Journal of Robotics and Control (JRC)
Volume 6, Issue 2, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i2.25245

Mobility Aid for the Visually Impaired Using Machine Learning and Spatial Audio

Wahyudi [1*], Rizkar Al Akbar [2], Daniel Witansa Samuel [3], Muhammad Favian Adinata [4], Denis [5]
[1, 2, 3, 4] Department of Electrical Engineering, Faculty of Engineering, Universitas Diponegoro, Semarang, Indonesia
[5] Department of Institute Production Engineering of E-Mobility Components (PEM) RWTH Aachen, Faculty of Mechanical Engineering, RWTH Aachen University, Aachen, Germany
Email: [1*] wahyuditinom@elektro.undip.ac.id, [2] rizkaralakbar@gmail.com, [3] danielwitansa@gmail.com, [4] favian.adi8888@gmail.com, [5] d.ginting@pem.rtwh-aachen.de
*Corresponding Author

***Abstract*—Assistive technology is crucial in enhancing the quality of life for individuals with disabilities, including the visually impaired. Many mobility aids lack advanced features such as real-time machine learning-based object detection and spatial audio for environmental awareness. This research contributes to developing more intelligent and adaptable assistive technology for visually impaired individuals, promoting improved navigation and environmental awareness. This research presents a head-mounted mobility aid that integrates a time-of-flight camera, a web camera, and a touch sensor with K-Means clustering, Convolutional Neural Networks (CNNs), and concurrent programming on a Raspberry Pi 4B to detect and classify surrounding obstacles and objects. The system converts obstacle data into spatial audio, allowing users to perceive their surroundings through sound direction and intensity. Object recognition is activated via a touch sensor, providing distance and directional information relative to the user using audio description. The concurrent programming implementation improves execution time by 50.22% compared to Infinite Loop Design (ILD), enhancing real-time responsiveness. However, the system has limitations, including object recognition limited to 80 predefined categories, a 4-meter detection range, reduced accuracy under high-intensity sunlight, and potential interference in spatial audio perception due to external noise. Assistive technology to help the mobility of blind people using advanced technology based on machine learning has developed in a form that can be used flexibly for the user's mobility.**

Keywords—Assistive Technology; Blind People; Time-of-Flight Camera; K-Means; Image Recognition; Concurrent Programming.

I. INTRODUCTION

Technological advances aim to simplify various aspects of human life, with assistive technology being one of its applications. Assistive technology includes devices, products, or software designed to improve the functional abilities of individuals with disabilities [1]-[5]. Disability is a condition that can limit mental and physical abilities, preventing an individual from performing specific tasks in the usual manner [6]. Disabilities arise from limiting conditions that affect an individual's vision, physical mobility, social activities, and self-care abilities. Assistive technologies can significantly enhance navigation and daily activities for individuals with visual impairments [7]-[10]. One common challenge for visually impaired individuals is the difficulty in moving from one location to another due to obstacles, uneven surfaces, or challenges in determining orientation for mobility, as well as other hard-to-detect hazards [11]-[18].

Assistive devices are tools for the visually impaired to enhance environmental awareness, mobility, and information acquisition [19]-[23]. Commonly used assistive technologies include navigation tools with cameras and audio to gather and convey information [24], [25]. One such example is an ultrasonic or radar navigation device. This device is like a cane with a sensor positioned in the center [26]-[28]. In addition to canes, some research explores the use of camera technology [29]-[35]. For instance, as in [36], assistive devices employ a smartphone camera with depth-sensing capabilities to detect obstacles for visually impaired individuals. However, these devices still have limitations, such as constrained obstacle detection, limited movement adaptation, and suboptimal use of sensory feedback systems for visually impaired users. Some assistive tools, such as canes, can only detect obstacles directly in front of the user, resulting in limited movement adaptation, which makes it difficult for users to detect obstacles on the sides, above or below them. Constrained obstacle detection is important when visually impaired individuals navigate new or dynamic environments since some assistive devices rely on predefined databases for obstacle detection. In such cases, unexpected obstacles may not be identified effectively. Moreover, sensory feedback systems in assistive devices enhance spatial awareness without disrupting the user's daily activities. These systems should provide clear and intuitive feedback regarding the spatial positioning of surrounding objects. Therefore, a device that can adapt movement direction, unconstraint obstacle detection, and provide clear, easily understood guidance is necessary.

To perform their intended function, a walking aid for the visually impaired must incorporate a distance measurement function to guide visually impaired users. According to various studies, there are two primary methods for acquiring distance information about an object: the passive and active methods [37]-[41]. The passive method measures distance by receiving information about an object’s position within a frame [41]-[45]. This system is based typically on cameras and computer vision, with one standard implementation being stereo vision using the triangulation method. Stereovision is a computer vision system that calculates

distances using stereoscopic measurement techniques. It uses two cameras as if they were a single camera to create a sense of depth. It utilizes the disparity between the two camera views to calculate distance accurately [46]-[49]. Distance estimation via stereoscopic measurement requires trigonometric equations separated into three phases. The first stage employs various image processing techniques to improve computing speed, such as decreasing the input image resolution and transforming the RGB input image to grayscale. The second stage extracts the object's position from both cameras. The third stage determines the object's position based on the extracted data and estimates the distance using trigonometric equations [50]-[53]. Despite its precision, the stereo vision method has several significant limitations. One major drawback is the need for exact camera calibration, both for intrinsic parameters such as focus and lens distortion and extrinsic parameters that define the relative positions of the cameras.

Additionally, the method is susceptible to lighting conditions or shadows and can disrupt feature matching between images. In areas with low texture, such as smooth surfaces, stereo vision struggles to find matching points, reducing distance measurement accuracy. Another challenge is the high computational demand for feature matching, mainly when working with high-resolution images, making real-time applications challenging without powerful hardware. The method also tends to be less accurate when measuring distances to objects that are either very close or far due to disparity errors. Occlusion, where one object blocks another in one camera's view, further complicates object matching between images, diminishing measurement accuracy. Finally, while stereo vision can be implemented with low-cost hardware, achieving optimal results requires high-quality cameras, increasing system costs. These limitations restrict the effectiveness of stereo vision, particularly in uncontrolled environments or resource-constrained applications [54]-[57].

On the other hand, this research uses an active method to measure distance by transmitting a signal to the target. This system typically calculates the time-of-flight of laser beams, ultrasonic waves, or radio signals to detect and locate objects. Time-of-flight systems estimate the distance to an object by measuring the time a signal pulse travels to the object and returns. A primary drawback of this method is the potential confusion caused by echoes from previous or subsequent pulses and a limited accuracy range, typically between one to four meters [58]-[60]. Active distance measurement can be performed using ultrasonic waves, which are sound waves with frequencies above 20 kHz, to determine the distance to an object without physical contact. In this process, an ultrasonic sensor emits sound waves propagating through the air until they reach an object. The sensor then reflects and receives the waves, and the round-trip travel time is measured using a specific formula [61]. In addition to ultrasonic pulses, the active method can utilize infrared light. One example is the use of a time-of-flight camera. Time-of-flight cameras measure distance based on the time it takes for infrared light to be reflected off an object and captured by the receiver, providing a distance output for each camera pixel. Unlike the stereo vision method, time-of-flight cameras do not require training data, have lower processing overhead, function effectively in low-light conditions, and can avoid issues with object occlusion. Although it does not require training data, the use of a time-of-flight camera in previous research could only provide the distance for each pixel of the captured image. This distance information remains raw and cannot be a reference for visually impaired individuals [62]-[64]. In this research, specific obstacle distance information is obtained by integrating a time-of-flight camera with the K-Means algorithm. The detection results from the time-of-flight camera are processed using the K-Means algorithm to determine the distance and position of the nearest obstacle by clustering the distance values of each pixel and selecting the cluster that most accurately represents the distance. Subsequently, the clustered object is further processed to provide the obstacle's position [65]. The calculated object position and distance are then conveyed through audio output, providing real-time guidance to users. However, using the time-of-flight camera still has limitations when operating outdoors under high sunlight, as it can obscure the reflection results from the infrared laser. In contrast, the time-of-flight camera performs better under low-light conditions or even in complete darkness. Therefore, this research would yield optimal results if conducted indoors.

Previous research on assistive technology for the visually impaired has frequently employed audio cues or verbal descriptions to convey information about obstacles. This research proposed using audio descriptions to inform users of obstacle locations. The coordinates of obstacle points trigger warnings for obstacles in directions such as full right, entire left, below, or directly in front of the body. The system's output consists of audio feedback based on the obstacle's direction, delivering phrases like "Left Torso," "Full Right," or "Left Ground" to alert the user [36]. However, the audio cue method has several limitations. These include restricted object description, the use of non-universal language, and lengthy descriptions that can interfere with the primary auditory function, such as engaging in conversations or listening to other sound sources [66]. In this research, development is conducted over previous research that relied on audio cues; a spatial audio-based navigation system is used to convey information about the position and distance of obstacles. A spatial audio navigation system uses the Head-Related Transfer Function (HRTF) to characterize audio by implementing spatial sound. Implementing the spatial audio system involves creating a spatial audio dataset using a Digital Audio Workstation (DAW) equipped with spatial audio plugins. Mono audio data is manipulated within DAW using these plugins to produce spatial audio outputs. The audio outputs generated by the DAW are organized into a spatial audio dataset categorized by distance and direction. This data set serves as the system's output, allowing the software to call the appropriate audio files based on the detected distance and direction, streamlining the system's final output. One key advantage of this method is the reduced processing load on the system, enabling the system to operate more efficiently. The spatial audio quality is superior because the audio data is pre-rendered, ensuring high fidelity [67]-[70].

In addition, previous assistive technologies for the visually impaired also included functionality to identify objects essential for daily activities—machine learning algorithms based on CNNs commonly used for object recognition. CNNs are a type of artificial neural network capable of identifying and classifying objects in images with high accuracy. Images are captured by extracting frames from video footage recorded by a web camera at regular intervals. A commonly used library for executing CNN is YoloV8, with a limitation in object detection, without new training data, which is the detection of only 80 object types. However, the object recognition typically only provides the object's name [71]-[74]. In this research, a method is implemented to obtain object position information by incorporating distance data from the time-of-flight camera to determine the distance of each detected object. This research tested four random objects to evaluate the system's reliability. Using this system, once an object is identified, it will classify its position based on the pixel coordinates of the object's center of mass and convert this information into audio output. The type and position of the object will be conveyed to the user via earphones in the form of audio, providing real-time guidance for enhanced spatial awareness.

In addition to detection systems, technological advancements can be applied to processing systems. Assistive technologies for the visually impaired primarily enhance environmental awareness, mobility, and information acquisition. Typically, these technologies utilize embedded systems that sequentially execute programs, beginning with data collection from cameras or sensors, then object or obstacle recognition, and concluding with audio output to convey the results before restarting the cycle. This approach is known as the ILD, where detection, computation, and actuator execution programs run repeatedly in a continuous loop.

An example of assistive technology employing infinite loop design is in research with the Raspberry Pi, a camera, and a speaker in its design. In this research, Raspberry Pi executed the program using the ILD methodology [75]. However, this method has notable limitations, as it can only execute tasks serially, processing one instruction at a time. This method becomes problematic when the mobility aid must handle multiple sensors and perform high-demand processing tasks, where responsiveness and rapid execution are critical. The limitations of assistive technologies for the visually impaired in previous research necessitate the development of more advanced methods for controlling program execution [76], [77]. This research implemented multitasking control through concurrent programming. Concurrent programming improves system processing time by distributing tasks across each core, which operates in parallel, to maximize the processing speed of a multi-core system. This system can be achieved using Real-Time Operating Systems (RTOS), threading, and multiprocessing. Research has shown that RTOS, threading, and multiprocessing can execute multitasking systems with improved processing times and responsiveness. However, unlike threading and multiprocessing, which are designed to handle multiple tasks simultaneously for computationally intensive operations, RTOS focuses primarily on scheduling time-sensitive tasks with strict reliability requirements. RTOS operates on a priority-based task-scheduling mechanism. When an interruption occurs, the current task is delayed until the necessary conditions are met, which prevents true parallelism from being achieved. Consequently, threading and multiprocessing are more suitable for multitasking in mobility aids for the visually impaired, as they allow concurrent execution of multiple tasks without compromising system responsiveness [78]-[81]. This research designed an appropriate implementation of threading and multitasking to improve processing times. Therefore, concurrent programming is adopted to overcome the limitations of the infinite loop design. Threading and multiprocessing improve processing speed and system response time, ensuring a more efficient and reliable assistive device.

The novelty of this research lies in the integration of a time-of-flight camera with the K-Means algorithm to determine specific obstacle positions and distances, the incorporation of distance data acquired from the time-of-flight camera to complement object detection by CNN, the use of a spatial audio-based navigation system to convey this information, and the implementation of concurrent programming techniques, such as threading and multiprocessing, to enhance system responsiveness and processing efficiency in mobility aids for the visually impaired. This research aims to design and test mobility aids for the visually impaired with better performance in detection, audio navigation, and speed processes using machine learning based on time flight cameras and spatial audio. This research contributes to developing mobility aids for the visually impaired to implement better design, sensor, algorithm, and processing techniques. The design of this device is worn on the user's head and can detect obstacles around the user during mobility. This study utilizes a time-of-flight camera, web camera, and touch sensor to acquire data and output it as spatial audio and object descriptions through a headset. This research employs machine learning, specifically the K-means algorithm, to process depth information from the time-of-flight camera, allowing for determining the distance and direction of the nearest obstacles. The study also incorporates an object recognition system that identifies obstacles' type, position, and distance by applying CNNs to the images captured by the web camera. All systems implemented using concurrent programming to achieve better task execution performance compared to the infinite loop design, enhancing system efficiency and responsiveness.

II. METHOD

The program in this research consists of an obstacle detection system and an object detection system, which are integrated using concurrent programming. The obstacle detection results from the K-Means algorithm, which provides the position and distance of the nearest objects, are processed and then conveyed to the user in the form of spatial audio. The spatial audio is designed to create the effect of the sound source appearing to come from a specific direction in 3D space, relying on the HRTF in humans. Additionally, object recognition results are provided through audio

descriptions. The hardware block diagram is represented in Fig. 1.

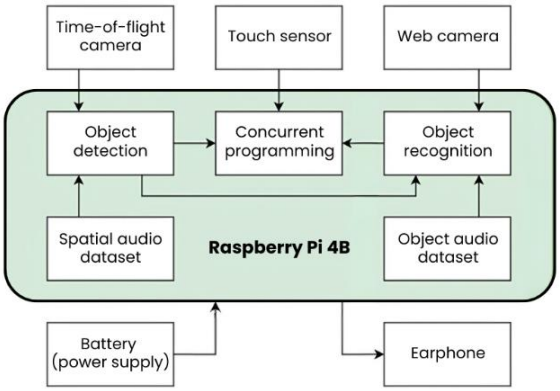


Fig. 1. Block diagram of the assistive system

This research uses the Raspberry Pi 4B as a single-board computer (SBC) to execute previously designed programs. The sensors employed in this study include the Logitech C270 HD Webcam and the Arducam time-of-flight camera. The inputs for the object recognition subsystem consist of image data generated by the web camera and distance data provided by the time-of-flight camera. A 15-watt power bank battery powers the device. The data read by the Raspberry Pi 4B is MJPEG (compressed video data), an efficient data type that retains its quality. The design of the mobility aid for the visually impaired in this research is shown in Fig. 2.

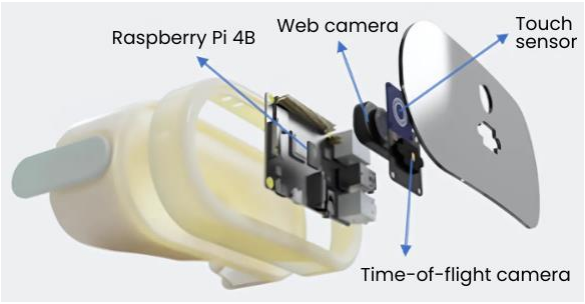


Fig. 2. Hardware design of the mobility aid for the visually impaired

The mobility aid for the visually impaired is designed with two main features based on continuous real-time processing. The first feature is a tool that guides the condition of the pathway and the environment encountered by the visually impaired person based on obstacles around them. The second feature is a tool capable of recognizing objects useful for the daily life of the visually impaired. In the first feature, the object detection subsystem provides information on the distance and direction of obstacles, which is then passed to the audio subsystem for processing and characterizing the information into spatial audio. In the second feature, the object recognition subsystem interrupts the looping process in the object detection subsystem, triggered by the touch sensor. The object recognition subsystem provides information about the object's name through audio descriptions. Each system will run simultaneously using a concurrent programming method. This method allows the system to acquire data, process the collected data, and output the results simultaneously without waiting for each other. The workflow diagram of the mobility aid for the visually impaired in this paper is shown in Fig. 3.

To illustrate further, the operational scheme of the mobility aid in this paper is shown in Fig. 4. The system initiates its operation by performing object detection using the K-Means clustering algorithm, which identifies and categorizes objects based on their spatial distance and position. Subsequently, the system evaluates whether the detected object has changed its position. If no movement is detected, the system continuously monitors the object's location.

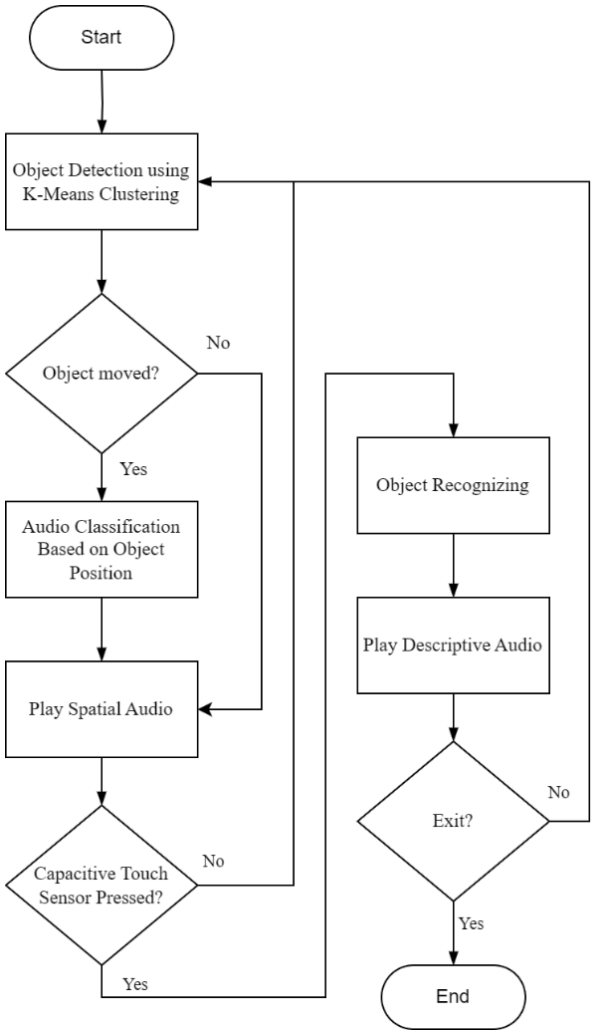
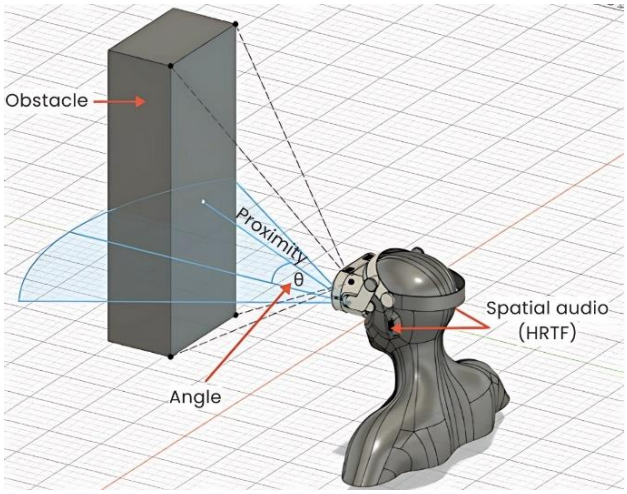
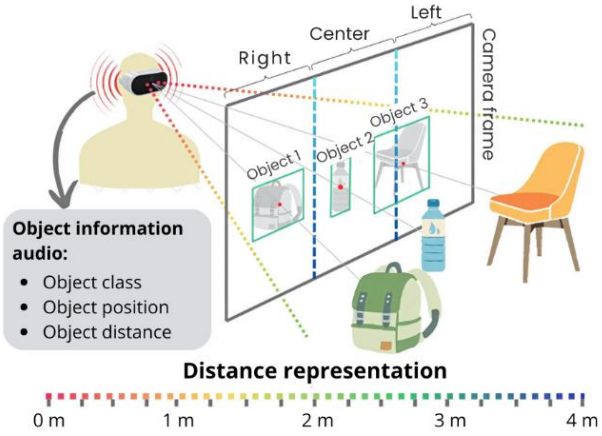


Fig. 3. Workflow diagram of the mobility aid for the visually impaired

Conversely, if movement is detected, the system proceeds to classify the object's position and generates spatial audio feedback to inform the user of the object's relative location. The workflow then includes assessing whether the capacitive touch sensor has been activated. If the sensor is not activated, the system returns to resume object detection and spatial audio feedback. Upon activating the touch sensor, the system transitions to the object recognition process, identifying the specific object detected. Following recognition, descriptive audio feedback is provided to convey detailed information about the identified object to the user. Finally, the system evaluates whether the user intends to terminate the process. If the termination condition is not met, the system continues its operations; otherwise, it concludes. This architecture effectively integrates object detection, spatial and descriptive audio feedback, and user interaction to enhance the usability and responsiveness of assistive systems for visually impaired individuals.



(a) Object detection scheme



(b) Object recognition scheme

Fig. 4. Operational scheme of mobility aid for the visually impaired

A. *Obstacle Detection System*

The limited detection range in previous technologies, which used one-directional sensors, can be addressed using more advanced sensors, such as the time-of-flight camera. A time-of-flight camera can directly measure the distance to an object by calculating the time taken for an infrared laser to travel to the object and reflect on the sensor, providing distance data that can work even in the absence of light. Based on the known speed of light, the distance of each pixel in the camera frame can be calculated using the equation (1), where d is the distance (m), t is the time taken by the light to travel from the source to the sensor (s), and c is the constant speed of light (m/s). The distance measurements create a map that provides 3D information in a 2D matrix.

$$d = \frac{1}{2}ct \tag{1}$$

The time-of-flight camera can be used in conjunction with the K-Means clustering algorithm. The K-Means clustering algorithm divides data into many clusters depending on their inherent distance from one another. The K-Means algorithm's final step is to group the data based on their proximity to the centroids identified in earlier iterations. The formula for calculating the new centroid is given by equation (2), where c_j is the new centroid for cluster j, N_j is the number of data points in cluster j, and x_i is the data point i in cluster j.

$$c_j = \frac{1}{N_j}\sum_{i=1}^{N_j} x_i \tag{2}$$

The K-Means algorithm detects the nearest objects without requiring prior training. This research utilizes the K-Means clustering method because it allows the nearest objects to be detected by clustering different depth values. Another advantage of using K-means clustering in this study is that it does not require training data, meaning the types of objects that can be detected are not limited if the object has a measurable distance or depth value that can be measured using light. K-Means clustering method is employed to group the distance data from the time-of-flight camera based on the distance value of each pixel. The value of K is determined based on the need to detect various distance differences of objects that may fall within the camera's range. The cluster with the smallest distance values, indicating the closest objects, serves as the reference cluster for detecting objects.

Object detection coordinates limit the measurement area within the depth matrix from the time-of-flight camera, where each element in the array contains a distance value. The distance value of the object is obtained from the computation of the distance data within that boundary. The distance measurement calculates the second quartile (median) of the sorted distance values at each pixel within the detected object area. The object's angle is determined by converting the centroid position of the object within the camera frame into an angle measured in degrees. The centroid position represents the angle derived from the ratio of the maximum number of pixels to the camera's maximum field of view angle. This direct ratio yields the detection angle distorted by the camera lens. Therefore, an equation is applied to compensate for this distortion. The distortion compensation equation is shown in equation (3).

$$\theta_H = \tan^{-1}\left(\frac{2(x - c_x)}{W_H}\tan\left(\frac{FOV_H}{2}\right)\right) \tag{3}$$

Several variables must be calibrated, including the sample size (pixels) captured in a single frame, the number of clusters in the K-Means algorithm, and the distance measurement calibration across the system to achieve optimal results. Sample size calibration is performed by trying several sample sizes, such as 3200, 6400, 12800, 25600, and 43200, to balance computational efficiency, object detection accuracy, and noise captured by the time-of-flight camera. Larger samples increase computational load and noise but improve accuracy, while smaller samples reduce noise and processing load but decrease accuracy. Based on these considerations, a sample size of 6400 pixels was determined with a smaller computational load, noise, and enough accuracy. Similarly, the number of clusters on K-Means was calibrated by trying several numbers of clusters from 3, 5, and 7 clusters, considering computational efficiency and the clarity of object selection based on distance. A higher number of clusters increases computational load but improves object selection, although excessive clustering can make the system overly sensitive to surfaces protruding toward the time-of-flight camera. Consequently, five clusters were chosen for the K-Means algorithm with a smaller computational load and more apparent object selection. Distance measurement calibration

was conducted using linear regression, comparing the system's distance measurements to tape measurements, resulting in an equation incorporated into the algorithm to correct measurement results.

Despite being reliable for detecting nearby obstacles without requiring training data and offering lightweight processing, time-of-flight cameras and K-Means have limitations. Time-of-flight cameras, which operate using infrared laser emissions, have a maximum detection range of 4 meters and are sensitive to external light sources, such as sunlight in outdoor environments. High-intensity sunlight can overshadow the infrared signals emitted by the time-of-flight camera, causing false detections of the closest object. Consequently, this system is unsuitable for environments with direct or intense sunlight. Additionally, reflected light from mirrors or bright-colored objects such as white can generate noise, reducing accuracy. Therefore, this system is better suited for indoor use or in dark-to-moderate lighting conditions.

Additionally, the use of a time-of-flight camera combined with the K-Means algorithm can prevent occlusion, where a smaller object in front of it obstructs a detected object. This is because the K-Means algorithm utilizes clusters based on the average nearest distance within each cluster, causing objects with minor anomalies to be overshadowed by the dominant members of the cluster [82]-[85]. Several strategies are employed to mitigate the impact of sunlight on time-of-flight cameras. Narrow-band optical filters allow only the specific wavelength emitted by the camera, reducing interference from ambient sunlight. Increasing the intensity of the emitted infrared (IR) signal ensures that reflected signals are more potent than background noise. Advanced algorithms are implemented to separate relevant signals from sunlight-induced noise, enhancing depth data accuracy. Physical adjustments, such as placing the camera in shaded areas or at optimal angles, also help minimize sensor saturation. Finally, modern time-of-flight systems utilize frequency modulation to distinguish reflected signals from environmental noise, including sunlight interference [86]-[88].

B. Spatial Audio-Based Navigation System

The spatial audio-based navigation system manages the output through spatial audio, which is delivered to the user through earphones. The audio system is configured based on input from the object detection system from the previous process. Technology for delivering object position information in assistive technology for the visually impaired is developed through spatial audio based on HRTF. HRTF is a mathematical function that describes how sound from its source is altered by parts of the human body, such as the head, ears, and torso, before reaching the eardrum [89], [90]. As a result of these changes, the listener, in this case, a human, can perceive the position of the sound source [91]-[93].

Implementing HRTF on binaural sound sources such as earphones, headphones, and stereo speakers manipulates the audio signal so that the direction of the sound source can be determined. The HRTF mathematical function alters the amplitude and phase at several frequency bands, mimicking changes caused by human anatomy [94]. The altered amplitude and phase allow the listener to perceive differences in the direction of the sound source using binaural sound sources. The frequency response graph shows differences at specific frequencies [95]. These changes are caused by acoustic interference due to the shape of the human anatomy, enabling humans to distinguish the direction of the sound source.

The spatial audio-based navigation system manages the output through spatial audio, which is delivered to the user through earphones. The audio system is configured based on input from the object detection system from the previous process. This process generates spatial audio output using the input coordinates of the centroid and the distance to the detected object. In this design, the system calls audio from a dataset, which is then adjusted in intensity and playback interval based on the object's distance. The adjustment of intensity and interval is intended to represent the object's distance from the user. The dataset consists of 21 “beep” audio files with a duration of 200 milliseconds, which have been manipulated using the HRTF method. The audio dataset includes 7 variations of azimuth: -30°, -20°, -10°, 0°, 10°, 20°, and 30°. Elevation varies into three levels: -15°, 0°, and 15°, representing the maximum field of view angle of the camera. The azimuth and elevation are distinguished based on the angle of the detected object.

Although spatial audio provides more efficient information by leveraging the heightened auditory perception of visually impaired individuals, it requires acclimatization for users to perceive the position and distance of obstacles accurately. The duration and effectiveness of this acclimatization process may vary depending on the user's auditory sensitivity and prior experience with spatial audio systems. External sounds, particularly those louder than the spatial audio signals, can interfere with the system's effectiveness by masking its output. To mitigate this, it is crucial to use audio output devices that are best suited to the user's needs. The flexibility of the device allows users to choose from a range of compatible audio output options, including earbuds, bone-conduction headphones, or over-ear headphones. For environments with high ambient noise, devices equipped with Active Noise Cancelling (ANC) technology can enhance spatial audio experience by reducing background noise, especially low-frequency sounds like engine hums. However, users should note that ANC may not entirely suppress higher-frequency or sudden noises. Furthermore, volume levels and audio sampling rates can be adjusted to improve user comfort and enhance the clarity of spatial audio cues. It is essential to maintain safe volume levels to avoid auditory fatigue or potential long-term hearing damage. These considerations ensure that the system remains practical, adaptable, and effective for diverse environmental conditions and user preferences.

C. Object Recognition System

The object recognition system can provide the type of object, its position, and its distance based on data from the time-of-flight and web cameras. The object recognition system in the mobility aid for the visually impaired can be implemented using a machine learning algorithm based on CNNs to identify the object type. CNNs are artificial neural

networks that can accurately identify objects in images [4]. CNNs can be developed using vision-based navigation. Vision-based navigation is a technique that uses camera visual data to determine position and orientation relative to the environment. Objects can be detected using a camera to find the centroid, representing the object's location. The results are conveyed to the user as an audio description through earphones. The flowchart of the object recognition system is shown in Fig. 5. Segmentation is performed on the camera frame to play audio based on the detected object's horizontal center of mass position (`xc`). The frame is divided into three equal horizontal sections. These sections are defined as the left, center, and right areas. All three areas are of equal size. The distance units in meters are converted into indices and then used to select the appropriate audio file corresponding to the detected object's distance. The distance classification process aims to simplify the grouping of certain distance ranges to improve computational efficiency. The audio for the object's distance is limited within each 25 cm range, so the distance data from the time-of-flight camera is divided every 25 cm.

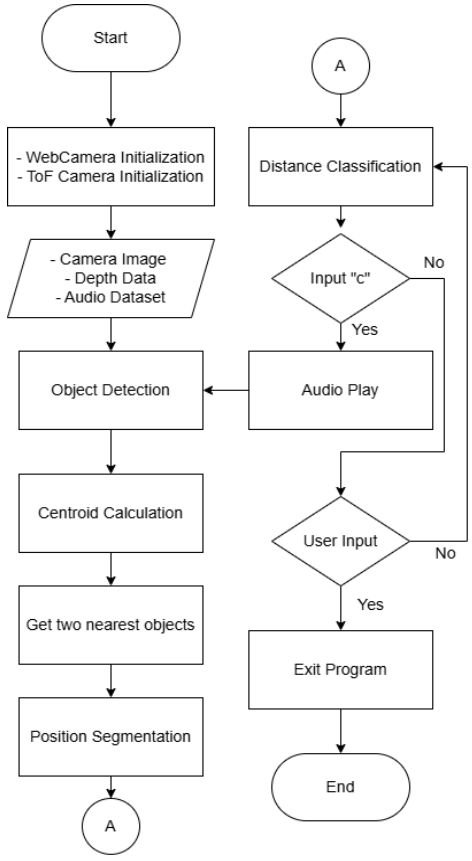
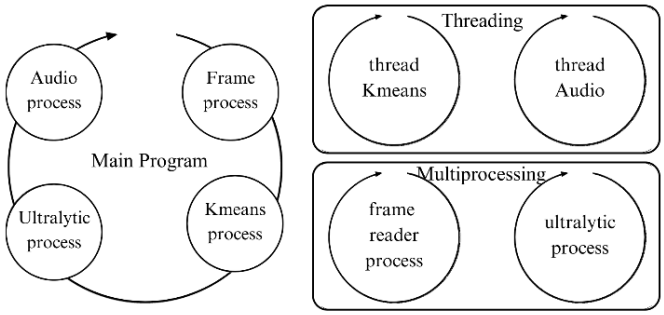


Fig. 5. Flowchart of the object recognition system

There are limitations in object type recognition within this system. The object recognition system utilizes YoloV8, which can detect 80 different types of objects without additional training. However, the testing in this study focused on five randomly selected objects: Bottle, Cup, Scissors, Human, and Mobile Phone. Moreover, the number of object types that can be recognized significantly affects computational load. The recognized object types could be specified to include only those frequently encountered by visually impaired individuals, thereby reducing computational demands to mitigate the impact in future research.

D. Multitasking System using Concurrent Programming

The program consists of threads and subprocesses that allow each system to operate simultaneously. The program will create two threads: one for performing calculations using the K-Means algorithm to compute the distance and position of object obstacles and another for playing audio based on the data generated by the system. The program will also create two subprocesses: one for recording raw data from the sensor and performing distance and position calculations for object recognition and another for object recognition to process the images captured by the sensor. A comparison of the concurrent programming architecture and the ILD is shown in Fig. 6.



(a) ILD (b) Concurrent

Fig. 6. Comparison of concurrent program and ILD

Fig. 6 shows the fundamental differences in the main functions between systems using ILD and concurrent programming systems. It can be observed that the ILD system executes the main functions sequentially with only one processing loop. In contrast, the concurrent programming system has multiple loops, including threads and subprocesses, that communicate and operate based on their respective inputs. A limitation of the infinite loop design method is its suboptimal use of multi-core capabilities on the Raspberry Pi and its inability to handle inputs or interrupts. In contrast, concurrent programming allows the program to receive input from cameras or sensors while performing other tasks without waiting for the previous task to be completed. This feature is critical for mobility aids for the visually impaired, as such assistive technologies require dynamic systems that can respond quickly to obstacles.

In concurrent programming, risks such as race conditions and crashes frequently arise. A race condition occurs when multiple threads simultaneously access and modify shared resources, leading to unpredictable outcomes. This issue is mitigated using mechanisms like threading. Lock and queue—queue to ensure thread-safe operations. Crashes can also result from deadlocks, where threads are stuck waiting for each other to release resources, or from resource contention, where multiple threads compete for limited system resources. Maintaining a consistent locking order or using timeouts on locks is essential to prevent deadlocks. Multiprocessing is preferred over threading to bypass Python's Global Interpreter Lock (GIL) for CPU-bound tasks. Proper thread management and thorough testing ensure stability and reliability in concurrent programs. Moreover,

the computational load of the system is not significantly affected when detecting a single object compared to multiple objects, as it utilizes the same inference model. The system still processes the entire frame if it detects one or multiple objects. However, slight differences occur in the post-processing stage when detecting multiple objects. In this case, the system generates a more significant number of bounding boxes and requires more memory for labeling. Nevertheless, there is no significant difference in computational load in the overall process, which primarily focuses on object recognition [96]-[98].

III. RESULTS AND DISCUSSION

A. Obstacle Detection

Using the K-Means algorithm, the object detection system was tested to assess the success of detecting the nearest objects. The testing was conducted by positioning the time-of-flight camera at various angles toward several obstacles. Success parameters were determined based on two factors: success in detecting the presence of obstacles and the object selection results. The success of detecting the obstacle presence indicates that the object detection system can detect the nearest object. The object selection result shows how accurately the system can select an object as a detected object. The test evaluated five objects of different sizes at 1.5 meters. The objects tested were a human, a wall, a wooden rack, a bucket, and a door. The results of the testing are presented in Table I.

TABLE I. OBJECT DETECTION SYSTEM TESTING

Object Type	Description	Object Selection
Human	Successfully detected	Perfect
Wall	Successfully detected	Imperfect
Wooden Rack	Successfully detected	Perfect
Bucket	Successfully detected	Imperfect
Door	Successfully detected	Perfect

Table I shows that the system successfully detected all objects. However, there are imperfections in the Bucket and Wall object selection. In the case of Wall selection, inaccuracies arise due to the extensive surface area of the wall, causing the system to primarily detect the nearest portion, which is the section directly in front of the user. Wall surfaces located to the side or beyond the detection frame of the time-of-flight camera remain undetected because their absolute distances from the time-of-flight camera do not fall within the nearest cluster. For the bucket, selection imperfections occur at certain distances where the bucket, positioned on the floor, does not exhibit significant depth variation compared to the surrounding floor. Consequently, the system may partially misidentify floor regions with similar absolute distances as part of the closest detected object. However, when detected at closer distances, the object selection process for the bucket improves, as the absolute distance of the floor in that region then falls within the nearest cluster. The Wall selection imperfection generally does not require mitigation, as the detected central portion of the wall sufficiently represents its distance. Furthermore, objects with extensive surface areas, such as walls, do not necessitate full selection. Conversely, inaccuracies in Bucket selection are more challenging to mitigate solely through calibration and may require additional algorithms or systems to prevent floor misclassification. Nevertheless, the inaccuracies in bucket selection do not significantly affect the overall distance accuracy, as the system calculates the mean distance of the second quadrant, and the selected floor and bucket pixels generally share similar depth values. The screenshot captured on Raspberry Pi in Fig. 7 shows an example of the test. A silhouette resembling a human figure can be observed in Fig. 7, representing the first cluster of the K-Means algorithm. This silhouette is composed of points corresponding to individual pixels within the frame. Pixels closer to the sensor appear brighter, whereas those at greater distances appear darker. Objects beyond the time-of-flight camera's maximum detection range of 4 meters are rendered in black. The bounding box is also constructed using the outermost points of the detected silhouette, forming the bounding box edges.

Fig. 7. Testing with human obstacle objects

Compared to the AI-Based Visual Aid with Integrated Reading Assistant [99] and the Assistive Cane with Visual Odometry [100], the proposed obstacle detection system demonstrates a key advantage in its adaptability to diverse environments due to its unsupervised K-Means clustering approach. Unlike deep-learning-based methods, which require extensive dataset training and rely on predefined object labels for classification, the proposed system autonomously segments objects based on depth information without prior knowledge of their specific characteristics. This eliminates the need for labeled training data, allowing the system to generalize more effectively across different environments and obstacle types. In contrast, the AI-based visual Aid [99] employs a combination of camera and ultrasonic sensors, which, while effective for static object detection, lacks the flexibility to classify unknown obstacles dynamically. Similarly, the Assistive Cane with Visual Odometry [100] relies on a structured approach to processing visual data, necessitating optimal camera positioning and parameter adjustments to enhance detection accuracy. The unsupervised nature of the proposed system ensures greater robustness and scalability, enabling real-time adaptation to varied environmental conditions without the computational overhead of deep learning models.

B. Accuracy of Obstacle Detection Distance Measurement

The accuracy of the obstacle detection distance measurement was tested to determine the accuracy of area-based distance measurements. The testing involved comparing the actual measured distance of an object with the

measurement made by the prototype device. The test was conducted using a measurement tape aligned with the sensor as a reference for the actual distance of the object from the sensor. The reference object measured was cardboard with dimensions 65×90cm. The test was conducted by taking 12 data points at distances ranging from 0.25 to 3 meters in increments of 0.25 meters using three measurement methods. Distance measurements were taken from direct measurements displayed in the Raspberry Pi terminal. Fig. 8 shows the calibrated measurement graph using linear regression. The testing resulted in an average calibration error of 0.026 m compared to the actual measurement values using the measurement tape. This result provides an average error value like the distance measurement results using the stereo vision method [38].

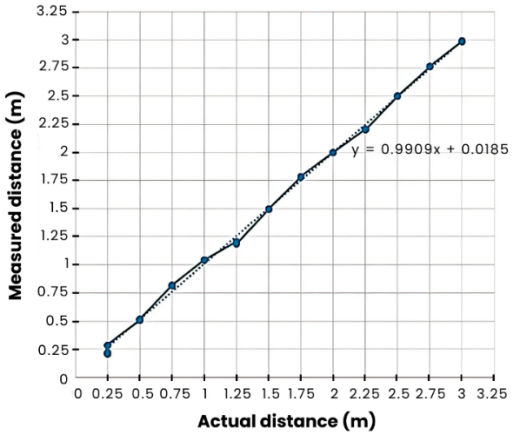


Fig. 8. Calibrated measurement graph

In the distance calculation algorithm, the second quadrant filters the time-of-flight camera distance measurements when there are pixels within the bounding box that are either too close, such as noise, or too far, such as regions outside the nearest cluster. Additionally, based on measurement results, the second quadrant exhibits the smallest error compared to the first, third, and fourth quadrants. However, errors still occur because the system relies on the average distance of the second quadrant. This quadrant is assumed to represent the distance of each pixel on the obstacle's surface. Since obstacle surfaces are not always perfectly flat, the average value from the second quadrant may not always correspond to the nearest point of the object. It is recommended that a new algorithm be developed that replaces the bounding box with an object selection method that conforms to the surface shape of the obstacle to mitigate this error. This approach would increase the likelihood that the second quadrant accurately represents the closest distance to the object.

Compared to existing assistive technologies, the proposed system demonstrates better distance measurement accuracy and broader detection coverage through its time-of-flight-based sensing approach and quadrant-based filtering technique. The Millimeter-Wave Radar Cane [27] employs a 122 GHz radar sensor for distance measurement, leveraging range alignment techniques to enhance detection accuracy. While radar-based systems exhibit high robustness in varied environmental conditions, their distance resolution is lower, making them less effective for precise near-field measurements. In contrast, the proposed time-of-flight system achieves an average calibration error of 0.026 m, offering superior precision in detecting stationary obstacles within a 0.25 to 3 meters range. Similarly, the Smart Assistive System for Visually Impaired People [33] utilizes ultrasonic sensors for obstacle detection. However, ultrasonic sensors have a narrow detection angle, typically ranging between 15° and 30°, meaning they can only detect obstacles directly in front of the sensor. In contrast, the proposed time-of-flight camera-based system provides a significantly wider field of view, allowing for the simultaneous detection of objects across a broader spatial range. This advantage ensures that users receive more comprehensive environmental awareness, reducing the risk of undetected obstacles outside the narrow sensing path of ultrasonic-based systems.

Additionally, the object distance measurement system refines distance calculations by reducing noise from outliers and ensuring that only the most reliable depth values contribute to the final measurement. Unlike visual odometry-based approaches, such as the Assistive Cane with Visual Odometry [27], which require camera motion tracking for distance estimation, the proposed method does not rely on positional changes, making it more stable for static object detection. Therefore, the proposed time-of-flight camera-based approach offers a more precise, computationally efficient, and spatially comprehensive distance measurement method, outperforming radar, ultrasonic, and vision-based alternatives in near-field obstacle detection.

C. Accuracy of Object Detection Angle Measurement

The device tested the angle measurement system to evaluate the performance of the angle measurements. The testing was conducted only on the horizontal axis due to the limitations of the measuring tool, and the horizontal axis measurements can represent the angle measurements on the vertical axis since lens distortion is identical on both axes. The angle measurements were carried out using a 5×3 cm object placed 2 meters away from the sensor at a radius of 40 cm, 60 cm, 80 cm, and 1 meter, with a horizontal angle range from -20° to 20° at 10° intervals. Three trials were conducted for each angle. The test compared the measured angles with the actual angles. Fig. 9 shows the angle measurement system test. The accuracy of the angle measurement was evaluated by calculating the difference between the actual and measured angles. The test result yielded the maximum average error and was calculated at 2.80% in measurement testing in 40 cm. The error occurred due to the limitations of the actual angle measuring equipment and the imperfections in object detection, where the centroid may not be positioned precisely at the object's center. The average error of measurement is shown in Table II.

Fig. 9. The angle measurement system testing

TABLE II. OBJECT DETECTION ANGLE MEASUREMENT

Distance (cm)	Average Error (°)	Average Error (%)
40	1.12	2.80%
60	1.19	1.98%
80	1.08	1.35%
100	1.21	1.21%

In the object detection angle measurement algorithm, the angle is determined based on the center point of the bounding box. Consequently, error increases when an object deviates significantly from the rectangular shape of the bounding box. Conversely, if the object closely aligns with the bounding box shape, the angle detection error decreases. As shown in Table II, the average error decreases as the object distance increases. This phenomenon occurs due to human error during testing. The evaluation was conducted by detecting a vertically positioned cane. If the cane was not perfectly perpendicular to the surface or was tilted, the bounding box, which should ideally enclose the cane accurately, became distorted, leading to detection inaccuracies. It is recommended that a new algorithm, like the distance calculation method, be developed to mitigate this error. Replacing the bounding box with an object selection algorithm that conforms to the obstacle's surface shape can reduce errors, as the algorithm would detect the object's center of mass rather than relying solely on the bounding box centroid.

Compared to existing systems, the proposed Object Detection Angle Measurement demonstrates superior precision and stability in static object detection. The Assistive Cane with Visual Odometry [100] estimates object angles using camera-based motion tracking, but its reliance on continuous movement introduces accumulative errors over time, making it less reliable for precise static angle measurements. In contrast, the proposed system directly calculates angles from bounding box centroids, ensuring consistent accuracy without dependence on motion. Similarly, the Integrating Wearable Haptics and Obstacle Avoidance System [21] employ an RGB-D camera for depth-based object detection, but it lacks a dedicated angle computation framework, limiting its effectiveness in providing precise angular positioning. The proposed time-of-flight-based approach achieves a low average error, offering higher spatial resolution and structured angle estimation compared to these vision-based systems. While minor errors arise from bounding box centroid misalignment, integrating an object selection algorithm that conforms to obstacle contours could further enhance precision. Thus, the proposed system provides a more stable, accurate, and computationally efficient alternative for real-time angular measurement in assistive visual aid applications.

D. HRTF Spatial Audio

Spatial audio testing was conducted by comparing the frequency response graphs of the spatial audio channels on the left and right. The frequency response graphs were displayed using the DAW Adobe Audition, with frequency responses taken at the peak amplitude of the audio. The frequency response graph shows the left channel in red and the right channel in yellow. Fig. 10 displays the spatial audio frequency response graph with an azimuth of -30° and an elevation of 0°. The graph indicates a significant amplitude drop on the right channel at frequencies below 30 Hz and a slight amplitude drop across the overall frequencies.

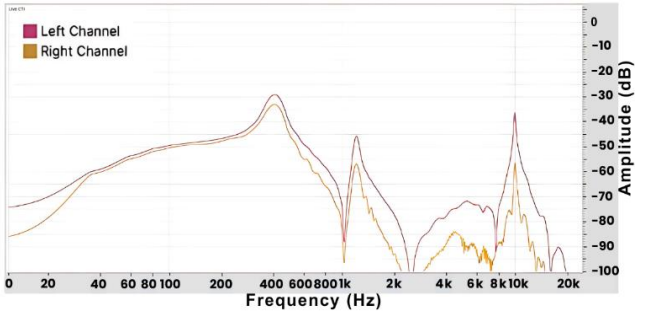


Fig. 10. Frequency response graph of spatial audio with an azimuth of -30° and an elevation of 0°

Fig. 11 displays the frequency response graph of HRTF audio with an azimuth of 0°and elevation of 0°. Visually, there is no significant difference in the frequency response because the azimuth and elevation are set to 0°, meaning the sound source is directly in front of the listener, resulting in the sound being perceived similarly by both the right and left ears.

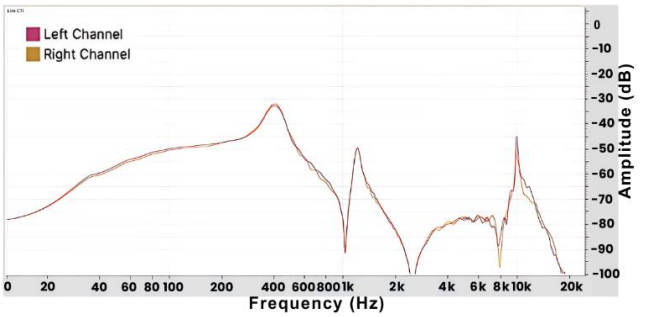


Fig. 11. The frequency response graph of HRTF audio has an azimuth of 0° and an elevation of 0°.

The implementation of spatial audio uses a dataset consisting of 21 datasets. Each dataset, lasting less than 1 second, provides dynamic information about the object's position, enabling the user to track changes in the object's movement. Spatial audio offers a more effective representation of an object's position than audio cues, as it continuously updates the object's position information when played repeatedly, according to the input data from the obstacle detection system [11]. Spatial audio can also blend with ambient sounds since it is static audio, like a "beep," which does not interfere with conversations, in contrast to audio cues or audio descriptions in previous studies that would be disruptive if played repeatedly. There will be variable delays in repeated playback. The farther the obstacle, the greater the delay, as its value is determined by the obstacle detection algorithm. This effect is intended to enhance depth perception by reinforcing the sense of distance.

The proposed HRTF-based spatial audio system enhances spatial awareness and depth perception for visually impaired users by dynamically adjusting frequency responses based on object azimuth and elevation. Unlike the AI-Based Visual Aid with Integrated Reading Assistant [99], which relies on static audio cues or verbal descriptions, the proposed system continuously updates the spatial positioning of obstacles

through directional filtering, allowing for real-time object tracking without disrupting the user's auditory environment. Additionally, while the Integrating Wearable Haptics and Obstacle Avoidance System [21] employs haptic feedback for navigation, tactile-based approaches may become cognitively demanding in complex environments with multiple obstacles. In contrast, the HRTF spatial audio method provides a non-intrusive auditory representation, allowing users to perceive both direction and distance without requiring direct physical interaction. Furthermore, the incorporation of variable delay effects, where delays increase with obstacle distance, reinforces depth perception, a feature that is not present in traditional binaural or cue-based audio systems. This ensures that the proposed system delivers a more immersive and continuous spatial awareness experience, facilitating real-time obstacle tracking while minimizing interference with natural environmental sounds and conversations.

E. Accuracy of Object Type Recognition

The accuracy of object type detection is tested to obtain the percentage of success in detecting and identifying each object class. In the tests conducted, objects from five classes were tested, each evaluated 20 times. The detection accuracy of object type is shown in Fig. 12. The results show that several objects, such as bottles, cups, and humans, have a high recognition rate in the object type identification process, with 100% accuracy at close and far distances. The accuracy of scissors is higher at 0.5 meters, at 80%, but only 30% at 1.5 meters. The accuracy for a mobile phone at 0.5 meters is 50%, and 40% at a far distance.

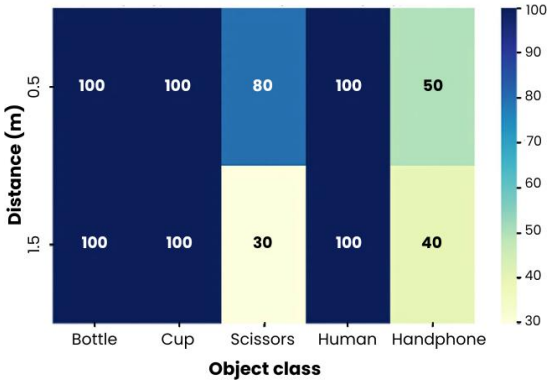


Fig. 12. Detection accuracy of object type

The low accuracy of the scissors aligns with the findings in [36], which were caused by the object's detailed shape and the small, more complex size of the scissors. Smaller objects at greater distances exhibit lower accuracy because they occupy fewer pixels, leading to a loss of fine-grained details crucial for recognition. Smaller objects like scissors are particularly affected, as their intricate shapes blend with the background, reducing detection accuracy. Lower resolution and sensor limitations can cause blurring, while perspective distortions and lighting variations further complicate object recognition. As a result, larger and well-defined objects like bottles and humans maintain high accuracy, while smaller, complex objects experience significant detection drops. However, through various enhancements such as attention mechanisms, multi-scale feature fusion, and improved loss functions, the detection accuracy loss for small objects can be mitigated. These advancements are crucial for applications requiring precise small object detection in complex scenarios [101]-[104].

F. Accuracy of Object Centroid Detection

The accuracy of object centroid detection was tested to obtain the percentage of successful object centroid detections. The objects used for accuracy testing were in 5 object classes and three angular variations, namely -15º, 0º, and 15º, where 0º is the central viewpoint, positive angles are measured to the right of the Line of Sight (LOS), and negative angles are measured to the left of the LOS. Each object class was tested 18 times. The test data are shown in Table III. The results show that the error in detecting the object's centroid varies. It can be observed in Table III that the centroid error for the mobile phone was detected with RMSE 0.07 at an angle of 0º, while for other angle variations, the mobile phone object was 3.52 and 2.08.

TABLE III. ACCURACY OF OBJECT CENTROID DETECTION WITH A DISTANCE OF 0.5 METERS

Object Type	RMSE of Centroid Position Detection		
	-15 º	**0 º**	**15 º**
Bottle	2.69	3.36	5.38
Cup	6.11	2.04	1.82
Scissors	5.66	0.36	1.51
Human	2.18	2.88	5.67
Mobile Phone	3.52	0.07	2.08

The influence of position variations, whether left, center or correct, does not show a significant difference in the error magnitude. The variations in the accuracy of object centroid detection using YoloV8 can be attributed to several factors, including object size, shape, viewing angle, distance from the camera, and dataset limitations [105]. Smaller objects, such as scissors and mobile phones, tend to have higher centroid detection errors at non-central angles because fewer pixels represent the object in the frame, making precise localization more difficult. Additionally, irregularly shaped objects like scissors may result in inconsistent bounding box placements, leading to centroid misalignment. Viewing angle effects also play a significant role, as objects positioned at -15º and 15º relative to the Line of Sight (LOS) may experience perspective distortion, altering their perceived shape and position. This is evident in the mobile phone detection results, where the RMSE at 0º was only 0.07 but significantly increased to 3.52 at -15º and 2.08 at 15º, likely due to perspective changes affecting bounding box positioning. Another contributing factor is the bounding box detection limitations of YoloV8, which relies on learned features from training data that may not always align perfectly with real-world conditions. Variations in lighting, background clutter, or object occlusion can lead to inconsistent bounding box placements, further impacting centroid calculations [106], [107].

G. Accuracy of Object Position Classification

The accuracy of object position classification was tested to determine the percentage of success in classifying the position of objects. The objects tested for accuracy were in the bottle, cup, scissors, human, and mobile phone categories,

with variations of close distance (0.5 meters) and far distance (1.5 meters). Each object category was tested 20 times. The test data visualization is shown in Fig. 13.

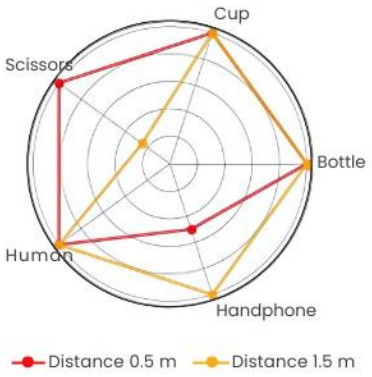


Fig. 13. Classification accuracy of object in center position

The accuracy of all five types of objects was 100% when positioned in the center. However, when the objects were positioned to the left or right, some objects, specifically scissors and mobile phones, did not achieve 100% accuracy. The inability to detect scissors and mobile phones resulted in lower accuracy for position classification. The object position recognition system identifies objects, determines their center point, and retrieves the distance data from the time-of-flight camera for that point. Errors may occur due to misalignment between the web camera and the time-of-flight camera. Calibration of both cameras is performed only initially using trigonometric calculations based on the actual distance and their relative positioning, resulting in a formula for frameshift and scaling in the time-of-flight camera. Shifts in camera placement, such as those caused by vibrations, can lead to distance discrepancies when retrieving object distance data from the time-of-flight camera. Implementing a dynamic calibration algorithm that adjusts distance when necessary without modifying the code is recommended to mitigate this error. Additionally, a more stable camera mounting can help minimize shifts due to vibrations, ensuring consistent camera positioning within the system.

H. Accuracy of Object Distance Detection

The accuracy of object distance detection was tested to determine the percentage of success in detecting the distance of each object. The objects tested for accuracy were a human, a bottle, a chair, a bag, and a laptop. Each object category was identified 45 times with five distance variations (0.5 m, 0.75 m, 1 m, 1.25 m, and 1.5 m) and three position variations (left, center, right). The detection accuracy of object distance is shown in Fig. 14. The accuracy of distance detection for each type of object varied and fluctuated. However, there was a tendency for the accuracy to decrease as the distance increased. According to polynomial regression, the optimal object distance detection range is between 0.5 and 1 meter, as it resulted in a relatively high average accuracy. Fig. 15 shows the distribution of object distance detection data. The data distribution shows varying detected distance values, but there is a tendency for the values to cluster around the actual distance. Based on the graph visualization, it is observed that the deviation in detected distance values increases as the actual distance increases with the testing distance. The average detection results obtained at the distance variations of 0.5 m, 0.75 m, 1 m, 1.25 m, and 1.5 m are 0.51 m, 0.75 m, 1 m, 1.27 m, and 1.56 m, respectively.

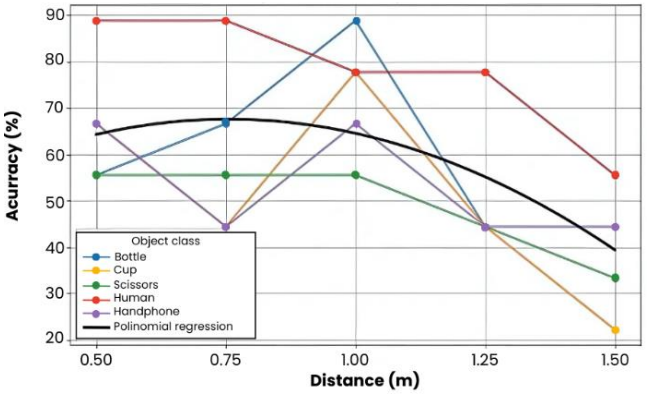


Fig. 14. Detection accuracy of object distance

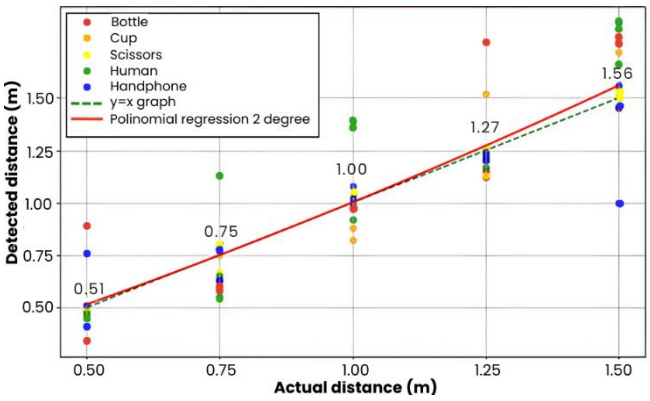


Fig. 15. Distribution of distance data compared to actual distance

Based on Fig. 14, the highest accuracy is achieved for humans, while the lowest accuracy is observed for scissors. Additionally, as the object's distance increases, the accuracy of distance detection decreases. This result is influenced by the object's size and distance, which affects the number of pixels captured in the frame. Larger and closer objects produce more pixels, reducing the likelihood of incorrect distance estimation. Moreover, the calibration between the time-of-flight and web cameras can also contribute to lower accuracy when only a few pixels represent the object. The object's distance is obtained by retrieving the pixel's distance, representing the object's center point. Since calibration is performed only initially, any camera displacement may reduce accuracy over time. Several improvements are suggested, including developing an additional algorithm to compare distances from multiple pixels instead of relying solely on the center point, implementing a dynamic calibration algorithm for both the time-of-flight camera and web camera to adjust for positional shifts, and enhancing the camera mounting system to prevent displacement and maintain alignment to mitigate this issue. Moreover, based on Fig. 15, there is a linear relationship between the actual object distance and the distance detected by the system. Therefore, linear regression can be applied to improve the accuracy of object distance detection, and the resulting regression equation can be integrated into the code for more precise distance estimation.

I. Overall Object Recognition Accuracy

The overall testing of the object recognition subsystem aims to determine the accuracy of detecting the object type, object center of mass location, object position, and object distance. The overall testing accuracy is obtained using Mean Absolute Percentage Error (MAPE) to determine the error in detecting the object type and position. The testing is conducted using five variations of object types (bottle, cup, scissors, human, and bag), three variations of position (left, center, right), and variations of short distance (0.5 m) and long distance (1.5 m). The visualization of the test data is shown in Fig. 16. The highest accuracy value is observed for the human object. In contrast, other object types exhibit fluctuating values depending on the distance variation. The accuracy is better when the object is positioned in the center than when positioned on the left or right. The accuracy is generally higher at a short distance (0.5 meters) than at a long distance (1.5 meters). Detection accuracy decreases as the distance increases, though the extent of the effect varies across different object types. The differences in the impact of distance on accuracy are likely due to the varying dimensions of the objects tested. The accuracy tends to be higher when the object is positioned at the center compared to the left or right positions. The result indicates that the overall testing accuracy is 54%.

Fig. 16. Detection accuracy with variations of objects and distances 1.5 meters

The error in testing results for object recognition accuracy can be attributed to several factors, including the number of pixels representing the object at different distances, camera displacement, and variations in object size within the frame. When objects are farther away (1.5 meters), fewer pixels represent them in the captured image, reducing the system's ability to accurately detect the object's type, centroid, and position. This is particularly evident in smaller objects like scissors and mobile phones, which suffer from greater detection errors at increased distances. Additionally, inaccuracies in distance estimation can occur due to slight shifts in camera positioning, which affect calibration and lead to inconsistencies in object localization. Camera displacement may result from minor vibrations or misalignment after initial calibration, impacting the accuracy of distance measurement and object placement within the frame. The size of an object also plays a crucial role, as larger objects like humans and bags tend to maintain higher detection accuracy due to their significant presence in the frame, whereas smaller objects contribute to more fluctuating results. Improvements should be made to each subsystem, including Object Type Recognition, Object Centroid Detection, Object Distance Detection, and Object Position Classification to mitigate these issues. Enhancing the accuracy of each component will contribute to a more reliable overall system.

The proposed object recognition subsystem demonstrates strength in integrating multi-parameter detection, encompassing object type classification, centroid localization, position estimation, and distance measurement, making it more comprehensive than traditional detection methods. Compared to the Design of Blind Guiding Robot Based on Speed Adaptation and Visual Recognition [31], which employs YOLOv5 for object classification, the proposed system provides a broader range of spatial information rather than solely focusing on object presence. Similarly, the Millimeter-Wave Radar Cane excels in distance estimation and motion tracking [27] but lacks the capability to perform detailed object classification and precise position detection. Despite achieving an overall accuracy of 54%, the proposed system offers a more holistic approach to scene understanding, capturing multiple attributes rather than prioritizing a single detection metric.

J. Obstacle Detection Processing Time

The testing compares the processing time of the concurrent programming system and the infinite loop program. The test is conducted by varying the constant movement of objects detected by both systems to determine the average processing time. Fig. 17 shows the time testing results of the obstacle detection system with variations in obstacle movement.

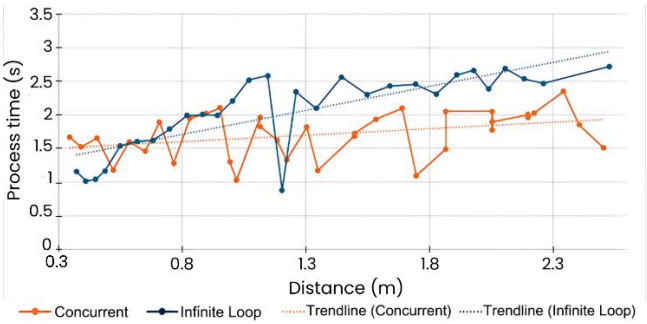


Fig. 17. Graph of obstacle detection system time process

Fig. 17 demonstrates that the processing time of the infinite loop program increases as the object distance grows. This increase is attributed to the extended 3D audio delay, which postpones subsequent processing steps. In contrast, the concurrent programming system separates 3D audio processing from distance detection tasks, including frame acquisition and K-Means algorithm implementation. This separation ensures that the 3D audio playback does not delay distance detection. Additionally, for distances below 0.6 meters, the infinite loop program processes data faster, achieving faster execution, as its processing time depends on the object's proximity. The system's processing time in an infinite loop remains constant and is longer than that of concurrent programming. The object's lateral movement distance remains unchanged, ensuring the 3D audio delay

remains consistent. However, concurrent programming achieves better processing efficiency by dividing the workload into three parallel processes: frame acquisition, K-Means algorithm execution, and 3D audio playback. These processes run concurrently across multiple cores while sharing the same memory through shared variables. Utilizing multiple cores maximizes CPU workload distribution compared to a single-core approach, improving processing time. However, based on Table IV, the overall average processing time of the concurrent programming approach remains faster.

Based on testing the obstacle detection system control with variations in movement direction, the concurrent programming system can respond to obstacle changes more quickly than the infinite loop system design. Concurrent programming can improve the average processing time by up to 19.23%. This performance improvement in concurrent programming demonstrates that implementing concurrent programming provides significant benefits in real-time applications where rapid response is critical.

K. Object Recognition Processing Time

The test compares the processing time between the concurrent programming system and the infinite loop program. The test was conducted by presenting variations of objects in a video for recognition by both systems, with the same playback duration for each object, to determine the average processing time. Fig. 18 compares processing times between the two systems for object recognition in the video. Table IV shows the time improvement in the system process. Concurrent programming offers significant time improvement in object recognition compared to conventional programming. During the scissor recognition test, concurrent programming improved the object recognition processing time by 55.39%. The process saw an improvement of 47.75% for the cup object test while recognizing another cup object yielded a time improvement of 57.15%. Overall, concurrent programming has proven effective in accelerating the object recognition process, with an average time improvement of approximately 53.43%.

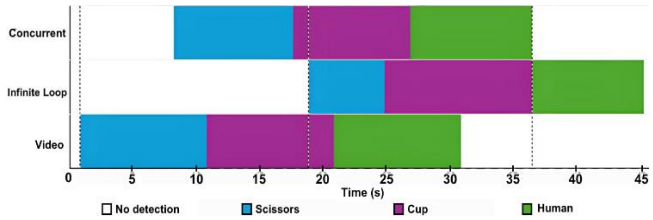


Fig. 18. Object detection program results for objects in video based on time

TABLE IV. OBJECT RECOGNITION PROCESSING TIME

Object	Process Time (s)		Time Improvement (%)
	Infinite Loop	*Concurrent*	
Human	16.50	7.07	57.15
Cup	14.91	7.79	47.75
Scissor	18.92	8.44	55.39
Average			53.43

Object recognition using CNN operates continuously without being triggered by a sensor, ensuring that both systems consistently detect objects in front of them. In CNN-based processing, frame acquisition and object recognition using YoloV8 impose a significant computational load on the system. When these processes are executed on the same core, the system's FPS decreases, and frame acquisition experiences delays during YoloV8 processing, leading to delayed or incorrect object recognition. In contrast, the concurrent programming system assigns frame acquisition and object recognition to separate cores, improving the system's FPS and preventing delays during YoloV8 processing. This parallel execution enhances real-time performance and ensures accurate and timely object detection.

L. Overall System Processing Time

This section creates a timing diagram for the concurrent programming system, starting from the timing diagram for each thread and subprocess to the touch sensor timing. The *threadK-means* is a thread that works based on input from the *frame_reader_process* to calculate the distance to the nearest object. The *threadAudio* is a thread responsible for calculating the index in calling audio samples and looping through those samples with a frequency that adjusts to the distance of the obstacle. The *frame_reader_process* is a subprocess responsible for capturing data from the time-of-flight and web cameras and calculating the distance to objects recognized by the *ultralytic_process. Ultralytic_process* is a subprocess that works when given an image input, recognizes the image, and provides the result for the following process. The touch sensor is the input that provides the start signal for the object recognition system. The four processes and the response from the touch sensor are shown in Fig. 19.

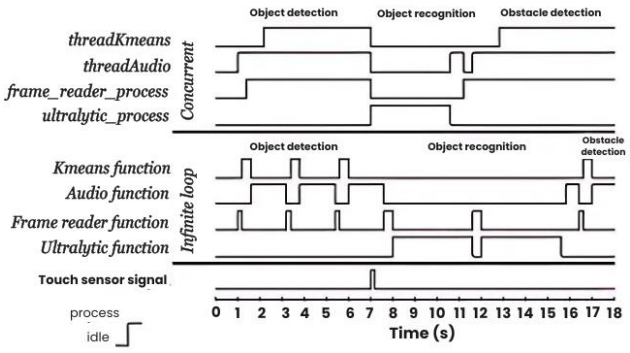


Fig. 19. Comparison timing diagram of concurrent and infinite loops

In Fig. 19, the black dashed line indicates when the object recognition process begins, at which point processes other than *ultralytic_process* will start. When the object recognition process finishes, the *threadAudio* will run at the black dashed line to provide the object description. The red dashed line shows when the obstacle detection process resumes after playing the description audio. Additionally, it can be observed that *thread K-means* can run concurrently with other processes and will start processing once *frame_reader_process* begins. The *threadAudio* will always run unless the *ultralytic_process* is active, in which case *threadAudio* will enter an idle state. Similarly, the *frame_reader_process* will always run unless the *ultralytic_process* is active; in this case, the *frame_reader_process* will also enter an idle state. The *ultralytic_process* will run when the touch sensor enters a high state, causing the other systems to enter an idle state. The

timing diagram shows that all systems have been controlled as intended and can operate concurrently to maximize efficiency. The overall processing times for the infinite loop and concurrent system are shown in Table V and Table VI, respectively.

TABLE V. OVERALL PROCESSING TIME FOR INFINITE LOOP

Object	Processing Time (s)		Total Time (s)
	Obstacle detection	*Object recognition*	
Human	2.20	8.60	10.80
Cup	2.24	8.63	10.87
Scissor	2.40	8.65	11.05

TABLE VI. OVERALL PROCESSING TIME FOR CONCURRENT

Object	Processing Time (s)		Total Time (s)
	Obstacle detection	*Object recognition*	
Human	1.64	3.66	5.30
Cup	1.65	3.86	5.51
Scissor	1.73	3.75	5.48

Table V and Table VI show that the system with concurrent programming can respond to changes in obstacles and recognize objects more quickly than the infinite loop system. A comparison of overall processing time for infinite loop and concurrent programming is shown in Table VII.

TABLE VII. COMPARISON OF OVERALL PROCESSING TIME FOR INFINITE LOOP AND CONCURRENT

Object	Overall Processing Time (s)		Time Improvement (%)
	Infinite Loop	*Concurrent*	
Human	10.80	5.30	50.93
Cup	10.87	5.51	49.31
Scissor	11.05	5.48	50.41
Average			50.22

Table VII shows that the overall processing time for obstacle detection and object recognition with concurrent programming is quicker than the infinite loop system. For three objects, concurrent programming has a process time improvement with an average of 50.22% compared to the infinite loop program. The difference in processing time between the infinite loop and concurrent programming is primarily due to differences in core utilization and task distribution within the system. Concurrent programming achieves faster distance change detection and object recognition by fully utilizing the cores of the Raspberry Pi 4B, eliminating task waiting times that would otherwise cause tasks to halt while others are being executed. This allows tasks to run concurrently, significantly improving overall efficiency.

Additionally, concurrent programming enhances object recognition performance as it can successfully identify objects on the first attempt. In contrast, the infinite loop system often fails to do so due to frame delays. In the infinite loop system, the initial object recognition attempt may result in no detection because the object has not yet entered the frame, requiring the process to be repeated. Due to these advantages, concurrent programming delivers more real-time and responsive results, making it particularly beneficial for applications requiring fast processing and immediate feedback.

M. Testing with Visually Impaired Users

The testing was conducted by allowing visually impaired users to use the device directly in a controlled and supervised environment. The test involved two fully blind participants aged between 20 and 30 years. In this controlled setting, the users navigated the environment using the device while several individuals stood stationary to act as obstacles, in addition to environmental objects such as parked cars, walls, fences, and other abstract obstacles. When encountering an obstacle, users were instructed to point at it and maneuver around it to ensure the device functioned effectively as a mobility aid. Testing with visually impaired users is shown in Fig. 20.

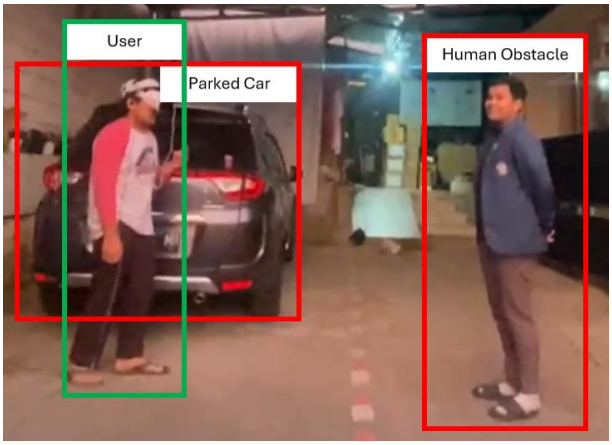


Fig. 20. Picture of testing setup

Fig. 20 shows that encounter 1 standing human obstacle and a parked car obstacle. The test results showed that users could avoid and locate stationary obstacles. However, some adjustments were required to improve their ability to detect fast-moving objects approaching from the side, while slow to normal speed moving objects were still identifiable. The device's design, which conforms to the user's facial structure, was found to be comfortable during movement, followed head movements well, and remained securely in place. Additionally, the system was powered by a 10,000mAh power bank, which provided sufficient energy without issues throughout the test period.

Despite these advantages, there are limitations in the use of spatial audio. Users require an adaptation period to become familiar with the spatial audio. Moreover, overlapping environmental sounds can reduce the effectiveness of the device. Another challenge is that the device currently requires assistance from a caregiver to properly fit it onto the user due to the specific placement of the headset and power cables. Several improvements can be implemented to mitigate these challenges. First, integrating Active Noise Cancelling (ANC) technology can help reduce background noise, mainly low-frequency sounds like engine hums. However, users should be aware that ANC may not completely suppress sudden or high-frequency sounds. Adjusting the volume levels and audio sampling rates can also enhance the clarity of spatial audio cues while maintaining safe listening levels to prevent auditory fatigue or potential long-term hearing damage.

In addition, improving the detection of fast-moving objects can be achieved by implementing predictive motion

tracking, which can further anticipate the movement of approaching objects, allowing users more time to react. The headset and power connection layout can be redesigned to make it more intuitive for visually impaired users to wear and adjust without assistance to increase user independence in setting up the device. Implementing voice-guided setup instructions can also provide step-by-step guidance on how to wear and position the device correctly. Furthermore, a structured spatial audio training program can be introduced to mitigate the challenge of adapting to spatial audio cues. This program would involve step-by-step familiarization exercises in controlled environments before real-world use. Users can begin by recognizing directional sounds in quiet settings, then gradually progress to noisier environments to improve their ability to differentiate relevant audio cues from background noise. Additionally, interactive training sessions, such as virtual simulations or gamified exercises, could help users develop a stronger spatial awareness through repeated practice. Implementing these strategies allows the system to become more effective, user-friendly, and adaptable to various real-world conditions. These improvements will not only enhance the device's usability but also contribute to greater mobility, confidence, and independence for visually impaired users.

Compared to the Design of Blind Guiding Robot Based on Speed Adaptation and Visual Recognition [31], which utilizes YOLOv5 for object detection and motion adaptation, the proposed system focuses on real-time spatial awareness through a combination of time-of-flight camera and HRTF spatial audio feedback. This enables more natural and intuitive navigation, particularly in environments where static and dynamic obstacles coexist. Similarly, while the Millimeter-Wave Radar Cane [27] excels in motion tracking and obstacle differentiation, its lower spatial resolution limits fine-grained object classification. In contrast, the proposed system offers a broader range of environmental perception through object type recognition, centroid detection, and position estimation. The Smart Assistive System for Visually Impaired People [33] relies on ultrasonic sensors for obstacle detection, which are effective for general navigation but lack detailed spatial representation. In contrast, the proposed system enhances depth perception and localization accuracy by combining a time-of-flight camera with K-Means Clustering.

Additionally, wearable haptic systems like the Integrating Wearable Haptics and Obstacle Avoidance System [21] offer direct tactile feedback but may become overwhelming in complex environments, whereas the proposed HRTF-based spatial audio feedback provides a non-intrusive, ambient guidance mechanism, allowing users to maintain situational awareness without excessive cognitive load. Also, user testing with visually impaired participants demonstrated the system's effectiveness in navigating static obstacles but highlighted areas for improvement, particularly in detecting fast-moving objects approaching from the side. While the system remains securely positioned and aligns with head movements, the spatial audio component requires an adaptation period, like challenges noted in other audio-based navigation aids [99].

IV. CONCLUSIONS

Mobility aids for the visually impaired based on machine learning using a time-of-flight camera and spatial audio have been described in this research, integrating a time-of-flight camera, web camera, and touch sensor with K-Means, CNNs, and concurrent programming on the Raspberry Pi 4B to detect objects and obstacles around the user. The novelty of this research lies in the obstacle detection system; this research excels in detecting dynamic objects using the time-of-flight camera and the K-Means algorithm without requiring training data or an internet connection. The use of spatial audio distinguishes this research from previous studies that utilized audio descriptions. Spatial audio modifies sound to include direction and distance, leveraging the blind's heightened sensitivity to sound. Spatial audio allows users to estimate the position of the nearest obstacle using a universal language, making it accessible to everyone. In the object recognition system, users can recognize objects that are useful for daily activities by touching a sensor to trigger the system. Unlike previous studies, objects are recognized and described in terms of distance and position relative to the user. All systems run concurrent programming, offering better execution time than previous studies based on infinite loop programming. The device is designed to conform to the user's facial structure and incorporates flexible audio output, ensuring adaptability to individual needs. This research advances mobility aids for the visually impaired by refining system design, sensor integration, algorithmic efficiency, and overall user experience. However, several limitations remain. The device has a maximum detection range of 4 meters, reduced performance under high-intensity sunlight, and requires user adaptation to spatial audio, which may be affected by external noise. Additionally, object recognition is limited to 80 predefined categories, and minor inaccuracies persist in obstacle distance measurement and object recognition. Future research should focus on enhancing the bounding box algorithm to better conform to object contours, optimizing spatial audio samples for improved listener comfort, and implementing a robust trigonometric calibration algorithm between the time-of-flight camera and the web camera. Furthermore, expanding the training dataset to improve object recognition across different perspectives and lighting conditions, designing a more stable camera mounting system to mitigate direct sunlight exposure, and developing a user-friendly interface tailored for the visually impaired will further enhance the device’s usability. Lastly, providing structured training guidelines will help users familiarize themselves with the system’s unique features, particularly spatial audio, ensuring optimal adoption and effectiveness.

REFERENCES

[1] A. Bonello, E. Francalanza, and P. Refalo, “Smart and Sustainable Human-Centred Workstations for Operators with Disability in the Age of Industry 5.0: A Systematic Review,” *Sustainability*, vol. 16, no. 1, p. 281, Dec. 2023, doi: 10.3390/su16010281.

[2] N. Jayasekara, B. Kulathunge, H. Premaratne, I. Nilam, S. Rajapaksha, and J. Krishara, “Revolutionizing Accessibility: Smart Wheelchair Robot and Mobile Application for Mobility, Assistance, and Home Management,” *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 27–53, Dec. 2023, doi: 10.18196/jrc.v5i1.20057.

[3] M. H. Abidi, A. Noor Siddiquee, H. Alkhalefah, and V. Srivastava, “A comprehensive review of navigation systems for visually impaired

individuals," *Heliyon*, vol. 10, no. 11, p. e31825, Jun. 2024, doi: 10.1016/j.heliyon.2024.e31825.

[4] J. Madake, S. Bhatlawande, A. Solanke, and S. Shilaskar, "A Qualitative and Quantitative Analysis of Research in Mobility Technologies for Visually Impaired People," *IEEE Access*, vol. 11, pp. 82496–82520, 2023, doi: 10.1109/ACCESS.2023.3291074.

[5] Y. Lei, S. L. Phung, A. Bouzerdoum, H. Thanh Le, and K. Luu, "Pedestrian Lane Detection for Assistive Navigation of Vision-Impaired People: Survey and Experimental Evaluation," *IEEE Access*, vol. 10, pp. 101071–101089, 2022, doi: 10.1109/ACCESS.2022.3208128.

[6] M. Itair, I. Shahrour, and I. Hijazi, "The Use of the Smart Technology for Creating an Inclusive Urban Public Space," *Smart Cities*, vol. 6, no. 5, pp. 2484–2498, Sep. 2023, doi: 10.3390/smartcities6050112.

[7] M. Singh, J. Chauhan, M. S. Kanroo, S. Verma, and P. Goyal, "IPCRF: An End-to-end Indian Paper Currency Recognition Framework for Blind and Visually Impaired People," *IEEE Access*, 2022, doi: 10.1109/ACCESS.2022.3202007.

[8] G. I. Okolo, T. Althobaiti, and N. Ramzan, "Assistive systems for visually impaired persons: challenges and opportunities for navigation assistance," *Sensors*, vol. 24, no. 11, p. 3572, 2024.

[9] J. Wang, E. Liu, Y. Geng, X. Qu, and R. Wang, "A Survey of 17 Indoor Travel Assistance Systems for Blind and Visually Impaired People," *IEEE Trans Hum Mach Syst*, vol. 52, no. 1, pp. 134–148, Feb. 2022, doi: 10.1109/THMS.2021.3121645.

[10] S. M. Aslam and S. Samreen, "Gesture Recognition Algorithm for Visually Blind Touch Interaction Optimization Using Crow Search Method," *IEEE Access*, vol. 8, pp. 127560–127568, 2020, doi: 10.1109/ACCESS.2020.3006443.

[11] K. M. Masal, S. Bhatlawande, and S. D. Shingade, "Development of a visual to audio and tactile substitution system for mobility and orientation of visually impaired people: a review," *Multimed Tools Appl*, vol. 83, no. 7, pp. 20387–20427, Aug. 2023, doi: 10.1007/s11042-023-16355-0.

[12] T. Halbach, K. S. Fuglerud, T. Fyhn, K. Kjæret, and T. A. Olsen, "The Role of Technology for the Inclusion of People with Visual Impairments in the Workforce," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13309 LNCS, pp. 466 – 478, 2022, doi: 10.1007/978-3-031-05039-8_34.

[13] B. Leporini, M. Rosellini, and N. Forgione, "Designing assistive technology for getting more independence for blind people when performing everyday tasks: an auditory-based tool as a case study," *J Ambient Intell Humaniz Comput*, vol. 11, no. 12, pp. 6107–6123, Dec. 2020, doi: 10.1007/s12652-020-01944-w.

[14] M. Tuttle and E. W. Carter, "Examining High-Tech Assistive Technology Use of Students With Visual Impairments," *J Vis Impair Blind*, vol. 116, no. 4, pp. 473–484, Jul. 2022, doi: 10.1177/0145482X221120265.

[15] V. H. Le, "Visual Slam and Visual Odometry Based on RGB-D Images Using Deep Learning: A Survey," *Journal of Robotics and Control (JRC)*, vol. 5, no. 4, pp. 1050–1079, 2024, doi: 10.18196/jrc.v5i4.22061.

[16] F. Merchan, M. Poveda, D. E. Cáceres-Hernández, and J. E. Sanchez-Galan, "Indoor Navigation Aid Systems for the Blind and Visually Impaired Based on Depth Sensors," *Examining Optoelectronics in Machine Vision and Applications in Industry 4.0*, pp. 187–223, 2021, doi: 10.4018/978-1-7998-6522-3.ch007.

[17] A. Paramarthalingam, J. Sivaraman, P. Theerthagiri, B. Vijayakumar, and V. Baskaran, "A deep learning model to assist visually impaired in pothole detection using computer vision," *Decision Analytics Journal*, vol. 12, p. 100507, Sep. 2024, doi: 10.1016/j.dajour.2024.100507.

[18] J. H. Han *et al.*, "Mobility Support with Intelligent Obstacle Detection for Enhanced Safety," *Optics*, vol. 5, no. 4, pp. 434–444, Oct. 2024, doi: 10.3390/opt5040032.

[19] P. Powell, F. Pätzold, M. Rouygari, M. Furtak, S. M. Kärcher, and P. König, "Helping Blind People Grasp: Evaluating a Tactile Bracelet for Remotely Guiding Grasping Movements," *Sensors*, vol. 24, no. 9, May 2024, doi: 10.3390/s24092949.

[20] S. Alzalabny, O. Moured, K. Müller, T. Schwarz, B. Rapp, and R. Stiefelhagen, "Designing a Tactile Document UI for 2D Refreshable Tactile Displays: Towards Accessible Document Layouts for Blind People," *Multimodal Technologies and Interaction*, vol. 8, no. 11, Nov. 2024, doi: 10.3390/mti8110102.

[21] F. Barontini, M. G. Catalano, L. Pallottino, B. Leporini, and M. Bianchi, "Integrating Wearable Haptics and Obstacle Avoidance for the Visually Impaired in Indoor Navigation: A User-Centered Approach," *IEEE Trans Haptics*, vol. 14, no. 1, pp. 109–122, Jan. 2021, doi: 10.1109/TOH.2020.2996748.

[22] F. E. Z. El-Taher, L. Miralles-Pechuan, J. Courtney, K. Millar, C. Smith, and S. McKeever, "A Survey on Outdoor Navigation Applications for People With Visual Impairments," 2023, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2023.3244073.

[23] Z. Yu and M. Hu, "Real Environment Warning Model for Visually Impaired People in Trouble on the Blind Roads Based on Wavelet Scattering Network," *IEEE Access*, vol. 12, pp. 82156–82167, 2024, doi: 10.1109/ACCESS.2024.3412328.

[24] K. C. Shahira and A. Lijiya, "Towards Assisting the Visually Impaired: A Review on Techniques for Decoding the Visual Data from Chart Images," *IEEE Access*, vol. 9, pp. 52926–52943, 2021, doi: 10.1109/ACCESS.2021.3069205.

[25] K. M. Safiya and R. Pandian, "Real-Time Photo Captioning for Assisting Blind and Visually Impaired People Using LSTM Framework," *IEEE Sens Lett*, vol. 7, no. 11, pp. 1–4, Nov. 2023, doi: 10.1109/LSENS.2023.3327565.

[26] S. Malla, P. K. Sahu, S. Patnaik, and A. K. Biswal, "Obstacle Detection and Assistance for Visually Impaired Individuals Using an IoT-Enabled Smart Blind Stick," *Revue d'Intelligence Artificielle*, vol. 37, no. 3, pp. 783–794, Jun. 2023, doi: 10.18280/ria.370327.

[27] E. Cardillo, C. Li, and A. Caddemi, "Millimeter-wave radar cane: A blind people aid with moving human recognition capabilities," *IEEE J Electromagn RF Microw Med Biol*, vol. 6, no. 2, pp. 204–211, Jun. 2022, doi: 10.1109/JERM.2021.3117129.

[28] B. Mangesh, K. Shruti, P. Gaurav, S. Mahek, and P. Jay, "Next Generation Smart Stick for Blind People using Assistive Technology," *International Journal of Performability Engineering*, vol. 20, no. 5, p. 282, 2024, doi: 10.23940/ijpe.24.05.p3.282291.

[29] M. Bamdad, D. Scaramuzza, and A. Darvishy, "SLAM for Visually Impaired People: A Survey," *IEEE Access*, 2024, doi: 10.1109/ACCESS.2024.3454571.

[30] S. Khan, S. Nazir, and H. U. Khan, "Analysis of Navigation Assistants for Blind and Visually Impaired People: A Systematic Review," *IEEE Access*, vol. 9, pp. 26712–26734, 2021, doi: 10.1109/ACCESS.2021.3052415.

[31] L. Zhang, K. Jia, J. Liu, G. Wang, and W. Huang, "Design of Blind Guiding Robot Based on Speed Adaptation and Visual Recognition," *IEEE Access*, vol. 11, pp. 75971–75978, 2023, doi: 10.1109/ACCESS.2023.3296066.

[32] P. Mejia, L. C. Martini, F. Grijalva, and A. M. Zambrano, "CASVI: Computer Algebra System Aimed at Visually Impaired People. Experiments," *IEEE Access*, vol. 9, pp. 157021–157034, 2021, doi: 10.1109/ACCESS.2021.3129106.

[33] U. Masud, T. Saeed, H. M. Malaikah, F. U. Islam, and G. Abbas, "Smart Assistive System for Visually Impaired People Obstruction Avoidance Through Object Detection and Classification," *IEEE Access*, vol. 10, pp. 13428–13441, 2022, doi: 10.1109/ACCESS.2022.3146320.

[34] J. Guerreiro, Y. Kim, R. Nogueira, S. A. Chung, A. Rodrigues, and U. Oh, "The Design Space of the Auditory Representation of Objects and Their Behaviours in Virtual Reality for Blind People," *IEEE Trans Vis Comput Graph*, vol. 29, no. 5, pp. 2763–2773, May 2023, doi: 10.1109/TVCG.2023.3247094.

[35] S. B. Sukhavasi, S. B. Sukhavasi, K. Elleithy, A. El-Sayed, and A. Elleithy, "A hybrid model for driver emotion detection using feature fusion approach," *International journal of environmental research and public health*, vol. 19, no. 5, p. 3085, 2022.

[36] A. R. See, B. G. Sasing, and W. D. Advincula, "A Smartphone-Based Mobility Assistant Using Depth Imaging for Visually Impaired and Blind," *Applied Sciences*, vol. 12, no. 6, p. 2802, Mar. 2022, doi: 10.3390/app12062802.

[37] O. Duran and B. Turan, "Vehicle-to-vehicle distance estimation using artificial neural network and a toe-in-style stereo camera,"

Measurement, vol. 190, p. 110732, Feb. 2022, doi: 10.1016/j.measurement.2022.110732.

[38] A. Zaarane, I. Slimani, W. Al Okaishi, I. Atouf, and A. Hamdoun, "Distance measurement system for autonomous vehicles using stereo camera," *Array*, vol. 5, p. 100016, Mar. 2020, doi: 10.1016/j.array.2020.100016.

[39] P. Johanns, T. Haucke, and V. Steinhage, "Automated distance estimation for wildlife camera trapping," *Ecol Inform*, vol. 70, p. 101734, Sep. 2022, doi: 10.1016/j.ecoinf.2022.101734.

[40] J. Wei *et al.*, "Dual UAV-based cross view target position measurement using machine learning and Pix-level matching," *Measurement*, vol. 236, p. 115039, Aug. 2024, doi: 10.1016/j.measurement.2024.115039.

[41] X. Li *et al.*, "Three-dimensional reconstruction based on binocular structured light with an error point filtering strategy," *Optical Engineering*, vol. 63, no. 3, Mar. 2024, doi: 10.1117/1.OE.63.3.034102.

[42] J. Kim, "Camera-Based Net Avoidance Controls of Underwater Robots," *Sensors*, vol. 24, no. 2, p. 674, Jan. 2024, doi: 10.3390/s24020674.

[43] X. Geng *et al.*, "A Lightweight Approach for Passive Human Localization Using an Infrared Thermal Camera," *IEEE Internet Things J*, vol. 9, no. 24, pp. 24800–24811, Dec. 2022, doi: 10.1109/JIOT.2022.3194714.

[44] C. Vela, G. Fasano, and R. Opromolla, "Pose determination of passively cooperative spacecraft in close proximity using a monocular camera and AruCo markers," *Acta Astronaut*, vol. 201, pp. 22–38, Dec. 2022, doi: 10.1016/j.actaastro.2022.08.024.

[45] Y. Yin, D. Gao, K. Deng, and Y. Lu, "Vision-based autonomous robots calibration for large-size workspace using ArUco map and single camera systems," *Precis Eng*, vol. 90, pp. 191–204, Oct. 2024, doi: 10.1016/j.precisioneng.2024.08.010.

[46] E. Adil, M. Mikou, and A. Mouhsen, "A novel algorithm for distance measurement using stereo camera," *CAAI Trans Intell Technol*, vol. 7, no. 2, pp. 177–186, Jun. 2022, doi: 10.1049/cit2.12098.

[47] A. Zaarane, I. Slimani, W. Al Okaishi, I. Atouf, and A. Hamdoun, "Distance measurement system for autonomous vehicles using stereo camera," *Array*, vol. 5, p. 100016, Mar. 2020, doi: 10.1016/j.array.2020.100016.

[48] O. Duran, B. Turan, and M. Kaya, "Machine-learning-based ensemble regression for vehicle-to-vehicle distance estimation using a toe-in style stereo camera," *Measurement*, vol. 240, p. 115540, Jan. 2025, doi: 10.1016/j.measurement.2024.115540.

[49] A. Abdelsalam, M. Mansour, J. Porras, and A. Happonen, "Depth accuracy analysis of the ZED 2i Stereo Camera in an indoor Environment," *Rob Auton Syst*, vol. 179, p. 104753, Sep. 2024, doi: 10.1016/j.robot.2024.104753.

[50] P. K. Duba, N. P. B. Mannam, and R. P, "Stereo vision based object detection for autonomous navigation in space environments," *Acta Astronaut*, vol. 218, pp. 326–329, May 2024, doi: 10.1016/j.actaastro.2024.02.032.

[51] J. Wang, Y. Guan, Z. Kang, and P. Chen, "A Robust Monocular and Binocular Visual Ranging Fusion Method Based on an Adaptive UKF," *Sensors*, vol. 24, no. 13, p. 4178, Jun. 2024, doi: 10.3390/s24134178.

[52] M. Carfagni *et al*., "Metrological and critical characterization of the Intel D415 stereo depth camera," *Sensors*, vol. 19, no. 3, p. 489, 2019.

[53] X. Ding, L. Xu, H. Wang, X. Wang, and G. Lv, "Stereo depth estimation under different camera calibration and alignment errors," *Applied Optics*, vol. 50, no. 10, pp. 1289-1301, 2011.

[54] A. Simonelli, S. R. Buló, L. Porzi, E. Ricci, and P. Kontschieder, "Towards Generalization Across Depth for Monocular 3D Object Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 767–782. doi: 10.1007/978-3-030-58542-6_46.

[55] B. Wei *et al.*, "Remote Distance Binocular Vision Ranging Method Based on Improved YOLOv5," *IEEE Sens J*, vol. 24, no. 7, pp. 11328–11341, Apr. 2024, doi: 10.1109/JSEN.2024.3359671.

[56] W. Wang *et al.*, "A multi-degree-of-freedom monitoring method for slope displacement based on stereo vision," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 13, pp. 2010–2027, Jul. 2024, doi: 10.1111/mice.13173.

[57] G. Li, Z. Xu, Y. Zhang, C. Xin, J. Wang, and S. Yan, "Calibration method for binocular vision system with large field of view based on small target image splicing," *Meas Sci Technol*, vol. 35, no. 8, p. 085006, Aug. 2024, doi: 10.1088/1361-6501/ad4381.

[58] M. H. Conde, T. Kerstein, B. Buxbaum, and O. Loffeld, "Near-Infrared, Depth, Material: Towards a Trimodal Time-of-Flight Camera," in *2020 IEEE SENSORS*, IEEE, Oct. 2020, pp. 1–4. doi: 10.1109/SENSORS47125.2020.9278760.

[59] M. H. Conde, T. Kerstein, B. Buxbaum, and O. Loffeld, "Live Demonstration: a Trimodal Time-of-Flight Camera Featuring Material Sensing," in *2020 IEEE SENSORS*, IEEE, Oct. 2020, pp. 1–1. doi: 10.1109/SENSORS47125.2020.9278928.

[60] Y. Liu, Y. Fan, Z. Wu, J. Yao, and Z. Long, "Ultrasound-Based 3-D Gesture Recognition: Signal Optimization, Trajectory, and Feature Classification," *IEEE Trans Instrum Meas*, vol. 72, pp. 1–12, 2023, doi: 10.1109/TIM.2023.3235438.

[61] L. Qi, T. Zhang, K. Xu, H. Pan, Z. Zhang, and Y. Yuan, "A novel terrain adaptive omni-directional unmanned ground vehicle for underground space emergency: Design, modeling and tests," *Sustain Cities Soc*, vol. 65, p. 102621, Feb. 2021, doi: 10.1016/j.scs.2020.102621.

[62] V. A. Grishin, "Accuracy of Relative Navigation Using Time-of-Flight Cameras," *J Spacecr Rockets*, vol. 60, no. 2, pp. 471–480, Mar. 2023, doi: 10.2514/1.A35079.

[63] Y. Song, C. Lu, F. Wu, Z. Cao, and X. Liang, "A method for evaluating 3D-TOF camera ranging performance," in *Sixth Symposium on Novel Optoelectronic Detection Technology and Applications*, H. Jiang and J. Chu, Eds., SPIE, Apr. 2020, p. 179. doi: 10.1117/12.2564703.

[64] M. H. Conde, "A Material-Sensing Time-of-Flight Camera," in *IEEE Sensors Letters*, vol. 4, no. 7, pp. 1-4, July 2020, doi: 10.1109/LSENS.2020.3005042.

[65] C. Mao, Y. Song, and J. Chen, "A lightweight adaptive random testing method for deep learning systems," *Softw Pract Exp*, vol. 53, no. 11, pp. 2271–2295, Nov. 2023, doi: 10.1002/spe.3256.

[66] K. Manjari, M. Verma, and G. Singal, "A survey on Assistive Technology for visually impaired," *Internet of Things*, vol. 11, p. 100188, Sep. 2020, doi: 10.1016/j.iot.2020.100188.

[67] D. Zhu *et al.*, "Unified Audio-Visual Saliency Model for Omnidirectional Videos With Spatial Audio," *IEEE Trans Multimedia*, vol. 26, pp. 764–775, 2024, doi: 10.1109/TMM.2023.3271022.

[68] T. Rudzki, D. Murphy, and G. Kearney, "A DAW-Based Interactive Tool for Perceptual Spatial Audio Evaluation," in *145th Audio Engineering Society Convention*, Oct. 2018.

[69] C. Schissler, A. Nicholls, and R. Mehra, "Efficient HRTF-based Spatial Audio for Area and Volumetric Sources," *IEEE Trans Vis Comput Graph*, vol. 22, no. 4, pp. 1356–1366, Apr. 2016, doi: 10.1109/TVCG.2016.2518134.

[70] N. Javeri, P. B. Dutta, K. Sunder, and K. Jain, "A Machine Learning Approach to Predicting Personalized Head Related Transfer Functions and Headphone Equalization from Video Capture Data," in *2023 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*, pp. 1–9, 2023, doi: 10.1109/I3DA57090.2023.10289448.

[71] M. I. Thariq Hussan, D. Saidulu, P. T. Anitha, A. Manikandan, and P. Naresh, "Object Detection and Recognition in Real Time Using Deep Learning for Visually Impaired People," *International Journal of Electrical and Electronics Research*, vol. 10, no. 2, pp. 80–86, 2022, doi: 10.37391/IJEER.100205.

[72] J. Cruz Antony, G. M. Karpura Dheepan, V. K, V. Vikas, and V. Satyamitra, "Traffic sign recognition using CNN and Res-Net," *EAI Endorsed Transactions on Internet of Things*, vol. 10, Feb. 2024, doi: 10.4108/eetiot.5098.

[73] M. Fuad *et al*., "Towards Controlling Mobile Robot Using Upper Human Body Gesture Based on Convolutional Neural Network," *Journal of Robotics and Control (JRC)*, vol. 4, no. 6, pp. 856–867, Dec. 2023, doi: 10.18196/jrc.v4i6.20399.

[74] A. H. N. Hidayah, S. Ahmad Radzi, N. A. Razak, W. H. M. Saad, Y. C. Wong, and A. A. Naja, "Disease Detection of Solanaceous Crops Using Deep Learning for Robot Vision," *Journal of Robotics and Control (JRC)*, vol. 3, no. 6, pp. 790–799, Dec. 2022, doi: 10.18196/jrc.v3i6.15948.

[75] K. Zhang, Y. Wang, S. Shi, Q. Wang, C. Wang, and S. Liu, "Improved yolov5 algorithm combined with depth camera and embedded system for blind indoor visual assistance," *Sci Rep*, vol. 14, no. 1, p. 23000, Oct. 2024, doi: 10.1038/s41598-024-74416-2.

[76] M. S. Khoirom, M. Sonia, B. Laikhuram, J. Laishram, and D. Singh, "Comparative Analysis of Python and Java for Beginners," *International Research Journal of Engineering and Technology*, 2020.

[77] J. Sundnes, *Introduction to Scientific Programming with Python*. Cham: Springer International Publishing, 2020, doi: 10.1007/978-3-030-50356-7.

[78] A. Castello, R. M. Gual, S. Seo, P. Balaji, E. S. Quintana-Orti, and A. J. Pena, "Analysis of Threading Libraries for High Performance Computing," *IEEE Transactions on Computers*, vol. 69, no. 9, pp. 1279–1292, Sep. 2020, doi: 10.1109/TC.2020.2970706.

[79] T. Triwiyanto, W. Caesarendra, V. Abdullayev, A. A. Ahmed, and H. Herianto, "Single Lead EMG signal to Control an Upper Limb Exoskeleton Using Embedded Machine Learning on Raspberry Pi," *Journal of Robotics and Control (JRC)*, vol. 4, no. 1, pp. 35–45, Feb. 2023, doi: 10.18196/jrc.v4i1.17364.

[80] S. Venkateshalu and S. Deshpande, "Optimized CNN Learning Model With Multi-Threading for Forgery Feature Detection in Real-Time Streaming Approaches," in *Digital Twin and Blockchain for Smart Cities*, pp. 101–116, 2024, doi: 10.1002/9781394303564.ch6.

[81] V. Manjunath and M. Baunach, "A framework for static analysis and verification of low-level RTOS code," *Journal of Systems Architecture*, vol. 154, p. 103220, Sep. 2024, doi: 10.1016/j.sysarc.2024.103220.

[82] Z. Yan, H. Wang, Z. Wang, X. Liu, and Q. Ning, "Imaging simulation of the AMCW ToF camera based on path tracking," *Appl Opt*, vol. 61, no. 18, p. 5474, Jun. 2022, doi: 10.1364/AO.458940.

[83] N. Sanmartin-Vich, J. Calpe, and F. Pla, "Shot Noise Analysis for Differential Sampling in Indirect Time of Flight Cameras," *IEEE Signal Process Lett*, vol. 30, pp. 46–49, 2023, doi: 10.1109/LSP.2023.3236263.

[84] Y. Fang, X. Wang, Z. Sun, K. Zhang, and B. Su, "Study of the depth accuracy and entropy characteristics of a ToF camera with coupled noise," *Opt Lasers Eng*, vol. 128, p. 106001, May 2020, doi: 10.1016/j.optlaseng.2020.106001.

[85] Y. Du, Z. Jiang, J. Tian, and X. Guan, "Modeling, analysis, and optimization of random error in indirect time-of-flight camera," *Opt Express*, vol. 33, no. 2, p. 1983, Jan. 2025, doi: 10.1364/OE.547731.

[86] J. Lee and M. Gupta, "Mitigating AC and DC Interference in Multi-ToF-Camera Environments," *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 12, pp. 15005–15017, Dec. 2023, doi: 10.1109/TPAMI.2023.3307564.

[87] W. Zhang, P. Song, Y. Bai, H. Geng, Y. Wu, and Z. Zheng, "Non-systematic noise reduction framework for ToF camera," *Opt Lasers Eng*, vol. 180, p. 108324, Sep. 2024, doi: 10.1016/j.optlaseng.2024.108324.

[88] F. Ahmed, M. H. Conde, P. L. Martinez, T. Kerstein, and B. Buxbaum, "Pseudo-Passive Time-of-Flight Imaging: Simultaneous Illumination, Communication, and 3D Sensing," *IEEE Sens J*, vol. 22, no. 21, pp. 21218–21231, Nov. 2022, doi: 10.1109/JSEN.2022.3208085.

[89] D. Poirier-Quinot and B. F. G. Katz, "Assessing the Impact of Head-Related Transfer Function Individualization on Task Performance: Case of a Virtual Reality Shooter Game," *Journal of the Audio Engineering Society*, vol. 68, no. 4, 2020, doi: 10.17743/jaes.2020.0004ï.

[90] J. Wang, K. Qian, Y. Qiu, H. Zhang, and X. Xie, "A multi-attribute subjective evaluation method on binaural 3D audio without reference stimulus," *Applied Acoustics*, vol. 200, p. 109042, Nov. 2022, doi: 10.1016/j.apacoust.2022.109042.

[91] J. Zhao, D. Yao, J. Gu, and J. Li, "Efficient prediction of individual head-related transfer functions based on 3D meshes," *Applied Acoustics*, vol. 219, p. 109938, Mar. 2024, doi: 10.1016/j.apacoust.2024.109938.

[92] W. Ryu, S. Lee, and E. Park, "The Effect of Training on Localizing HoloLens-Generated 3D Sound Sources," *Sensors*, vol. 24, no. 11, p. 3442, May 2024, doi: 10.3390/s24113442.

[93] J. Wang, M. Liu, X. Wang, T. Liu, and X. Xie, "Prediction of head-related transfer function based on tensor completion," *Applied Acoustics*, vol. 157, p. 106995, Jan. 2020, doi: 10.1016/j.apacoust.2019.08.001.

[94] H. Liu, P. Yuan, B. Yang, G. Yang, and Y. Chen, "Head-related transfer function–reserved time-frequency masking for robust binaural sound source localization," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 1, pp. 26-33, 2022.

[95] J. M. Arend, F. Brinkmann, and C. Pörschmann, "Assessing spherical harmonics interpolation of time-aligned head-related transfer functions," *AES: Journal of the Audio Engineering Society*, vol. 69, no. 1–2, pp. 104–117, Feb. 2021, doi: 10.17743/JAES.2020.0070.

[96] R. R. de Alvarenga, L. A. V. Dias, and A. M. da Cunha, "Multtestlib: A Python package for performing unit tests using multiprocessing," *SoftwareX*, vol. 29, p. 101986, Feb. 2025, doi: 10.1016/j.softx.2024.101986.

[97] S. Yu. Gordleeva *et al.*, "Real-Time EEG–EMG Human–Machine Interface-Based Control System for a Lower-Limb Exoskeleton," *IEEE Access*, vol. 8, pp. 84070–84081, 2020, doi: 10.1109/ACCESS.2020.2991812.

[98] O. V. Doronin, "Improvement and comparison the performance of fuzzing testing algorithms for applications in Google Thread Sanitizer," *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, vol. 22, no. 4, pp. 734–741, Aug. 2022, doi: 10.17586/2226-1494-2022-22-4-734-741.

[99] M. A. Khan, P. Paul, M. Rashid, M. Hossain, and M. A. R. Ahad, "An AI-Based Visual Aid with Integrated Reading Assistant for the Completely Blind," *IEEE Trans Hum Mach Syst*, vol. 50, no. 6, pp. 507–517, Dec. 2020, doi: 10.1109/THMS.2020.3027534.

[100] J. Tang *et al.*, "Design and Optimization of an Assistive Cane With Visual Odometry for Blind People to Detect Obstacles With Hollow Section," *IEEE Sens J*, vol. 21, no. 21, pp. 24759–24770, 2022, doi: 10.1109/JSEN.2021.3115854.

[101] J. Cao, T. Zhang, L. Hou, and N. Nan, "An improved YOLOv8 algorithm for small object detection in autonomous driving," *J Real Time Image Process*, vol. 21, no. 4, p. 138, Aug. 2024, doi: 10.1007/s11554-024-01517-6.

[102] J. Qu *et al.*, "SS-YOLOv8: small-size object detection algorithm based on improved YOLOv8 for UAV imagery," *Multimed Syst*, vol. 31, no. 1, p. 42, Feb. 2025, doi: 10.1007/s00530-024-01622-3.

[103] G. Yao, S. Zhu, L. Zhang, and M. Qi, "HP-YOLOv8: High-Precision Small Object Detection Algorithm for Remote Sensing Images," *Sensors*, vol. 24, no. 15, p. 4858, Jul. 2024, doi: 10.3390/s24154858.

[104] T. Wu and Y. Dong, "YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition," *Applied Sciences*, vol. 13, no. 24, p. 12977, Dec. 2023, doi: 10.3390/app132412977.

[105] M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," *IEEE Access*, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[106] M. Talib, A. H. Y. Al-Noori, and J. Suad, "YOLOv8-CAB: Improved YOLOv8 for Real-time object detection," *Karbala International Journal of Modern Science*, vol. 10, no. 1, Jan. 2024, doi: 10.33640/2405-609X.3339.

[107] Q. Su and J. Mu, "Complex Scene Occluded Object Detection with Fusion of Mixed Local Channel Attention and Multi-Detection Layer Anchor-Free Optimization," *Automation*, vol. 5, no. 2, pp. 176–189, Jun. 2024, doi: 10.3390/automation5020011.