

An Integrated IoT-Enabled Smart Stick for Visually Impaired Navigation

Mai Samir Mohamed^{1*}, Mohamed Kayed², Al Hussien Seddik Saad³, Marwa Hashem Shehata⁴

Mathematics and Computer Science Department-Faculty of Science, Beni-Suef University, Beni-Suef, Egypt^{1,4}

Faculty of Computers and Artificial Intelligence, Beni-Suef University, Beni-Suef, Egypt²

Computer Science Department-Faculty of Science, Minia University, Al Minya, Egypt³

Abstract—Visual impairment is a profound global health challenge that significantly impacts the quality of life, independence, and safety of millions of individuals. This paper presents an integrated IoT-enabled smart stick for visually impaired people to navigate their surroundings, with sensing modes for enclosed and open spaces. The proposed smart stick system consists of six subsystems: 1) a light detection subsystem using an LDR sensor; 2) an obstacle detection subsystem using IR and ToF sensors; 3) an object recognition subsystem using a Pi Camera Module and YOLOv8n; 4) an alert management subsystem that manages all warning signals generated by the other subsystems; 5) an embedded processing unit based on a Raspberry Pi 4 Model B and a Raspberry Pi Pico; and 6) an IoT-based connectivity subsystem that connects the sensing and processing modules to a remote server. Furthermore, the user is signalled about various obstacles using different types of alerts based on obstacle distance, object type, and environmental conditions. Preliminary experimental results obtained in an indoor test room indicate good performance. In a room with desks, chairs and a fan, ten participants tested the stick; the system achieved an obstacle detection accuracy of 97.0% and an object recognition accuracy of 96.5%. Outdoor operation was not evaluated. Furthermore, the overall response time estimated from component-level measurements was below 250 ms, which supports near real-time assistance for visually impaired users. The prototype integrates all sensing, processing and feedback components in a single stick.

Keywords—Assistive technology; smart stick; visually impaired; YOLOv8; Raspberry Pi; Time-of-Flight (ToF); distributed computing

I. INTRODUCTION

Visual impairment is a profound global health challenge that significantly impacts the quality of life, independence, and safety of millions of individuals. According to the World Health Organization (WHO), a substantial portion of the global population suffers from vision loss, creating an urgent demand for effective Assistive Technologies (AT) [20]. While the traditional white cane remains the most common mobility aid, it suffers from inherent limitations: it can only detect obstacles upon direct contact or within a very short range, and it fails to identify the nature of the obstacle (e.g., distinguishing a person from a wall) or environmental hazards like head-level obstacles.

Traditionally, the white cane has been the most widely used mobility aid for visually impaired individuals due to its simplicity and affordability. However, despite its widespread use, the traditional cane suffers from several inherent limitations.

It can only detect obstacles within a very short range or after direct physical contact, and it cannot identify the type of obstacle or recognize dangerous environmental conditions such as head-level obstacles, stairs, holes, or moving objects. Consequently, researchers have focused on developing intelligent navigation systems capable of improving user safety and mobility.

Recent advances in the Internet of Things (IoT), Artificial Intelligence (AI), and embedded systems have enabled the development of smart assistive devices for blind people. IoT technology allows sensors, processing units, cloud services, and mobile applications to communicate efficiently in real time. A typical IoT system consists of four main modules: data gathering, data storage, communication, and data processing. Sensors are used to collect environmental data, while communication technologies facilitate information exchange between hardware devices, databases, and mobile applications. These technologies have significantly contributed to the emergence of “Smart Sticks” and wearable assistive systems designed to support safe navigation.

Several studies, surveyed in [20], have proposed different approaches for assisting blind individuals. Some researchers developed wearable smart glasses combined with intelligent walking sticks to improve obstacle detection and navigation [1]. Other systems [2] relied on computer vision techniques, ultrasonic sensors, laser sensors, GPS-GSM modules, and beacon-based localization systems to detect surrounding obstacles and guide users safely. In addition, modern object detection techniques such as YOLO [17],[18] have been integrated into assistive systems [3] to recognize objects and analyze captured images in real time.

The development of smart navigation aids has been a subject of extensive research. Various approaches have been proposed, utilizing different combinations of sensors and processing units. Our proposed system presents a novel multi-functional smart stick based on a distributed computing architecture that aims to improve environmental awareness, response speed and user comfort. The proposed system detects obstacles, recognizes objects, and warns the user about hazards, using a close-range (IR) mode for enclosed spaces and a long-range (ToF) mode for open spaces; only indoor operation was evaluated in this study. indoor and outdoor environments.

The main contributions of this study are summarized as follows:

*Corresponding author.

1) *Hybrid distributed processing architecture*: The proposed system introduces a dual-layer processing framework in which a Raspberry Pi Pico is responsible for low-level sensor fusion and immediate haptic feedback (Reflex Layer), while a Raspberry Pi 4 Model B is dedicated exclusively to artificial intelligence inference and object recognition tasks (Cognitive Layer). This separation is intended to reduce computational overhead and system latency; a controlled comparison with a single-board configuration is reported in Section V.F.

2) *Efficient AI-Based object recognition*: The system employs the lightweight YOLOv8n model for real-time object detection, enabling environmental perception while maintaining efficient computational performance suitable for embedded systems.

3) *High-precision distance measurement*: Time-of-Flight (ToF) laser sensors are integrated to provide accurate and fast obstacle distance estimation, improving navigation safety and obstacle awareness for visually impaired users.

4) *Hardware-based ambient light detection*: Unlike conventional software-dependent approaches, the proposed design implements a pure hardware logic circuit for the LDR subsystem. This enables instantaneous ambient light detection without consuming microcontroller processing cycles, thereby improving response time and overall energy efficiency.

5) *Low-latency distributed processing*: By offloading sensor management and low-level processing tasks to the Raspberry Pi Pico, the proposed system reduces the computational burden on the main processor, allowing the Raspberry Pi 4 to focus exclusively on AI inference. This distributed architecture is designed to minimize system latency and to enhance real-time responsiveness.

6) *Integrated lightweight design*: The proposed system eliminates the dependency on external wearable modules by embedding all sensing, processing, and feedback components directly within the smart stick. This design enhances usability, minimizes user discomfort, and improves portability while targeting a lightweight form factor (the measured weight of the prototype is given in Section IV.A.3).

Following the introduction, the rest of this paper is organized as follows: Section II describes the related work. Our proposed system is introduced in Section III. Section IV describes the experimental setup, and Section V presents the experimental results and analysis. The conclusion and future work are presented in Section VI.

II. RELATED WORK

Visually impaired people face many difficulties in their daily routine. They face difficulties while navigating, and they cannot move freely in an environment known or unknown because of the lack of sight. They need support to know if they are going in the right direction or not, whether they are going to hit obstacles, or if there is water in the way, if the environment is dark or not, etc. Thanks to the rapid technological advancements in both hardware and software, obstacles can now be identified automatically. It is challenging for visually impaired people to walk comfortably and without any fear without human help in

urban or ambiguous environments. To travel independently, they usually use a cane or a guide dog. Smart tools such as sticks and glasses can help visually impaired people navigate the city during their day-to-day movement. Building these types of smart tools would require the utilization of ultrasonic, passive infrared (PIR) sensors, water sensors, vibration motors, camera, processors, a speaker with voice messages, and also GPS module helpful to determine the current location of the users. In this regard, several approaches and navigation techniques have been presented in this section to aid blind people in avoiding obstacles along their path.

Rahman and Sadi [4] proposed an IoT-enabled system for object and currency recognition using a spectacle-based design. While their system utilized laser sensors and cameras effectively, the reliance on smart glasses introduces ergonomic challenges. Users often report discomfort and fatigue during prolonged use due to the added weight on the head. Annamalai et al. [5] utilized a Raspberry Pi 4B to manage both peripheral sensor data and heavy AI computations using TensorFlow. Such monolithic architecture often suffers from processing bottlenecks, leading to high system latency and delayed feedback. In contrast, Paul et al. [16] developed a multifunctional stick using an Arduino Mega and ultrasonic sensors. However, ultrasonic sensors are prone to environmental noise and have wide beam angles that lack precision in complex layouts. Jeeva et al. [6] proposed wearable glasses and a walking assistant design based on ultrasonic sensors. Chibambararaj et al. [7] also used an ultrasonic sensor to detect obstacles and alert blind people via a smartphone.

According to Adil et al. [8], an Arduino Uno Microcontroller (AUM) and ultrasonic sensors are integrated with a voice module to guide blind people. The smart stick proposed in [9] can detect obstacles within 4 meters based on an ultrasonic sensor and a vibrator motor. Also, Khan [10] proposed a stick that can discover obstacles using an ultrasonic sensor and then send a signal to a buzzer and a vibrator. A wireless bracelet proposed in [11] can guide a blind person to the optimal position. This system employs ultrasonic sensors to detect various obstacles in an indoor environment. It also makes use of the Firebase database and NodeMCU WiFi module to enable real-time communication with a blind individual. An intelligent cane designed using SolidWorks 3D modelling software was proposed in [12]. It detects water, obstacles, and fire. This smart stick employs ultrasonic sensors and an Arduino module. Farooq et al. [13] proposed an IoT-enabled smart stick that used ultrasonic sensors for obstacle detection, a water sensor for sensing the puddles and wet, a high-definition video camera for object recognition, and earphones for voice alert. The smart stick has two modes; one uses ultrasonic sensors for detection and feedback through vibration motors, and the second mode is the detection and recognition of obstacles through voice feedback. They [13] also used GPS for mobile communication.

Most of the research makes use of ultrasonic sensors for detecting obstacles, vibration, beeps, or audio messages for alerting users, water sensors for detecting water, and GPS in combination with the main system. Additionally, it is important to know that the GPS receiver is helpful to determine the current location of visually impaired people. Such a system was implemented by Kunta et al. [14], who presented a smart stick

for helping visually impaired people by integrating distance sensors, speakers for alerts, an Arduino processor, and GPS and GSM modules for sending a message to the caretaker of the user on their mobile phone by getting coordinates from the GPS module and sending them through the GSM module. They also developed a simple user interface that allows the user to change the mobile number for the message. The study in [15] is similar to [14], but it uses a Raspberry Pi instead of an Arduino, since the Raspberry Pi is better suited to the use of multiple sensors.

In summary, most of the reviewed systems rely on ultrasonic ranging with vibration, buzzer or voice output on low-cost microcontrollers [8]–[10], [12], [16]; fewer systems add cameras [4], [5], [13] or GPS/GSM modules [13]–[15]; and several are wearable [4], [6]. Their common limitations are (a) a single ranging modality that cannot recognize the type of obstacle, (b) a monolithic design in which one board performs sensing, inference and alerting [5], and (c) the absence of a reproducible quantitative evaluation protocol. The proposed system addresses (a) and (b) by combining IR, ToF and camera sensing on a two-processor architecture. A feature-by-feature comparison of representative works [4], [5], [13], [14], [16] with the proposed stick is given in Table VII (Section V.E); the figures in that table are those reported in the original papers and were obtained with different sensors and metrics, so Table VII compares features, not performance.

III. PROPOSED SYSTEM

Our system presents a novel IoT-enabled smart stick for visually impaired people to navigate their surroundings with its ability to detect and warn about obstacles. It integrates multiple sensing technologies, embedded artificial intelligence, and cloud-based connectivity to provide real-time environmental awareness, obstacle avoidance, and remote monitoring. The system architecture consists of six major subsystems: Light Detection, Obstacle Detection, Object Recognition, Alert Management, Embedded Processing Unit, and IoT-Based Connectivity; the embedded processing unit coordinates all the other subsystems. Fig. 1 shows the block diagram of the proposed smart stick.

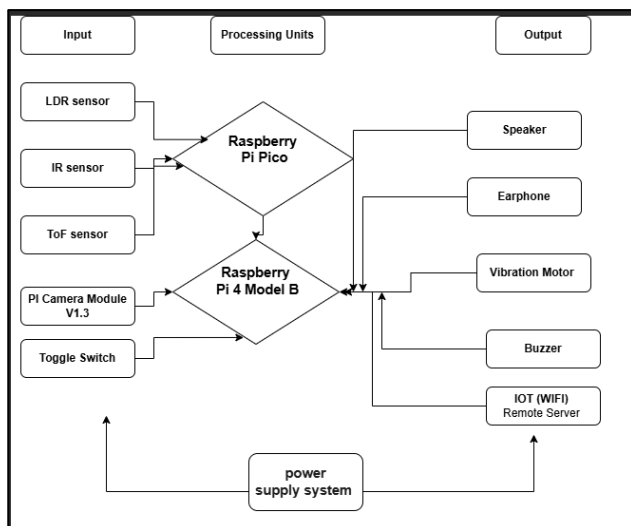


Fig. 1. Block diagram of the proposed smart stick.

A. System Architecture

1) *Light detection subsystem*: The light detection subsystem assists visually impaired users in recognizing changes in ambient lighting conditions. It helps the user distinguish between day and night in outdoor environments and determine whether lights are ON or OFF indoors. The proposed system utilizes an LDR (Light Sensitive Relay Module, 5V) sensor to monitor surrounding light levels. A Buzzer Alarm (3–24V, 95 dB) is integrated to alert the user when the light level falls below a preset threshold, indicating darkness. This buzzer produces a loud and clear audible signal (operated at 12 V in our prototype) suitable for dark environments. Additionally, a manual toggle switch is included to allow the user to manually turn the subsystem ON or OFF as needed.

2) *Obstacle detection subsystem*: Obstacle detection is the cornerstone of our system. To ensure safe navigation and avoid collisions, we employ infrared (IR) sensors alongside laser-based Time-of-Flight (ToF) sensors. The IR sensor (a 4-pin module) is used to detect very close obstacles—specifically within a range of 4 to 15 cm. It is mounted on the lower part of the cane to identify ground-level hazards, such as furniture, walls, or the edge of a staircase. When an object enters this 15 cm range, the Raspberry Pi Pico microcontroller triggers a 5V relay, which in turn activates a vibration motor, providing a strong alert to indicate an imminent danger. Meanwhile, the Time-of-Flight sensor (model TOF10120) is used to detect obstacles at medium to long distances; according to the manufacturer, it has a range of up to 4 meters, making it well-suited for open environments like streets. The system operates in two modes, selected based on the surrounding environment: in "short-range mode"—intended for indoor settings such as a home—the IR sensor handles the detection of nearby obstacles, whereas in "open-space mode," the Time-of-Flight sensor monitors for obstacles at distances of up to 4 meters. It should be noted that the stated sensing ranges are those specified by the manufacturers; since individual sensors are not calibrated separately, the actual ranges may be slightly shorter (Section IV.A.3). In "short-range mode," the warning range is limited (a 35 cm range corresponds to approximately 0.3 seconds when moving at a speed of 1.2 m/s)—a limitation discussed in Section V.G. The sensor measures distance by calculating the time it takes for a light pulse to travel to an object and back, in accordance with Eq. (1) [21]:

$$\text{Distance} = (S \times t) / 2 \quad (1)$$

where, S is the speed of light and t is the elapsed time. The measured data is sent to the Raspberry Pi Pico via a digital interface. When an obstacle is detected within the 15–35 cm range, the system activates the vibration motor with lower vibration feedback, indicating a medium-risk level and allowing the user more time to react.

3) *Object recognition subsystem*: The Raspberry Pi 4 Model B and Pi Camera Module v1.3 are integrated into the object recognition subsystem to recognize various objects in the target environments. The proposed system recognizes the

object categories supported by its detection model, e.g., chairs, vehicles, and pedestrians (see “Detection model” below). The comprehensive process of this subsystem is detailed in the following steps:

a) Image capturing: The Pi Camera Module v1.3 is utilized to capture real-time images from the surroundings. In the proposed system, the camera is mounted on the front side of the stick to provide a forward field of view [(mounting height 80 cm; downward tilt 15°; horizontal field of view 90°)]. Because of this placement, overhead (head-level) obstacles lie outside the evaluated detection scope (Section V.G). Each real-time instance is captured as a frame; to optimize the processing cycle and ensure low latency, a fixed resolution of 640 × 480 pixels has been implemented. Subsequently, the captured frame is transmitted to the feature extraction module for further analysis

b) Feature extraction: A pretrained YOLOv8n model [17] is utilized to extract features from the input frames. The 'Nano' version was specifically selected to maintain an optimal balance between detection accuracy and inference speed on the embedded CPU.

Detection model. the camera branch uses the COCO-pretrained YOLOv8n checkpoint (80 classes) without fine-tuning. The classes used by the system are: person, chair, car, bus, truck, motorcycle, bicycle (add or remove). Stairs and walls are not COCO classes; they are detected only by the IR/ToF sensors, not by the camera. The classes are mapped to the hazardous set H_{haz} and the informational set H_{info} used by the algorithm of Section III.C.

c) Audio feedback generation: This final stage of the Object Recognition subsystem translates YOLOv8n outputs into spoken alerts. Using a Text-to-Speech (TTS) engine, detected class labels are converted into clear voice commands. To ensure user comfort and environmental adaptability, the system supports dual output modes: an integrated speaker for private environments and earphones via the Raspberry Pi audio jack for noisy public spaces. This auditory feedback works in synchronization with the vibration motors and buzzers managed by the Alert Management Subsystem, providing a redundant safety mechanism that prevents sensory overload while ensuring that the user does not miss a critical alert.

d) Alert management subsystem: The Alert Management Subsystem is responsible for managing all warning signals generated by the smart stick's subsystems. It is distributed over the two processors (see the allocation below) and manages alerts from the obstacle detection (IR & ToF), object recognition (YOLOv8n), and light detection (LDR) modules. Based on obstacle distance, object type, and environmental conditions, the system dynamically selects the most appropriate feedback modality, including:

- Strong vibration for immediate hazards.
- Low vibration for medium-distance obstacles.
- Voice alerts for recognized hazardous objects.

To avoid unnecessary sensory overload, the subsystem activates multiple alert modalities simultaneously only during critical situations involving multiple hazards.

Allocation of responsibilities: 1) The Raspberry Pi Pico acquires the IR, ToF and LDR readings, classifies the proximity risk with the thresholds of Table II, and drives the vibration motors directly (reflex layer), so that haptic alerts do not depend on the Raspberry Pi 4, the camera or the network. 2) The Raspberry Pi 4 runs YOLOv8n inference, merges the proximity risk received from the Pico with the camera result, generates the voice alerts (TTS), drives the low-light buzzer and logs the data. 3) When the two processors report different risk levels, the higher level is applied (Section III.C). The communication protocol and the behaviour at start-up, shutdown and subsystem failure are described in Section III.D.

4) Embedded processing unit: The central processing architecture of the proposed smart stick consists of Raspberry Pi 4 Model B and Raspberry Pi Pico, which collaboratively manage sensor acquisition, intelligent processing, and communication tasks. The Raspberry Pi 4 serves as the primary edge-computing unit responsible for image processing, object recognition, alert prioritization, and IoT communication. Meanwhile, the Raspberry Pi Pico [19] handles low-level real-time sensor interfacing, including obstacle and light detection, ensuring fast response and reduced computational load on the main processor. This distributed architecture is designed to improve system efficiency, reduce latency, and enable seamless integration between sensing, artificial intelligence, and cloud services.

5) IoT based connectivity subsystem: To enhance system functionality and enable remote monitoring, the proposed smart stick incorporates an Internet of Things (IoT) framework that connects all sensing and processing modules to a remote server. The Raspberry Pi 4 acts as the IoT gateway, collecting processed data from the obstacle detection, object recognition, and light detection subsystems. The collected data is transmitted through Wi-Fi connectivity to a cloud database for storage and monitoring purposes.

The IoT subsystem provides:

- Real-time sensor data logging.
- System status monitoring.
- Event history storage.
- Cloud-based data management.

In the current prototype the cloud connection is used only for logging and remote monitoring; no alert decision depends on it. Records are first written to the local SQLite database and are uploaded in batches when Wi-Fi is available, so a network or server outage delays the upload but does not interrupt local alerting.

B. System Workflow

The workflow of the proposed smart stick integrates sensing, embedded artificial intelligence, alert management, and IoT

communication into a unified assistive system. The operational sequence is described as follows:

Step 1: Power Initialization: The power supply subsystem provides stable electrical power to all hardware components. During laboratory testing, 220V AC power is converted into 12V DC using an SMPS module, then regulated to 5V through a buck converter to power the Raspberry Pi units, sensors, and peripheral modules.

Step 2: Sensor Data Acquisition: The Raspberry Pi Pico continuously acquires readings from:

- IR sensor (4–15 cm range)
- ToF sensor (15–35 cm range)
- LDR light sensor

The Pico preprocesses the collected sensor readings and transmits them to the Raspberry Pi 4 through a UART serial link, using the message format and error-handling rules described in Section III.D.

Step 3: Image Capturing: The Pi Camera Module v1.3 captures real-time environmental images at a resolution of 640×480 pixels and forwards the captured frames to the Raspberry Pi 4 for processing.

Step 4: Object Recognition: The Raspberry Pi 4 executes the YOLOv8n model to detect and classify surrounding objects such as stairs, vehicles, chairs, walls, and pedestrians.

Step 5: Risk Assessment: The Alert Management Subsystem integrates sensor readings and object recognition outputs to determine the environmental risk level.

- High Risk Conditions.
- Obstacle detected within the IR threshold (≤ 15 cm).
- Hazardous objects detected by YOLOv8n (e.g., stairs or vehicles).
- Low-light environment detected by the LDR sensor.
- Medium Risk Conditions.
- Obstacles detected only within the ToF sensing range (15–35 cm).
- Recognized non-hazardous objects (e.g., chairs).

Step 6: Alert Generation: Based on the detected risk level, the system dynamically generates the appropriate feedback response as follows:

- High Risk
- Strong vibration
- Voice alert
- High-sound buzzer activation
- Medium Risk
- lower vibration feedback
- Short voice prompt (e.g., “Obstacle ahead”)

Step 7: Data Logging and IoT Synchronization: All sensor readings, detected object labels, risk levels, and generated alerts are stored locally in an SQLite database on the Raspberry Pi 4. The Raspberry Pi 4 then transmits the stored data to a remote cloud server through Wi-Fi connectivity using HTTP requests for real-time monitoring and cloud synchronization.

C. Proposed Algorithm

Algorithm 1 summarizes the control logic of the proposed smart stick. It runs in a continuous loop that has four layers. In the sensor-collection layer, the Raspberry Pi Pico reads the IR distance D_{IR} , the ToF distance D_{ToF} and the ambient light level L , and sends them to the Raspberry Pi 4. In the vision-processing layer, the Raspberry Pi 4 captures a 640×480 frame I_{frame} and runs YOLOv8n to obtain the recognized class C . In the risk-assessment layer, the risk level is reset to Safe at the beginning of every cycle and is then raised according to the rules of Table II: an IR distance of at most 15 cm, a hazardous object class or a low light level gives High risk, whereas a ToF distance between 15 and 35 cm or a non-hazardous recognized object gives Medium risk only if no High-risk condition is present. When the sources disagree, the most conservative (highest) risk level is applied, and a camera prediction older than T_{stale} is discarded so that stale detections never trigger alerts. The selected risk level determines the feedback (strong or lower vibration, voice, buzzer), and several modalities are combined only in critical situations. Finally, in the logging layer, every cycle record $\{D_{IR}, D_{ToF}, L, C, Risk\}$ is stored in a local SQLite database and uploaded to the cloud server whenever Wi-Fi is available. The behaviour at start-up, in case of sensor or link failure, and at shutdown is described in Section III.D.

Algorithm 1

Input:

$DIR \rightarrow$ IR sensor distance
 $D_{TOF} \rightarrow$ ToF sensor distance
 $L \rightarrow$ Light intensity from LDR
 $I_{frame} \rightarrow$ Captured image frame
 $H_{haz}, H_{info} \rightarrow$ hazardous / informational object classes
 $T_{stale}, T_{sensor} \rightarrow$ maximum age of a camera prediction / of a sensor reading

Output:

Vibration Alerts
Voice Alerts
Buzzer Alerts
IoT Cloud Synchronization
Initialize system power
Initialize communication between:
- Raspberry Pi Pico
- Raspberry Pi 4

$Risk \leftarrow Safe; C \leftarrow None;$ run start-up self-test (Section III.D)

WHILE (System == ON)

$Risk \leftarrow Safe$ // reset at the start of every cycle
// Sensor collecting Layer
Read DIR from IR sensor
Read D_{TOF} from ToF sensor
Read L from LDR sensor
IF (a reading is missing or invalid for longer than T_{sensor})
Mark that sensor FAULT, announce it once, ignore it in the rules
END IF
Transmit sensor readings

```
from Raspberry Pi Pico
to Raspberry Pi 4
// Vision Processing Layer
Capture Iframe using Pi Camera
(640 × 480 resolution)
Execute YOLOv8n inference on Iframe
Recognize object class C
IF (age of C > T_stale) THEN C ← None // discard stale
prediction
// Risk Assessment Layer
IF (DIR ≤ 15 cm)

    Risk = High
    Activate strong vibration motor
END IF
IF (C belongs to hazardous object classes H_haz)
    Risk = High
    Generate voice alert
END IF
IF (C ∈ H_info AND Risk != High)
    Risk = Medium
    Generate short voice prompt
END IF
IF (L < predefined threshold)
    Risk = High
    Activate high-sound buzzer
END IF
IF (DToF ≥ 15 cm AND DToF ≤ 35 cm)
    AND (Risk != High)
    Risk = Medium
    Activate lower vibration feedback
END IF
IF no hazard detected
    Risk = Safe
END IF
// Multiple Hazard Management
Multiple alerts may operate simultaneously
when multiple critical conditions occur.
Contradictory sources → apply the highest Risk
// Data Logging & IoT Synchronization
Store:
{DIR, DToF, L, C, Risk}
in local SQLite database
on Raspberry Pi 4
IF (Wi-Fi connection is available)
    Retrieve stored records from SQLite
    Send data packets to remote server
    using HTTP requests
END IF
END WHILE
```

D. Communication Protocol, Operating States and Fault Handling

1) *Pico–Pi 4 communication protocol*: The Raspberry Pi Pico and the Raspberry Pi 4 communicate over a UART serial link at [115200 baud, 8N1]. The Pico sends one ASCII frame per sampling cycle, in the form [SEQ,D_IR,D_ToF,L,FLAGS*CHK], where SEQ is a rolling sequence number, D_IR and D_ToF are distances in cm (−1 when no valid reading is available), L is the light level, FLAGS reports the status of each sensor, and CHK is an XOR checksum. The Pico samples the IR, ToF and LDR sensors at [

20 Hz] and transmits at [20 Hz]; the camera pipeline runs at [15 frames/s]. Synchronization: each camera prediction is time-stamped by the Raspberry Pi 4 and paired with the latest valid sensor frame that differs by less than [100 ms]. Error handling: frames with a wrong checksum or a non-consecutive SEQ are discarded and counted. Recovery: if no valid frame arrives for [500 ms], the Raspberry Pi 4 declares the link failed, announces it by voice, and tries to re-open the serial port every [2 s]; meanwhile the Pico keeps running the proximity alerts autonomously, because they do not depend on the Raspberry Pi 4.

2) *Start-up and shutdown*: At power-on, the Pico runs a self-test (sensor read-back and a short motor pulse) while the Raspberry Pi 4 loads the YOLOv8n model. The system announces “System ready” once the model is loaded and a valid sensor frame has been received; until then, only the proximity alerts of the Pico are active. At shutdown or power loss, the Raspberry Pi 4 closes the SQLite database safely and flushes the pending records, and the Pico switches all outputs off.

3) *Subsystem-failure behaviour*: Camera or inference failure the Pico alerts continue, the Raspberry Pi 4 announces “vision unavailable”, and the risk is computed from the IR, ToF and LDR readings only. Failure of a distance sensor: the sensor is marked FAULT, the user is informed once, and the remaining sources are used. Pico link failure: see Section III.D.1; voice alerts continue from the camera branch. Wi-Fi or cloud unavailability: records stay in SQLite and are uploaded later; local alerts are not affected. Contradictory readings (e.g., the IR sensor reports an obstacle that the camera does not recognize): the highest risk level among the valid sources is applied. Stale camera predictions (older than T_stale) are discarded. These behaviours are verified by the fault-injection tests of Section V.G.

IV. EXPERIMENTAL SETUP

A. Smart Stick Component

The proposed smart stick integrates both hardware and software components to provide safe and adaptive navigation for visually impaired users. Fig. 2 shows the hardware components of our smart stick for visually impaired users.

1) *Hardware*: The proposed smart stick integrates multiple hardware components organized into the following layers: 1) the sensing layer, which consists of an IR sensor (Infrared Obstacle Avoidance Sensor, 4-pin), an LDR module (Light Sensitive Relay Module, 5 V), a ToF sensor (laser-based Time-of-Flight range sensor, TOF10120) and a camera module (Pi Camera Module v1.3); 2) the processing layer, which consists of a microcontroller unit (MCU, Raspberry Pi Pico) and an AI processing unit (Raspberry Pi 4 Model B); 3) the feedback layer, which consists of a vibration motor (haptic feedback), a buzzer alarm (3–24 V, 95 dB, for audible alerts) and an audio output (speaker and earphones for voice alerts); and 4) the power management layer, which consists of an SMPS (for laboratory testing), and buck converters (for voltage regulation).



Fig. 2. Hardware components of the proposed smart stick.

2) *Software*: The software architecture of the smart stick is built using the following tools and libraries: Python v3.9, YOLOv8n (Nano version) for object detection, implemented via the Ultralytics framework and OpenCV v4.8, PyTorch library, pyttsx3 for generating real-time voice notifications, SQLite for local database storage on the Raspberry Pi and HTTP protocols for IoT data transmission to the remote server.

3) *Sensor placement, calibration and physical characteristics*: Sensor placement. With the stick held upright and its tip on the ground, the ToF sensor is mounted at a height of 30 cm and the IR sensor at 35 cm above the ground. The electronics box, which houses the LDR module and the camera, is mounted at a height of 1 m, and the vibration motors are located at the top of the stick, at the user's hand, at a height of 140 cm.

Sensor ranges and calibration. The ranges of the IR, ToF and LDR modules are the values rated in the manufacturers' datasheets. No individual calibration against a reference instrument was performed, and the practical ranges may be slightly lower than the rated ones; this is listed as a limitation (Section V.G). The light threshold of Table II was set using the LDR module potentiometer.

Physical characteristics. The stick is 150 cm long. It is powered from the mains through an SMPS and a detachable 3 m cable (no battery in the prototype). The electronics are housed in a waterproof box in the upper part of the stick, whereas the sensors and the vibration motors are outside it; the water resistance of the enclosure was not tested.

B. Environment

Our experiments were conducted in an indoor environment: a room containing desks, chairs and a fan. Outdoor operation (sidewalks, vehicles, pedestrians, varying lighting) was not evaluated and is left to future work.

Environmental conditions. The experiments were carried out in an indoor room measuring 7 m × 5 m (35 m²). The room contained four desks, about six chairs, two fans, an air conditioner, a whiteboard, four laptops, a monitor and a projector, and people were present in the room during the tests. A staircase of six steps leads down from the room.

C. Evaluation Metrics

We evaluate the proposed system using different metrics described in Table I.

TABLE I. EVALUATION METRICS

Metric	Description	Unit
Detection Accuracy	Percentage of obstacles correctly detected by the proposed smart stick.	%
Missed Obstacles	Number of obstacles that were not detected by the system.	Count
False Alerts	Number of alerts generated without the presence of an actual obstacle.	Count
Average Detection Distance	Average distance at which obstacles were successfully detected.	cm
System Response Time	Time elapsed from obstacle detection to user notification.	ms
Alert Correctness	Ability of the alert management subsystem to generate the appropriate risk level and feedback for different scenarios.	Qualitative
Ease of Navigation	User rating of how easily the smart stick supports safe navigation.	Likert (1–5)
Clarity of Alerts	User rating of the clarity and understandability of system alerts.	Likert (1–5)
Confidence in Obstacle Detection	User confidence in the reliability of obstacle detection.	Likert (1–5)

D. Risk Assessment Criteria

To ensure reliable obstacle detection and timely user notifications, threshold values were predefined for each sensing module according to sensor specifications, previous studies, and preliminary experimental observations. These criteria are used by the alert management subsystem to classify the detected situation into different risk levels. The following criteria are described in Table II.

E. Experimental Protocol

Obstacle tests. The tests were carried out in the indoor room described in Section IV.B (7 m × 5 m), which contained desks, chairs, fans, an air conditioner, a whiteboard, laptops, a monitor, a projector, people, and a staircase of six steps. Ten participants walked through the room while holding the stick, and the 100 obstacles of Table V were encountered during these walks.

TABLE II. RISK ASSESSMENT CRITERIA

Sensor	Measurement Range	Threshold Value	Risk Level Triggered	Notes
IR Sensor	4–15 cm	≤ 15 cm	High	Detects very close obstacles such as stairs, steps, and furniture.
ToF Sensor	15–35 cm	15–35 cm	Medium	Detects medium-range obstacles, allowing early reaction.
LDR Sensor	0–1000 lux	< 100 lux	High	Indicates low-light or dark environments.

Ground truth and definitions. An obstacle is counted as correctly detected if the system produced an alert of the appropriate risk level before the stick reached it; a missed obstacle is an obstacle for which no alert was produced; and a false alert is an alert produced when no obstacle was present in the walking direction. Object-recognition accuracy is the percentage of recognition instances in which the spoken label matched the real object. Detection accuracy is reported with its 95% Wilson confidence interval (91.5–99.0%).

User study. Ten participants [state whether they were visually impaired or sighted and blindfolded] took part in the study. After the walks, they rated the three Likert items of Table I (1 = very poor, 5 = excellent).

V. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed smart stick was tested in an indoor environment to evaluate its performance in detecting obstacles, recognizing objects, and providing appropriate alerts to visually impaired users. The evaluation focused on five aspects: 1) detection performance, 2) alert generation, 3) system response time, 4) user experience, and 5) comparison with existing systems, in an indoor test room. Sections V.F–V.G report the contribution of the individual subsystems, fault-tolerance tests, long-duration operation and the limitations. Fig. 3 presents a real snapshot of the proposed smart stick.

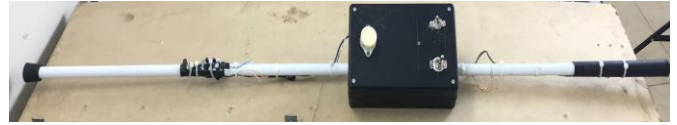


Fig. 3. Real snapshot of the proposed smart stick.

A. Alert Generation Evaluation

To verify the correctness of the proposed alert management subsystem, the smart stick was evaluated under a set of representative environmental scenarios. These scenarios simulate different combinations of sensor readings and object detection outcomes to assess whether the system generates the appropriate risk level and multimodal feedback. The evaluated scenarios are presented in Table III.

As shown in Table III, the proposed system successfully generated appropriate alerts according to the detected environmental conditions. High-risk situations, such as nearby stairs, pedestrians, and low-light environments, triggered strong vibration, text-to-speech (TTS), and buzzer notifications to maximize user awareness. In contrast, medium-risk scenarios activated milder feedback, including vibration and short voice prompts, thereby reducing unnecessary alarms while maintaining safe navigation. These results demonstrate the effectiveness of the proposed alert management subsystem in adapting the feedback according to the surrounding context.

TABLE III. ALERT GENERATION EVALUATION

Scenario	IR (cm)	ToF (cm)	LDR (lux)	YOLO Detection	Risk Level	Alert Type	Message / Feedback
1	10	30	50	Stairs	High	Strong Vibration + TTS + Buzzer	“Stairs ahead. Low light detected.”
2	–	25	200	None	Medium	Mild Vibration + Short TTS	“Obstacle ahead.”
3	20	40	150	Chair	Medium	Mild Vibration	“Chair detected ahead.”
4	–	–	50	None	High	Loud Buzzer + TTS	“Low light environment detected.”
5	12	20	40	Person	High	Strong Vibration + TTS	“Person ahead. Low light detected.”

B. System Response Time Evaluation

To evaluate the real-time capability of the proposed smart stick, the response time of each system component was measured from sensor activation to user feedback generation. The evaluation includes sensing, object recognition, alert processing, and the overall system latency as in Table IV.

As shown in Table IV, the IR and ToF sensing modules provide rapid obstacle detection with response times of 50 ms and 60 ms, respectively. The YOLOv8n object recognition module requires approximately 200 ms per frame due to the computational cost of deep learning inference. In contrast, the alert management subsystem introduces only about 5 ms of additional processing time. The sensor-to-vibration path (IR/ToF → Pico → motor, 50–60 ms) and the camera-to-voice path (YOLOv8n and alert processing, about 205 ms) run in

parallel, so the response time of the stick is bounded by the slower path, i.e., below 250 ms. These figures are separately measured component times; they do not include the camera capture time or the start-up time of the speech synthesis.

TABLE IV. AVERAGE RESPONSE TIME OF THE PROPOSED SMART STICK COMPONENTS

Component	Average Response Time
IR Sensor + Vibration Motor	50 ms
ToF Sensor + Vibration Motor	60 ms
YOLOv8n Object Recognition	200 ms per frame
Alert Management Processing	~5 ms
Overall System Response Time (derived from the component times above)	≤ 250 ms

C. Detection Performance Evaluation

We evaluated the performance of our smart stick for detecting and recognizing the obstacles in an indoor environment. The evaluation considered the number of encountered obstacles, correctly detected obstacles, missed detections, false alerts, and the average detection distance, using the protocol of Section IV.E. The results are summarized in Table V.

TABLE V. OBSTACLE DETECTION AND OBJECT RECOGNITION PERFORMANCE IN THE INDOOR ENVIRONMENT

Metric	Indoor
Total Obstacles Encountered	100
Correctly Detected Obstacles	97
Obstacle Detection Accuracy (%)	97.0
95% CI of detection accuracy (Wilson, %)	91.5–99.0
YOLO Object Recognition Accuracy (%)	96.5
Missed Obstacles	3
False Alerts	2
Average Detection Distance (cm)	388

As shown in Table V, in the indoor test room the proposed system achieved an obstacle detection accuracy of 97.0% (95% Wilson CI 91.5–99.0%) and a YOLOv8n object recognition accuracy of 96.5%, with three missed obstacles and two false alerts. The average detection distance was [388 cm]: obstacles were detected from more than 2 m, consistent with the ToF mode (rated range up to 4 m) [confirm]. These results come from a single room with desks, chairs and a fan, ten participants and one prototype; outdoor conditions (vehicles, pedestrians, varying lighting) were not evaluated.

D. User Evaluation

Ten visually impaired volunteers (22–45 years old) participated in the user evaluation in the indoor test room. Participants assessed the proposed system using a 5-point Likert scale based on ease of navigation, alert clarity, and confidence in obstacle detection, following the protocol of Section IV.E.

TABLE VI. SUMMARY OF USER EVALUATION RESULTS (N = 10)

Evaluation Metric	Mean $\pm$ SD
Ease of Navigation	4.8 $\pm$ 0.42
Clarity of Alerts	4.9 $\pm$ 0.32
Confidence in Obstacle Detection	4.8 $\pm$ 0.40

As summarized in Table VI, the proposed system achieved high usability ratings across all evaluation criteria. Alert clarity received the highest mean score (4.9 $\pm$ 0.32), indicating that participants clearly perceived the multimodal feedback. Ease of navigation and confidence in obstacle detection both achieved a mean score of 4.8, suggesting that participants perceived the system as supporting safe and intuitive navigation. These ratings are subjective, come from a small sample (n = 10), show a ceiling effect, and were not accompanied by objective measures (collisions, walking speed, task completion time) or by a white-

cane control condition; they should therefore be read as preliminary usability feedback.

E. Comparative Analysis With Existing Systems

To position the proposed smart stick with respect to the literature, we compare its features with those of existing systems in Table VII.

As presented in Table VII, the proposed system integrates multiple sensors (IR, ToF, LDR and a camera), providing a broader set of sensing modalities than most of the listed systems. In contrast to approaches relying solely on ultrasonic or laser sensors, the proposed system combines sensor fusion with YOLOv8n-based object recognition to improve obstacle awareness.

Furthermore, the proposed system provides multimodal feedback through multi-level vibration, voice, and a buzzer, enabling adaptive user notifications according to the detected risk level. The integration of Raspberry Pi 4 and Raspberry Pi Pico supports real-time processing, while Wi-Fi connectivity enables IoT-based communication. Table VII is a feature-level comparison: the values of the other systems are those reported in their papers (NR = not reported) and were obtained with different sensors, environments and metrics; the table therefore does not establish that the proposed system outperforms them. Overhead (head-level) obstacle detection was not evaluated for the proposed stick (Section V.G).

F. Contribution of the Individual Subsystems and System Behaviour

The proposed stick combines three sensing branches whose roles are complementary. The short-range IR branch detects near obstacles and drives the strong-vibration motor through a relay, the laser ToF branch is read by the Raspberry Pi Pico 2 W and drives the gentle-vibration motor, and the camera branch runs on the Raspberry Pi 4 and reports recognized objects by voice. The light-detection branch switches the buzzer through the LDR relay module. Table VIII summarizes the role of each branch and the response times measured for its components (Table IV).

The detection accuracy (97.0%), the number of missed obstacles (3) and the number of false alerts (2) in Table V refer to the complete system with the fused decision. The accuracy of each branch alone, and the latency of a single-board configuration, were not evaluated, so the individual contribution of each branch and the latency benefit of the distributed architecture are not quantified in this study.

By design, the IR and light alerts are generated by hardware relays and therefore do not depend on the Raspberry Pi 4, the camera or the network. The ToF alert depends on the Pico only, and the voice alerts depend on the Raspberry Pi 4.

Records are first stored in the local SQLite database and uploaded when Wi-Fi is available, so a network outage does not interrupt local alerting. These behaviours follow from the design and were not verified by fault-injection tests, and the behaviour during long continuous operation (processor load, temperature, drift of the inference time) was not evaluated. The Raspberry Pi 4 is housed in an enclosure with a cooling fan, and the prototype is powered from the mains. Both points are listed as limitations in Section V.G.

TABLE VII. COMPARISON OF THE PROPOSED SMART STICK WITH EXISTING SYSTEMS

Ref.	Sensors	Output	Controller	GPS/GSM	IoT Enabled?	Image Processing ?	Detection Range (Max)	Overhead Detection?	Water-Proof ?	Detection Accuracy	Response Time
[4] Rahman & Sadi	Laser + Camera	Voice	Raspberry Pi	No	No	Yes	~200 cm	No	No	NR	NR
[5] Annamalai et al.	Ultrasonic + Camera	Voice + Vibration	Raspberry Pi 4B	No	No	Yes (TensorFlow)	~300 cm	No	No	NR	NR
[16] Paul et al.	Ultrasonic	Vibration	Arduino Mega	No	No	No	~250 cm	No	No	NR	NR
[13] Farooq et al.	Ultrasonic + Water + Camera	Voice + Vibration	Arduino + Camera	Yes (GPS)	No	Yes	~400 cm	No	Yes	NR	NR
[14] Kunta et al.	Ultrasonic + GPS	Voice + GSM Message	Arduino	Yes	No	No	~350 cm	No	No	NR	NR
Proposed System (Our Work)	IR + ToF + LDR + Camera	Multi-level Vibration + Voice (TTS) + Buzzer	Raspberry Pi 4 + Pico	No	Yes	Yes (YOLOv8n Nano)	≤ 400 cm (ToF mode, rated)	No	Not Tested	97% (indoor test room)	≤250 ms

TABLE VIII. ROLES OF THE SENSING BRANCHES AND THEIR COMPONENT RESPONSE TIMES

Branch	Processing	Output	Component response time (ms)
IR (short range)	Relay (hardware)	Strong vibration	50
ToF (15–35 cm)	Raspberry Pi Pico 2 W	Gentle vibration	60
Camera (YOLOv8n)	Raspberry Pi 4	Voice alert	200
Alert management	Raspberry Pi 4	Alert selection	5

G. Limitations

The present study has the following limitations: 1) Evaluation scope: all tests were done in one indoor room (desks, chairs and a fan) with ten participants and one prototype; outdoor operation, for which the ToF mode is intended, was not evaluated. 2) Sensor ranges are the manufacturers' rated values and were not individually calibrated, so practical ranges may be slightly shorter. 3) In the close-range IR mode the warning horizon is short (about 0.3 s at 1.2 m/s for 35 cm), so for static obstacles it does not extend the user's reaction time beyond that of a white cane. 4) Object recognition is limited to the classes of the detection model, and overhead obstacles lie outside the camera's field of view. 5) No white-cane control condition and no collision, walking-speed or task-completion measures were collected, and the Likert ratings show a ceiling effect. 6) The comparison with other systems (Table VII) is feature-based, because the original papers use different sensors and metrics.

7) The IoT subsystem currently provides logging and remote monitoring only, and its usefulness for caregivers has not been tested.

VI. CONCLUSION AND FUTURE WORK

This study presents the design and implementation of an integrated AI-powered smart stick specifically designed to assist blind and visually impaired individuals with navigation. By integrating infrared and time-of-flight sensors with a lightweight YOLOv8n deep learning model, the system provides obstacle detection, object recognition, and ambient light perception in near real time. The distributed architecture—where the Raspberry Pi Pico manages low-level sensor data integration, while the Raspberry Pi 4 handles AI inference—was designed to reduce latency and improve response speed; its benefit over a single-board configuration is quantified in Section V.F. Experimental results in an indoor test room showed an obstacle detection accuracy of 97.0% and an object recognition accuracy of 96.5%, with a component-based response time below 250 ms; outdoor operation was not evaluated. User ratings ($n = 10$) indicated that the multimodal feedback (vibration, buzzer, and voice alerts) was clear and supported confidence and ease of navigation. Furthermore, the integration of IoT connectivity expands the system's utility by enabling remote monitoring and data logging. Overall, the proposed smart stick is a promising prototype of an assistive technology for visually impaired individuals; the limitations discussed in Section V.G must be addressed before its effect on mobility and independence can be established.

A. Future Work

Although the proposed system has achieved promising results, there are several areas that require improvement and development. Future work will focus on the following:

- Expanding the Warning Horizon: Replacing or complementing the IR and ToF sensors (currently limited to 35 cm) with long-range ToF sensors or stereoscopic vision modules to provide warnings several meters ahead.
- Improving Object Recognition: Training the YOLOv8n system using larger and more diverse datasets to improve the recognition of complex or rare objects (including stairs, kerbs and head-level hazards) in crowded urban environments.
- GPS Navigation: Integrating GPS and mapping services to provide route guidance and location-based alerts for outdoor navigation.

By addressing these aspects, the smart stick can be developed into a comprehensive navigation system, providing greater independence and safety for visually impaired users.

REFERENCES

- [1] Chang, Wan-Jung, et al. "Design and implementation of an intelligent assistive system for visually impaired people for aerial obstacle avoidance and fall detection." *IEEE Sensors Journal* 20.17, 2020, 10199-10210.
- [2] Abidi, Mustufa Haider, et al. "A comprehensive review of navigation systems for visually impaired individuals." *Heliyon* 10.11, 2024.
- [3] Alruwaili, Madallah, et al. "Deep learning and ubiquitous systems for disabled people detection using YOLO models." *Computers in Human Behavior* 154, 2024, 108150.
- [4] Rahman, Md Atikur, and Muhammad Sheikh Sadi. "IoT enabled automated object recognition for the visually impaired." *Computer methods and programs in biomedicine update* 1, 2021, 100015.
- [5] M. Annamalai, P. S. Tale, N. S. Pawar, and V. T. Shelke, "Smart Stick for Blind People using Raspberry Pi," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, Apr. 2025, 5, 4.
- [6] Jeeva, V., et al. "Smart Glasses with Voice Assistance and Sensors for the Visually Impaired." *2026 International Conference on Emerging Trends in Mobile Computing and Sustainable Informatics (ICEMCSI)*. IEEE, 2026.
- [7] Chibambararaj, N., and Mr K. Aravindhan. "An Embedded Vision and Ultrasonic Sensor Fusion Approach for Real-Time Assistive Navigation using Smart Glasses." *2026 8th International Conference on Inventive Material Science and Applications (ICIMA)*. IEEE, 2026.
- [8] Md. Adil, T. Rafa, J. Ferdoush, A. Mahmud and A. Pathak, "An IoT based Voice Controlled Blind Stick to Guide Blind People". *International journal of engineering inventions*. 9, 1, 2020, 9-14.
- [9] K. Loganathan, N. Lakshmi, N. Chandrasekaran, S.R. Cibisakaravarthi, R.H. Priyanga and K.H. Varthini, "Smart Stick for Blind People," in *Proc. 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020, 65-67.
- [10] N.S. Khan, A.A. Gaud, H.R. Khanvilkar and R.N. Shelar, "Design and modification of smart stick for visually impaired with stick finder," *International Journal of Advance Research, Ideas and Innovations in Technology*. 6. 2, 2020, 326-328.
- [11] Nitha, T. M., et al. "NETRA-An IoT Based Smart Glass System for Visually Impaired Individuals." *2026 International Conference on AI-Driven Solutions for Sustainable Smart Cities: Challenges and Opportunities (ADSSSC)*. IEEE, 2026.
- [12] Singh, Shashank, et al. "Assistive Technologies for Visually Impaired Using Arduino: A Conceptual Study." *Progressive Computational Intelligence, Information Technology, and Networking*. CRC Press, 2026. 604-610.
- [13] Farooq, M. S., Shafi, I., Khan, H., Diez, I. D. L. T., Breñosa, J., Espinosa, J. C. M., & Ashraf, I. "IoT enabled intelligent stick for visually impaired people for obstacle recognition", *Sensors*, 22(22), 2022, 8914.
- [14] Kunta, V.; Tuniki, C.; Sairam, U. "Multi-functional blind stick for visually impaired people". In *Proceedings of the IEEE 2020 5th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 10–12 June 2020, 895–899.
- [15] Mind, P.; Palkar, G.; Mahamuni, A.; Sahare, S. "Smart stick for visually impaired". *Int. J. Eng. Res. Technol.* 10, 2021, 196–198.
- [16] D. Paul et al., "AI-Enhanced Multifunctional Smart Assistive Stick for Enhanced Mobility and Safety of the Visually Impaired," *Journal of Computer Science and Technology Studies*, 7(1), 283–301, 2025.
- [17] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>.
- [18] Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, 2023, 1-33.
- [19] Dogan Ibrahim, *Raspberry Pi Pico Essentials: Introduction to Programming with MicroPython*, Elektor, 2021.
- [20] N. El-Bendary et al., "Smart Assistive Systems for Visually Impaired People: A Survey," *IEEE Access*, 9, 2021, 12150-12170.
- [21] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect v2 for mobile edge computing: A survey of Time-of-Flight sensors," *IEEE Sensors Journal*, 15. 6, 2015.