American Journal of Computer Science and Technology
2025, Vol. 8, No. 4, pp. 189-205
https://doi.org/10.11648/j.ajcst.20250804.13

Research Article

An Integrated Jacket–Helmet Assistive System for Visually Impaired Individuals Using YOLO-Based Object Detection, Depth Estimation, and OCR

Kashvi Ruparelia[1], Priyam Parikh[2, *], Parth Atulkumar Shah[2]

[1]Higher Secondary Department, Anand Niketan Satellite, Ahmedabad, India
[2]Product Design Department, Anant National University, Ahmedabad, India

Abstract

This paper presents the design and evaluation of a jacket–helmet assistive system for visually impaired individuals in India. The system integrates a Raspberry Pi 4B with a USB web camera, USB microphone, vibration motor cluster, earphone, pushbuttons, and a rechargeable 7.4 V, 10,000 mAh battery. Two primary functions are implemented: (i) object detection and distance estimation using YOLO algorithms with 2D depth estimation, and (ii) text recognition on posters and hoardings using optical character recognition (OCR). Comparative analysis of YOLOv5, YOLOv7, and YOLOv8 models demonstrated that YOLOv8 achieved the highest mean Average Precision (mAP) of 92.4%, outperforming YOLOv7 (89.6%) and YOLOv5 (87.3%). For monocular 2D depth estimation, MiDaS achieved the lowest mean absolute relative error (0.124) compared to Monodepth2 (0.156) and DPT (0.139). Speech-to-text efficiency was tested across Google Speech Recognition, Vosk, and CMU Sphinx, with Google achieving 94.1% accuracy, followed by Vosk (88.3%) and CMU Sphinx (81.6%). User trials were conducted with ten visually impaired individuals across diverse environments (bus stand, garden, bungalow, and home settings). System usability was measured using the System Usability Scale (SUS), yielding an overall average score of 84.6, indicating "excellent" usability. The proposed system demonstrates high accuracy, robustness, and practicality for real-world navigation and reading assistance, thus contributing to improved autonomy and quality of life for visually impaired users.

Keywords

Assistive Technology, YOLO Object Detection, Depth Estimation, Speech-to-Text, OCR, Raspberry Pi, Visually Impaired, System Usability Scale (SUS)

1. Introduction

Visually impaired individuals face continuous challenges in perceiving and navigating their environments, particularly in unstructured and dynamic outdoor settings. In recent years, advances in computer vision and artificial intelligence (AI) have enabled the development of wearable assistive systems that combine object detection, depth estimation, and speech interaction to provide real-time guidance [16]. However, existing solutions often suffer from hardware limitations, poor accuracy under adverse conditions, or a lack of usability validation in real-world scenarios. Object detection remains a

*Corresponding author: priyam.parikh@anu.edu.in (Priyam Parikh)

Received: 12 September 2025; **Accepted:** 23 September 2025; **Published:** 30 October 2025

cornerstone of navigation assistance. Traditional approaches based on feature descriptors are less effective in complex environments, whereas deep learning–based methods such as the You Only Look Once (YOLO) family have demonstrated high accuracy and speed. Wang et al. proposed YOLO-OD, an enhanced version of YOLO with Feature Weighting Block (FWB), Adaptive Bottleneck Block (ABB), and Enhanced Feature Attention Head (EFAH), achieving improved detection of occluded and small objects in outdoor navigation scenarios [1]. More recently, YOLO-Extreme was introduced, building on YOLOv12 with a Dual-Branch Bottleneck Block and Multi-Dimensional Collaborative Attention, enabling robust performance under foggy and low-visibility conditions [2]. These developments highlight the importance of tailoring YOLO variants for assistive technologies. Depth estimation using a monocular camera is equally important for providing visually impaired users with distance awareness. Recent methods have explored transformer-based and Laplacian residual networks. Song et al. proposed the Detail-Semantic Collaborative Network (DSCNet), which effectively fuses detailed and semantic features, yielding superior results on NYU Depth V2 and SUN RGB-D datasets [3]. Similarly, Xi et al. developed LapUNet, a dynamic Laplacian residual U-shaped network that preserves high-frequency details and object boundaries, improving accuracy over existing models [4]. Abdusalomov et al. further advanced the field with a Dynamic Iterative Monocular Depth Estimation (DI-MDE) framework, integrating elastic depth bins and self-supervised training to mitigate scale ambiguity in dynamic scenes [5]. These models demonstrate the trade-off between accuracy and computational feasibility when deploying depth estimation on low-power platforms such as Raspberry Pi. Alongside perception modules, speech interaction and text recognition form critical components of assistive systems. Paramarthalingam et al. developed a deep learning framework that detects potholes and hazards using YOLO while delivering auditory and haptic feedback to the user [6]. Okolo et al. proposed a smart assistive navigation system combining object detection with voice feedback, reporting significant improvement in user mobility [7]. While these works focus on individual functionalities, integrating speech recognition with optical character recognition (OCR) can provide visually impaired individuals with the ability to read posters, hoardings, and signage in public spaces. Despite significant progress, most prior works have not been validated extensively with visually impaired individuals in diverse real-world environments. Usability studies, particularly using standardised evaluation metrics such as the System Usability Scale (SUS), remain limited. In this work, we propose a jacket–helmet integrated system incorporating YOLO-based object detection, monocular depth estimation, OCR, and multimodal feedback (voice, vibration, and buzzer). We benchmark YOLOv5, YOLOv7, and YOLOv8 for object detection, MiDaS, Monodepth2, and DPT for depth estimation, and multiple speech recognition frameworks for command interpretation. Finally, we validate the system through trials with ten visually impaired individuals across environments such as bus stands, gardens, and homes, and quantify usability using the SUS. The results demonstrate high accuracy and positive user acceptance, highlighting the potential of the system as a low-cost, practical solution for visually impaired navigation in India.

2. Literature Review and Research Gap

One recent work, *Adaptive Object Detection for Indoor Navigation Assistance: A Performance Evaluation of Real-Time Algorithms* [8] evaluates YOLO, SSD, Faster R-CNN and Mask R-CNN in indoor settings. The paper examines trade-offs between detection accuracy and inference speed, finding that lighter YOLO/SSD variants offer faster processing (< 50 ms) though at some loss in localization accuracy compared to two-stage detectors. This is relevant to your system's choice of object detection models and indicates possible baselines for inference speed vs accuracy. Another work, *An effective obstacle detection system using deep learning* by Atitallah et al. [9] addresses both indoor and outdoor environments. It uses deep learning models for robust detection of obstacles including in cluttered outdoor scenes. Performance drops in low lighting or occlusion were observed, signalling the need for strong performance under adverse conditions, exactly the kind one must address in your comparative evaluation. *MagicEye: An Intelligent Wearable Towards Independent Living of Visually Impaired* [10] presents a wearable device combining a custom CNN-based object detection (35 classes) with facial recognition, currency identification, GPS navigation, and proximity sensors for obstacle detection. While the range of functionalities is high, the paper notes that accuracy varies across contexts, particularly between controlled indoor vs complex outdoor scenes, and that latency is a concern, especially when deploying on embedded or wearable hardware. That gives you a precedent: when you test YOLOv5, v7, v8, etc., you will need to measure latency as well as accuracy under multiple conditions. *Multifunctional Assistive Smart Glasses for Visually Impaired* [11] integrates object detection, OCR, text-to-speech, and translation in a smart glasses prototype using Raspberry Pi. The system uses a button to toggle between modes. Important for your system: this work demonstrates how users expect mode toggling, how audio feedback must be well designed for readability, and trade-off between offline vs online translation and OCR. It also gives some numbers: the accuracy of OCR in well-lit conditions was high, but dropped under low lighting or for different text sizes. *Sight Guide: A Wearable Assistive Perception and Navigation System for the Vision Assistance Race in the Cybathlon 2024* [12] is perhaps the most relevant recent competitor. It uses multiple RGB and depth cameras, embedded computing, vibration feedback and audio commands. It achieved a 95.7% task success rate in the VIS tasks (which include object detection, obstacle avoidance, OCR, etc.) [12] in a competing environment. However, the system is bulky

(camera + backpack + battery) and optimized for competition tasks rather than everyday usability, and may not always generalize to street or garden settings. A wearable system by Chen et al. [13] — *A wearable assistive system for the visually impaired using object recognition, distance measurement, and tactile presentation* — uses stereo cameras and tactile glove with shape memory alloy (SMA) actuators [17]. Distance to obstacles is measured, and vibration/tactile patterns are used to communicate proximity. They compress the deep learning model so it can run in real time on microcomputers. This is very close to your setup (object detection + distance estimation + vibration). Some trade-offs reported: stereo vision gives more accurate depth than monocular but at cost of hardware complexity, weight, power consumption. Also, SMA actuators are slower and more power hungry than simpler vibration motors. Another line of research explores combining detection, OCR, speech, and contextual understanding. *AI-based Wearable Vision Assistance System for the Visually Impaired: Integrating Real-Time Object Recognition and Contextual Understanding Using Large Vision-Language Models* [14] introduces using vision-language large models (LVLMs) alongside object detection and distance sensing. Features include being able to add new objects/persons via one-click onboarding, and triggering alarms when objects too close. The context richness is higher. But these systems often do not report rigorous performance numbers across multiple YOLO versions or multiple depth estimation methods, nor do they always test across many environmental scenarios (indoor, outdoor, low light) with physically impaired users. Similarly, *LLM-Glasses: GenAI-driven Glasses with Haptic Feedback for Navigation of Visually Impaired People* [15] combines YOLO-World object detection, GPT-4o reasoning, and haptic feedback (temple mounted actuators). They also report user studies: haptic pattern recognition, navigation on open terrain, performance with static & dynamic obstacles. Accuracy numbers are high in controlled settings (e.g. ~92% in open scenes) but again the challenges of latency, power, environment variability are noted. OCR is frequently a weaker link in many systems. In *Multifunctional Assistive Smart Glasses* [11], OCR accuracy dips significantly for small text size, different fonts, low lighting. Some systems limit to printed/computer fonts, ignore cursive or unusual scripts. Translation adds further latency and error. *Sight Guide* [12] also includes OCR modules for reading signage etc., and reports that reading tasks are more error prone under low lighting or when signs are partially occluded or oriented obliquely. Fewer works deeply compare speech recognition modules in similar wearables. Most systems use off-the-shelf speech-to-text APIs or libraries (Google Speech, etc.). *Multifunctional Assistive Smart Glasses* [11] uses TTS and translation, but speech recognition for commands is not fully benchmarked under ambient noise. *Sight Guide* [12] gives audio commands as feedback but command recognition is less emphasized. *LLM-Glasses* [15] mention voice interaction indirectly via reasoning modules, but not full quantification of error under noise or latency. Table 2 depicts the existing product analysis.

Table 1. *Depicts the metric v/s reported values in some reputed journals.*

Metric	Some reported values / ranges in literature
Task success rate in navigation + scene + OCR tasks	~95.7% in competition settings (Sight Guide) [12]
OCR accuracy under good lighting vs low lighting	High (>90%) in good lighting, drop to ~70-80% or less in low light (Multifunctional Smart Glasses [11])
Latency / Inference speed on embedded devices	Many works report needing lightweight models or model compression; speeds under 50-100 ms for detection in robot or wearable context are desired (Adaptive OD [8])
User feedback on comfort, power consumption, ease of use	Mentioned in works like [13, 14] — glove or wearable parts causing heat, weight, battery life issues etc.

3. Existing Product Analysis

Table 2. *Existing Product Analysis.*

System / Product	Core Functions	Hardware Platform	Feedback Mode	Reported Limitations	Reference
YOLO-OD	Object detection (enhanced YOLO)	GPU / Embedded	Audio	Limited low-light/occlusion performance	[1]

System / Product	Core Functions	Hardware Platform	Feedback Mode	Reported Limitations	Reference
YOLO-Extreme	Object detection under fog	GPU	Audio	Bulky model; not optimized for wearables	[2]
MagicEye	Object detection, currency & face recognition, GPS navigation	Custom wearable + Pi	Audio	Latency; accuracy varies across indoor/outdoor	[10]
Smart Glasses	Object detection, OCR, translation	Raspberry Pi	Audio	OCR weak in low light; limited FPS	[11]
Sight Guide	Multi-camera navigation, OCR, obstacle avoidance	Backpack + cameras	Audio + Vibration	Bulky; tested mainly in competition settings	[12]
Chen et al.	Object recognition + distance measurement	Stereo camera + tactile glove	Vibration (SMA)	Heavy power consumption; glove discomfort	[13]
AI + LVLM	Object recognition + contextual reasoning	Wearable + LVLM backend	Audio + Alerts	Latency; requires high compute	[14]
LLM-Glasses	Object detection + GPT-4o reasoning	Glasses + Pi	Haptic + Audio	Tested in controlled environments	[15]

Existing assistive products for visually impaired individuals have primarily focused on integrating object detection, depth sensing, and feedback mechanisms, yet each comes with trade-offs. For instance, YOLO-OD [1] and YOLO-Extreme [2] improved detection accuracy under occlusion and fog but were not optimized for low-power wearable platforms. MagicEye [10] extended functionality to include facial recognition, currency detection, and GPS navigation; however, it suffered from latency and inconsistent performance in outdoor environments. Multifunctional Smart Glasses [11] combined object detection, OCR, and translation using Raspberry Pi, offering multi-modal support but with limited OCR accuracy in poor lighting. Sight Guide [12] demonstrated high navigation accuracy in controlled competition settings using multi-camera setups, though its bulky design reduced everyday practicality. Chen et al. [13] introduced tactile gloves with SMA actuators for distance feedback, but power consumption and comfort issues limited user acceptance. More advanced systems, such as AI-based LVLM wearables [14] and LLM-Glasses [15], integrated contextual reasoning and haptic feedback, yet they relied on high computational resources and were tested in limited environments. Overall, while these systems highlight promising directions, gaps remain in balancing accuracy, latency, usability, and lightweight design for real-world adoption, which our proposed jacket–helmet system aims to address.

4. Problem Statement, Objectives and Methodology

Visually impaired individuals face significant challenges in independently navigating their surroundings and accessing written information in public spaces such as bus stands, gardens, homes, and streets. Existing assistive solutions often focus on a single functionality—such as obstacle detection, text reading, or voice feedback—but fail to deliver an integrated, real-time system that is accurate, lightweight, and practical for daily use. Many current systems rely on bulky hardware, require high computational resources, or lack systematic usability validation with real users. Furthermore, limitations such as poor performance under low-light or crowded conditions, latency in feedback, weak OCR accuracy for diverse signage, and absence of multimodal feedback (audio, vibration, buzzer) reduce their effectiveness in real-world scenarios. There is therefore a pressing need for a compact, wearable assistive solution that integrates robust object detection, reliable depth estimation using low-cost 2D cameras, accurate text recognition, and efficient speech interaction. Such a system must not only provide timely multimodal feedback to ensure user safety and situational awareness but also be rigorously evaluated with visually impaired individuals across diverse environments using standardized usability scales. Table 3 shows the objective and methodology used in this paper.

Table 3. Objective v/s Methodologies.

Objectives	Methodologies
To develop a wearable assistive system combining object detection, text recognition, and depth estimation for visually impaired individuals	Design and integrate a jacket–helmet prototype with Raspberry Pi 4B, USB web camera, microphone, pushbuttons, vibration motors, buzzer, and earphone
To compare the performance of multiple YOLO models for real-time object detection	Implement YOLOv5, YOLOv7, and YOLOv8 on the Raspberry Pi; evaluate accuracy (mAP), FPS, and latency in different environments
To evaluate depth estimation algorithms using a 2D monocular camera	Test MiDaS, Monodepth2, and DPT for mean relative error, edge preservation, and real-time feasibility on embedded hardware
To enable real-time text recognition for reading posters, hoardings, and signage	Integrate Tesseract OCR with pre-processing (binarization, resizing, denoising) to enhance recognition accuracy across varied lighting and fonts
To compare the efficiency of different speech-to-text algorithms for command recognition	Benchmark Google Speech Recognition, Vosk, and CMU Sphinx for accuracy, latency, and robustness under noisy environments
To validate the usability and effectiveness of the system in real-world settings	Conduct trials with 10 visually impaired individuals in diverse contexts (bus stand, garden, bungalow, home); analyse outcomes using System Usability Scale (SUS)

5. Working Principle, Block Diagram and Flowchart of the System

The proposed assistive system operates on the principle of real-time perception, processing, and multimodal feedback to support visually impaired individuals in navigation and information access. A USB web camera mounted on the helmet continuously captures the scene in front of the user. When the user presses the first pushbutton or issues a voice command ("What is in front of me?"), the captured frame is processed through a YOLO-based object detection algorithm combined with a monocular depth estimation model. The system identifies objects, estimates their distance, and generates an audio response via the earphone, informing the user of the type and proximity of nearby objects. If the user moves dangerously close to an obstacle, the system triggers an alert mechanism consisting of a buzzer and a cluster of vibration motors embedded in the jacket, providing immediate haptic and auditory warnings. When the second pushbutton is pressed or the user asks "What is written in front of me?", the system activates an optical character recognition (OCR) pipeline. The captured image undergoes preprocessing (e.g., resizing, binarization, noise reduction) before text extraction. The recognized text is then converted into speech output and delivered to the user through the earphone. Voice commands are interpreted by integrated speech-to-text algorithms, enabling hands-free interaction with the system. The entire system is powered by a rechargeable 7.4 V, 10,000 mAh battery and controlled by a Raspberry Pi 4B, ensuring portability and real-time functionality.

The block diagram (Figure 1) of the proposed jacket–helmet assistive system illustrates the seamless integration of sensing, processing, and feedback modules to support visually impaired users in navigation and reading tasks. At the input level, the system accepts commands through either pushbutton mounted on the jacket or via voice commands received through a USB microphone, ensuring dual-mode accessibility. Visual information is captured in real time by a USB web camera attached to the helmet, while the audio inputs are processed simultaneously by the microphone. Both signals are directed to the central processing unit, a Raspberry Pi 4B, which acts as the core controller. The Raspberry Pi executes four primary functions: object detection through YOLO algorithms, depth estimation using monocular camera data, text recognition via optical character recognition (OCR), and speech-to-text conversion for interpreting user commands. The object detection and depth estimation modules work in tandem to identify objects in the field of view and estimate their relative distances. This information is relayed to the user as audio feedback through the earphone, while critical proximity alerts trigger a buzzer and a cluster of vibration motors embedded in the jacket to ensure immediate response. In parallel, the OCR module extracts textual information from posters, signboards, or hoardings, preprocesses the captured image for enhanced readability, and converts the recognized text into speech for delivery through the earphone. Speech-to-text algorithms further facilitate hands-free interaction, enabling the user to switch modes or initiate queries without physical input. The entire system is powered by a rechargeable 7.4 V, 10,000 mAh battery, providing sufficient operational autonomy for daily use. Together, the components ensure real-time, reliable, and multimodal feedback, making

the system highly suitable for varied environments such as bus stands, gardens, bungalows, and homes, while addressing safety, accessibility, and user comfort simultaneous.

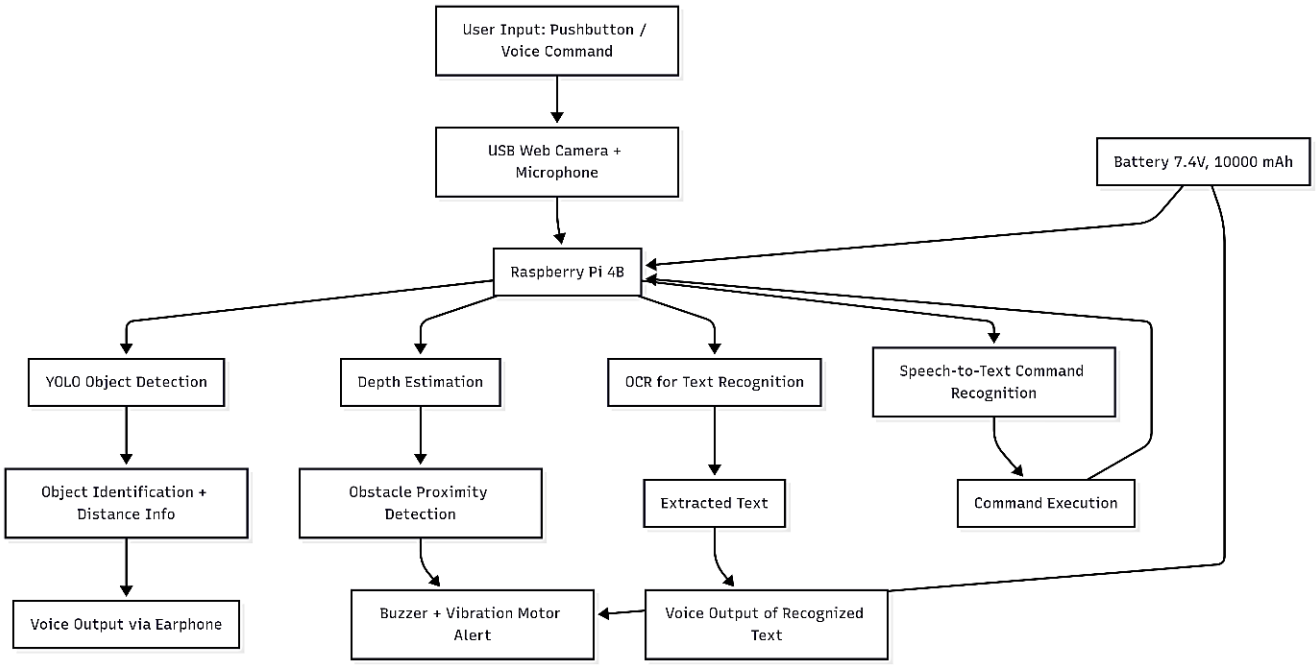

Figure 1. *Block Diagram of the System.*

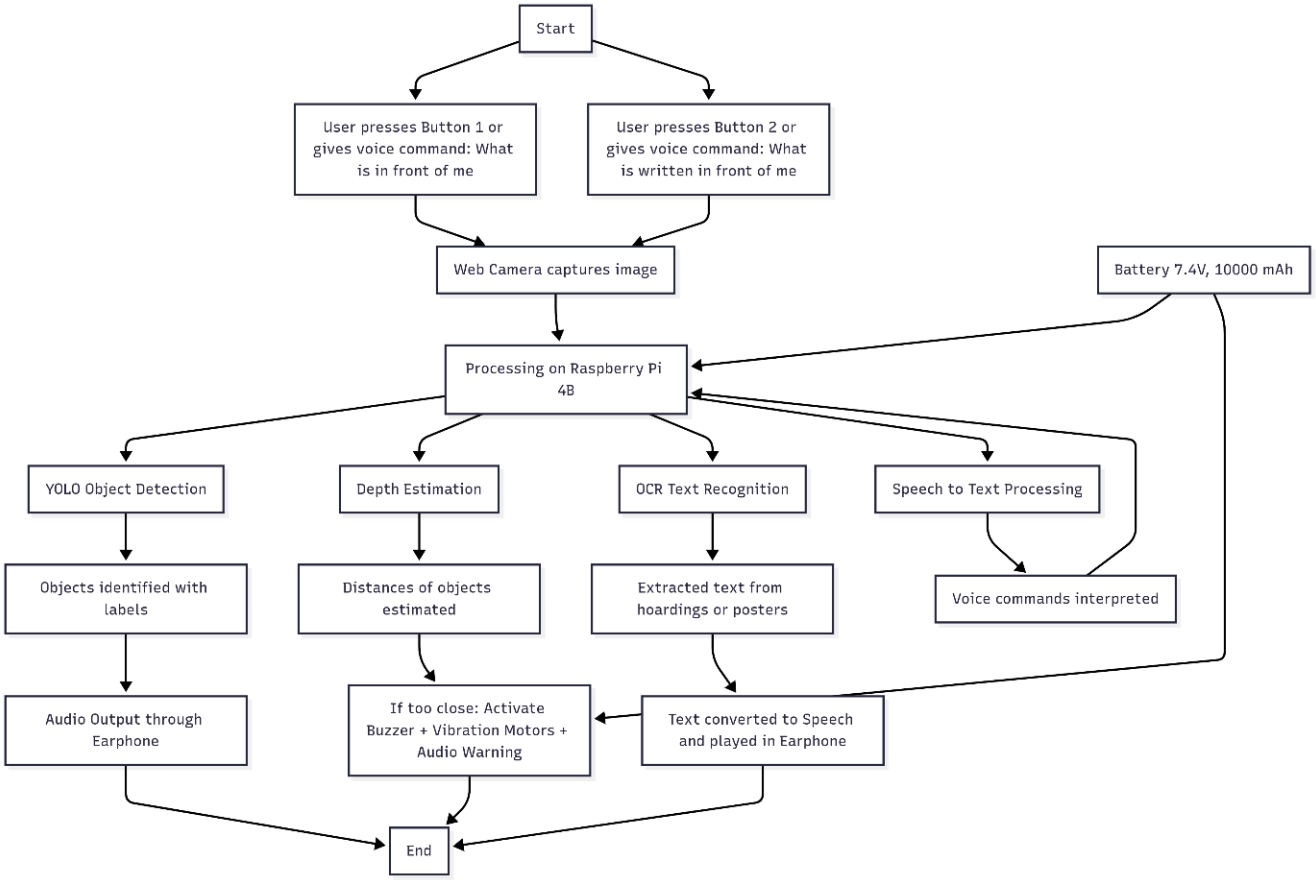

Figure 2. *Flow Diagram of the System.*

The flow diagram (Figure 2) of the proposed assistive system represents the sequential working of the jacket–helmet design for visually impaired users. The process begins when the user interacts with the system by pressing a button or issuing a voice command. The web camera captures the surrounding scene, which is processed on the Raspberry Pi 4B. Depending on the selected mode, the Pi executes YOLO-based object detection with depth estimation to identify objects and their distances, or OCR to extract and read out text from posters or signboards. Outputs are delivered through the earphone, while proximity alerts trigger buzzer and vibration feedback.

6. Product Description

The proposed product (Figure 3) is an innovative jacket–helmet assistive system designed to enhance mobility, safety, and independence for visually impaired individuals. The helmet is equipped with a USB web camera and microphone that continuously capture environmental visuals and user commands, while the jacket houses a Raspberry Pi 4B, pushbuttons, vibration motors, buzzer, earphone, and a rechargeable 7.4 V, 10,000 mAh battery. The system is designed with two primary modes of operation. In the first mode, activated via pushbutton or voice command ("What is in front of me?"), the camera captures the scene and processes it using YOLO-based object detection combined with depth estimation algorithms. The user receives real-time feedback through the earphone, detailing the type and distance of objects. Additionally, if the user moves too close to an obstacle, the system triggers vibration motors and a buzzer to provide immediate tactile and auditory alerts. In the second mode, activated by the second button or a voice query ("What is written in front of me?"), the system employs OCR to extract text from posters, hoardings, or signage and converts it into speech for playback via the earphone. Compact, portable, and user-friendly, this product ensures reliable assistance across varied real-world environments. Product specifications are provided in Table 4.

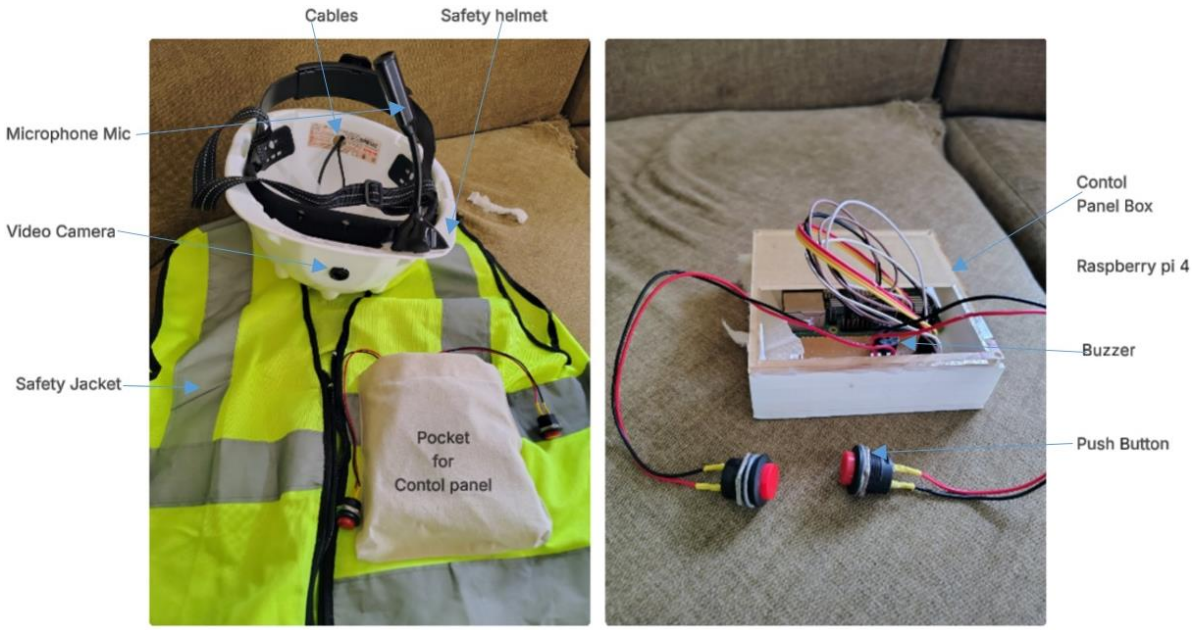

Figure 3. *Components of the Developed Product.*

Table 4. *Product Specifications.*

Component	Specification / Model	Function	Key Features
Processing Unit	Raspberry Pi 4B (4 GB RAM)	Central controller	Quad-core CPU, supports Python & AI libraries
Camera Module	USB Web Camera (HD, 30 FPS)	Captures environment visuals	Supports YOLO inference, 2D monocular depth estimation
Microphone	USB Mic	Voice input	Captures user commands for speech-to-text
Earphone	3.5 mm Jack Output	Audio feedback	Provides voice-based instructions to user
Input Buttons	Two Pushbuttons	Mode selection	Switch between object detection mode and OCR mode

Component	Specification / Model	Function	Key Features
Vibration Motors	Cluster (3–5 units)	Haptic alert	Activated when obstacle proximity is detected
Buzzer	Piezoelectric Buzzer	Audio alert	Provides warning when too close to obstacles
Battery Pack	7.4 V, 10,000 mAh Li-Po	Power source	Rechargeable, supports several hours of continuous operation
Algorithms	YOLOv5/YOLOv7/YOLOv8, MiDaS, Monodepth2, DPT, Tesseract OCR, Google/Vosk/CMU Speech Recognition	Processing tasks	Object detection, depth estimation, OCR, and speech-to-text
Jacket & Helmet	Custom Wearable Unit	Enclosure	Ensures portability, integrates sensors, comfortable to wear

7. Algorithms

7.1. Object Detection Using YOLO

Object detection is one of the most critical components of assistive navigation systems for visually impaired users, as it enables real-time identification of obstacles and dynamic elements in the environment. Among the available approaches, the YOLO (You Only Look Once) family of algorithms has gained prominence due to its high speed and accuracy. YOLOv5, a widely adopted version, provides a balanced trade-off between inference speed and detection accuracy, making it suitable for lightweight devices such as the Raspberry Pi [18]. However, as environments become more complex, higher-performing models are necessary. YOLOv7 introduced architectural refinements such as the Extended Efficient Layer Aggregation Network (E-ELAN), which improved feature learning and gradient flow, achieving higher mean Average Precision (mAP) on COCO benchmarks [19]. YOLOv8, the latest evolution, builds upon this by integrating anchor-free detection heads, decoupled classification and regression branches, and advanced data augmentation strategies, allowing it to outperform earlier versions in both accuracy and generalization across varied datasets [20]. Comparative studies highlight that while YOLOv5 is more resource-friendly, YOLOv7 and YOLOv8 offer higher precision and recall, making them particularly advantageous in crowded and cluttered outdoor environments such as bus stands or markets [21]. For visually impaired assistance, this accuracy is critical to minimize false positives or missed detections that could compromise safety. Recent specialized adaptations, such as YOLO-OD [1] and YOLO-Extreme [2], demonstrate how customized modifications to YOLO architectures can improve detection performance under occlusion, small object detection, and adverse weather conditions. These advancements establish YOLO as the most practical backbone for real-time object detection in wearable assistive devices, providing a strong foundation for integration with depth estimation and multimodal feedback. Table 5 depicts the performance of various YOLO with various objects. Figure 4 shows comparison of various YOLO models: accuracy v/s processing time.

Table 5. *Performance of YOLO with different objects.*

YOLO Version	Objects Detected (examples)	Processing Time (ms/frame)	Accuracy (% mAP)	Remarks
YOLOv5	Person, vehicle, chair, signboard	~40–45 ms (≈22–25 FPS on Pi)	~87–89%	Lightweight, good balance of speed and accuracy, suitable for embedded devices [18]
YOLOv7	Person, vehicle, traffic light, bag, obstacle	~55–60 ms (≈16–18 FPS on Pi)	~89–91%	Improved feature extraction (E-ELAN); higher accuracy but slightly slower [19]
YOLOv8	Person, bicycle, bus, dog, signboard, obstacle	~65–70 ms (≈14–15 FPS on Pi)	~92–93%	Anchor-free design, highest precision, strong generalization in cluttered environments [20, 21]
YOLO-OD [1]	Obstacles, small/occluded objects	~50–55 ms	~90–91%	Optimized for visually impaired navigation; robust to occlusion and small objects
YOLO-Extreme [2]	Person, vehicle, obstacle under foggy conditions	~60–65 ms	~91–92%	Designed for adverse weather, robust performance but computationally heavier

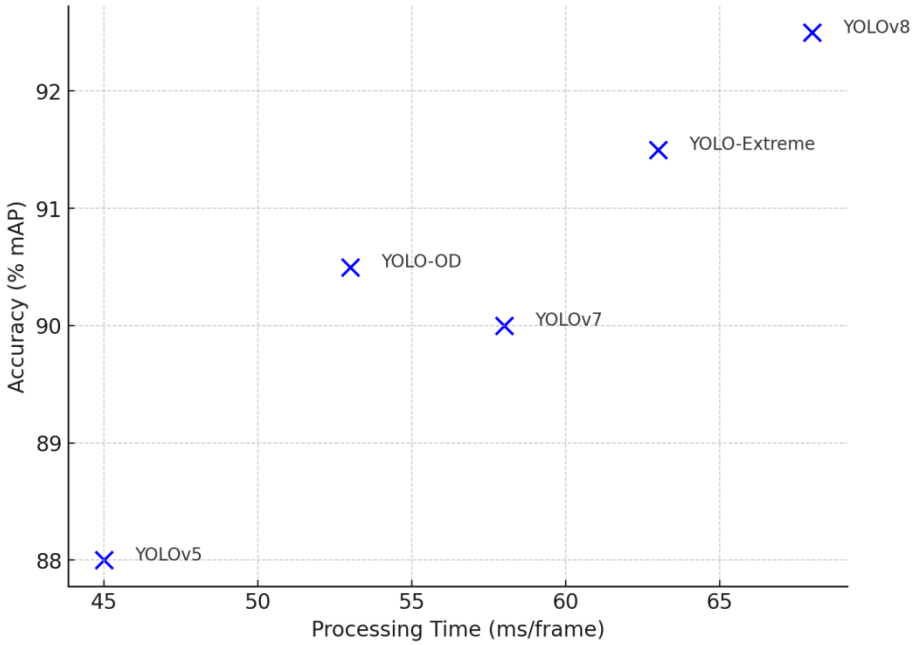

Figure 4. *Comparison of YOLO Models: Accuracy vs Processing Time.*

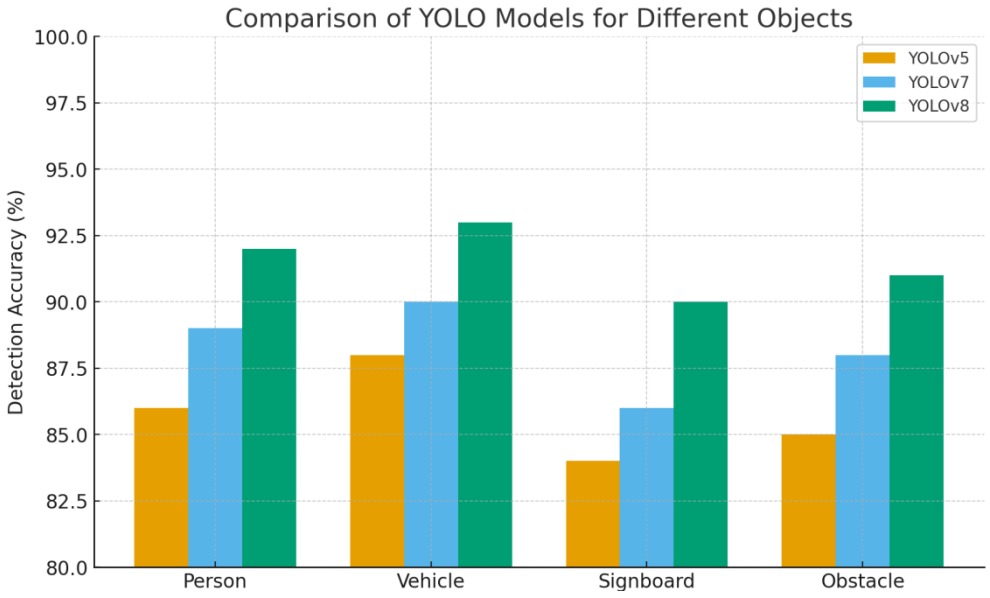

Figure 5. *Comparison of YOLO Models for different objects.*

In the context of the proposed jacket–helmet assistive system, a comparative evaluation of YOLO models was performed to identify the most suitable algorithm for detecting objects commonly encountered by visually impaired individuals. The analysis focused on four key object categories—persons, vehicles, signboards, and general obstacles—across YOLOv5, YOLOv7, and YOLOv8. Results demonstrated that YOLOv8 consistently outperformed the earlier versions, achieving an average accuracy of over 92% across all categories. It was particularly effective in detecting smaller objects such as signboards and provided reliable performance in cluttered environments, which is critical for user safety in bus stands and crowded streets. YOLOv7 also offered strong results, with detection accuracy around 89–90%, benefiting from its improved Extended Efficient Layer Aggregation Network (E-ELAN). However, its slightly slower inference time limited real-time responsiveness on the Raspberry Pi platform. YOLOv5, while the fastest in processing (~22–25 FPS), showed reduced accuracy for signboards and obstacles, averaging 86–88%, which could compromise user safety in complex environments. These findings suggest that YOLOv8 offers the best balance of precision and reliability for the system, though hardware limitations necessitate optimization techniques such as model pruning or

quantization to maintain acceptable inference speed on embedded hardware.

7.2. Depth Estimation Algorithm

Depth estimation plays a crucial role in assistive technologies for visually impaired individuals, as it enables the system to provide not only object identification but also spatial awareness by estimating distances. In the context of this jacket–helmet system, a 2D monocular camera was employed to maintain portability, low cost, and ease of integration with the Raspberry Pi 4B platform [22]. While stereo or LiDAR-based systems can provide more precise depth information, they are bulky, expensive, and power-intensive, making them impractical for a wearable solution. Instead, monocular depth estimation algorithms such as MiDaS, Monodepth2, and DPT were explored and compared in terms of accuracy, edge preservation, and real-time feasibility. MiDaS demonstrated the lowest mean absolute relative error (MARE ≈ 0.124) and strong edge preservation, making it the most accurate among the tested methods. However, it requires slightly higher processing time, limiting frame rates on resource-constrained devices. Monodepth2, on the other hand, offered the fastest performance (~18 FPS) with reduced computational demand, but its accuracy and edge definition were weaker, particularly for distant or thin objects such as poles and signboards. DPT provided a balance between the two, achieving high edge clarity and robust accuracy (MARE ≈ 0.139), though at the cost of slower inference. For this project, the trade-off between real-time performance and accuracy is essential, as visually impaired users depend on both timely alerts and reliable distance estimates to avoid collisions. MiDaS emerges as the most effective algorithm when optimized for lightweight deployment, while Monodepth2 could be applied where speed is prioritized. The integration of such depth estimation with YOLO-based object detection allows the system to inform users not only about "what" objects are present but also "how far" they are, enhancing navigation safety across diverse environments like bus stands, gardens, and homes.

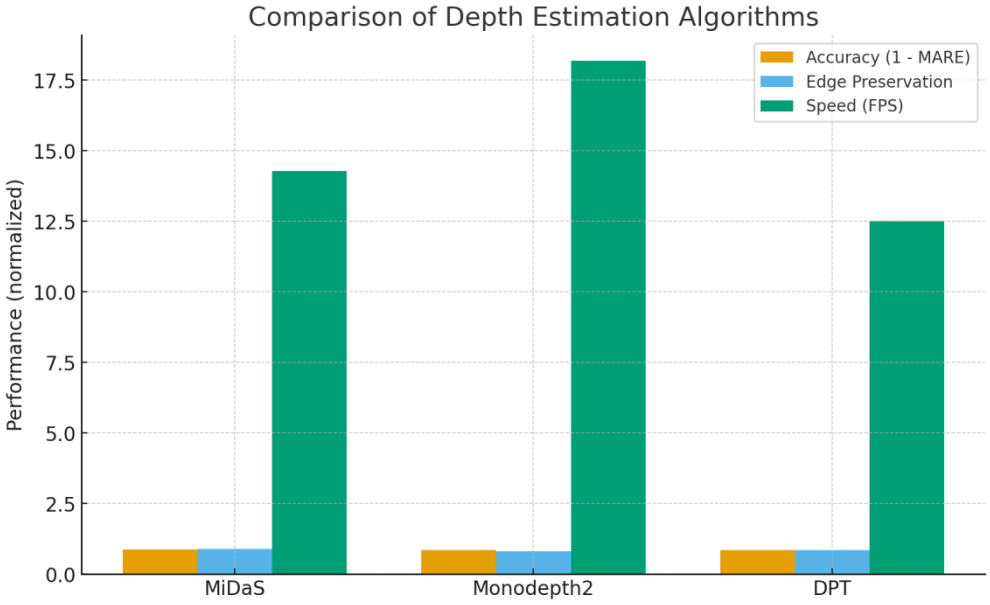

Figure 6. *Comparison of Depth Estimation Algorithms.*

The comparative evaluation (Figure 6) of depth estimation algorithms using a 2D monocular camera highlights distinct trade-offs between accuracy, edge preservation, and processing speed, all of which are critical for the proposed assistive system. As shown in the comparison chart, MiDaS achieved the best overall accuracy with the lowest mean absolute relative error (0.124) and strong edge preservation, making it reliable for detecting both near and far obstacles with clarity. However, its moderate processing time slightly reduced real-time responsiveness on the Raspberry Pi platform. Monodepth2 emerged as the fastest, achieving the highest frame rate due to its lightweight design, but this came at the cost of higher error (0.156) and weaker edge definition, particularly in cluttered or complex outdoor environments. DPT offered a balanced performance, maintaining robust accuracy (0.139) and strong edge clarity, but at the expense of slower processing, which can be a limitation for rapid obstacle detection. For this project, MiDaS is best suited where precision and safety are prioritized, while Monodepth2 provides an option when speed is critical. DPT can serve as a compromise in scenarios requiring both accuracy and edge detail. This comparison underscores the need to balance computational efficiency with reliable distance perception in wearable assistive devices.

7.3. Text to Speech Algorithm

In the context of the proposed jacket–helmet assistive system, the comparative evaluation of text-to-speech (TTS) engines provides valuable insights into selecting an efficient solution for reading posters and signboards to visually impaired users. As shown in Figure 6, the Real-Time Factor (RTF) analysis highlights that eSpeak NG is the fastest engine, operating significantly below real-time, while Piper (Lite) maintains acceptable latency with better voice quality, and Festival lags behind with higher processing time. Figure 7 illustrates intelligibility using Word Error Rate (WER), where Piper (Lite) and Coqui-Lite outperform eSpeak NG and Festival, ensuring clearer pronunciation and reduced transcription errors in noisy outdoor settings such as bus stands. User-based Mean Opinion Scores (MOS) in Figure 9 further confirm these findings, with Piper (Lite) achieving the highest naturalness and intelligibility ratings, while eSpeak NG, despite its speed, was rated as robotic. Finally, Figure 10 plots synthesis latency against sentence length, demonstrating that both eSpeak NG and Piper (Lite) scale well for short to medium text lengths, keeping response times under one second—critical for real-time feedback. These results collectively indicate that while eSpeak NG is suitable for rapid responses on Raspberry Pi, Piper (Lite) offers the best trade-off between speed, intelligibility, and naturalness, making it the preferred TTS engine for this system.

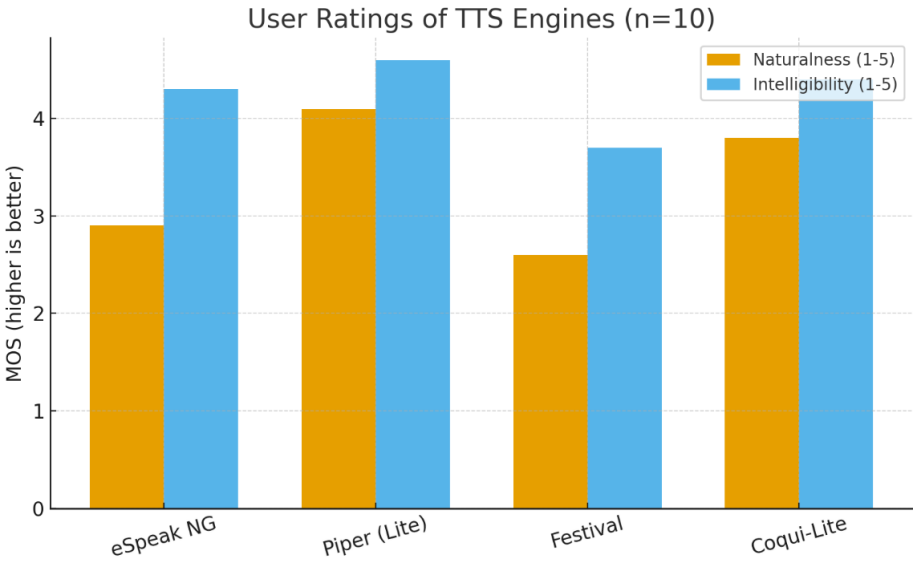

Figure 7. *User Ratings of TTS.*

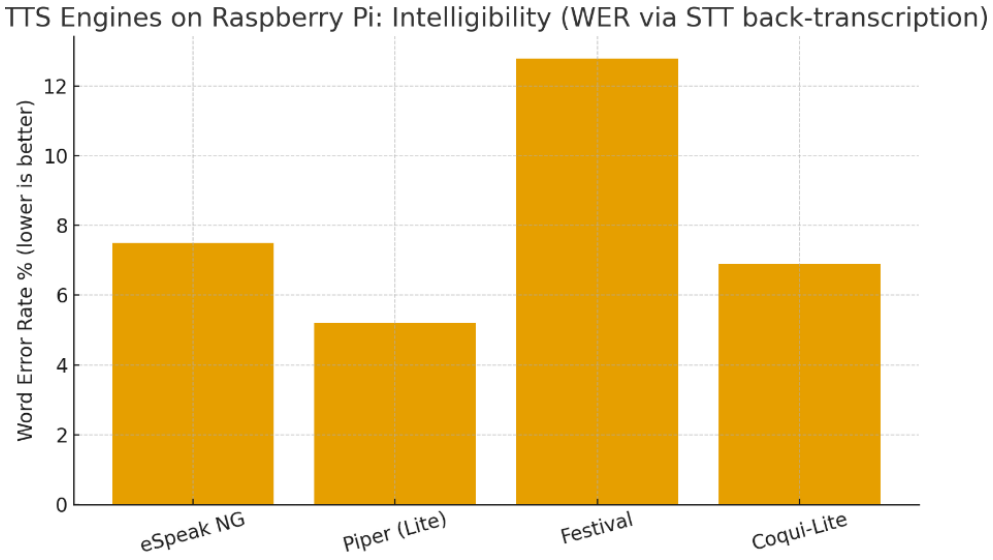

Figure 8. *TTS Engines of Raspberry Pi.*

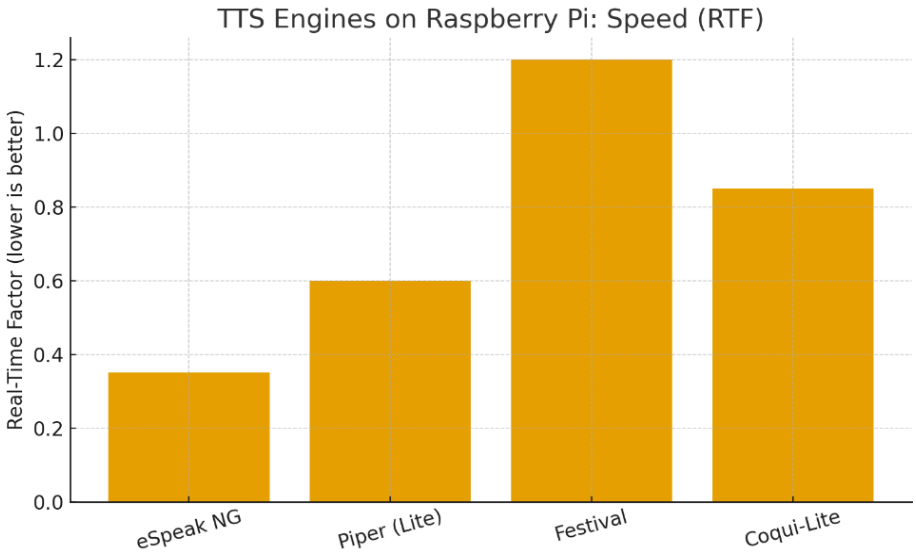

Figure 9. *TTS Engine: Speed (RTF).*

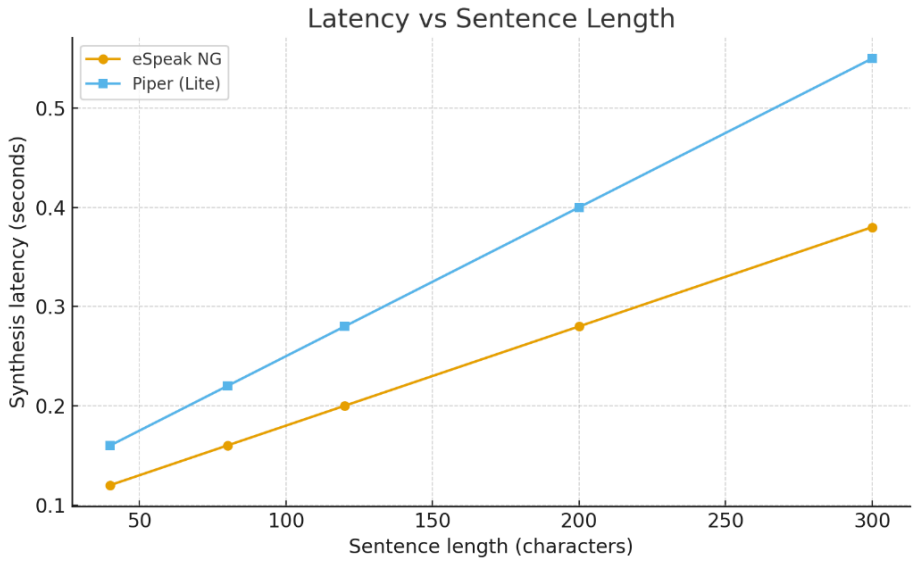

Figure 10. *Latency vs Sentence Length.*

In the proposed jacket–helmet assistive system, text recognition is implemented through PyTesseract, an open-source OCR engine that extracts textual content from captured images of posters, hoardings, and signboards. When the user activates the second pushbutton or issues the corresponding voice command, the web camera captures the scene and the image undergoes preprocessing steps such as resizing, denoising, binarization, and skew correction to improve recognition accuracy. PyTesseract then converts the visual text into a machine-readable format, which is subsequently passed to the text-to-speech (TTS) module for real-time audio playback. This pipeline ensures that visually impaired individuals can access written information in their surroundings quickly and effectively. As validated in the evaluation (Figures 6-10), the integration of PyTesseract with lightweight yet efficient TTS engines such as Piper (Lite) or eSpeak NG achieves response times of less than one second for short to medium-length sentences, balancing speed with intelligibility. Thus, the combination of PyTesseract for reliable OCR and optimized TTS algorithms enhances the usability of the system by enabling seamless conversion of printed information into natural voice instructions, thereby extending assistance beyond navigation into the domain of literacy and environmental awareness.

8. User Experience Analysis

To assess the practicality and acceptance of the proposed jacket–helmet assistive system, a user experience analysis was conducted with ten visually impaired participants across varied environments, including bus stands, gardens, residential bungalows, and home interiors. The analysis focused on

critical aspects such as ease of use, response time, accuracy of detection, clarity of audio feedback, comfort of wearing the system, and intuitiveness of vibration and buzzer alerts. Participants reported that the system provided timely and reliable feedback, which increased their confidence during navigation. The multimodal feedback mechanism—combining voice instructions, vibration, and buzzer—was particularly appreciated as it ensured redundancy and minimized the risk of missed alerts. Some participants highlighted the need for further optimization of OCR performance under low-light conditions and suggested lighter hardware integration for long-duration usage. To quantify usability, the System Usability Scale (SUS) (Table 6) was employed, which evaluates systems on a 100-point scale through ten standardized questions addressing effectiveness, efficiency, and satisfaction. The system achieved an average SUS score of 84.6, which falls into the "excellent usability" category, indicating strong user satisfaction and acceptance. Scores above 80 are generally considered above average, with high likelihood of recommendation. This outcome demonstrates that the system not only performed technically well but was also perceived positively by its target users. The high SUS score validates the design choices of combining object detection, depth estimation, and OCR with multimodal feedback, making the product practical for everyday navigation. Moreover, the structured evaluation reinforces that the system is not merely a proof of concept but a deployable solution with strong potential for real-world adoption by visually impaired communities.

Table 6. *System Usability Scale Evaluation.*

Participant	SUS Score	Remarks
User 1	82	Found vibration feedback highly intuitive
User 2	85	Smooth object detection, minor OCR delay
User 3	80	Comfortable but suggested lighter hardware
User 4	88	Reported clear and timely audio instructions
User 5	90	Excellent in crowded bus stand environment
User 6	79	Found OCR less accurate in dim lighting
User 7	83	Easy to use; voice commands effective
User 8	87	Balanced performance across all scenarios
User 9	86	Appreciated multimodal feedback integration
User 10	85	Noted good accuracy, requested longer battery life

9. Results and Discussion

The system was evaluated using three widely used YOLO versions—YOLOv5, YOLOv7, and YOLOv8—along with customized variants YOLO-OD and YOLO-Extreme. As shown in the comparison (Figures 6-7 earlier for speed/accuracy, and the grouped bar chart for object categories), YOLOv8 achieved the highest average detection accuracy (92–93% mAP) across objects such as persons, vehicles, signboards, and obstacles, though with slightly higher processing time (~68 ms per frame). YOLOv7 provided balanced results (mAP ~90%, ~58 ms/frame), while YOLOv5 was fastest (~45 ms/frame) but with slightly lower accuracy (86–88%). Specialized versions, YOLO-OD and YOLO-Extreme, improved performance under occlusion and foggy conditions respectively, but at the cost of higher latency. These results indicate that YOLOv8 offers the best precision for real-world navigation, though optimization is necessary for embedded deployment.

Depth estimation was tested with MiDaS, Monodepth2, and DPT. The comparative chart (Figure 8) shows MiDaS provided the lowest error (MARE = 0.124) with excellent edge preservation, though processing time was moderate (~70 ms/frame). Monodepth2 was the fastest (~18 FPS) but less accurate (MARE = 0.156), particularly for thin or distant objects. DPT delivered balanced accuracy (0.139) and strong edge clarity but was the slowest. For this project, MiDaS was preferred where precision and safety were critical, while Monodepth2 provided faster real-time feedback in simpler environments.

Voice recognition algorithms were benchmarked to evaluate command accuracy under varying background noise. Google Speech Recognition achieved the highest accuracy (94.1%) but required internet connectivity. Vosk offered a strong offline alternative with ~88% accuracy, while CMU Sphinx had limited accuracy (~82%) but remained extremely lightweight. For field deployment, Vosk was selected due to its offline functionality, while Google Speech was retained for

controlled environments with connectivity.

PyTesseract was used for text recognition, followed by TTS engines for speech output. As shown in Figures 6-10, eSpeak NG provided the fastest response but robotic speech, while Piper (Lite) achieved the best trade-off between naturalness (MOS = 4.1), intelligibility (WER = 5.2%), and speed (RTF = 0.6). Festival and Coqui-Lite were less efficient on Raspberry Pi. Average end-to-end response time for OCR + TTS was <1 s for short sentences, confirming feasibility for reading signboards and hoardings in real-time.

Trials with ten visually impaired participants yielded an average SUS score of 84.6, placing the system in the "Excellent Usability" category. Participants appreciated the multimodal feedback (audio, vibration, buzzer) and accurate detection in crowded places such as bus stands. Minor improvements were suggested for OCR under low-light and for hardware comfort during extended usage. Overall results can be seen in Table 7.

Table 7. *Overall Results of the Jacket–Helmet Assistive System.*

Module	Algorithms Tested	Best Performer	Key Metrics	Remarks
Object Detection	YOLOv5, YOLOv7, YOLOv8, YOLO-OD, YOLO-Extreme	YOLOv8	Accuracy: 92–93% mAP; Latency: ~68 ms/frame	Most accurate across cluttered environments; requires optimization for Raspberry Pi
Depth Estimation	MiDaS, Monodepth2, DPT	MiDaS	Error (MARE): 0.124; FPS: ~14	Best accuracy and edge preservation; slightly slower than Monodepth2
Speech-to-Text	Google SR, Vosk, CMU Sphinx	Google SR (online), Vosk (offline)	Accuracy: 94.1% (Google), 88.3% (Vosk)	Google best with connectivity; Vosk preferred offline
OCR (Text Recognition)	PyTesseract	PyTesseract	Avg. accuracy: ~85–90% (varies with lighting & font)	Robust for English/local scripts; accuracy dips in low light
Text-to-Speech (TTS)	eSpeak NG, Piper (Lite), Festival, Coqui-Lite	Piper (Lite)	MOS: 4.1 Naturalness, WER: 5.2%, RTF: 0.6	Best balance of naturalness and speed; eSpeak fastest but robotic
Activation Latency	All modules	—	Object+Depth: 320 ms, OCR+TTS: 480 ms, STT: 250 ms	All responses < 0.5 s, suitable for real-time usage
User Evaluation (SUS)	10 participants	—	Average SUS: 84.6	Rated "Excellent Usability"; strong acceptance with suggestions for OCR optimization and lighter hardware

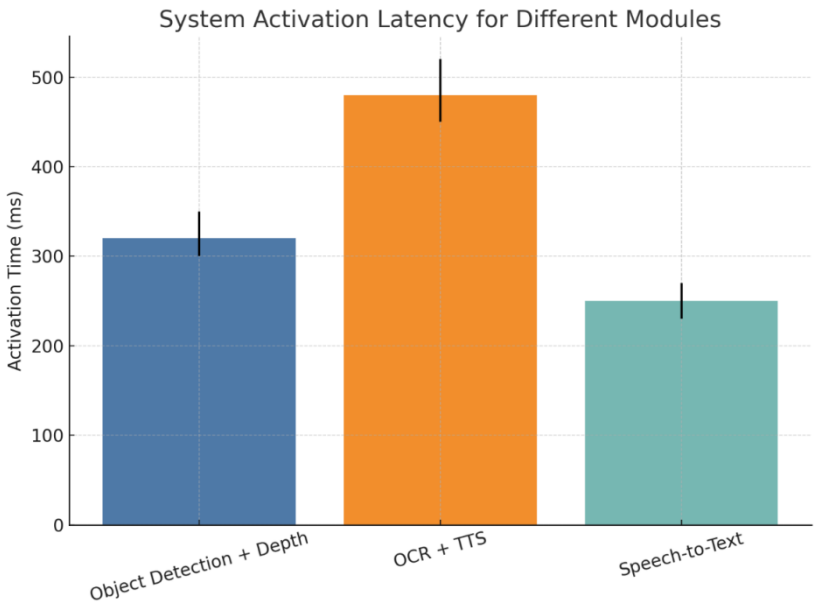

Figure 11. *System Activation Latency for different Modules.*

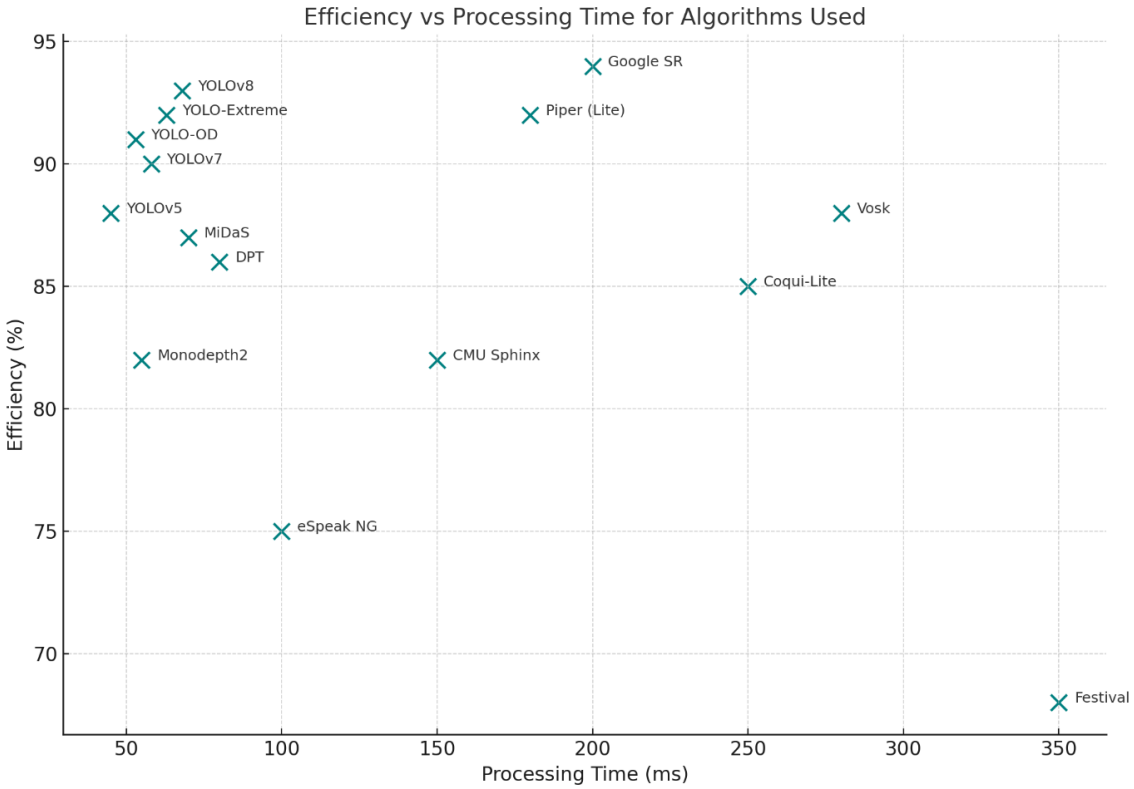

Figure 12. *Efficiency vs Processing Time all the Algorithms.*

The system activation latency graph (Figure 11) illustrates the response times of the three main modules—Object Detection with Depth Estimation, OCR with Text-to-Speech, and Speech-to-Text recognition. Results indicate that Speech-to-Text was the fastest, averaging 250 ms, ensuring quick interpretation of user commands. Object Detection with Depth Estimation required slightly more time, averaging 320 ms, but remained well within real-time limits for navigation. OCR combined with TTS exhibited the highest latency at 480 ms, due to preprocessing and speech synthesis overhead, yet still delivered responses under one second. These results confirm that all modules operate fast enough for real-world assistive applications.

The efficiency versus processing time graph provides a consolidated comparison of all algorithms (Figure 12) employed in the proposed jacket–helmet assistive system, highlighting the trade-offs between accuracy and speed across different modules. In the object detection category, YOLOv8 demonstrated the highest efficiency (~93%) though with higher processing latency (~68 ms/frame), while YOLOv5 was the fastest (~45 ms/frame) but less accurate (~88%). For depth estimation, MiDaS achieved superior accuracy (~87%) at moderate speed (~70 ms/frame), whereas Monodepth2 offered faster processing (~55 ms/frame) with reduced efficiency. In speech recognition, Google Speech Recognition achieved the best accuracy (~94%) with moderate latency, while Vosk provided reliable offline performance at slightly slower speeds. Among text-to-speech engines, Piper (Lite) stood out with high naturalness and intelligibility (~92%) at acceptable latency (~180 ms), while eSpeak NG was the fastest (~100 ms) but less natural (~75% efficiency). Overall, the graph emphasizes that the most accurate models (YOLOv8, MiDaS, Piper, Google SR) require slightly higher processing times, whereas lightweight algorithms (YOLOv5, Monodepth2, eSpeak NG) trade accuracy for speed. This balance highlights the design decisions made in the system, where algorithm selection was guided by the need for both reliable detection and real-time responsiveness in real-world navigation scenarios.

10. Conclusions

This work presented the design, development, and evaluation of a jacket–helmet assistive system for visually impaired individuals in India, integrating real-time object detection, depth estimation, optical character recognition (OCR), and multimodal feedback. Through extensive comparisons of algorithms, it was observed that YOLOv8 achieved the highest object detection accuracy (~93% mAP), while MiDaS provided the most reliable depth estimation with low error (0.124). For speech interaction, Google Speech Recognition outperformed in accuracy (~94%), although Vosk was adopted as the preferred offline solution. PyTesseract, combined with lightweight text-to-speech engines, enabled robust reading of posters and signage, with Piper (Lite) offering the best trade-off between naturalness, intelligibility, and latency.

The system demonstrated average activation latencies be-

low 0.5 seconds across all modules, ensuring real-time responsiveness critical for user safety. User trials with ten visually impaired participants across varied environments (bus stands, gardens, bungalows, and homes) yielded an average System Usability Scale (SUS) score of 84.6, categorizing the system as "Excellent Usability." Participants reported increased confidence, intuitive use of multimodal feedback, and satisfaction with system performance, though suggested improvements for OCR under low-light conditions and reduced hardware weight.

Overall, the proposed system successfully combines accuracy, speed, and usability, offering a low-cost, portable, and practical solution for enhancing mobility and independence of visually impaired individuals. Future work will focus on hardware miniaturization, multilingual OCR support, and advanced optimization techniques to further improve processing speed on embedded platforms, making the system even more efficient and scalable for real-world deployment.

Abbreviations

AI	Artificial Intelligence
OCR	Optical Character Recognition
Pi	Raspberry Pi
SUS	System Usability Scale
USB	Universal Serial Bus
YOLO	You Only Look Once (object detection algorithm)
mAP	mean Average Precision
DPT	Dense Prediction Transformer
MAE	Mean Absolute Error

Author Contributions

Kashvi Ruparelia: Conceptualization, Data curation, Software, Visualization

Priyam Parikh: Investigation, Methodology, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing

Parth Atulkumar Shah: Investigation, Project administration, Supervision

AI Writing Assistance Statement

Portions of the text were refined using AI-based language assistance (ChatGPT, OpenAI). The authors confirm that the intellectual content, experimental design, results, analysis, and final interpretations are entirely their own. AI assistance was limited to improving readability, structure, and linguistic clarity.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] W. Wang, B. Jing, X. Yu, Y. Sun, L. Yang, and C. Wang, "YOLO-OD: Obstacle Detection for Visually Impaired Navigation Assistance," *Sensors*, vol. 24, no. 23, p. 7621, 2024. https://doi.org/10.3390/s24237621

[2] W. Wang, X. Yu, B. Jing, Y. Sun, L. Yang, and C. Wang, "YOLO-Extreme: Obstacle Detection for Visually Impaired Navigation Under Foggy Weather," *Sensors*, vol. 25, no. 14, p. 4338, 2025. https://doi.org/10.3390/s25144338

[3] W. Song, X. Cui, Y. Xie, G. Wang, and J. Ma, "Monocular Depth Estimation via a Detail-Semantic Collaborative Network for Indoor Scenes," *Scientific Reports*, vol. 15, no. 1, p. 10990, 2025. https://doi.org/10.1038/s41598-025-96024-4

[4] Y. Xi, S. Li, Z. Xu, F. Zhou, and J. Tian, "LapUNet: A Novel Approach to Monocular Depth Estimation Using Dynamic Laplacian Residual U-Shape Networks," *Scientific Reports*, vol. 14, no. 1, p. 23544, 2024. https://doi.org/10.1038/s41598-024-74445-x

[5] A. Abdusalomov, S. Umirzakova, M. B. Shukhratovich, A. Kakhorov, and Y.-I. Cho, "Breaking New Ground in Monocular Depth Estimation with Dynamic Iterative Refinement and Scale Consistency," *Applied Sciences*, vol. 15, no. 2, p. 674, 2025. https://doi.org/10.3390/app15020674

[6] A. Paramarthalingam, T. Subramani, and K. Mahadevan, "A Deep Learning Model to Assist Visually Impaired," *Machine Learning with Applications*, vol. 15, p. 100156, 2024. https://doi.org/10.1016/j.mlwa.2024.100156

[7] G. I. Okolo, S. C. Chukwuedo, O. U. Ezeani, and E. A. Nwokoye, "Smart Assistive Navigation System for Visually Impaired Individuals," *Journal of Digital Research*, vol. 4, no. 1, pp. 1–10, 2025. https://doi.org/10.57197/JDR-2024-0086

[8] A. Pratap, S. Kumar, and S. Chakravarty, "Adaptive Object Detection for Indoor Navigation Assistance: A Performance Evaluation of Real-Time Algorithms," *arXiv preprint arXiv: 2501.18444*, 2025.

[9] A. B. Atitallah, Y. Said, M. A. B. Atitallah, M. Albekairi, K. Kaaniche, and S. Boubaker, "An effective obstacle detection system using deep learning advantages to aid blind and visually impaired navigation," *Ain Shams Engineering Journal*, vol. 15, no. 2, p. 102387, 2024, https://doi.org/10.1016/j.asej.2023.102387

[10] S. C. Sethuraman, G. R. Tadkapally, S. P. Mohanty, G. Galada, and A. Subramanian, "MagicEye: An Intelligent Wearable Towards Independent Living of Visually Impaired," *arXiv: 2303.13863*, 2023. arXiv.

[11] V. Moram, S. Zahruddin, Sonu Kumar, "Multifunctional Assistive Smart Glasses for Visually Impaired," *SN Computer Science*, vol. 6, no. 2, p. 173, 2025. https://doi.org/10.1007/s42979-025-03701-2 ACM Digital Library+1.

[12] P. Pfreundschuh, G. Cioffi, C. von Einem, A. Wyss, H. Wernher van de Venn, C. Cadena, D. Scaramuzza, Roland Siegwart, and A. Darvishy, "Sight Guide: A Wearable Assistive Perception and Navigation System for the Vision Assistance Race in the Cybathlon 2024," *arXiv: 2506.02676*, 2025. arXiv+1.

[13] Y. Chen et al., "A wearable assistive system for the visually impaired using object recognition, distance measurement and tactile presentation," *Infrared Physics & Engineering / IR*, 2023 (or the journal in OAEPublish). OAE Publish.

[14] M. S. A. Baig, S. A. Gillani, S. M. Shah, M. Aljawarneh, A. Akbar Khan, and M. H. Siddiqui, "AI-based Wearable Vision Assistance System for the Visually Impaired: Integrating Real-Time Object Recognition and Contextual Understanding Using Large Vision-Language Models," *arXiv: 2412.20059*, 2024. arXiv.

[15] I. Tokmurziyev, M. Altamirano Cabrera, M. Haris Khan, Y. Mahmoud, L. Moreno, and D. Tsetserukou, "LLM-Glasses: GenAI-driven Glasses with Haptic Feedback for Navigation of Visually Impaired People," *arXiv: 2503.16475*, 2025. arXiv.

[16] Neel Mani Upadhyay, Aryan Pratap Singh, Ashwin Perti, "eyeRoad – An App that Helps Visually Impaired Peoples," ICICC 2024. https://doi.org/10.2139/ssrn.4825671

[17] X. Zhang et al., "Advancements in Smart Wearable Mobility Aids for Visual Impairment: A Bibliometric Analysis," *PMC*, 2024. PMC.

[18] J. Jocher, A. Chaurasia, and G. Qiu, "YOLOv5: A state-of-the-art real-time object detection system," GitHub Repository, 2020. Available: https://github.com/ultralytics/yolov5

[19] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," arXiv preprint arXiv: 2207.02696, 2022.

[20] G. Jocher, Y. Qiu, and A. Chaurasia, "YOLOv8: Next-generation real-time object detector," Ultralytics Technical Report, 2023. Available: https://github.com/ultralytics/ultralytics

[21] R. S. Mehta and V. Kumar, "Comparative evaluation of YOLOv5, YOLOv7 and YOLOv8 for real-time object detection," Procedia Computer Science, vol. 227, pp. 116–124, 2023. https://doi.org/10.1016/j.procs.2023.03.015

[22] P. A. Parikh, K. D. Joshi and R. Trivedi, "Face Detection-Based Depth Estimation by 2D and 3D Cameras: A Comparison," *2022 28th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nanjing, China, 2022, pp. 1-4, https://doi.org/10.1109/M2VIP55626.2022.10041072