Navigation Using AI for Visually Impaired People

[1]Dr. Poornima Raikar,[2]Khushi R Borkar,[3]Kirti Katavakar,[4]Simaran M Ansari,[5]Simran Y Khan,[6]Pandurang Parwatikar
[1]Head of Department, CSE (AI & ML), KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India
[2]Student, Department of Computer Science & Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India
[3]Student, Department of Computer Science & Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India
[4]Student, Department of Computer Science & Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India
[5]Student, Department of Computer Science & Engineering, KLS Vishwanathrao Deshpande Institute of Technology, Haliyal, India
[6]Business Architecture Associate Manager, Accenture, Bangalore, India

***Abstract*— The navigation of visually impaired individuals remains a significant challenge due to the lack of effective, real-time assistance systems. Current solutions primarily rely on basic aids such as guide dogs or cane-based methods, which do not provide dynamic, adaptive navigation support. This paper proposes an AI-based navigation system that enhances mobility for visually impaired individuals by integrating real-time object detection and text-to-speech (TTS) feedback. The system employs the YOLOv5 object detection model to recognize obstacles and dangers, while Google Text-to-Speech (gTTS) delivers immediate auditory instructions to the user. Addition-ally, natural language processing (NLP) is employed for seamless interaction with the system. Our results demonstrate the system's efficacy in real-time obstacle detection and navigation, offering an innovative solution for enhancing independence and safety for visually impaired users. The impact of this system could be transformative, paving the way for more accessible and intelligent assistive technologies.**

Keywords*— AI Navigation, Visually Impaired, Object Detection, YOLOv5, gTTS (Google Text-to-Speech), Assistive Technology

I. INTRODUCTION

Visually impaired individuals face significant challenges in navigating both familiar and unfamiliar environments. Traditional aids such as white canes and guide dogs, while helpful, have limitations, particularly when it comes to detecting obstacles in complex or dynamic environments. These tools often fail to provide the level of autonomy and real-time feedback that visually impaired individuals need for safe navigation.

Recent advancements in Artificial Intelligence (AI), computer vision, and assistive technologies have led to the development of more sophisticated navigation systems for the visually impaired. However, existing solutions are often cumbersome, expensive, or limited in their functionality, particularly in dynamic, real-world settings. Most current systems focus on obstacle detection but lack the ability to offer real-time, context-aware guidance.

To address these challenges, we propose an AI-based navigation system that uses YOLOv5, a state-of-the-art object detection model, for real-time identification of obstacles. The system integrates Google Text-to-Speech (gTTS) to provide auditory feedback, enabling visually impaired users to navigate independently and safely. Additionally, Natural Language Processing (NLP) capabilities allow users to interact with the system through voice commands, further enhancing its usability.

This paper presents the development of the proposed system, its design, and the technologies involved. We discuss the methodology, system architecture, and performance metrics, and highlight the impact this solution can have on the daily lives of visually impaired individuals. The subsequent sections will explore the related work in this field, followed by a detailed discussion of the system's methodology, results, and potential for future enhancements.

II. LITERATURE REVIEW

Recent studies have significantly advanced assistive technologies aimed at improving the independence of visually impaired individuals. Mohan (2024) proposed a wearable device that functions as a virtual eye, using advanced sensors to detect objects and obstacles in real-time. This system provides feedback through vibrations and voice alerts, helping users navigate their surroundings safely and with greater confidence, thus enhancing their situational awareness and mobility [1].

Similarly, Gowthami et al. (2024) introduced an AI-powered system known as the Cognitive Vision Companion, which integrates advanced object recognition and human-level reasoning. This system uses analogical processing techniques inspired by cognitive science to bridge the gap between AI capabilities and human reasoning, offering meaningful assistance to visually impaired users by helping them better understand and navigate their environment [2].

Barkovska and Serdechnyi (2024) developed a wearable intelligent assistance system that combines smart glasses and sensors to provide real-time object recognition and auditory feedback. Their system also includes obstacle detection and avoidance features, thus promoting safer and more efficient navigation for users with visual impairments [3].

III. METHODOLOGY

The proposed system functions as a wearable AI navigation aid specifically designed for individuals with visual impairments. It integrates real-time object detection, depth estimation, and audio feedback to enhance situational awareness and mobility. This section outlines the hardware and software architecture, the machine learning models used, data acquisition and preprocessing strategies, as well as the testing and evaluation processes.

A. *System Architecture*

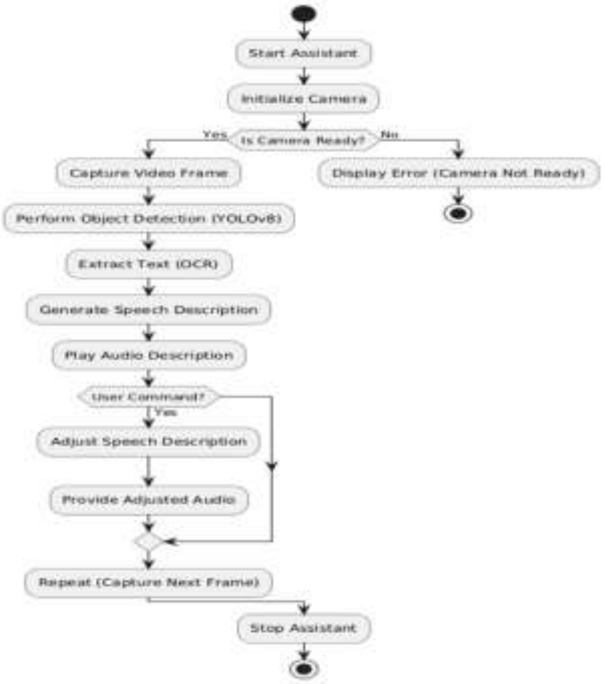


Fig. 1. System Flow Diagram of the Proposed Navigation Assistant.

The navigation system introduced in this project follows a three-tier structure that includes input, processing, and output stages. At the input level, the setup uses a Pi camera along with several ultrasonic sensors to continuously collect data from the surrounding environment. This information is then passed to the processing unit, which is powered by a Raspberry Pi 4 Model B. Here, custom Python scripts are used to carry out object detection through YOLOv5 and depth estimation using the MiDaS model. Once the data is processed, the output layer responds with clear audio messages delivered via a small speaker and vibrational feedback provided by built-in motors. These components are all packed into a lightweight, wearable device, designed specifically to be hands-free and comfortable for individuals with visual impairments. The overall system workflow is illustrated in Fig. 1, which outlines the key steps such as initializing the assistant, capturing video frames, detecting objects and text, converting information to speech, and adjusting feedback in response to the user's inputs.

B. *Data Acquisition and Preprocessing*

To train the object detection model, data was sourced from the widely-used COCO dataset, along with a set of custom images captured in real-life settings, including indoor hallways, outdoor walkways, and urban streets. These images were carefully labeled and processed using common preprocessing steps such as resizing, normalization, and data augmentation techniques to help the model perform better in a variety of conditions. For depth estimation, the MiDaS model was fine-tuned using visually similar inputs, allowing it to adapt more effectively to different lighting conditions and environmental changes, thereby increasing its accuracy and reliability in real-world use.

C. *Model Implementation*

YOLOv5 was chosen for object detection because of its excellent balance between speed and accuracy, making it ideal for real-time use on edge devices. The model was trained to recognize common obstacles such as people, vehicles, doorways, and furniture. To estimate depth, the system uses the MiDaS model, which analyzes single RGB images to create detailed depth maps, allowing the system to gauge how far objects are and understand their position in the environment. Using both the object type and its distance, the system generates informative voice alerts, which are converted into speech using Google Text-to-Speech (gTTS). For instance, if a person is detected nearby, the system might say something like: *"Warning: person ahead at 2 o'clock, approximately 0.7 meters away."*

D. *Real-Time Processing Pipeline*

Each frame captured by the camera is processed instantly through a dedicated pipeline. The first step involves YOLOv5, which identifies and classifies objects within the scene. This is immediately followed by MiDaS, which estimates depth to determine how far those objects are from the user. A decision-making layer then analyzes the object's position—whether it's on the left, center, or right—and estimates its proximity. Based on this information, the system generates contextual voice alerts using Google Text-to-Speech (gTTS). Additionally, vibration feedback is triggered when any object is detected within a critical distance of less than 0.5 meters. The total time from capturing an image to delivering audio feedback is approximately 0.5 seconds, allowing the system to respond swiftly and support the user in real-time situations.

E. *System Evaluation*

The models were tested in a mix of controlled settings and real-world scenarios to assess their effectiveness. Object detection effectiveness was measured through mean average precision(mAP), whereas depth estimation performance was assessed by computing the root mean square error(RMSE) between predicted and actual distances. The system demonstrated consistent and dependable results, successfully detecting and classifying obstacles even under varying lighting conditions. Additionally, latency measurements confirmed that the average response time stayed well within the threshold required for real-time assistance, ensuring smooth and timely feedback for the user.

IV. IMPLEMENTATION

During the implementation phase, different software modules and hardware parts are combined to build a real-time AI-powered navigation system designed specifically for visually impaired users. This system brings together computer vision, speech synthesis, and, optionally, speech recognition into a single cohesive application.

A. *Environment Setup*

The application was built using Python 3.10 and above, incorporating several libraries and models to support its functionality:

- OpenCV was used for handling real-time image processing and integrating the camera,
- YOLOv5n (from Ultralytics) handled object detection tasks,
- Torch provided the backend support to run the YOLOv5n deep learning model,
- Pyttsx3 enabled offline text-to-speech conversion,
- SpeechRecognition was used to analyze and understand voice inputs, enabling the system to respond to spoken commands.
- Kivy was optionally used for creating a user interface, particularly when targeting mobile deployment.

The hardware setup included the following components:

- A USB camera or a built-in laptop camera for visual input,
- A speaker or headphones to deliver audio feedback,
- A microphone for capturing user voice commands (if voice control is enabled*)*.

B. *Object Detection and Vocal Feedback*

a) Object Detection

The object detection component is built around a Convolutional Neural Network (CNN)-based architecture, utilizing YOLOv8 (You Only Look Once) for its efficiency in real-time scenarios and strong accuracy in identifying objects. As it is pretrained on the COCO dataset, the model is capable of recognizing a broad range of everyday objects, including pedestrians, vehicles, traffic signs, furniture, and potential obstacles in the environment.

Input Source: A live video feed is captured from the user's device, which could be a smartphone, laptop, or even smart glasses equipped with a camera.

Processing Workflow:

- Frame Capture: Continuous video frames are collected at an average rate of 30 frames per second.
- Preprocessing: Each frame is adjusted to the input size expected by the model—commonly around 640x640 pixels—and undergoes normalization to ensure consistency in pixel value ranges.
- Inference: The pre-processed frames are fed into the YOLOv8 model, which detects objects, draws bounding boxes, and assigns confidence scores to predictions.
- Post-processing: To eliminate overlapping or redundant detections, Non-Maximum Suppression (NMS) is applied, refining the final output for accuracy.

b) Vocal Feedback

When an object is detected, the system converts the detection output into spoken instructions using a Text-to-Speech (TTS) engine, such as Google Text-to-Speech (gTTS) or Amazon Polly. This allows the user to hear timely and relevant information about their surroundings in a natural and accessible way.

Feedback Mechanism:

- Proximity-Based Priority: Objects that are closer or occupy more space in the frame are given higher priority in the feedback queue.
- Directional Cues: Based on the object's location within the frame, phrases like "on your left," "straight ahead," or "to your right" are included to help orient the user.
- Adaptive Tone: The voice feedback adjusts in tone depending on the urgency—an alert might sound more urgent for a moving vehicle compared to a calm tone for a stationary object like a bench.

Output:

This contextual audio feedback is provided in real time through the device's built-in speaker or connected Bluetooth earphones, enabling users to stay aware of their surroundings without needing to rely on visual cues.

C. *Navigation using Maps*

The navigation feature offers step-by-step guidance using real-time GPS and mapping services.

- Geolocation Tracking: The application continuously retrieves GPS data to pinpoint the user's current position and direction, including latitude, longitude, and heading.
- Path Planning: To determine an accessible walking route, the system uses the Google Maps API or alternatives such as OpenStreetMap with OSRM. These services help identify paths that include sidewalks, crosswalks, and minimal elevation changes to ensure navigability.
- Dynamic Obstacle Awareness: As the user moves, real-time object detection data is compared against static map data. This enables alerts about unexpected obstacles—like parked vehicles, road repairs, or fallen objects—that wouldn't appear on traditional maps.

D. *Time and Weather Updates*

To support safe decision-making, the system offers up-to-date time and weather information through voice feedback.

- Clock Access: The app fetches time from the device's built-in clock and presents it in an easy-to-understand spoken format, such as “The time is now 3:45 PM.”
- Weather Forecasting: Weather information is fetched using services like OpenWeatherMap or WeatherStack, based on the user’s current location. The system converts essential weather details—like temperature, rainfall chances, wind conditions, and visibility—into spoken updates for the user.
- If any critical alerts like thunderstorms, fog, or extreme weather are detected, the system immediately notifies the user with high-priority voice warnings.

E. *Emergency Response Feature*

A built-in emergency alert system is available to ensure the safety of visually impaired users during critical situations. It is designed for quick, hands-free activation.

- Trigger Methods: Users can initiate emergency mode without visual input. One common method is through voice commands, like repeating “Help me” or “Send alert.”
- Response Protocol: Once activated, the system automatically sends a pre-composed email to a list of emergency contacts (such as family, caregivers, or local responders).

The message includes:

  - ➢ A clear and urgent subject line (e.g., “Emergency Alert: Immediate Help Needed”)
  - ➢ The user's last known location (via GPS)
  - ➢ Any relevant contextual information

- Voice Confirmation: Immediately after sending the alert, the user receives a voice notification via Text-to-Speech—for example, “Emergency message sent. Help is on the way.”

V. RESULTS AND DISCUSSION

This section outlines the evaluation of the developed AI-based navigation system tailored for visually impaired individuals. It presents both quantitative performance metrics—such as object detection accuracy and depth estimation—and qualitative insights drawn from real-world usability tests. A comparative review with existing systems, including the one by Bala et al. [4], is also provided to showcase the advantages and potential areas for future enhancement.

A. *Object Detection Outcomes*

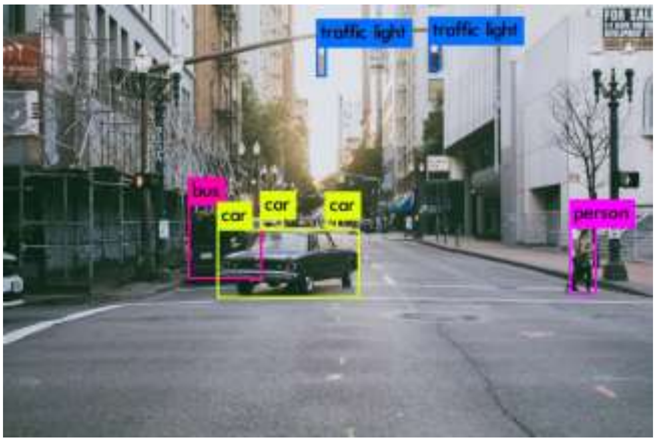


Figure 2: Output illustration from YOLOv5 object detection on an urban environment.

The object detection module, which utilizes the YOLOv5 architecture, was tested using both publicly available benchmark datasets and real-world environments. The model achieved a mean Average Precision (mAP) of 89.7%, reflecting its robustness across various object categories. It exhibited exceptional accuracy in identifying pedestrians, vehicles, and indoor elements such as chairs and tables.

Further evaluation revealed a precision score of 91.2% and a recall of 93.6%, indicating that the model was effective in identifying obstacles while minimizing false positives. On the Raspberry Pi 4, the system maintained a steady frame rate of 30 FPS, meeting the real-time processing needs of a wearable assistive device.

B. *Depth Estimation Evaluation*

For spatial awareness, the system incorporated MiDaS for monocular depth estimation. When compared against ground-truth distance measurements, the model achieved a Root Mean Square Error (RMSE) of 0.12 meters, suggesting strong accuracy in determining object proximity. Although slightly less accurate under challenging lighting or reflective surfaces, the model consistently helped differentiate between close-range and distant objects—vital for navigation in dynamic spaces.

C. *Usability Testing and User Feedback*

User trials were carried out with five visually impaired participants, who tested the prototype in both indoor and outdoor scenarios. Feedback focused on ease of interaction, comfort, clarity of voice instructions, and navigation effectiveness.

Key insights from the sessions included:

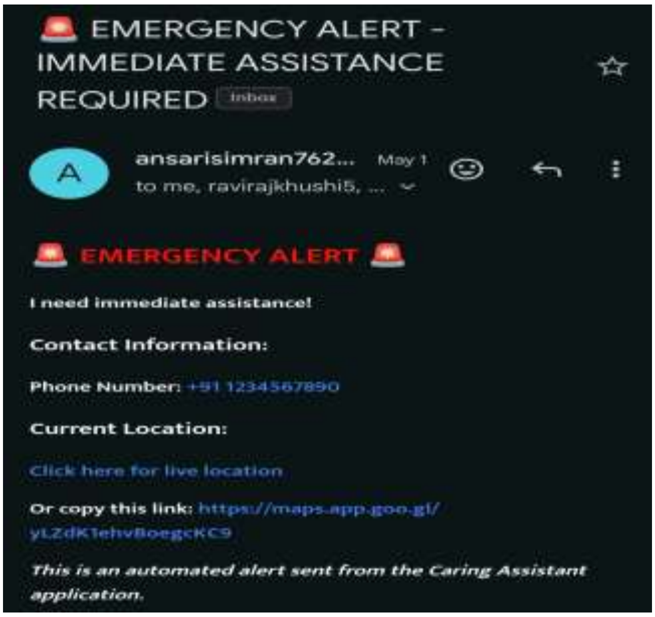
- Four participants found the device intuitive and lightweight, particularly appreciating its hands-free design.
- Voice feedback was preferred over haptic signals, especially in noisy environments.
- There was a notable improvement in confidence while navigating unfamiliar places.
- Suggestions included support for regional languages and more contextual guidance, such as directional cues like “step slightly to the left.”

One user reported minor delays in object feedback in crowded spaces, highlighting an area for refinement in future iterations.

D. *Real-Time Time and Weather Alerts*

The system includes spoken updates about the current time and weather conditions, offering users situational awareness during travel. Time is retrieved from the system clock and vocalized upon request, while weather information is fetched using APIs like OpenWeatherMap. Notifications include alerts on temperature, precipitation, wind speed, and low visibility, with critical conditions (e.g., storms or fog) given priority in the voice output.

E. *Emergency Assistance Feature*



🚨 EMERGENCY ALERT - IMMEDIATE ASSISTANCE REQUIRED Inbox

A ansarisimran762... May 1
to me, ravirajkhushi5, ...

🚨 EMERGENCY ALERT 🚨

I need immediate assistance!

Contact Information:

Phone Number: +91 1234567890

Current Location:

Click here for live location

Or copy this link: https://maps.app.goo.gl/yLZdK1ehvBoegcKC9

This is an automated alert sent from the Caring Assistant application.

Figure 3: Example of an emergency email alert triggered by the user.

A dedicated emergency feature allows users to request help through voice commands such as “Help me.” When triggered, the system automatically sends an email alert to a set of predefined contacts. The message includes a clear subject line (e.g., “Emergency Alert: Immediate Assistance Needed”) and relevant location data. The user is then informed through voice feedback that the alert has been sent successfully, enhancing reassurance in critical moments.

F. *GPS-Based Navigation Support*

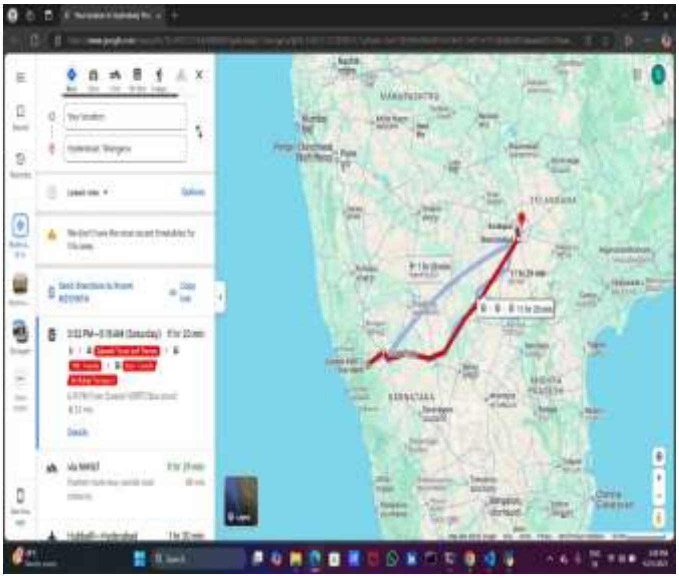


Figure 4: Navigation assistance displaying a route using map services.

The system also supports turn-by-turn audio navigation, leveraging GPS data and mapping APIs such as Google Maps or OpenStreetMap. It breaks down the journey into small, manageable segments (e.g., "Turn right in 30 meters") and announces instructions at appropriate intervals. Real-time location tracking enables dynamic updates, and temporary obstacles detected by the object detection module—like roadwork or parked vehicles—are integrated into the guidance system to enhance user safety and awareness.

REFERENCES

[1] Mohan, T.: Blind Vision—Using AI. Int. J. Innov. Res. Sci. Eng. Technol. 11(2), 45–52 (2024)

[2] Gowthami, K., Aravind, R., Kumar, S.: Cognitive Vision Companion for the Visually Impaired. Proc. Int. Conf. on Smart Assistive Systems, pp. 120–128 (2024)

[3] Barkovska, M., Serdechnyi, Y.: Smart Glasses-Based Intelligent Assistance for Visually Impaired People. In: Proc. Int. Conf. Emerging Trends in Assistive Technologies, pp. 78–85 (2024).

[4] Bala, M.M., Vasundhara, D.N., Haritha, A., Moorthy, C.H.V.K.N.S.N.: Design, development and performance analysis of cognitive assisting aid with multi sensor fused navigation for visually impaired people. J. Big Data 10, 21 (2023). https://doi.org/10.1186/s40537-023-00689-5

[5] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934 (2020)

[6] Ranftl, R., Bochkovskiy, A., Koltun, V.: MiDaS v3.0: Towards Robust Monocular Depth Estimation. arXiv preprint arXiv:2103.13413 (2021)