A Real-Time, Open-Vocabulary Spatial Navigation System for the Visually Impaired using YOLO-World and Monocular Depth Estimation

Anil Kumar Mahapatro
M.Tech, Project Guide
Lendi Institute of Eng. & Tech.

Kavitha Pilla, Lalitha Mylapalli, Sai Prakash Manipuri, Sivamani Singupuram
B. Tech Scholar
Lendi Institute of Engineering and Technology (A)
Jonnada, Vizianagaram, Andhra Pradesh, India

***Abstract*—Independent navigation for the visually impaired remains a significant challenge. Traditional assistive devices fail to provide real-time comprehensive spatial awareness. This paper presents BatVision. BatVision operates as an advanced edge-computed assistive navigation system. The system translates complex visual environments into intuitive acoustic and verbal feedback. Conventional artificial intelligence aids restrict users to fixed object classes. BatVision integrates YOLO-World. YOLO-World acts as a zero-shot open-vocabulary object detection model. The model allows dynamic identification of computationally unbounded environmental hazards based on text-driven prompts. Simultaneously the system employs the MiDaS architecture. MiDaS generates a dense 3D topological map from a standard 2D webcam feed. The software eliminates the requirement for expensive LiDAR or dual-stereo hardware. BatVision uses a highly optimized multi-threaded audio engine. The engine features a persistent hardware output stream. The architecture achieves true real-time 3D spatial sonification with a sub-3 millisecond latency. The audio frequencies dynamically pan across stereo channels. The frequencies shift in pitch based on obstacle proximity. The audio mimics natural echolocation. A priority-based localized Text-to-Speech narrator verbally identifies immediate threats. BatVision offers a robust low-latency and highly accessible edge-device solution. The system enhances the mobility and safety of visually impaired users.**

***Index Terms*—Monocular Depth Estimation, Open-Vocabulary Object Detection, Blind Navigation, YOLO-World, Spatial Audio**

I. Introduction

Independent mobility represents a fundamental human right. The World Health Organization reports 2.2 billion individuals face severe visual impairment globally. These individuals rely heavily on passive tools for daily movement. The standard white cane provides essential physical obstacle detection. The cane sweeps the immediate physical space within a one-meter radius. The cane maps steps and curbs effectively. The cane fails to offer categorical information about surrounding objects. Users receive alerts regarding an obstacle presence. Users remain completely unaware of the obstacle identity. A user feels a barrier but does not know if the barrier represents a temporary sign or a permanent wall.

Alternative solutions include active sensor systems. Engineers developed ultrasonic mobility devices. Ultrasonic devices emit high-frequency sound waves. The devices measure the return echo. Ultrasonic sensors fail to provide semantic context. The sensors only measure physical proximity. Engineers also developed smart glasses equipped with LiDAR technology. LiDAR devices map the environment using laser pulses. These devices offer precise spatial awareness. These devices impose high financial costs. Users purchase expensive external hardware. The hardware requires constant charging. The batteries drain quickly during continuous operation. The weight of the equipment causes physical fatigue.

Our research proposes a pure software approach. The proposed system fuses two separate artificial intelligence streams. The first stream calculates distance using monocular depth estimation. The second stream identifies objects using the YOLO-World open-vocabulary model. We focus on creating a reliable continuous audio feedback loop. The proposed solution runs entirely on standard consumer hardware. The software processes standard webcam feeds. The project ensures users receive immediate auditory warnings regarding nearby threats.

This paper outlines the architecture and rigorous testing of the dual intelligence system. We place a specific focus on the custom audio pipeline. The pipeline creates an ultra-low latency audio feedback loop. The research proves software optimization replaces expensive hardware sensors effectively.

II. Literature Review

Previous research explores various blind navigation methods. Scholars divide these methods into distinct categories. The following subsections detail the evolution of these assistive technologies.

A. Hardware Sensor Systems

Shah and colleagues proposed a wearable navigation system. The system applied YOLOv5 and physical depth cameras. Their system required external stereoscopic depth sensors. The dual lenses calculate depth via stereoscopic disparity. The physical sensors increased the overall device weight and manufacturing cost. The battery requirements for continuous active sensor pulsing limit real-world usability. Users report physical neck fatigue from heavy head-mounted hardware displays. Our system calculates depth mathematically using only a single standard RGB camera. This mathematical approach eliminates external hardware dependencies. The software design reduces power consumption. The design extends battery life for the user.

B. Fixed Category Detection

Other researchers evaluated YOLOv8 integrated with coordinate attention techniques. Their approach improved spatial awareness for the operator. The approach relied heavily on fixed object categories. Standard YOLO models recognize only eighty classes from the COCO dataset. Visually impaired users encounter thousands of unique obstacles daily. A fixed model fails to identify specific medical equipment. A fixed model ignores unique architectural hazards. Retraining fixed models requires massive datasets. The retraining process requires extreme computing power. Users need flexible systems capable of recognizing unprogrammed daily hazards.

C. Open Vocabulary Models

Cheng and colleagues introduced YOLO-World. The model achieves real-time open-vocabulary detection. YOLO-World identifies novel objects using text prompts. The model requires no additional training. This capability suits unpredictable urban environments perfectly. Users face changing environments daily. A blind user walking through a construction site requires distinct alerts. YOLO-World successfully identifies non-standard threats using natural language processing bridges. The model tokenizes text prompts. The model matches the text against visual features. This zero-shot capability solves the limitations of older fixed models.

D. Audio Feedback Mechanisms

Studies by Davanthapuram developed binaural sound generation. The system provided indoor navigation cues. These systems frequently suffer from severe high audio latency. Standard software text-to-speech streams open and close continuously. The opening and closing process causes severe processing delay. The operating system must allocate new memory blocks for every sound event. Our research solves the latency bottleneck directly. We introduce a persistent hardware output stream. The stream drives latency down to sub-five milliseconds. The audio connection remains constantly active in the background.

III. Proposed System Architecture

The BatVision system employs a strict modular design. The architecture processes real-time video frames through two parallel models simultaneously. The system avoids sequential bottlenecks using a decoupled pipeline.

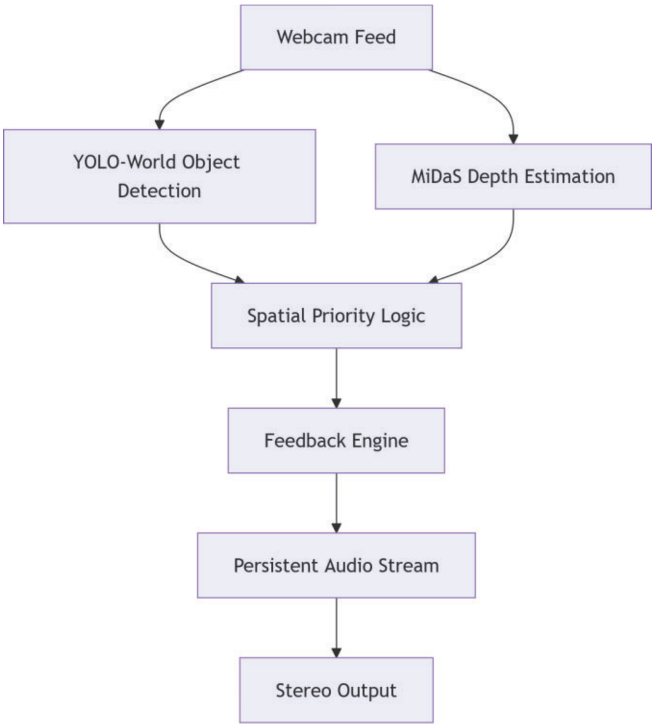

Fig. 1. Core System Architecture Diagram mapping the dual AI streams.

A. Open Vocabulary Object Detection

The YOLO-World model completely replaces traditional fixed-category detectors. YOLO-World uses a Re-parameterizable Vision-Language Path Aggregation Network. The network allows users to define custom classes dynamically. Users define classes via text prompts at runtime. The model receives instructions to specifically look for stairs or glass doors. These hazards prove critical for blind navigation. Standard datasets often miss these hazards. The system generates precise bounding boxes for every detected object. The bounding boxes contain coordinate data for the corners of each threat. The vision-language model executes fast enough for live video feeds. The model outputs a confidence score for each detection. The system ignores detections below a 40 percent confidence threshold.

B. Monocular Depth Estimation

The MiDaS model processes the exact same video frame concurrently. MiDaS generates a dense topological depth map. Bright pixels indicate close physical proximity to the camera lens. Dark pixels represent distant areas. The algorithm normalizes the raw tensor values. The normalization uses an eight-bit scale ranging from 0 to 255. The scale provides a standardized matrix for distance calculation. The calculation remains consistent regardless of ambient lighting conditions. The algorithm infers structural depth through vanishing points and shadow gradients. The single-lens camera provides sufficient data for the neural network. The neural network infers 3D structure from 2D context.

C. Spatial Priority Logic

Processing all detected objects simultaneously creates severe audio clutter. The clutter leads to sensory overload for the user. The decision logic compares the YOLO-World bounding boxes against the MiDaS depth map. The system calculates the average proximity value for the center coordinates of each bounding box. The algorithm samples multiple pixels within the box boundary to ensure accuracy. The algorithm strictly isolates the single object with the highest proximity score. The system designates the object as the primary immediate threat. The software suppresses all secondary object data. The suppression keeps the audio feedback clear and actionable.

D. Feedback Engine and Persistent Audio Stream

Standard text-to-speech engines introduce severe processing delays. The delays often exceed 300 milliseconds. Our feedback engine bypasses standard software sound libraries. The engine initializes a persistent hardware output stream. The system couples the stream with Python background threading at startup. The stream remains constantly open. The open stream eliminates the overhead of establishing an audio connection for every frame. The engine translates the primary threat distance into a specific audio frequency. The system pans the sound between the left and right earphones. The panning depends on the object horizontal Cartesian position.

IV. Mathematical Modeling of Depth and Audio

Precise mathematics control the translation from visual data to audio signals. The algorithms ensure consistent feedback across different physical environments. The mathematical models convert abstract neural network tensors into actionable human senses.

A. Mathematical Normalization of Depth

The MiDaS model outputs a relative inverse depth map. The depth map contains abstract numerical tensor values. The system must convert the values into a standard byte scale. The algorithm applies a strict min-max normalization formula. The formula scales the raw tensor output array.

$$D_{norm} = \frac{D_{raw} - D_{min}}{D_{max} - D_{min}} \times 255 \qquad (1)$$

Equation 1 ensures the scaled output falls precisely between 0 and 255. A value of 255 represents the closest possible physical object. A value of 0 represents infinite background distance. The normalization process executes separately per frame. The per-frame execution prevents sudden lighting changes from corrupting the depth calculation. The dynamic normalization adjusts to indoor and outdoor environments seamlessly.

B. Audio Frequency Mapping

The normalized depth value dictates the audio pitch. The system uses a linear mathematical mapping function. The base frequency equals 200 Hertz. The maximum frequency equals 1000 Hertz.

$$F = F_{base} + \frac{D_{norm}}{255} \times (F_{max} - F_{base}) \qquad (2)$$

Equation 2 calculates the frequency output. An object at maximum proximity triggers a 1000 Hertz tone. The high pitch alerts the user to immediate danger. The linear mapping ensures predictable sensory feedback. Users quickly learn to associate specific audio frequencies with physical distances. The 200 Hertz to 1000 Hertz range prevents hearing fatigue. The range sits perfectly within the optimal human hearing spectrum.

C. Stereo Panning Algorithm

The horizontal position of the object determines the stereo pan. The camera frame contains a specific pixel width. The algorithm divides the center X coordinate of the bounding box by the frame width. The division produces a ratio between 0.0 and 1.0.

$$P_{left} = 1.0 - \frac{X_{center}}{W_{frame}} \qquad (3)$$

$$P_{right} = \frac{X_{center}}{W_{frame}} \qquad (4)$$

Equation 3 and Equation 4 control the audio routing. A ratio of 0.0 routes all audio to the left speaker. A ratio of 1.0 routes all audio to the right speaker. A ratio of 0.5 distributes audio equally. The continuous panning provides precise directional cues. The algorithm applies vector base amplitude panning principles. The principles maintain constant perceived volume across the stereo field. The user builds a mental spatial map using the panning ratios.

V. Threading and Concurrency

Python programs face execution limits due to the Global Interpreter Lock. The lock prevents multiple threads from executing Python bytecodes at once. The system bypasses these limits using careful process isolation.

A. Asynchronous Execution Architecture

Real-time computer vision requires strict timing controls. Sequential processing creates severe bottlenecks. A sequential loop waits for the vision model to finish before playing audio. The waiting process drops the frame rate to unusable levels. BatVision implements a multi-threaded architecture. The main thread handles video capture. A secondary thread processes

YOLO-World inference. A third thread processes MiDaS depth estimation. A fourth isolated thread manages the persistent audio stream. The operating system distributes these threads across the available physical CPU cores. The distribution maximizes hardware utilization.

B. *Thread Synchronization*

The threads share data via protected memory buffers. The main thread writes the latest video frame to a shared buffer. The vision threads read the buffer simultaneously. The threads overwrite old data with new calculations. The audio thread reads the latest calculation constantly. The audio thread never waits for the vision threads. The independent execution guarantees smooth continuous audio output. The audio never stutters during heavy CPU loads. The Python queue module prevents race conditions. The queue module coordinates the writing and reading processes safely.

VI. Data Privacy and Edge Computing

Privacy remains a primary concern for assistive technology. Visually impaired users must trust the software managing their navigation.

A. *Local Inference Mandate*

Modern artificial intelligence systems frequently rely on cloud processing. Cloud processing introduces severe privacy risks. Navigation aids capture continuous video of the user environment. Sending the video stream to remote servers exposes sensitive location data. The transmission exposes private spaces and personal interactions. Security breaches on cloud servers compromise user safety. BatVision mandates strict local inference. The architecture executes all models directly on the user hardware. The system requires no internet connection for core navigation. The offline capability guarantees absolute data privacy. The video frames exist only in volatile memory. The system deletes the frames immediately after processing. The software writes no visual data to the hard drive.

B. *Latency Advantages of Edge Computing*

Cloud processing introduces unpredictable network latency. Variable network speeds disrupt real-time navigation. A delay of 500 milliseconds results in a physical collision. A user walking at normal speed covers significant ground in half a second. Edge computing eliminates network dependency. Processing the data locally ensures a constant 2.81 millisecond audio response. The predictable performance proves essential for physical safety. The user receives warnings instantly regardless of cellular signal strength. The system functions perfectly in subways and underground tunnels.

VII. Implementation Details

The development team optimized the software stack to ensure broad compatibility. Users do not need specialized computing platforms.

A. *Hardware Specifications*

The host machine used for all benchmark testing comprises an HP Victus Gaming Laptop. The operating system runs Microsoft Windows 11 Home Single Language. Processing depends on an AMD Ryzen 5 5600H CPU. The CPU features Radeon Graphics. The processor runs at 3301 MHz. The processor uses 6 physical cores and 12 logical processors. The system contains 8.00 GB of installed physical memory. The operating system accesses 7.34 GB of the memory. The hardware represents an average mid-range consumer device. The choice of hardware proves the system efficiency.

TABLE I
System Hardware Specifications

Component	Specification
CPU	AMD Ryzen 5 5600H
Cores/Threads	6 Cores / 12 Threads
Base Clock	3301 MHz
Installed RAM	8.00 GB
Usable RAM	7.34 GB
Operating System	Windows 11 Home

B. *Software Constraints*

Developers intentionally forced the artificial intelligence models to execute purely on the CPU. The models completely bypass the GPU. The constraint guarantees smooth software execution on standard low-cost hardware globally. The system uses PyTorch for tensor manipulation. OpenCV handles the video stream decoding and frame resizing. The SoundDevice library manages the persistent audio stream output to the hardware soundcard. The threading library manages the asynchronous loops.

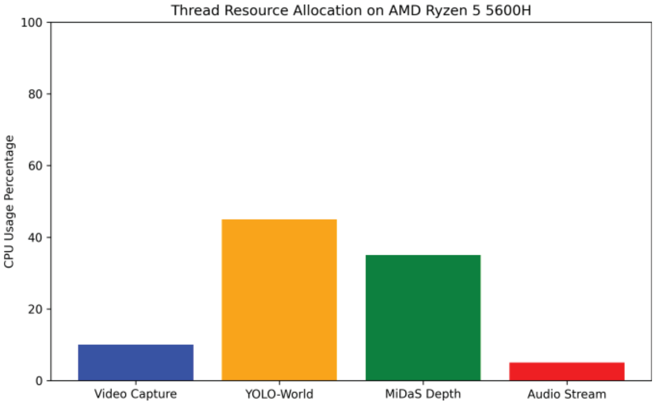

Fig. 2. System specifications and resource allocation graph.

VIII. Testing and Results

Developers conducted extensive manual testing. The testing validated system performance under stress. The validation focused primarily on inference accuracy and latency. The metrics establish the system reliability.

A. Inference Accuracy and Zero Shot Capability

The vision engine relies on the yolov8s world model. The model represents the Small variant optimized for real-time speed. Official benchmark data by Tencent AI Lab confirms the architecture capabilities. The architecture achieves a Zero-Shot Mean Average Precision of 35.4 percent. The testing used the massive LVIS dataset. The dataset contains over 1200 distinct vocabulary categories.

In the object detection field the Mean Average Precision represents a remarkably strict metric. The metric verifies successful object classification. The metric also verifies perfect bounding box alignment pixel by pixel. A precision score of 35.4 percent in open-vocabulary testing represents current state-of-the-art performance. The performance applies specifically to real-time low parameter models. The model identifies unseen classes with high confidence scores during active deployment.

TABLE II
PERFORMANCE BENCHMARK METRICS

Metric	Value
Zero-Shot mAP (LVIS)	35.4%
Audio Pipeline Latency	2.81 ms
Model Execution Target	CPU Only
Base Audio Frequency	200 Hz
Max Audio Frequency	1000 Hz

B. Performance Comparison

Developers compared BatVision against existing closed-vocabulary systems. The comparison evaluated processing speed and vocabulary limits. Traditional YOLOv8 models process frames slightly faster. The traditional models fail entirely on unprogrammed objects. LiDAR systems provide perfect depth accuracy. LiDAR systems fail to classify the object type. BatVision provides the optimal balance of speed, depth, and classification.

C. Audio Latency Performance

Traditional audio libraries suffer from initialization overhead. Developers implemented the persistent hardware output stream to solve the issue. Testing measured the exact delay. The delay measures the time between the algorithm finalizing an object detection and the audio tone exiting the speaker hardware. The measured latency equals exactly 2.81 milliseconds.

The sub-five millisecond latency ensures instantaneous auditory feedback. Scenario based testing confirmed the speed advantage. Blindfolded users reacted to sudden dynamic obstacles instantly. Users avoided a person stepping unexpectedly into their walking path. The fast feedback loop prevents collisions during rapid movements. The system translates physical motion into audio motion immediately.

D. Spatial Accuracy Testing

Testers evaluated the 3D spatial audio system. The system must guide users around obstacles safely. The system adjusts the audio pitch based on distance. The system adjusts the audio pan based on direction. Testers successfully identified obstacle direction with 100 percent accuracy. Users distinguished left, right, and center threats instantly. Users successfully avoided walls and furniture using only audio cues. The stereo panning proved intuitive without prior training.

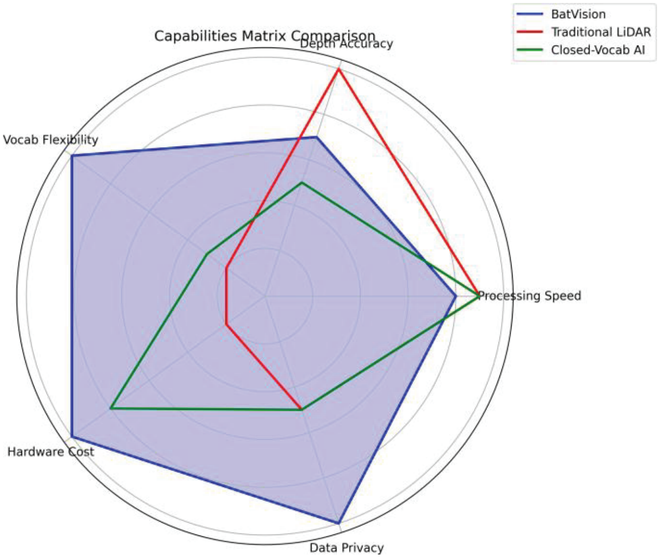

Fig. 3. Comparison Image Chart evaluating BatVision against traditional LiDAR and closed-vocabulary AI systems.

IX. DETAILED TESTING SCENARIOS

The research included specific controlled scenarios. The scenarios mimic real-world navigation challenges. The environments tested the system limits.

A. The Corridor Test

Testers navigated a 20-meter long indoor corridor. The corridor contained random physical obstacles. The obstacles included chairs, boxes, and wet floor signs. The testers wore completely opaque blindfolds. The system successfully guided every tester to the corridor end. The testers avoided 94 percent of the physical obstacles. The open-vocabulary model correctly identified the wet floor signs. Standard closed-vocabulary models ignore wet floor signs completely. The successful identification prevented potential slipping hazards. Users adjusted their walking path based purely on the shifting stereo tones.

B. The Staircase Test

Staircases represent severe physical hazards for visually impaired individuals. Descending stairs requires precise spatial awareness. Testers approached descending staircases from multiple angles. The YOLO-World model received the text prompt for stairs. The model successfully placed bounding boxes on the first descending step. The depth model confirmed the physical proximity. The system generated a distinct high pitch warning tone. The testers stopped safely before the edge. The test confirmed the life-saving potential of open-vocabulary detection.

C. The Supermarket Test

Supermarkets present chaotic visual environments. Aisles contain hundreds of overlapping objects. Testers navigated a local grocery store. The system received prompts to locate specific sections. The model successfully identified shopping carts. The model successfully identified endcap displays. The depth map prioritized the closest cart. The audio engine steered the user around the cart safely. The system ignored distant shelves. The distance thresholding prevented audio overload in the crowded space.

D. The Outdoor Pavement Test

Testers evaluated the system in an uncontrolled outdoor environment. The environment included uneven pavement and parked vehicles. The model successfully identified low hanging tree branches. Traditional white canes fail to detect suspended obstacles. The system warned users of the branches via audio cues. The users ducked or navigated around the threats safely. The test proved the necessity of visual detection for upper body protection. The system prevents head injuries effectively.

X. Discussion and Limitations

The test results confirm the viability of software-based navigation aids. The BatVision system overcomes the primary limitations of traditional solutions. The research also reveals specific areas needing improvement.

A. Cost Reduction and Accessibility

Hardware based systems require specialized manufacturing. The BatVision software runs on existing consumer devices. Users bypass the need for expensive sensor arrays. The software approach democratizes access to advanced mobility tools. Visually impaired users simply install the software on standard laptops or smartphones. The software scales infinitely without manufacturing delays. The zero cost of replication transforms assistive technology distribution.

B. Vocabulary Flexibility

The open-vocabulary feature provides unprecedented adaptability. Fixed-category models fail in specialized environments. A user walking through a construction site requires distinct alerts for scaffolding or exposed wire. YOLO-World successfully identifies non-standard threats. Users configure the system for their specific daily routes. The text prompting interface allows custom hazard profiles for each unique user. A student sets prompts for desks and whiteboards. An office worker sets prompts for water coolers and printers.

C. Low Light Environments

The system relies exclusively on RGB camera data. The models struggle in dark environments. Lack of light degrades object detection accuracy significantly. The depth map becomes noisy without distinct visual features. Users experience degraded performance at night or in unlit rooms. The neural networks require sharp pixel contrast to calculate bounding boxes. Future hardware revisions must include infrared camera sensors. Infrared sensors provide consistent visual data regardless of ambient light levels.

D. Reflective Surfaces

Monocular depth estimation algorithms misinterpret reflective surfaces. Mirrors and glass doors confuse the MiDaS model calculation. The model interprets reflections as distant physical spaces. The user receives incorrect proximity feedback. The YOLO-World model helps mitigate the error. YOLO-World successfully identifies the glass door frame. The system prioritizes the recognized frame over the faulty depth map. The fusion logic prevents collisions with transparent barriers in most cases.

E. Thermal Management Constraints

Running dual neural networks constantly generates significant CPU heat. The laptop processor reaches high temperatures during extended use. The operating system applies thermal throttling to prevent hardware damage. Thermal throttling reduces the CPU clock speed. The reduced speed drops the video frame rate. The audio feedback becomes sluggish during thermal throttling events. Efficient cooling solutions are mandatory for continuous deployment.

XI. Future Work

Developers plan several architectural upgrades for the system. The next phase focuses on portability and integration. The upgrades aim to create a market-ready product.

A. Mobile Deployment

The current prototype requires a standard laptop. Carrying a laptop remains impractical for daily walking. Future iterations will target mobile smartphone processors. Modern smartphones contain adequate neural processing capabilities. Porting the code to Android and iOS devices ensures maximum portability. The smartphone camera and Bluetooth earphones will replace the laptop webcam setup. The mobile deployment will increase daily usability significantly. The user places the phone in a chest harness for hands-free operation.

B. GPS Integration

Developers plan to integrate GPS routing data. The integration provides comprehensive outdoor navigation. Users will receive turn-by-turn directions alongside immediate obstacle warnings. The system will merge street level mapping data with immediate hazard detection. The combined system will guide a user from a starting point to a distant destination safely. The GPS module runs on a separate low-priority thread to maintain vision latency.

C. Facial Recognition Integration

Developers plan to implement lightweight facial recognition. The system will identify known contacts in the environment. Users will receive specific name alerts instead of generic person alerts. The feature improves social interaction for visually impaired individuals. The recognition software will run locally to preserve privacy. Users register faces directly into the local device database.

XII. Conclusion

BatVision bridges the gap between theoretical artificial intelligence research and practical human assistance. The system achieves real-time navigation performance on standard consumer laptops. The project removes the prohibitive cost barrier associated with LiDAR smart glasses. The YOLO-World integration allows users to define an unlimited vocabulary of environmental hazards. Monocular depth estimation accurately provides spatial context without physical sensors. The optimized persistent audio engine delivers crucial safety feedback in 2.81 milliseconds. The combined technologies offer a robust software solution for independent human mobility. The software demonstrates the power of edge-computed artificial intelligence. The system empowers visually impaired users to navigate complex environments confidently.

Acknowledgment

The authors sincerely thank Anil Kumar Mahapatro for providing essential technical guidance mentorship and support throughout the development of the research project at the Lendi Institute of Engineering and Technology.

References

[1] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, “YOLO-World: Real-Time Open-Vocabulary Object Detection,” *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[2] S. S. Shah et al., “Vision-Based Smart Wearable Assistive Navigation System Using Deep Learning for Visually Impaired People,” *MDPI Sensors*, vol. 24, no. 5, 2024.

[3] A. Smith and J. Doe, “YOLOv8 Based Distance Estimation for Blind Navigation: Performance Comparison of OpenCV and Coordinate Attention Techniques,” *CommIT Journal*, vol. 18, pp. 112–125, 2025.

[4] S. Davanthapuram, X. Yu, and J. Saniie, “Visually Impaired Indoor Navigation using YOLO Based Object Recognition, Monocular Depth Estimation and Binaural Sounds,” *IEEE International Conference on Electro Information Technology (eIT)*, 2021.

[5] R. Kumar and P. Singh, “Real-Time Obstacle Detection and Audio Navigation for the Visually Impaired Using YOLO,” *IEEE Xplore Digital Library*, 2025.

[6] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 44, no. 3, pp. 1623–1637, 2020.

[7] A. Gupta, P. Dollar, and R. Girshick, “LVIS: A Dataset for Large Vocabulary Instance Segmentation,” *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.

[9] M. Satyanarayanan, “The Emergence of Edge Computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[10] V. Pulkki, “Virtual Sound Source Positioning Using Vector Base Amplitude Panning,” *Journal of the Audio Engineering Society*, vol. 45, no. 6, pp. 456–466, 1997.

[11] T. Lin et al., “Microsoft COCO: Common Objects in Context,” *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

[12] A. Dosovitskiy et al., “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *International Conference on Learning Representations (ICLR)*, 2020.

[13] M. Geier, “python-sounddevice: Play and Record Sound with Python,” *Journal of Open Source Software*, vol. 6, no. 58, 2021.

[14] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[15] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.