual-band Wi-Fi (2.4/5 GHz), Bluetooth 5.0, and Gigabit Ethernet. A Raspberry Pi Camera Module 3 (12 MP Sony IMX708 sensor, autofocus) mounted at eye level supplies the visual input to mimic a natural, egocentric field of view. Voice input is received via a Bluetooth microphone. Audio output and status/haptic feedback are provided via a Bluetooth speaker headset and two SG90 micro servo motors, respectively. Figure *3* shows the physical wiring diagram, and Table *3* lists the corresponding Raspberry Pi 4 GPIO pin assignments.

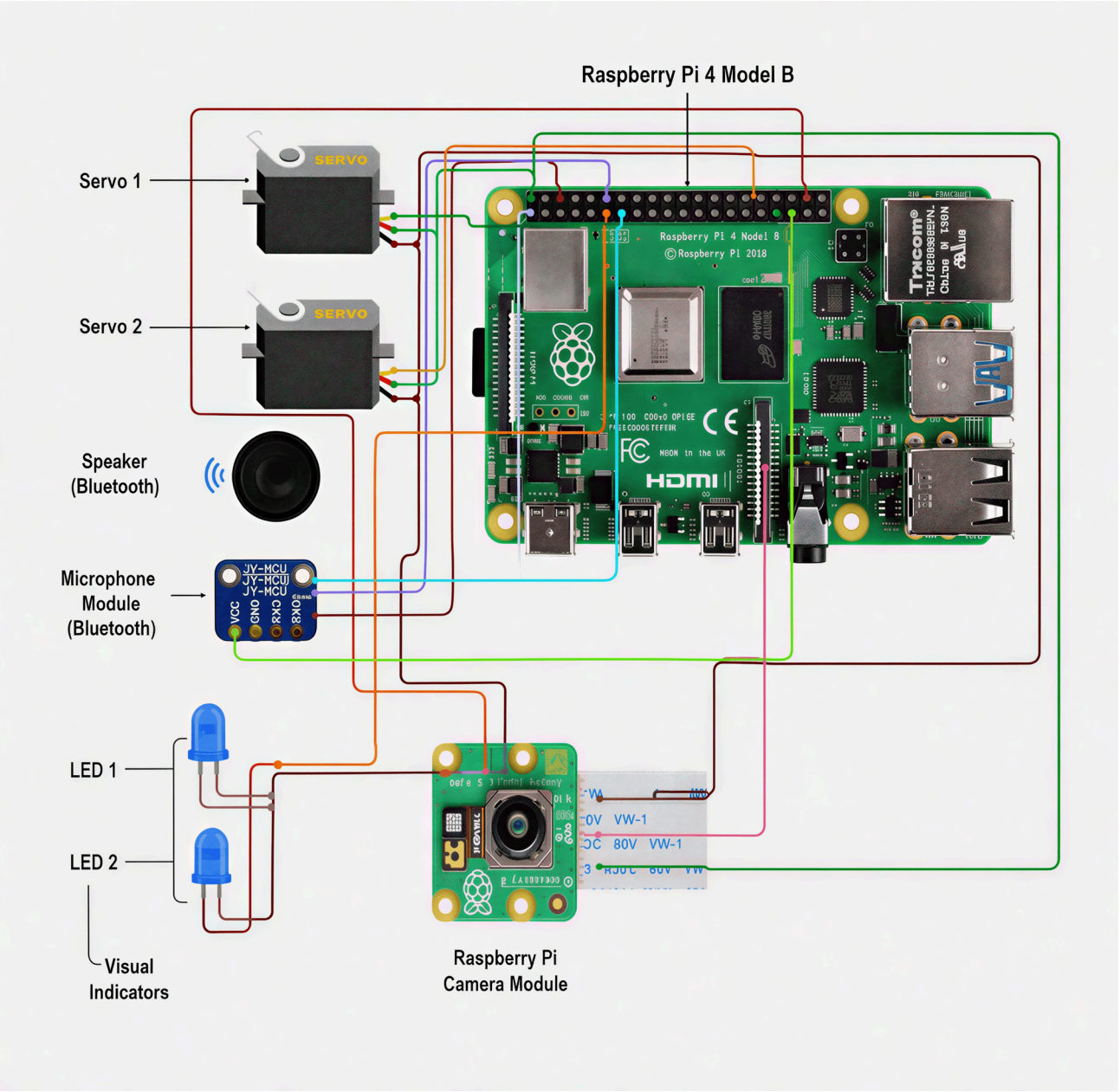

FIGURE 3: Hardware implementation diagram

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Component	Signal	Raspberry Pi 4 Connection	Physical Pin
CAM1 – Pi Camera Module 3	CSI	CSI ribbon connector	CSI port
SRV1 – SG90 Servo	PWM	GPIO17	Pin 11
SRV1 – SG90 Servo	VCC	External regulated 5V supply	–
SRV1 – SG90 Servo	GND	Common ground	Pin 30
SRV2 – SG90 Servo	PWM	GPIO13	Pin 33
SRV2 – SG90 Servo	VCC	External regulated 5 V supply	–
SRV2 – SG90 Servo	GND	Common ground	Pin 34
D1 – Blue LED	Anode	GPIO27 through 220–330 Ω resistor	Pin 13
D1 – Blue LED	Cathode	Ground	Pin 39
D2 – Blue LED	Anode	GPIO23 through 220–330 Ω resistor	Pin 16
D2 – Blue LED	Cathode	Ground	Pin 25
Bluetooth Microphone	Bluetooth	Internal Bluetooth interface	Internal
Bluetooth Speaker	Bluetooth	Internal Bluetooth interface	Internal
5V/3A Power Adapter	+5V	Raspberry Pi USB-C power input	USB-C
5V/3A Power Adapter	+5V	External servo power rail	–
5V/3A Power Adapter	GND	Common ground, shared with Pi and servo circuitry	–

TABLE 3: Prototype hardware connections

CSI, Camera Serial Interface; GND, Ground; GPIO, General-Purpose Input/Output; PWM, Pulse-Width Modulation; USB-C, Universal Serial Bus Type-C; VCC, Voltage at Common Collector

All servo motors are powered from an external regulated 5 V supply, rather than the Raspberry Pi's own 5 V rail, to avoid the problem of brownout at peak servo current draw. The Raspberry Pi, servo power supply, and peripheral circuitry are all commonly grounded to maintain stable pulse-width-modulation signal integrity.

The camera angle, field of view, and capture frequency all affect the object-detection and scene-understanding performance and were explicitly considered in the hardware design. The Camera Module 3 is mounted at approximately eye level and angled to correspond to a natural forward gaze, which alleviates the perspective distortion and downward-tilted framing a chest- or waist-mounted camera would introduce and keeps objects of interest closer to the center of the frame, where the vision-language model's attention is typically most reliable.

The standard Camera Module 3 used for this study has a narrower horizontal field of view than the wide-angle variant of the same module, which prioritizes detection accuracy for objects directly in front at the expense of peripheral coverage, something that participants also commented on in their feedback, asking for a wider camera field of view. Instead of streaming video continuously, the system captures a single frame when a scene-description or object-identification tool call is issued. This query-time capture strategy reduces bandwidth and processing load relative to continuous streaming, but means that transient or fast-moving objects present only between capture events will not be detected. This constraint becomes more significant outdoors or in dynamic environments than in the static indoor test conditions used for the present evaluation.

Table *4* summarizes the system's power budget using manufacturer datasheet ratings for each component; these figures represent a conservative design-time budget rather than a benchmarked power-measurement study, which was outside the scope of the present evaluation.

Component	Typical/Rated Current Draw (5V)	Description
Raspberry Pi 4 Model B	~600 mA (idle) to ~1.2 A (sustained load)	Increases under sustained CPU/network load
Pi Camera Module 3	~250 mA	Active capture
SG90 micro servo (×2, active)	~100–250 mA each; up to ~650 mA momentary stall	Powered from external regulated rail, not the Pi rail
Bluetooth microphone + speaker headset	~100 mA (host-side radio)	Peripheral battery is independent of the Pi power budget
Status LEDs (×2)	~20 mA each through current-limiting resistor	Negligible relative to total budget
Estimated system total (Pi + camera + LEDs, excluding externally powered servos)	~1.0–1.7 A	Informs power-supply sizing; the 5V/3A supply used provides adequate headroom

TABLE 4: Indicative power budget

The power budget shown in Table *4* has an immediate implication for the wearable ergonomics. The current draw of the Raspberry Pi 4, camera module, and servo motors must be met by a tethered power bank or onboard battery. The mass of the power source, as well as the Raspberry Pi enclosure, camera mount, and mechanical faceplate assembly, is the primary contributor to the overall head-worn weight of the prototype. The total mass was cited by some users as a source of discomfort during extended wear, detailed with specific numbers in the "Discussion of User Experience" subsection, and represents a general trade-off in single-board-computer wearables between processing headroom and wearable

comfort. The "Toward Reduced Cloud Dependency" subsection discusses how offloading part of the compute stack to a low-power, NPU-accelerated module could reduce power draw and battery mass, allowing for a lighter enclosure in future revisions.

Methodology

Environment and Network Conditions

All functional and usability testing was conducted indoors in a single controlled testing environment, with consistent artificial ambient lighting and a stable Wi-Fi connection with a minimum bandwidth of 10 Mbps to the cloud services described above. Standard-condition trials were performed with background noise below 45 dB. Moderate-noise trials, used only for the noise-sensitivity comparisons discussed below, were performed with ambient noise in the 60-70 dB range.

Given that the system relies on cloud-based speech-to-text and LLM services for most of its response pipeline, network quality directly affects both response time and reliability. The response times were in accordance with the numbers provided in the "Latency and Processing Analysis" subsection under the stable Wi-Fi conditions used throughout the testing. If network latency increased or bandwidth was decreased, we would expect the speech-to-text and LLM-reasoning stages to take longer (these are the two stages that rely on the cloud) and that, outside of the bounded timeouts described in the "System State Machine and Control Flow" subsection, this would trigger the ERROR_RECOVERY state instead of a successful response. This is the expected behavior by design of the bounded-timeout state machine, not a measured outcome, as degraded-network conditions were not part of the present experimental protocol. This study did not include formal testing under intentionally degraded network conditions (e.g., throttled bandwidth, simulated packet loss, etc.) and is noted as a limitation. The testing was carried out only under controlled indoor conditions; the results presented here characterize feasibility under favorable network and acoustic conditions and should not be extrapolated to outdoor, low-bandwidth, or high-noise real-world deployment without further testing.

Participants and Task Protocol

The pilot usability evaluation included ten participants with visual impairments, including four adolescents (9-17 years), four adults (18-40 years), and two older adults (55 years and above). All adult participants provided informed consent prior to testing. For the four adolescent participants (ages 9-17), parental/guardian consent was obtained in addition to participant assent. No personally identifiable information (names, images, or audio recordings linking responses to identity) was collected or stored; participant identifiers in this manuscript (P01-P10) are study-assigned codes with no external key connecting them to real identities. This study was conducted as a voluntary, non-invasive usability evaluation of a prototype device; the authors' institution does not maintain a formal ethics-board review process applicable to this type of pilot study, and consent and participant-safety procedures were therefore handled directly by the research team as described above. We note the absence of formal institutional ethics review as a limitation of the present pilot (see the "Limitations of the Experimental Design" subsection).

All participants completed a structured task protocol in one supervised session, consisting of four tasks completed in a fixed order to enable comparable exposure across participants: (i) scene description, where the participant asked the system to describe the immediate surroundings of a designated indoor test area; (ii) object identification, where the participant asked the system to identify a specific object placed within the camera's field of view; (iii) e-mail composition, where the participant dictated a short message for the system to compose and send through the Gmail-API tool; and (iv) media playback, where the participant requested a specific audio track through the YouTube-Data-API tool. After the four tasks, participants rated their overall satisfaction with the system on a 10-point scale (1 = not at all satisfied, 10 = extremely satisfied) and answered short structured questions on the intuitiveness of the voice interaction and their confidence in understanding their surroundings when using the system; these ratings form the basis of the quantitative findings reported in the "User Evaluation and Participant Feedback" subsection. Each session was supervised by a member of the research team who did not participate in the task but operated recording equipment, monitored participant safety, and recorded task-completion outcomes.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

In order to reach the sample sizes needed for the confidence intervals reported below, three core functions were benchmarked independently of the participant sessions using fixed, repeated-trial protocols rather than testing with the participant in the loop. Voice activation was evaluated over 50 wake-word trials for each of the quiet and moderate-noise conditions. We measured speech recognition with 50 speech samples for each condition and visual scene analysis with 50 images from eight common indoor objects (chairs, tables, doors, bottles, laptops, mobile phones, backpacks, and televisions). Object-recognition accuracy was calculated as the percentage of objects correctly identified across this image set. Speech-recognition accuracy was calculated as the percentage of commands correctly transcribed and correctly interpreted by the downstream tool-invocation layer. Success for message delivery, audio playback, and information retrieval was defined as task completion without system failure.

Statistical Analysis

Each functional benchmark is based on a fixed number of discrete trials, n = 50 per condition, not a large sample, so binomial accuracy metrics are reported with 95% Wilson score confidence intervals instead of the normal (Wald) approximation, which is known to perform poorly, producing intervals that may extend outside the valid range or have coverage below the nominal level for proportions near 0 or 1 or for small n. The Wilson interval for a sample proportion p observed over n trials, with z the standard normal critical value, is given by:

$$CI = \frac{\hat{p} + \frac{z^2}{2n} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \quad (2)$$

where z = 1.96 for 95% confidence. For the continuous end-to-end response-latency measurements, n = 20 independent runs, and a t-distribution-based confidence interval was used in place of a normal-approximation interval, appropriate for the small sample size, with $t(19, 0.975)$ = 2.093.

$$CI = \bar{x} \pm t_{(19,\,0.975)} \cdot \frac{s}{\sqrt{n}} \quad (3)$$

All confidence intervals reported in the "Results and Discussion" section were computed directly from the sample sizes and observed proportions or means already established in the functional and latency benchmarks; no additional data collection beyond the originally reported trials was required to compute them.

Results And Discussion

Functional performance evaluation

Table 5 reports functional performance across the core assistive functions, together with 95% Wilson score confidence intervals computed from the trial counts described above.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Function	Trials (*n*)	Observed Accuracy (%)	95% CI	Primary Source
Voice activation, quiet (<45 dB)	50	98	89.5–99.6	Third-party wake-word engine
Voice activation, moderate noise (60–70 dB)	50	85	72.6–92.4	Third-party wake-word engine
Speech recognition, quiet	50	93	82.5–97.4	Third-party STT service
Speech recognition, moderate noise	50	75	61.5–84.9	Third-party STT service
Visual scene analysis (8 object categories)	50 images	88	76.2–94.4	Third-party vision-language service
Message delivery (e-mail composition task)	10 (participant sessions)	No failures observed	Not computed (task-completion criterion;)	Original tool-invocation layer
Audio playback (media playback task)	10 (participant sessions)	No failures observed	Not computed (task-completion criterion)	Original tool-invocation layer
Information retrieval	Not independently fixed	Successful in observed interactions	Not computed (no fixed trial count)	LLM reasoning + original routing

TABLE 5: Functional performance evaluation with 95% confidence intervals

CI, Confidence Interval; dB, Decibel; LLM, Large Language Model; STT, Speech-To-Text

The moderate-width confidence intervals in Table *5*, such as a 95% CI of 61.5-84.9 around the 75% moderate-noise speech-recognition figure, are a consequence of the $n = 50$ trial-count protocol described above and should be taken into account when interpreting the point estimates; a larger multi-session, multi-site trial would be necessary to narrow these intervals sufficiently to support deployment-readiness claims.

Latency and processing analysis

As shown in Figure *4*, the average total system response time was 3.8 s, comprising wake-word detection (0.6 s), speech-to-text (1.2 s), LLM processing (1.3 s), and text-to-speech (0.8 s). The average shown here was computed over 20 independent test runs under stable Wi-Fi conditions. The response times ranged from 3.2 s to 4.7 s, with a mean of 3.8 s and a standard deviation of 0.4 s (95% CI 3.61-3.99 s, computed as described in the "Statistical Analysis" subsection). This latency can be experienced as a slight lag in fast-task switching, while it is acceptable for assistive conversational interaction. The major source of latency was the inference of LLMs in the cloud. The system performance is directly affected by the quality of the network. All response-time measurements reported above were obtained under stable Wi-

Fi conditions (minimum 10 Mbps); decreased bandwidth or higher network latency is expected to increase response time and may temporarily affect the reliability of cloud-based services such as speech processing and contextual reasoning, though this was not directly tested.

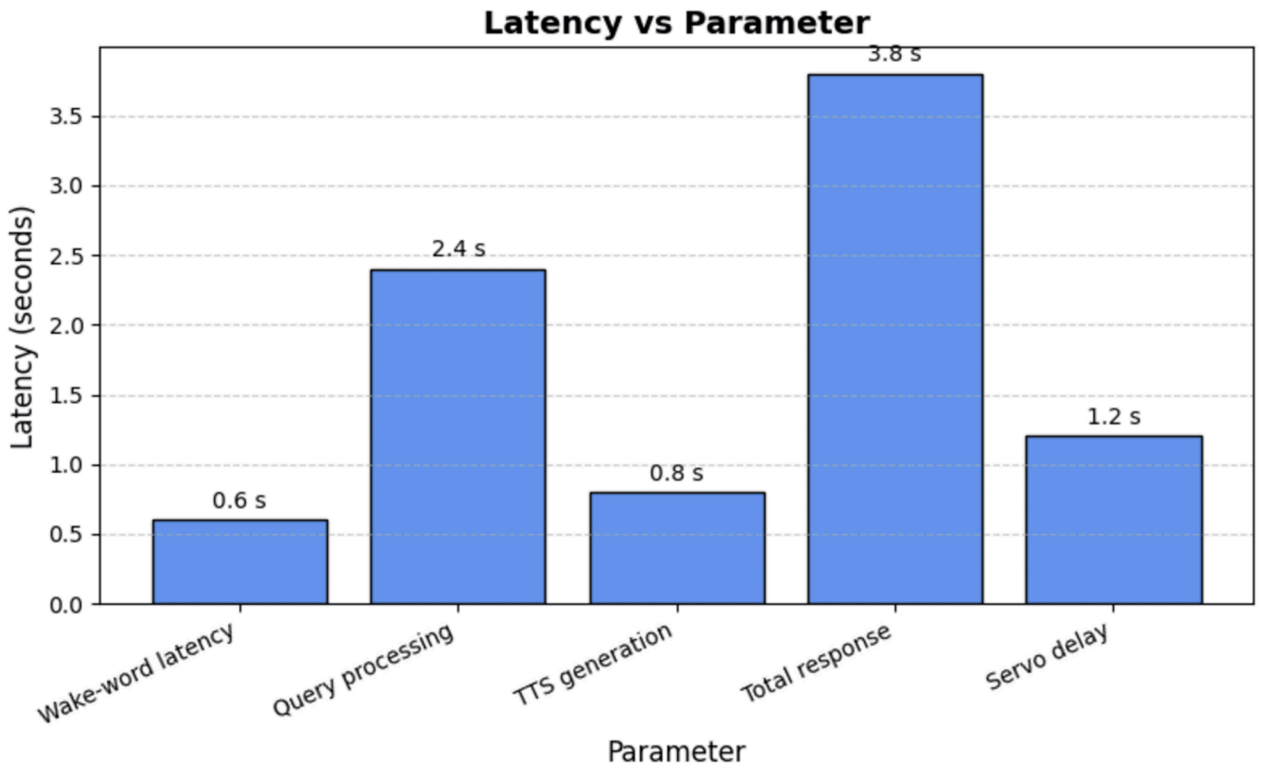

FIGURE 4: Response time graph

User evaluation and participant feedback

A pilot usability test was conducted with the 10 visually impaired participants summarized in Table *6* under monitored test conditions to evaluate the clarity of interaction, comfort, and perceived usefulness in the four tasks described in the "Participants and Task Protocol" subsection. The average user satisfaction score was 8.5/10; participants reported that voice interaction was intuitive (80%), and they felt more confident in understanding their surroundings (70%).

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Participant	Age Group	Age	Gender	Vision-Loss Onset	Tasks Completed	Satisfaction (1–10)	Voice Interaction Intuitive	Confidence in Surroundings
P01	Adolescent (9–17)	12	Male	Congenital	All 4 tasks	9	Yes	Yes
P02	Adolescent (9–17)	15	Female	Acquired (age 8)	All 4 tasks	8.5	Yes	Yes
P03	Adolescent (9–17)	10	Male	Congenital	All 4 tasks	9	Yes	No
P04	Adolescent (9–17)	16	Female	Congenital	All 4 tasks	8.5	No	Yes
P05	Adult (18–40)	28	Male	Acquired (age 22)	All 4 tasks	9	Yes	Yes
P06	Adult (18–40)	35	Female	Congenital	All 4 tasks	8	Yes	Yes
P07	Adult (18–40)	24	Male	Acquired (age 19)	All 4 tasks	9	Yes	No
P08	Adult (18–40)	31	Female	Acquired (age 25)	All 4 tasks	8	No	Yes
P09	Older adult (55+)	58	Male	Acquired (age 52)	All 4 tasks	8	Yes	Yes
P10	Older adult (55+)	62	Female	Congenital	All 4 tasks	8	Yes	No
Total/Average	**4 adolescents, 4 adults, 2 older adults**	**–**	**5 males, 5 females**	**–**	**All participants completed all 4 tasks**	**8.5**	**80% Yes (8/10)**	**70% Yes (7/10)**

TABLE 6: Participant-level demographics, task completion, and evaluation outcomes

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

Feedback from participants was mostly positive. Some participants noted that scene description via voice provided better spatial orientation than traditional obstacle warnings and that conversational interaction felt more natural than button-based assistive devices. Frequently mentioned liked features included hands-free operation, context-aware responses, and clear audio output. Limitations reported included a large lag in responses, decreased speech-recognition accuracy in noisy environments, some discomfort from prototype weight during extended wear, and decreased object-recognition performance in low-light environments. Proposed improvements were a smaller hardware form factor, faster response time, better noise filtering, and a wider camera field of view. Due to the small sample size and controlled testing conditions described above, these results should be interpreted as preliminary usability results rather than a full validation of real-world performance.

Discussion of user experience

Although participant feedback indicates a high level of acceptance of the proposed system, several practical limitations, identified through both direct participant reports and the functional and latency benchmarks reported above, would impact usability in real-world navigation scenarios, and each is linked here to a concrete measured value rather than discussed only in general terms.

The mean end-to-end response latency of 3.8 s (95% CI 3.61-3.99 s), primarily due to the cloud-based speech-to-text and LLM inference (see the "Latency and Processing Analysis" subsection), is consistent with participant reports of a noticeable response lag; for a user relying on the system for timely obstacle or hazard information, this delay may impair responsiveness in dynamic settings. Speech recognition accuracy decreased from 93% in quiet to 75% with moderate background noise (95% CI 61.5-84.9; Table *5*), which was consistent with participant reports that recognition was poor in noisy environments; this could lead to misinterpreted commands or repeated interactions in busy public spaces.

The object-recognition accuracy was 88% in normal indoor lighting conditions. The low-light performance was not separately benchmarked using a fixed trial protocol, but participants noted a loss of reliability in low-light conditions, which may constrain the effectiveness of scene understanding in nighttime or dim indoor use. As explained in the "Environment and Network Conditions" subsection, the system requires a stable internet connection to perform cloud-based reasoning, so network instability is expected to result in longer latencies or temporary limitations in accessing speech- and vision-dependent functionalities. Finally, a number of participants reported that the weight of the prototype was uncomfortable for extended wear, consistent with the prior discussion of the power budget relating the mass of the battery and enclosure to wearable comfort.

As described ahead in the "Toward Reduced Cloud Dependency" subsection, an important direction for future development will be to address these limitations with edge-based AI processing, better noise filtering, improved low-light vision capability, offline fallback functionality, and ergonomic hardware redesign.

Limitations of the experimental design

This work is explicitly a proof-of-concept study. It is intended to show that the proposed edge-cloud, LLM-integrated architecture is technically feasible and usable by visually impaired participants in a controlled setting, not to show real-world clinical efficacy, comparative superiority to existing commercial systems, or readiness for deployment. Concerning internal validity, all experiments were conducted in a single controlled indoor environment with good lighting, low noise, and stable network conditions; performance under variable outdoor lighting, intermittent connectivity, or ambient noise above the tested 60-70 dB moderate-noise condition has not been characterized.

In terms of external validity, the usability sample (n = 10) is small and from a single site, limiting statistical power and generalizability; the reported Wilson and t-distribution confidence intervals reflect this by remaining relatively wide, e.g., 76.2-94.4% for object-recognition accuracy. With respect to construct validity, the reported satisfaction score is a self-report taken in a single session immediately after exposure to the guided task and may not relate to sustained, unsupervised use in the real world. In terms of reliability, no test-retest or inter-rater reliability analysis was performed. Task success for the qualitative functions, i.e., message delivery, audio playback, and information retrieval, was scored by a single research team.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

In the *n* = 10 participant sessions, the message-delivery and audio-playback functions were measured as task-completion outcomes for metric coverage, rather than with an independently fixed trial count, as for the fixed n = 50 protocol used for voice activation, speech recognition, and visual scene analysis. The information-retrieval function was evaluated qualitatively, with no fixed trial count, and therefore accuracy percentages and confidence intervals are not reported for these three functions in Table *5*, and their evaluation should be considered less rigorous than the benchmarked perceptual functions.

As shown in Table *2* and the Third-Party Cloud Services subsection, the majority of the reported perceptual accuracy comes from third-party cloud services and not from novel algorithms developed by the authors. The evaluation measures the feasibility of the integrated system, rather than isolating the marginal contribution of the authors' own algorithmic work. These six points, taken together, delineate the boundary of what this study does and does not show: it offers evidence that the integrated pipeline works end-to-end and is usable in a supervised, controlled setting, and it does not offer evidence of outdoor robustness, large-population generalizability, long-term reliability, or effectiveness relative to alternative systems, each of which would require a separate, purpose-designed study.

Additionally, this pilot study, including sessions with adolescent participants, relied on researcher-administered informed consent (with parental consent for minors) and did not undergo external institutional ethics-board review, as no such review process was available through the authors' institution for this type of study. Future work involving human participants, particularly minors, will seek formal ethics review through an appropriate external board prior to data collection.

Toward reduced cloud dependency

Table *2* and the "Latency and Processing Analysis" subsection together indicate that cloud-hosted speech-to-text and LLM reasoning are both the major sources of the system's functional capability and its primary limitations in terms of latency and offline availability. The Raspberry Pi AI HAT+, based on the Hailo-8/8L NPU, offers 13-26 TOPS (tera-operations per second) of dedicated inference performance for a Raspberry Pi 5, fully integrated within the Raspberry Pi camera software stack [34], and could host an on-device object-detection model such as a YOLOv8-class detector, as validated in a comparable smart-glasses deployment [19], to provide near-instantaneous local object and obstacle feedback in parallel with cloud LLM reasoning, in line with the edge-cloud collaborative architectures reported in the LLM-serving literature [25,26].

The newer Raspberry Pi AI HAT+ 2 (January 2026) is based on the Hailo-10H NPU, provides 8 GB of on-module memory, and 40 TOPS (INT4) of inference performance sufficient to run small language and vision-language models locally, with reported token-generation throughput adequate for interactive, if not fully cloud-parity, local conversational responses [35]. The cloud LLM is kept for complex, open-ended reasoning and tool calls, which benefit from a larger foundation model, while wake-word confirmation, a lightweight intent-triage step, and a local object-detection pass are migrated onto such an NPU-accelerated Raspberry Pi 5 platform. This is consistent with the edge-cloud collaborative design pattern reported in the LLM-serving literature to reduce end-to-end latency by as much as 40-50% compared to an all-cloud baseline [25-27] and would additionally preserve basic obstacle and object feedback during network outages, directly addressing the offline-availability limitation identified above. Instead of claiming work already done in the current prototype, we propose the migration and a better thermally managed enclosure with a lighter form factor as the primary direction for the next iteration of this platform.

Conclusions

The proposed system demonstrates the feasibility of integrating AI, IoT connectivity, and embedded hardware to provide a proof-of-concept solution for visual impairment assistance. The modular architecture enables smooth interaction between on-device capabilities and actuation capabilities with cloud-based smart services.

The system is designed to allow the user to give voice commands and receive real-time feedback on the environmental conditions, facilitating interaction through natural and context-sensitive speech-based processing. The effectiveness of the suggested approach is demonstrated through reliable wake-word activation, accurate voice-driven speech control, and contextually appropriate natural-language response generation, validating the feasibility of the proposed architecture.

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Dhiraj Rajput, Sushant Kumar, Ashish Sharma, Yashraj Narke, Dhanashri Wategaonkar

Acquisition, analysis, or interpretation of data: Dhiraj Rajput, Sushant Kumar, Ashish Sharma, Yashraj Narke, Dhanashri Wategaonkar

Drafting of the manuscript: Dhiraj Rajput, Sushant Kumar, Ashish Sharma, Yashraj Narke, Dhanashri Wategaonkar

Critical review of the manuscript for important intellectual content: Dhiraj Rajput, Sushant Kumar, Ashish Sharma, Yashraj Narke, Dhanashri Wategaonkar

Supervision: Dhanashri Wategaonkar

Disclosures

Human subjects: Consent was obtained or waived by all participants in this study. Not applicable issued approval Not applicable. Informed consent was obtained from all adult participants and from parents/guardians of adolescent participants prior to participation. No formal institutional ethics-board review was available or obtained for this pilot study, as the authors' institution does not maintain a formal review process for this type of non-invasive usability testing. No personally identifiable information was collected, stored, or reported.

Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.

Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the following:

- **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work.
- **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work.
- **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

Data Availability Statements

The datasets (and/or code) supporting this study are available from the corresponding author upon reasonable request.

References

1. Kuriakose B, Shrestha R, Sandnes FE: Tools and technologies for blind and visually impaired navigation support: a review. IETE Technical Review. 2022, 39:3-18. 10.1080/02564602.2020.1819893

2. World Health Organization: Blindness and vision impairment. (2023). Accessed: December 25, 2025: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment.

3. Abidi MH, Noor Siddiquee A, Alkhalefah H, Srivastava V: A comprehensive review of navigation systems for visually impaired individuals. Heliyon. 2024, 10:e31825. 10.1016/j.heliyon.2024.e31825

4. Microsoft: Seeing AI: A Visual Assistant for the Blind. Accessed: April 30, 2026: https://www.microsoft.com/en-us/ai/seeing-ai.

5. OrCam: OrCam MyEye 2 Pro. Accessed: April 30, 2026: https://www.orcam.com/en-us/orcam-myeye-2-pro.

6. Abdel-Jaber H, Albazar H, Abdel-Wahab A, Amir ME, Alqahtani A, Alobaid M: Mobile based IoT solution for helping visual impairment users. Advances in Internet of Things. 2021, 11:141-152. 10.4236/ait.2021.114010

7. Zhang X, Huang X, Ding Y, Long L, Li W, Xu X: Advancements in smart wearable mobility aids for visual impairments: A bibliometric narrative review. Sensors. 2024, 24:7986. 10.3390/s24247986

8. Campero-Jurado I, Márquez-Sánchez S, Quintanar-Gómez J, Rodríguez S, Corchado JM: Smart helmet 5.0 for industrial Internet of Things using artificial intelligence. Sensors. 2020, 20:6241. 10.3390/s20216241

9. Ferrara E: Large language models for wearable sensor-based human activity recognition, health monitoring, and behavioral modeling: A survey of early trends, datasets, and challenges. Sensors. 2024, 24:5045. 10.3390/s24155045

10. Fu B, Hadid A, Damer N: Generative AI in the context of assistive technologies: Trends, limitations and future directions. Image and Vision Computing. 2025, 154:105347. 10.1016/j.imavis.2024.105347

11. Xu P, Zhu X, Clifton DA: Multimodal learning with transformers: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2023, 45:12113-12132. 10.1109/tpami.2023.3275156

12. Zong M, Hekmati A, Guastalla M, Li Y, Krishnamachari B: Integrating large language models with Internet of Things applications [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2410.19223

13. Zhao Y, Zhang Y, Xiang R, Li J, Li H: VIALM: A survey and benchmark of visually impaired assistance with large models [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2402.01735

14. Azzino T, Mezzavilla M, Rangan S, Wang Y, Rizzo JR: 5G edge vision: Wearable assistive technology for people with blindness and low vision. 2024 IEEE Wireless Communications and Networking Conference (WCNC), Dubai, United Arab Emirates. 2024, 1-6. 10.1109/WCNC57260.2024.10570607

15. Li X, Gandhi B, Zhan M, et al.: Fine-tuning vision-language models for visual navigation assistance [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2509.07488

16. Liu R, Zhang J, Schön A, et al.: ObjectFinder: open-vocabulary assistive system for interactive object search by blind people [PREPRINT]. arXiv. 2024, 10.48550/arXiv.2412.03118

17. Zhang H, Falletta NJ, Xie J, Yu R, Lee S, Billah SM, Carroll JM: Enhancing the travel experience for people with visual impairments through multimodal interaction: NaviGPT, a real-time AI-driven mobile navigation system. Companion Proceedings of the 2025 ACM International Conference on Supporting Group Work. 2025, 29-35. 10.1145/3688828.3699636

18. Pfreundschuh P, Cioffi G, von Einem C, et al.: SightGuide: A wearable assistive perception and navigation system for the Vision Assistance Race in the Cybathlon 2024 [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2506.02676

19. Kumari P, Hammady R: Assisting blind people with AI and audio using smart glasses: System design with YOLOv8 variants comparisons. Multimedia Systems. 2026, 32:73. 10.1007/s00530-025-02139-z

20. Trigka M, Dritsas E: A comprehensive survey of machine learning techniques and models for object detection. Sensors. 2025, 25:214. 10.3390/s25010214

21. Bradshaw J, Riahi Alam M, Ainary B, Kim M, Amini Salehi M: Audo-Sight: AI-driven ambient perception across edge-cloud for blind and low vision users [PREPRINT]. arXiv. 2026, 10.48550/arXiv.2603.13668

22. Waisberg E, Ong J, Masalkhi M, Zaman N, Sarker P, Lee AG, Tavakkoli A: Meta smart glasses—Large language models and the future for assistive glasses for individuals with vision impairments. Eye. 2024, 38:1036-1038. 10.1038/s41433-023-02842-z

23. Seiple W, van der Aa HPA, Garcia-Piña F, Greco I, Roberts C, van Nispen R: Performance on activities of daily living and user experience when using artificial intelligence by individuals with vision impairment. Translational Vision Science & Technology. 2025, 14:3. 10.1167/tvst.14.1.3

24. Zaman N, Potluri V, Biggs B, Coughlan JM: WhatsAI: Transforming Meta Ray-Bans into an extensible generative AI platform for accessibility [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2505.09823

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x

25. Chen Y, Zhao J, Han H: A survey on collaborative mechanisms between large and small language models [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2505.07460

26. Qu G, Chen Q, Wei W, Lin Z, Chen X, Huang K: Mobile edge intelligence for large language models: A contemporary survey. IEEE Communications Surveys & Tutorials. 2025, 27:3820-3860. 10.1109/comst.2025.3527641

27. Jiang S, Zhou X, Zhang M, Xu C, Liao G, Chen J, Cao J: Edge large language models: A comprehensive survey. CCF Transactions on Pervasive Computing and Interaction. 2026, 8:181-210. 10.1007/s42486-025-00227-7

28. Hu J, Zhong M, Chen K, Bai X, Zhang M: Agentic tool use in large language models [PREPRINT]. arXiv. 2026, 10.48550/arXiv.2604.00835

29. Ehtesham A, Singh A, Gupta GK, Kumar S: A survey of agent interoperability protocols: Model Context Protocol (MCP), Agent Communication Protocol (ACP), Agent-to-Agent Protocol (A2A), and Agent Network Protocol (ANP) [PREPRINT]. arXiv. 2025, 10.48550/arXiv.2505.02279

30. Wan Z, Wang X, Liu C, et al.: Efficient large language models: A survey [PREPRINT]. arXiv. 2023, 10.48550/arXiv.2312.03863

31. Qin Y, Hu S, Lin Y, et al.: Tool learning with foundation models. ACM Computing Surveys. 2024, 57:1-40.

32. Mukhopadhyay SC, Tyagi SKS, Suryadevara NK, Piuri V, Scotti F, Zeadally S: Artificial intelligence-based sensors for next generation IoT applications: A review. IEEE Sensors Journal. 2021, 21:24920-24932. 10.1109/jsen.2021.3055618

33. Aouedi O, Vu TH, Sacco A, Nguyen DC, Piamrat K, Marchetto G, Pham QV: A survey on intelligent Internet of Things: Applications, security, privacy, and future directions. IEEE Communications Surveys & Tutorials. 2024, 27:1238-1292. 10.1109/comst.2024.3430368

34. Raspberry Pi: AI HATs. Accessed: July 5, 2026: https://www.raspberrypi.com/documentation/accessories/ai-hat-plus.html.

35. Raspberry Pi: Introducing the Raspberry Pi AI HAT+ 2: Generative AI on Raspberry Pi 5. (2026). Accessed: July 5, 2026: https://www.raspberrypi.com/news/introducing-the-raspberry-pi-ai-hat-plus-2-generative-ai-on-raspberry-pi-5/.

How to cite this article:
Sharma A, Rajput D, Kumar S, et al. (September 07, 2026) Integrating Computer Vision and Large Language Models for Real-Time Blind Navigation. Cureus J Comput Sci 3 : es44389-026-00265-x. DOI https://doi.org/10.7759/s44389-026-00265-x