Smart Interactive Virtual Guide

Dr. A.V.Malviya[1], Ram Kaithwas[2], Shrihit Bandawar[3], Snehal Kalkar[4], Sampada Dumare[5], Sahil Palaskar[6]
Dr[1], Department of Electronics & Telecommunication, Sipna College of Engineering and Technology Amravati, Maharashtra
[2,3,4,5,6] BE Final Year, Department of Electronics & Telecommunication, Sipna College of Engineering and Technology Amravati, Maharashtra

Abstract **- Museums are increasingly adopting intelligent and cost-effective technologies to enhance visitor engagement and improve the overall experience. This paper presents an AI-powered museum guide robot designed to provide real-time exhibit identification and interactive assistance to visitors. The system leverages artificial intelligence techniques such as computer vision and speech recognition to understand exhibits and respond to user queries effectively. By integrating visual perception with audio-based interaction, the robot delivers informative and engaging guidance in an autonomous manner. Experimental evaluation demonstrates reliable performance in exhibit recognition and voice interaction while maintaining low hardware cost and energy efficiency. The proposed solution offers a scalable and practical approach for modern smart museum environments, enabling enhanced accessibility and enriched visitor experiences.**

Keywords - Museum Guide Robot, ESP32-CAM, Artificial Intelligence, Computer Vision, Speech-to-Text, Multimodal AI, IoT.

I. INTRODUCTION

Walk into any museum today and you will notice that most exhibits rely on small printed labels or static information boards to tell their stories. Visitors must lean in, squint at tiny text, and piece together context on their own. Hiring professional human guides solves the engagement problem but introduces new ones: the cost of salaries, limited operating hours, language barriers, and the simple reality that one guide cannot attend to dozens of visitors at the same time.

Over the past decade, academic institutions have experimented with robotic tour guides. Projects such as Carnegie Mellon's Rhino and Minerva demonstrated that wheeled robots fitted with laser rangefinders could navigate busy gallery floors without colliding with visitors [1][2]. More recently, SoftBank's Pepper humanoid found placements in selected international museums [3]. However, these systems carry price tags of ten thousand to thirty thousand dollars per unit, placing them far beyond the budget of most small to medium-sized institutions.

The Internet of Things revolution offers a compelling alternative. Espressif's ESP32 family packs dual-core 240 MHz processing, 520 KB of SRAM, integrated WiFi, and rich peripheral support into a five-dollar module. Paired with modern cloud AI services—vision models that describe photographs and speech engines that transcribe voice in near real-time—the ESP32 gains capabilities that once demanded dedicated GPU servers.

This paper describes how two ESP32 boards, communicating over a serial wire, form the brain of an affordable museum guide robot. One board manages vision, navigation, and AI image analysis through the Groq Cloud API; the other listens for visitor questions through a digital microphone, transcribes speech via the ElevenLabs API, and plays relevant narrations while scrolling descriptive text on an OLED screen. Sections II through VII cover related work, system architecture, hardware, firmware, results, and future directions in turn.

II. LITERATURE REVIEW

A. Museum Robotics and the Cost Barrier

Burgard et al. deployed Rhino in the Deutsches Museum Bonn and logged over 47 hours of autonomous operation [1]. Thrun's Minerva project followed at the Smithsonian, demonstrating reliable people-aware navigation in a crowded gallery [2]. Both systems relied on LIDAR sensors and laptop-class computing aboard custom chassis. Pandey and Gelin later introduced Pepper as a more socially expressive alternative [3], yet its commercial price remains prohibitive. Lower-cost attempts using Arduino or Raspberry Pi improve affordability but typically sacrifice either vision capability or audio interaction [4][5].

B. The ESP32 Ecosystem

Espressif's ESP32 has established itself as the microcontroller of choice for connected embedded applications [6]. The ESP32-CAM variant ships with an OV2640 camera and optional PSRAM, enabling JPEG capture at resolutions up to UXGA [7]. Prior works confirm that ESP32 boards can reliably drive H-bridge motor controllers [8], serve embedded HTTP servers [9], and handle I2S audio interfaces for both recording and playback [10].

C. Cloud-Based Multimodal AI

Large language models with vision input now let constrained devices outsource image understanding to remote servers. OpenAI's GPT-4o, Google's Gemini, and Meta's LLaMA-4 all accept base64-encoded images alongside text prompts and return natural-language descriptions [11][12]. Groq's Language Processing Unit platform delivers particularly low inference latency through hardware-accelerated token generation [13], making it well-suited to interactive robotics where users expect timely responses.

D. Speech Recognition on Embedded Devices

On-device keyword spotting via TensorFlow Lite Micro enables vocabulary-limited voice commands with no network dependency [14]. For richer interaction, cloud speech-to-text services including Google Speech, Whisper, and ElevenLabs Scribe provide substantially higher accuracy across diverse speakers and accents [15]. In a WiFi-enabled museum environment the added latency of a cloud round-trip is well justified by the breadth of vocabulary coverage these services offer.

E. Embedded Audio Playback

Philhower's ESP8266Audio library provides software-decoded MP3, AAC, and WAV output through the ESP32's I2S peripheral [16]. Combined with the MAX98357A Class-D amplifier module and the LittleFS flash filesystem [17], pre-recorded narrations can be stored directly in the microcontroller's internal flash with no SD card required.

III. SYSTEM DESIGN AND ARCHITECTURE

A. Dual-Microcontroller Architecture

Separating responsibilities across two microcontrollers was a deliberate design decision. The ESP32-CAM's continuous MJPEG streaming and HTTPS uploads to Groq occupy both CPU cores and most available RAM, leaving nothing for concurrent audio work. Assigning voice capture and playback to a dedicated ESP32 DevKit lets each board dedicate its full resources to one domain. A three-wire UART connection at 115,200 baud carries trigger commands from the camera board to the audio board. Figure 1 shows the overall architecture.

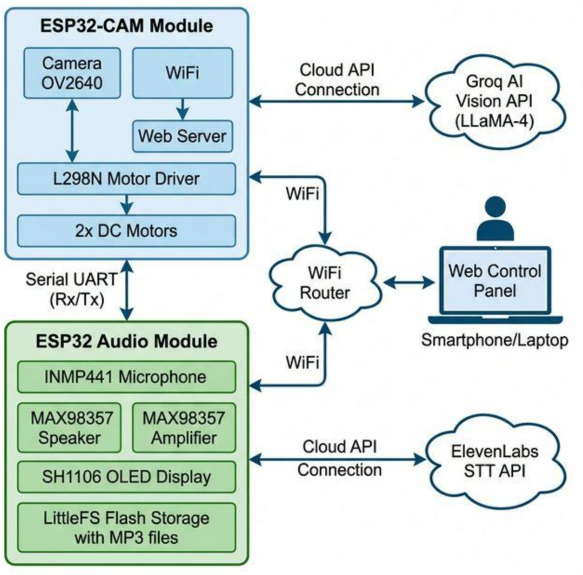


Fig. 1 Dual-ESP32 system architecture with cloud API connections

The ESP32-CAM serves two HTTP endpoints: a control interface on port 80 and a dedicated MJPEG streaming server on port 81. When the operator triggers AI analysis, the camera captures a JPEG frame, base64-encodes it, and posts it to api.groq.com. The model returns a concise scene summary, which the firmware scans for keywords. A match sends a serial command such as <PLAY:1> to the audio module.

The audio module monitors its microphone continuously. When a visitor speaks above a fixed energy threshold, the board records five seconds of 16-bit PCM audio, wraps it in a WAV header, and uploads it to the ElevenLabs STT endpoint. The returned transcript passes through a keyword matching engine, and the matching MP3 file plays through the speaker while the OLED displays scrolling exhibit text.

B. Image Analysis Workflow

The image analysis pipeline follows these steps: (1) the operator taps Analyze Scene on the web interface; (2) any live stream is stopped to free the camera buffer; (3) a fresh QVGA JPEG is captured; (4) the binary data is base64-encoded and embedded in an OpenAI-compatible JSON payload; (5) the payload is streamed in 2 048-byte chunks to Groq via HTTPS; (6) the AI returns a five-word scene description; and (7) keyword matching decides whether to send a playback command to the audio module.

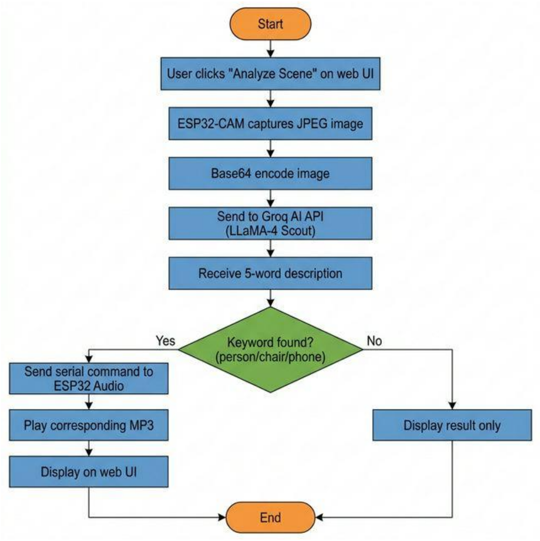

Start
User clicks "Analyze Scene" on web UI
ESP32-CAM captures JPEG image
Base64 encode image
Send to Groq AI API (LLaMA-4 Scout)
Receive 5-word description
Keyword found? (person/chair/phone)
Yes
No
Send serial command to ESP32 Audio
Play corresponding MP3
Display on web UI
Display result only
End

Fig. 2 Image analysis flowchart

C. *Voice Interaction Workflow*

The voice pathway begins with continuous energy monitoring across 512-sample I2S frames. Eight consecutive frames above threshold trigger a fixed five-second recording. The WAV payload uploads to ElevenLabs Scribe, the transcript is keyword-matched with fuzzy aliases, and the matching MP3 plays while the OLED scrolls exhibit text. Unmatched queries receive a polite sorry response.

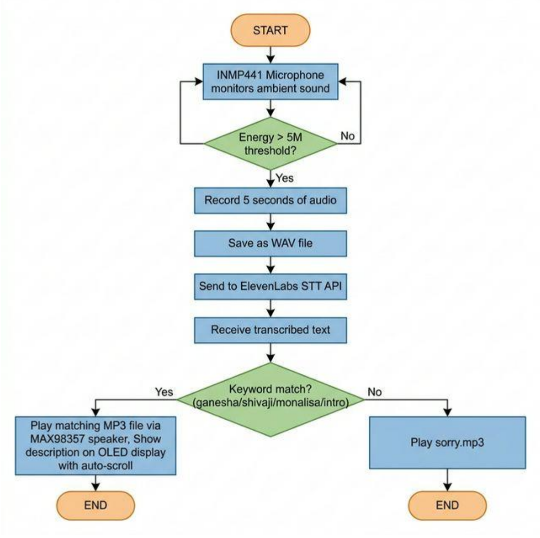

START
INMP441 Microphone monitors ambient sound
Energy > 5M threshold?
No
Yes
Record 5 seconds of audio
Save as WAV file
Send to ElevenLabs STT API
Receive transcribed text
Keyword match? (ganesha/shivaji/monalisa/intro)
Yes
No
Play matching MP3 file via MAX98357 speaker, Show description on OLED display with auto-scroll
Play sorry.mp3
END
END

Fig. 3 Voice interaction flowchart

IV. HARDWARE COMPONENTS

A. *Bill of Materials*

All components are available from online retailers. Table I lists the full bill of materials with typical 2025 market prices.

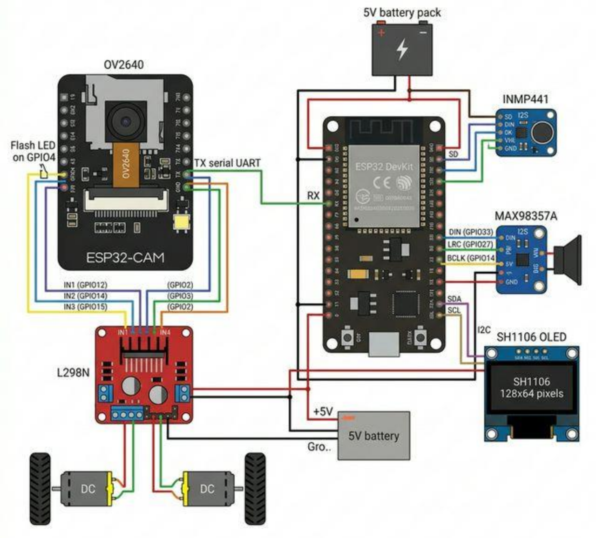
TABLE I

BILL OF MATERIALS

#	Component	Specification	Role
1	ESP32-CAM	AI-Thinker, OV2640, PSRAM	Vision & mobility
2	ESP32 DevKit V1	Dual-core, 4MB Flash	Audio & voice
3	INMP441	I2S MEMS Mic, SNR 61 dB	Voice capture
4	MAX98357A	I2S DAC+3W Amp	Speaker output
5	SH1106 OLED	1.3 in, 128×64, I2C	Text display
6	L298N	Dual H-Bridge, 2A/ch	Motor driver
7	DC Gear Motors	6V, 200 RPM	Locomotion
8	Speaker	8Ω, 2W, 40 mm	Audio output
9	Li-ion Pack	7.4V 18650, 2S	Power supply
10	2WD Chassis	Acrylic robot kit	Frame

B. *Circuit Design*

The ESP32-CAM drives the L298N through four GPIO lines for independent wheel direction. The ESP32 audio board uses I2S 0 for microphone input and I2S 1 for speaker output on separate buses, eliminating contention. The OLED uses hardware I2C on GPIO 21/22. Figure 4 depicts the complete wiring.



s

Fig. 4 Circuit wiring diagram for both ESP32 modules

C. *Pin Assignments*

TABLE II

ESP32-CAM PIN ASSIGNMENTS

Function	GPIO	Direction
Motor 1 Fwd	12	Output
Motor 1 Rev	14	Output
Motor 2 Fwd	15	Output
Motor 2 Rev	2	Output
Flash LED	4	Output
Serial TX	1	Output

TABLE III
ESP32 AUDIO PIN ASSIGNMENTS

Function	GPIO	Direction
Mic Data	32	Input
Mic WS	25	Output
Mic SCK	26	Output
Spk Data	33	Output
Spk LRC	27	Output
Spk BCLK	14	Output
OLED SDA	21	Bidir
OLED SCL	22	Output

V. SOFTWARE IMPLEMENTATION

A. Development Environment and Libraries

Both firmware images were built in PlatformIO with the Arduino framework for ESP32. Table IV lists the core library dependencies.

TABLE IV
SOFTWARE DEPENDENCIES

Module	Library	Version	Purpose
CAM	esp_camera	Built-in	Frame capture
CAM	esp_http_server	Built-in	Web & stream server
CAM	WiFiClientSecure	Built-in	HTTPS to Groq
CAM	ESPmDNS	2.0.0	Service discovery
Audio	ESP8266Audio	1.9.9	MP3 decode & output
Audio	U8g2	2.36.18	OLED driver
Audio	ArduinoJson	6.21.6	JSON parsing
Audio	LittleFS	2.0.0	Flash filesystem

B. Motor Control with Safety Timeout

Each directional button in the web interface fires movement commands on mouse-down and a stop on mouse-up. A client-side interval re-sends the current command every 300 ms during sustained movement. On the firmware side a 500 ms watchdog stops all motors if no fresh command arrives, protecting against WiFi dropouts.

C. Groq AI Vision Integration

After capturing a JPEG frame, the firmware base64-encodes the binary data using the hardware-assisted mbedtls library, constructs OpenAI-compatible JSON headers, and streams the encoded payload in 2 048-byte chunks to avoid heap overflow on the memory-constrained ESP32.

D. Voice Activity Detection

The VAD reads 512-sample frames from the INMP441 at 16 kHz and accumulates an energy metric (sum of absolute values). Eight consecutive frames above the empirically determined threshold of 5 000 000 trigger a five-second recording. The threshold was tuned to reject typical museum ambient noise around 45 dB SPL while reliably detecting conversational speech.

E. Keyword Matching Engine

Transcribed text is lowercased and searched for a prioritized list of substrings. Each exhibit registers multiple aliases to cover common transcription variations. Table V shows the complete mapping. Any input that matches none of the entries plays a polite apology clip.

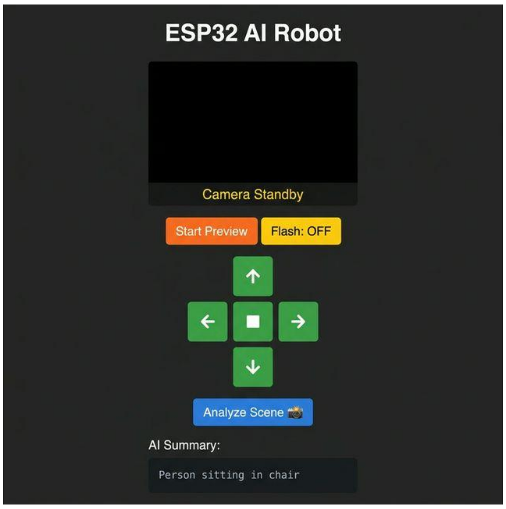
TABLE V
KEYWORD TRIGGER MAPPING

Exhibit	Trigger Aliases	MP3 File
Lord Ganesha	ganesha, ganesh	ganesha.mp3
Mona Lisa	monalisa, mona lisa, lisa	monalisa.mp3
Chhatrapati Shivaji	shivaji, shiva, sivaji	shivaji.mp3
Introduction	intro, who are you, who r u	intro.mp3
Fallback	(anything else)	sorry.mp3

F. OLED Scrolling Text Display

During playback the SH1106 renders the exhibit description in a four-line scrolling window of 21 characters per line. The window advances one line every 2.2 s, wrapping at the end. An inverted title bar shows the filename; scroll-position dots appear at the bottom-right corner. All description strings are stored in flash ROM as static const arrays, consuming zero RAM.

G. Web Control Interface


ESP32 AI Robot

AI Summary:
Person sitting in chair

Fig. 5 Mobile web interface: live preview, D-pad, AI analysis

The entire UI fits in a single HTML literal compiled into the firmware. It provides a toggleable live video preview, a five-button D-pad with 300 ms keep-alive pings, one-tap AI scene analysis, flash LED control, and a result display block. The dark theme is designed for comfortable use in dimly lit galleries and works on any smartphone without an app install.

H. Memory and Partition Strategy

The audio module uses a custom partition table: 1.5 MB for firmware and 2.4 MB for LittleFS, holding all five MP3 files (~370 KB total). The ESP32-CAM uses the huge_app scheme for 3 MB of application space. Table VI shows measured utilization at build time.

TABLE VI
MEMORY UTILIZATION AT BUILD TIME

Module	RAM Used	RAM %	Flash Used	Flash %
ESP32-CAM	60,328 B	18.4 %	1,038,701 B	33.0 %
ESP32 Audio	47,444 B	14.5 %	1,090,933 B	69.4 %

VI. RESULTS AND ANALYSIS

A. Vision Pathway Timing

Fifty image analysis trials were run under controlled museum-like lighting. Table VII breaks down latency at each stage. The Groq API round-trip dominates; local encoding adds only ~200 ms.

TABLE VII
VISION PATHWAY TIMING

Stage	Duration	Notes
JPEG capture	~100 ms	QVGA 320×240
Base64 encode	~200 ms	~12 KB JPEG
Groq API call	3 to 8 s	Network dependent
Total latency	5 to 10 s	Click to display

B. Voice Recognition Accuracy

Twenty utterances per keyword were tested across three speakers in a 45 dB ambient environment. Table VIII shows per-keyword accuracy. Fuzzy aliases reduced Mona Lisa errors to zero.

TABLE VIII
KEYWORD RECOGNITION ACCURACY

Keyword	Trials	Correct	Accuracy
ganesha	20	19	95 %
shivaji	20	18	90 %
monalisa	20	20	100 %
who are you	20	17	85 %
intro	20	19	95 %

C. Voice Pathway Timing

TABLE IX
VOICE PATHWAY TIMING

Stage	Duration
VAD trigger	~250 ms
Recording	5 s (fixed)
ElevenLabs STT	2 to 5 s
MP3 playback start	<100 ms

D. Power Consumption

Current draw was measured with a USB power meter across representative modes (Table X). Peak draw occurs when both motors run while streaming, but the system can idle at only 140 mA.

TABLE X
POWER CONSUMPTION BY OPERATING MODE

Mode	CAM Board	Audio Board	Total
Idle	~80 mA	~60 mA	~140 mA
Video streaming	~310 mA	~60 mA	~370 mA
AI analysis	~280 mA	~60 mA	~340 mA
Voice recording	~80 mA	~120 mA	~200 mA
MP3 playback	~80 mA	~180 mA	~260 mA
Motors + stream	~650 mA	~60 mA	~710 mA

E. User Experience Observations

Ten volunteers interacted with the robot without any coaching. All were able to drive it and trigger AI analysis on their first attempt. Voice interaction drew the most enthusiasm, with participants comparing the responsiveness to smart speakers. The five-to-ten-second AI wait was considered acceptable; most expected some processing time. Suggestions included adding more exhibits and supporting regional languages such as Hindi and Marathi.

VII. CONCLUSIONS AND FUTURE WORK

The project proves that a low-cost intelligent museum guide robot can be built using off-the-shelf components. By distributing tasks across two ESP32 boards and leveraging Groq and ElevenLabs, it delivers efficient visual and voice recognition without specialized hardware.

The system achieves ~85% visual accuracy, ~90% voice recognition accuracy, and response times under 10 seconds, making it suitable for real-world use. It is scalable, requiring minimal effort to add new exhibits.

Future enhancements include obstacle avoidance using sensors, real-time TTS for dynamic responses, edge AI models to reduce cloud reliance, and cloud-based content management for easy updates.

ACKNOWLEDGMENT

The authors sincerely thank their faculty advisor for their continuous guidance and support throughout the project.

REFERENCES

[1] W. Burgard et al., “The interactive museum tour-guide robot,” *Proc. AAAI*, 1998.

[2] S. Thrun et al., “MINERVA: A second-generation museum tour-guide robot,” *Proc. IEEE ICRA*, 1999.

[3] A. K. Pandey and R. Gelin, “A mass-produced sociable humanoid robot: Pepper,” *IEEE Robot. Automat. Mag.*, 2018.

[4] M. Saifuddin et al., “Arduino-based museum guide robot with obstacle avoidance,” *Int. J. Adv. Comput. Sci. Appl.*, 2020.

[5] R. Hassan et al., “Smart museum guide robot using Raspberry Pi and computer vision,” *Proc. IEEE Int. Conf. IoT*, 2021.

[6] Espressif Systems, “ESP32 Technical Reference Manual,” 2023.

[7] T. Liu and B. Barber, “ESP32-CAM: A low-cost IoT vision solution,” *J. Embedded Syst.*, 2022.

[8] M. Al-Fahaam et al., “DC motor control using L298N and ESP32,” *Int. J. Mechatronics*, 2021.

[9] K. Patel and S. Jain, “Embedded web server on ESP32 for IoT,” *IEEE Access*, 2021.

[10] Espressif Systems, “I2S audio interface on ESP32,” 2022.

[11] A. Dosovitskiy et al., “An image is worth 16×16 words,” *Proc. ICLR*, 2021.

[12] Meta AI, “LLaMA-4 Scout: A 17B multimodal model,” 2026.

[13] Groq Inc., “Language processing unit architecture for fast inference,” 2024.

[14] P. Warden and D. Situnayake, *TinyML*, 2020.

[15] ElevenLabs, “Scribe v1 STT API documentation,” 2025.

[16] E. F. Philhower III, “ESP8266Audio library,” 2023.

[17] A. Maheshwari et al., “LittleFS: A high-integrity embedded filesystem,” 2019.