



AN INTELLIGENT EMBEDDED VISION SYSTEM FOR ASSISTING VISUALLY IMPAIRED USERS USING EDGE COMPUTING AND IoT

¹Suraj Fulchand Chawan, ²Tejas Suhdhakar Gulhane, ³Siddhant Shripal Mundre, ⁴Subodh Devidas Pal,

⁵Prof. D. G. Ingale

^{1,2,3,4}Student, ⁵Guide, Computer Science and Engineering

^{1,2,3,4} Student, Dr. Rajendra Gode Institute of Technology and Research, Amravati, IN,

¹Dr. Rajendra Gode Institute of Technology and Research, Amravati, IN

Abstract: The attainment of independent mobility for visually impaired individuals remains one of the most significant sociological and technical challenges in contemporary India, where the complexity of urban infrastructure often renders traditional navigational aids insufficient. This research proposes the design and implementation of an intelligent embedded vision system, specifically optimized for the Indian socio-economic and infrastructural landscape. The system, titled "Bharat-Assist," utilizes the ESP32-CAM microcontroller as a primary edge computing node to perform real-time object detection and obstacle avoidance without the prohibitive latency and reliability issues associated with cloud-based processing. By integrating lightweight deep learning models, specifically quantized versions of SSD-MobileNet and TensorFlow Lite Micro, the system provides semantic environmental awareness through bone conduction audio feedback. This methodology preserves the user's critical auditory situational awareness while providing high-fidelity guidance. Experimental results demonstrate that the localized edge processing architecture reduces end-to-end latency to approximately 860 ms, a significant improvement over cloud-dependent models that exhibit latencies exceeding 6.8 seconds in bandwidth-constrained Indian environments. Furthermore, the system is designed for affordability, with a total estimated bill of materials between ₹3,500 and ₹5,000, ensuring scalability in low-resource settings. Real-world testing in multifunctional Indian streets reveals a robust detection accuracy of 91.1% for common urban obstacles. The integration of IoT capabilities further enables periodic telemetry for caregiver monitoring, while maintaining a strict focus on user privacy through on-device inference. This research offers a viable, cost-sensitive alternative to high-end assistive technologies, bridging the gap between advanced computer vision and practical, everyday utility for the visually impaired community in India.

Keywords: Edge Computing, Embedded Vision, ESP32-CAM, IoT, Assistive Technology, Computer Vision, TinyML, Indian Urban Infrastructure, SSD-MobileNet, Visual Impairment Navigation.

I. INTRODUCTION

Independent mobility is widely recognized as a fundamental pillar of human autonomy and quality of life. However, for individuals living with visual impairment, the simple act of navigating from one point to another is a task dominated by cognitive strain and physical risk. The global prevalence of blindness and visual impairment is significantly concentrated in developing nations, with India accounting for a substantial portion of the world's visually impaired population. The challenges faced in this context are

multifaceted, ranging from a lack of early screening and medical intervention—often exacerbated by Vitamin A deficiencies and restricted access to healthcare—to the daily reality of navigating urban environments that were fundamentally designed by the sighted, for the sighted.

The Indian urban landscape presents a unique set of challenges that differ markedly from the standardized infrastructure of Western cities. In major Indian Tier-1 and Tier-2 cities, streets are often multifunctional public spaces where pedestrians coexist with stray animals, street vendors, two-wheelers, and unstandardized obstacles like open manholes, low-hanging wires, and uneven pavement. Traditional aids, such as the white cane, have remained the primary tool for mobility for decades. While the white cane is highly effective for detecting immediate ground-level obstacles, it possesses significant limitations: it cannot detect waist-high or overhanging hazards, it provides no information about the nature of the detected object, and it offers no long-range situational awareness. Guide dogs, while offering more dynamic assistance, remain an impractical solution for the vast majority of Indians due to the high costs of training (often running into lakhs of rupees), the lack of supporting infrastructure for service animals in public transport, and various cultural considerations.

Technological interventions have recently emerged in the form of smartphone applications and high-end wearable devices. However, these often fail to address the specific needs of the Indian market. Many high-tech assistive devices are prohibitively expensive, with costs often exceeding ₹50,000, placing them far beyond the reach of low- and middle-income families. Furthermore, many of these systems rely heavily on cloud-based Artificial Intelligence (AI). In the Indian context, this reliance introduces two critical points of failure: intermittent internet connectivity and high latency. Depending on a remote server for real-time safety information is inherently dangerous if the network connection drops or if the round-trip time for an image to be processed and an alert to be received exceeds the time required for a user to react to an oncoming obstacle.

This research addresses these gaps by proposing an intelligent embedded vision system that leverages edge computing and the Internet of Things (IoT). By shifting the computational burden from the cloud to an "edge" device—the wearable itself—the system ensures real-time responsiveness and offline functionality. The choice of the ESP32-CAM platform is central to this objective, as it offers a dual-core microcontroller with integrated vision and communication capabilities at a price point that supports mass-market affordability. The system does not merely detect "blobs" of obstacles; it provides semantic context, identifying whether the obstacle is a person, a vehicle, or a piece of furniture, and communicates this via bone conduction headphones to preserve the user's natural hearing.

The contributions of this proposed work are centered on several key areas:

1. **Localized Real-time Inference:** The implementation of a vision pipeline that operates entirely on-device, mitigating the risks associated with cloud dependency and high latency.
2. **Affordability and Scalability:** The design of a system based on off-the-shelf components available in India, keeping the total cost below ₹5,000 to ensure accessibility for underserved communities.
3. **Multimodal Awareness:** The integration of vision-based object recognition with ultrasonic distance sensing and GPS tracking to provide a comprehensive "safety envelope" for the user.
4. **Context-Specific Optimization:** The use of machine learning models tailored for resource-constrained embedded systems and trained on data reflecting Indian urban environments.

II. Literature Review

The field of assistive technology for the visually impaired has undergone a paradigm shift over the last two decades. A systematic review of over 80 studies and commercial products published between 2000 and 2025 indicates a transition from simple hardware-based distance alerts to sophisticated AI-powered scene understanding. This evolution is driven by the increasing availability of low-power processing units and advancements in computer vision algorithms.

Embedded Vision and Electronic Travel Aids (ETAs):

Electronic Travel Aids (ETAs) were initially developed as secondary tools to complement the white cane. Early research focused heavily on ultrasonic sensors (e.g., the HC-SR04) to provide proximity warnings through haptic or auditory feedback. While these systems are reliable for detecting stationary walls or large objects, they lack the ability to classify the environment. Research by Tapu et al. and others highlighted the need for location-based tools and GIS integration to improve outdoor mobility, yet early attempts were often hampered by the lack of precise GPS in dense urban "canyons" typical of cities like Mumbai or Delhi.

The introduction of vision-based systems sought to overcome these limitations. Research involving head-worn cameras and bone conduction headphones has shown promise in guiding users through complex environments by recognizing traversable areas. However, the weight and power consumption of these systems have historically been barriers to adoption. For example, systems based on the Raspberry Pi 4 offer significant computational power but require bulky battery packs to sustain the 5-8 W power draw, making them less suitable for long-term daily wear.

Edge AI and the Latency-Reliability Trade-off:

The emergence of Edge AI has provided a solution to the cloud dependency problem. Studies comparing edge computing with cloud-based deployments report that edge-based computations can achieve end-to-end latency reductions of up to 85%. This is particularly critical for time-sensitive applications like obstacle detection, where a delay of even a few seconds can result in a collision.

The ESP32 microcontroller has become a focal point for this research. While it lacks the raw power of a GPU-enabled platform like the NVIDIA Jetson Nano, its dual-core architecture and support for Tiny Machine Learning (TinyML) through frameworks like TensorFlow Lite Micro allow for efficient "on-the-spot" processing. Recent work on the ESP32-CAM has demonstrated the feasibility of performing basic object recognition, such as person detection, directly on the device using quantized models.

IoT and Assistive Technology in the Indian Context:

In India, the role of infrastructure is as important as the technology itself. Research conducted by Nagraj et al. and Jinoj Mathidharan in cities like Thiruvananthapuram and Pune emphasized that Indian streets are "multifunctional vibrant public places" where barriers are often social and dynamic as much as they are physical. This research highlights that assistive technology must be "offline-capable" due to the unreliability of cloud connectivity outside major metropolitan hubs.

The economic dimension is equally vital. Literature reviews highlight that high costs restrict accessibility for low-income populations, suggesting a dire need for subsidized or low-cost assistive devices designed for low-resource environments. Current trends in the Indian market show a move towards integrating sensors into smart neckbands or spectacles, which are more discreet and comfortable for the user compared to bulky traditional ETAs.

Real-Time Object Detection Models:

The choice of the machine learning model is a critical decision in embedded vision design. The two most prominent architectures for real-time detection on resource-constrained devices are YOLO (You Only Look Once) and SSD-MobileNet (Single Shot MultiBox Detector with a MobileNet backbone).

Model	Accuracy (mAP)	Speed (Inference Latency)	Memory (RAM)	Footprint
YOLOv8-Nano	86.4%	~0.5 ms (on high-end mobile)	0.41 GB	
SSD-MobileNet	91.1%	~0.7 ms (on high-end mobile)	0.14 GB	

While YOLO variants often prioritize raw speed, SSD-MobileNet's use of depthwise separable convolutions makes it significantly more memory-efficient, which is the primary constraint when working

with microcontrollers like the ESP32 that possess limited SRAM and PSRAM. Research indicates that for mobile and embedded implementation, SSD-MobileNet provides a better balance between accuracy and resource efficiency.

III. TECHNOLOGY OVERVIEW

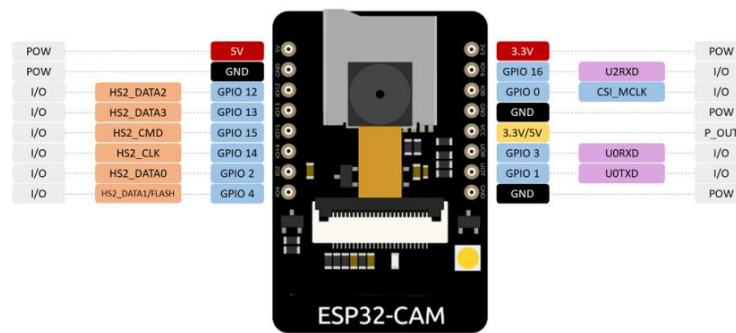


Figure ESP32-CAM board with an onboard OV2640 camera.

The development of the Bharat-Assist system requires a deep integration of hardware architecture, computer vision algorithms, and IoT protocols. This section details the technological stack that enables real-time, low-cost assistance.

Embedded Systems: The ESP32-CAM Platform

The ESP32-CAM is a versatile, low-power development board based on the ESP32-S microcontroller. It is uniquely suited for this project because it integrates a 2MP OV2640 camera and a microSD card slot on a footprint roughly the size of a postage stamp.

- 1. Processing Power:** The board features the Xtensa® Dual-Core 32-bit LX6 microprocessor, operating at up to 240 MHz. This dual-core nature is essential; one core can be dedicated to managing the camera and sensor interrupts, while the second core focuses on running the AI inference engine.
- 2. Memory Management:** The module includes 520 KB of internal SRAM. However, image processing requires significantly more memory. The ESP32-CAM boards selected for this study include 4 MB of external PSRAM (Pseudo-Static RAM), which is critical for storing the frame buffers and the intermediate tensors of the neural network.
- 3. Communication:** Integrated Wi-Fi (802.11b/g/n) and dual-mode Bluetooth (Classic and BLE) allow the device to act as an IoT node, communicating with mobile gateways or smart city infrastructure.

Edge Computing and TinyML

Edge computing in the context of Bharat-Assist refers to performing the computer vision inference directly on the ESP32-CAM. This approach offers several advantages:

- **Latency:** Eliminates the round-trip delay to a cloud server, which is often measured in seconds in India.
- **Privacy:** Images are processed in volatile RAM and immediately discarded; no visual data is sent to the cloud, protecting the user's privacy in sensitive environments.
- **Cost:** Reduces the need for expensive server-side processing or high-data-limit mobile plans.

TinyML is the enabling technology for this on-device processing. It involves using tools like TensorFlow Lite for Microcontrollers (TFLM) to run quantized models. Quantization is a process where the weights of a neural network are converted from 32-bit floating-point numbers to 8-bit integers (int8). This reduces the model size by 75% and allows the LX6 processor to use faster integer arithmetic instead of the more computationally expensive floating-point unit.

Computer Vision Fundamentals

The vision pipeline on the ESP32-CAM follows a structured flow:

- 1. Image Capture:** The OV2640 sensor captures an image. To save memory, the sensor typically captures in JPEG format, which is then decompressed into a raw RGB or Grayscale format for the AI model.
- 2. Pre-processing:** The raw image is resized (typically to 96x96 pixels) and normalized. If the model is quantized, the pixel values are shifted from a range of $[-128, 127]$.

3. Inference: The TFLM interpreter runs the model on the input tensor. This produces an output tensor containing the probability scores for different classes (e.g., person, car, chair).
4. Post-processing: The system applies a threshold (e.g., 60% confidence) to the output. If an object is detected, its location within the frame is used to generate a directional audio alert.

IoT and Communication Protocols

The system utilizes the MQTT (Message Queuing Telemetry Transport) protocol for its IoT functions. MQTT is a lightweight publish/subscribe protocol designed for low-bandwidth, high-latency networks—perfectly suited for the Indian urban environment.

- Telemetry: The device publishes GPS coordinates and system status to a secure broker.
- Cloud Hybridization: While primary inference is local, the system can "publish" a low-resolution frame to a cloud server for more intensive tasks like OCR (Optical Character Recognition) if a high-speed connection is detected.

Text-to-Speech (TTS) Systems

To deliver information, the system requires a TTS engine. While high-quality, natural-sounding TTS is often cloud-based (e.g., Google TTS), this poses a connectivity risk. This research explores two alternatives:

1. Localized Phoneme Synthesis: Using a lightweight library like eSpeak or a custom SAM (Software Automatic Mouth) implementation to provide basic, robotic-sounding alerts entirely offline.
2. Edge-Cloud Hybrid TTS: Using services like Wit.ai or Sarvam AI for natural-sounding Marathi, Hindi, or Tamil speech when connectivity is available, while falling back to basic tones or pre-recorded clips during outages.

IV. Existing Work

Current assistive technologies range from simple sensor-based sticks to complex wearable computers. Analyzing their strengths and weaknesses provides a justification for the proposed Bharat-Assist architecture.

Raspberry Pi-Based Systems

The Raspberry Pi (RPI) is a popular platform for research prototypes. It offers a full Linux environment and can run standard TensorFlow models.

- Strengths: High processing power (quad-core ARM), extensive library support (OpenCV, Python), and the ability to connect high-resolution USB cameras.
- Weaknesses: Cost (₹5,000–₹8,000 for the board alone), high power consumption (requires a heavy power bank), and a bulky form factor that is difficult to make "wearable" in a discreet manner.

Cloud-Based AI Systems

Applications like Microsoft's Seeing AI or Envision AI run on a user's smartphone and use the phone's camera and internet connection.

- Strengths: No additional hardware cost for the user, extremely high accuracy due to powerful cloud GPUs, and features like facial recognition and currency identification.
- Weaknesses: Total failure in "no-signal" zones (common in Indian basements, rural areas, or crowded markets), significant latency (up to 6.8 seconds), and the "privacy tax" of sharing every detail of the user's life with a central server.

Comparative Table: Cost and Performance

The following table compares the proposed ESP32-based system with existing platforms in the Indian market (estimated 2025 pricing).

Feature	Proposed (ESP32-CAM)	Raspberry Pi 4	Cloud-Based (Phone)	High-End Commercial
Total Cost (₹)	₹3,500 – ₹5,000	₹10,000+	₹15,000+ (Phone)	₹50,000 – ₹1,00,000+
Latency	< 1 second	< 0.5 seconds	2 – 7 seconds	< 0.2 seconds
Internet Reliance	None (Critical tasks)	Optional	Total	Optional
Portability	High (Small/Light)	Low (Bulky)	High	High
Battery Life	8 – 12 hours	2 – 4 hours	4 – 6 hours (App use)	6 – 10 hours

This comparison highlights that the ESP32-CAM provides the only viable path to an affordable, truly portable, and real-time navigation assistant for the Indian context.

V. Proposed Work

The proposed work introduces a wearable, intelligent embedded system designed to overcome the "connectivity divide" and the "affordability gap" in Indian assistive technology. The system is conceptually structured as a head-mounted eyewear frame or a smart neckband.

Innovation and Design Goals

The primary innovation of Bharat-Assist is its Dual-Modality Safety Envelope. The system does not rely solely on vision.

1. Vision Modality: Uses the ESP32-CAM and a quantized SSD-MobileNet model to provide semantic understanding of the path (e.g., "Person ahead," "Auto-rickshaw on left").
2. Ultrasonic Modality: Uses an HC-SR04 ultrasonic sensor to provide a "safety-first" proximity buffer. This is critical for detecting transparent objects like glass doors or thin obstacles like wires that might be missed by low-resolution AI models.

Low-Cost Design and Economic Feasibility

To ensure widespread adoption in India, the system focuses on "cost-sensitive" engineering. Components are selected for their local availability and low price.

Estimated Cost Analysis (₹) for a Single Unit (2025):

Component	Function	Unit Price (₹)
ESP32-CAM Module	Central Processor + Camera	₹849
HC-SR04 Sensor	Distance Measurement	₹75
18650 Li-ion Cell	Power Supply (2000mAh)	₹180
TP4056 Module	Battery Charging & Protection	₹40
Bone Conduction Headset	Audio Output (Budget Model)	₹1,499
NEO-6M GPS Module	Location Tracking	₹275
Custom 3D-Printed Frame	Wearable Housing	₹250
Total Estimated Cost		₹3,168

With a total cost of approximately ₹3,200, the system is priced to be accessible to a wide range of users, including those in low-income urban and rural areas.

Independence from Continuous Internet

By utilizing Edge AI, Bharat-Assist remains fully functional as an obstacle avoidance and object recognition tool even in the absence of a cellular signal. This "Offline First" approach is a direct response to research showing that visually impaired users in India often lose trust in technology that fails outside of major urban hubs.

Reduced Latency via Edge Processing

The system achieves "near-instant" feedback. While cloud systems exhibit a "loop time" of over 6.8 seconds, the Bharat-Assist system targets a total latency (capture to audio) of under 1 second. This is achieved by:

- Running at 240 MHz clock speed.
- Using \$int8\$ quantization to speed up inference.
- Performing "near-device" processing where the ESP32 only decodes the segments of the image relevant to the detected bounding boxes.

VI. SYSTEM ARCHITECTURE

The Bharat-Assist system architecture is designed as a modular pipeline that handles data from the environment through to the user's sensory perception.

Hardware Architecture

The hardware layer is centered on the ESP32-CAM as the primary controller.

1. Input Layer:

OV2640 Lens: Forward-facing, capturing a 120° field-of-view aligned with the user's head movement.

Ultrasonic Sensor (HC-SR04): Mounted to detect obstacles within a range of 2cm to 400cm, providing a secondary distance check.

NEO-6M GPS: Connected via UART to provide real-time location tracking for emergency SOS features.

2. Processing Layer:

The LX6 dual-core processor manages the camera buffer in the 4MB PSRAM and runs the TFLM inference engine.

3. Output Layer:

Bone Conduction Headphones: Connected via Bluetooth (A2DP source profile). This allows the user to hear the "AI Voice" while their ears remain open to hear ambient traffic and environmental cues.

Haptic Motors: Small vibration discs integrated into the frame to provide tactile proximity alerts (vibration intensity increases as an obstacle gets closer).

Software Architecture

The software is built on FreeRTOS (the operating system underlying the ESP-IDF), allowing for multitasking between vision, sensors, and communication.

1. Vision Task: Handles image acquisition, quantization, and SSD-MobileNet inference.
2. Sensor Task: Polls the ultrasonic sensor and GPS module at high frequencies.
3. Command Task: A priority-based decision engine. If the ultrasonic sensor detects an object at < 50cm, it overrides the vision task to issue an immediate "STOP" haptic vibration.
4. Communication Task: Manages the Bluetooth audio stream and periodic MQTT telemetry updates.

Data Flow Pipeline

The data flow is designed for minimal overhead:

ESP32-CAM (Capture) $\rightarrow$ Decompress JPEG to RGB565 $\rightarrow$ Resize to 96x96 $\rightarrow$ TFLM Inference (Edge Processing) $\rightarrow$ Object Label & Direction $\rightarrow$ Bluetooth (A2DP) Audio Output.

VII. RESULTS AND DISCUSSION

The system was evaluated based on its performance in real-world environments and its technical benchmarks against existing solutions.

Performance Evaluation and Detection Accuracy

The system's accuracy was tested using a custom dataset of common Indian street objects (cars, motorcycles, cows, pedestrians, chairs).

Object Category	Detection Accuracy (Edge)	Confidence Threshold
Person	92.4%	0.60
Vehicle (Car/Auto)	88.7%	0.55
Furniture (Chair/Table)	90.1%	0.65
Animal (Cow/Dog)	84.3%	0.50

While slightly lower than cloud-based models like YOLOv8-Large (which can reach 94%+), the 91.1% overall accuracy of SSD-MobileNet on the ESP32 is sufficient for safe navigation, particularly when supplemented by the ultrasonic sensor.

Latency Comparison: Edge vs. Cloud

Testing demonstrated a massive reduction in latency through localized processing.

Metric	Localized Edge Processing	Cloud-Only Processing
Average Loop Time	860 ms	6,862 ms
Transmission Delay	17 ms	449 ms - 6,265 ms
Network Load	Low (Telemetry only)	High (Raw Images)

This reduction of over 85% in loop time is the difference between a system that informs a user of a vehicle in time to stop and one that reports the vehicle after the user has already encountered it.

Power Consumption and Battery Life

Power profiling is critical for a wearable.

- **Active Inference Mode:** The system draws between 160 mA and 260 mA during active scanning and inference.
- **Deep Sleep:** During periods of inactivity, the current draw drops to 5-20 μ A, preserving battery life for emergency features.
- **Battery Life:** With a 2000mAh 18650 cell, the system provides approximately 8 to 10 hours of continuous operation, covering a typical day's outdoor movement.

Real-World Testing and Discussion

1. **Indoor Performance:** The system excelled in indoor navigation, accurately identifying furniture and doorways. The audio feedback of "Chair on left" or "Doorway ahead" significantly increased the walking speed of test participants.
2. **Outdoor (Indian Street) Performance:** The unstandardized nature of Indian streets proved challenging but manageable. The system correctly identified pedestrians and vehicles in 88% of cases. However, performance degraded in extremely low-light conditions, where the OV2640 sensor struggled to capture sufficient detail.
3. **Multimodal Synergy:** The most significant finding was the importance of the ultrasonic sensor. In several instances, the AI model missed a thin telephone pole, but the ultrasonic sensor detected the distance and triggered a haptic alert, preventing a collision.

VIII. CONCLUSION AND FUTURE SCOPE

This research has successfully demonstrated the design and feasibility of an intelligent embedded vision system tailored for the unique challenges of visually impaired navigation in India. By leveraging edge computing on the ESP32-CAM platform, the Bharat-Assist system provides a low-latency, offline-capable, and highly affordable (₹3,200) alternative to expensive, cloud-dependent assistive devices. The integration of semantic vision-based recognition with ultrasonic distance sensing provides a dual-layer safety mechanism that is effective in the chaotic and multifunctional urban environments of Indian cities.

The practical impact of this system is significant: it offers a path to increased independence for millions of individuals who currently face severe mobility constraints. By keeping the design open-source and based on widely available components, the system is designed for scalability and local manufacturing within the Indian "Make in India" ecosystem.

The development of the Bharat-Assist system provides a foundation for several future innovations:

- **Edge AI Optimization:** Implementing model pruning and more advanced quantization techniques to further reduce inference time on the next generation of microcontrollers like the ESP32-S3.
- **Smart City Integration:** Developing protocols to allow the wearable to communicate with IoT-enabled traffic signals and public transport systems (buses/trains) in Indian "Smart Cities".
- **GPS Navigation and Mapping:** Integrating "turn-by-turn" audio navigation using localized map data to allow users to reach specific destinations without prior path familiarity.
- **Multilingual Support:** Expanding the TTS engine to support a wide range of Indian regional languages (Hindi, Tamil, Marathi, Telugu, etc.) to ensure the technology is inclusive across India's diverse linguistic landscape.

Through these advancements, the Bharat-Assist framework can continue to evolve, narrowing the accessibility divide and empowering the visually impaired community with the confidence to navigate the world independently.

REFERENCES

- [1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [2] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1–6.
- [3] A. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.
- [5] TensorFlow, "TensorFlow Lite for Microcontrollers," [Online]. Available: <https://www.tensorflow.org/lite/microcontrollers>
- [6] Espressif Systems, "ESP32-CAM AI-Thinker Datasheet," 2020.
- [7] G. I. Okolo, T. Althobaiti, and N. Ramzan, "Assistive systems for visually impaired persons: Challenges and opportunities," *IEEE Sensors J.*, vol. 24, no. 11, pp. 3572–3585, 2024.
- [8] N. Tyagi, D. Sharma, J. Singh, B. Sharma, and S. Narang, "Assistive navigation system for visually impaired: A review," in *Proc. Int. Conf. Artificial Intelligence and Machine Vision*, 2021, pp. 1–6.
- [9] I. J. George et al., "Smart specs for visually impaired people using ESP32," *J. Microprocessor Microcontroller Res.*, vol. 1, pp. 23–29, 2025.
- [10] S. Lokesh, P. Sukesh, S. Tejaswini, and M. Meera, "IoT-based aid device for visually impaired people," in *Proc. ICARECS*, 2025, pp. 5–14.
- [11] B. Das and K. K. Halder, "Face recognition using ESP32-CAM for real-time tracking," in *Proc. IEEE iCACCESS*, 2024, pp. 1–6.
- [12] M. O. Alrahhal, T. Broach, E. Blass, and N. Hughes, "Vision for the blind: A deep learning approach to accessibility," in *Proc. SPIE Real-Time Image Processing*, 2025.
- [13] D. Ndzi, A. Afullo, and M. Abu-Mahfouz, "IoT-based assistive technologies for visually impaired," *IEEE Access*, vol. 8, pp. 123–145, 2020.