Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ENHANCING NAVIGATION FOR THE VISUALLY IMPAIRED THROUGH OBJECT DETECTION AND 3D AUDIO FEEDBACK

MONCEF AHARCHI[1], M'HAMED AIT KBIR[2]
[1 & 2] Intelligent Automation & BioMedGenomics Laboratory, STSM Doctoral center, Abdelmalek Essaadi University, Morocco
E-mail : [1]maharchi@uae.ac.ma , [2]maitkbir@uae.ac.ma

ID 55510 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
05-09-2024	05-09-2024	25-09-2024	29-09-2024

ABSTRACT

Assisting individuals with visual impairments in navigating their surroundings using technological equipment remains a challenging task due to challenges regarding movement, item and person identification, and engagement with the environment. Typically, these devices integrate sensors, visual mechanisms, and tactile or auditory feedback. This article proposes a vision system integrated with 3D audio feedback to improve navigation for the visually impaired people by providing a more intuitive knowledge of object placements along a path by modifying headphone sound level. This system consists of three primary components: firstly, depth calculation utilizing stereoscopic vision to generate a depth map; secondly, object recognition employing a YOLO neural network (CNN) for identifying common objects and Aruco tags for less common ones; and finally, the production of 3D audio based on the depth map and object locations. Subsequently, the user utilizes this spatial audio signal to navigate effectively. When an object is selected using voice commands, the system spell the detected objects names to provide users with direction and distance guidance. During real-world testing, this system has proven to be very helpful and precise in assisting visually impaired people with their navigation.

Keywords: *Visually Impaired, Navigation, Object Detection, 3D Sound, Stereoscopic Vision, Computer Vision, Neural Networks*

1. INTRODUCTION

Developing autonomous navigation tools for visually impaired individuals remains a significant challenge, despite advances in technology. According to World Health Organization (WHO) statistics from 2015, 39 million people are blind, and 256 million have vision impairments that, if untreated, could lead to permanent blindness. Currently, the most widely used self-navigation aid for the visually impaired is still the white cane, which has seen limited enhancements, such as the addition of ultrasonic sensors.

Despite innovations, including computer vision systems capable of recognizing path patterns [1], RFID systems detecting ground beacons for navigation assistance [2], and devices providing environmental information through tactile feedback on the tongue [3], precise navigation toward specific objects or destinations remains a largely unsolved issue. This paper focuses on addressing this gap by assisting visually impaired individuals in navigating toward specific objects or destinations along a route.

The system proposed in this research introduces a portable solution that provides navigation instructions through an audio-based feedback system. It incorporates advanced technologies such as neural networks, stereoscopic vision for depth estimation, and image processing techniques. By leveraging well-established stereoscopic vision concepts, the system estimates the depth of objects using cameras. Common objects are detected using widely adopted deep neural network architectures [5-10], which have demonstrated high precision across various industries. For less common objects, locations, or directions, ArUco markers can be affixed to enable identification through image processing techniques. This allows visually impaired individuals (PVI) to receive real-time navigation guidance and information about their surroundings.

The device alerts users to obstacles in their path via auditory feedback, tapping into the naturally heightened sensory abilities of visually impaired individuals [4]. By delivering 3D audio messages, users can better understand the spatial relationship between themselves and surrounding objects, allowing for more intuitive navigation.

This paper is structured as follows: Section 2 details the implementation of the proposed system, Section 3 evaluates its performance through test cases, and Section 4 discusses the limitations of this work and suggests potential improvements for future iterations of the system.

2. RELATED WORKS

Multiple tools leverage computer vision technologies to assist blind individuals. For instance, TapTapSee, a mobile app [11], employs computer vision and crowdsourcing to describe images captured by blind users within about 10 seconds. Blindsight's Text Detective [12] utilizes optical character recognition (OCR) to identify and read text from camera-captured images. Facebook is in the process of developing image captioning technology to enable blind users to engage in conversations about pictures [13]. Baidu recently showcased a demo video of the DuLight project [14], hinting at concepts of scene description and recognition of people, currency, merchandise, and crosswalk signals. However, these tools primarily focus on specific functionalities rather than offering a comprehensive visual sense for the blind. Moreover, they do not utilize spatial sound techniques to enhance the user experience.

In broader sensory substitution efforts, individuals like Daniel Kish, who is totally blind, have developed accurate echolocation abilities using "mouth clicks" for independent navigation tasks like biking and hiking [15]. Similarly, colorblind artist Neil Harbisson devised a device that translates color information into sound frequencies. The vOICe technology [16] takes an extreme approach by converting visual information into sound, associating height with pitch and brightness with loudness. However, these approaches are reported to involve challenging learning processes. In contrast, our approach utilizes visual recognition algorithms, enabling a more direct understanding of objects within a visual scene.

Moreover, researchers have explored the use of 3D sound technology to assist the blind. One study [17] introduced a system that employs spatial audio to aid in discovering points of interest in large, unfamiliar indoor environments, such as shopping malls. Another initiative [18] attempted to integrate 3D sound into GPS-based outdoor navigation. However, these approaches did not incorporate visual recognition techniques. The integration of object detection methods presents new possibilities for assisting indoor navigation among the blind and visually impaired [19].

A recent study proposes a novel deep learning-based approach to assist visually impaired individuals in identifying Indonesian banknotes, overcoming challenges related to varying denominations and imaging conditions. The system utilizes a Convolutional Neural Network (CNN) model specifically designed for banknote detection, incorporating modules for image capture, feature extraction, and classification. By leveraging a camera-based setup accessible via smartphones, the model achieved a high accuracy rate of 94.29% at the 60th epoch, using optimized kernel sizes of 3x3 and 2x2 for the convolutional layers. This approach demonstrates significant improvements over traditional methods, providing an efficient and accurate solution for real-time banknote recognition [37].

Understanding 2D images is a complex challenge in computer vision, extending beyond mere object identification to encompass scene comprehension. This capability is crucial for tasks like image captioning, visual question answering (VQA), and image retrieval. Recent advances have seen graph neural networks (GNNs) become integral to these tasks, providing a natural representation of object relationships within an image. A comprehensive survey reviews this evolving field, detailing various graph types, GNN models, and future directions. This survey is notable for being the first to focus extensively on the use of GNNs in image captioning, VQA, and image retrieval [38].

A recent study introduces a smart stick system designed to aid visually impaired individuals in navigating their surroundings. This system integrates multiple technologies—such as ultrasonic, infrared, and water sensors; alarm modules like buzzers and voice statements; and GPS/GSM systems—into a single device. The smart stick serves as a vision assistant, providing obstacle detection and location-tracking capabilities. The real-world tests showed positive results, achieving an average obstacle avoidance accuracy of 88.75%. The system offers a comprehensive solution for visually impaired people, combining all essential features to facilitate safer and more independent navigation [39].

Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED METHODOLOGY

Multiple The primary goal of this system is to guide the PVI during navigation by delivering directional cues, contextual information about the locations visited, and information about the distance and type of objects in front of them in order to make their navigation smoother and more autonomous.

Several queries concerning the functionality of the system arise in this context:

1. How the distance between the PVIs and the object is measured?
2. What are the methods used for object identification?
3. What are the mechanisms that facilitate communication between the PVIs and the system?
4. How does the system effectively convey information to the PVI?

In order to answer these important questions, the system is designed around three essential modules, as shown in following Figure:

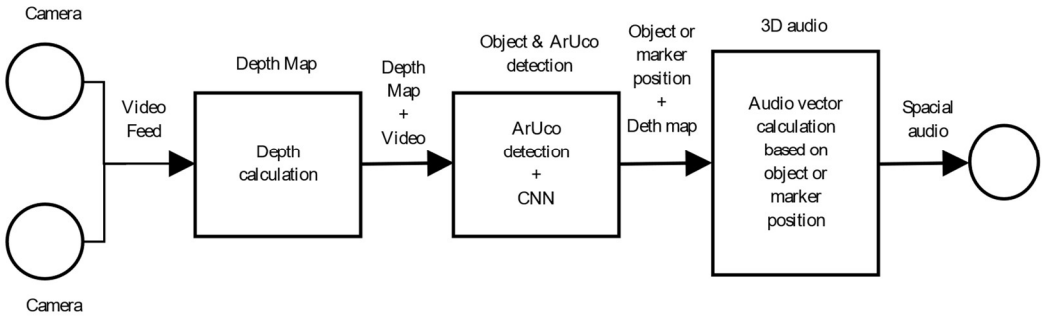

Figure 1: System Architecture

1. Depth calculation, focuses on precisely determining an object's depth or distance from the user.
2. Object detection: Employing advanced techniques, this module recognizes various objects within the user's surroundings.
3. System feedback through 3D audio: The system sends the user 3D audio messages in real time as feedback.

These integrated modules collectively serve to enhance the navigation experience and provide invaluable assistance to PVIs.

This project centered around a system meant to process images in real time. To achieve this demand, an efficient setup is required to ensure that results are delivered on time. Several continuous operations must be carried out by the system at the same time

Perform object detection using YOLO for particular classes, process video streams from two cameras, build depth maps, and detect Aruco markers. Originally designed for a Raspberry Pi 4, this vast number of procedures is a real challenge due to the platform's computing limitations.

In this sense, the use of a parallel server with the Raspberry Pi 4 has been considered. The server will handle object detection using YOLO, while the Raspberry Pi 4 will focus on acquiring video streams, creating depth maps, and forwarding these streams to the server for object detection operations. This approach aims to overcome the processing limitations of the Raspberry Pi 4.

However, Because of the time delays associated with sending and receiving data between the server and the Raspberry Pi 4, this method isn't optimal. Furthermore, users of this system will constantly require a reliable and stable internet connection, which might not always be feasible in everyday life. Hence, this method does not ensure complete autonomy given the user's dependency on internet connectivity.

The final explored and adopted solution involves using two Raspberry Pi 4 devices. The first is responsible for processing video streams and

generating depth maps. It transmits these streams to the second Raspberry Pi 4 through a local network. The second Raspberry Pi 4 then handles object and Aruco marker detection in these video streams and finally relays the results back to the first Raspberry Pi 4, which, in turn, verifies the response from the second Raspberry Pi 4, compiles, and communicates to the PVI the necessary indications and information about objects, locations, and directions through binaural audio in the earphones. This method eliminates response delays, as both Raspberry Pi communicates within the same local network via an access points, without requiring an internet connection. In other words, these two Raspberry Pi 4 devices act as local servers communicating with each other. This operational mode ensures real-time processing in this context.

The hardware used for this project includes:

- Two Raspberry Pi 4 Model B 8GB
- Two Raspberry Pi Camera Module v2
- One Earphone

The following figure illustrates the data flow circulating through various system components:

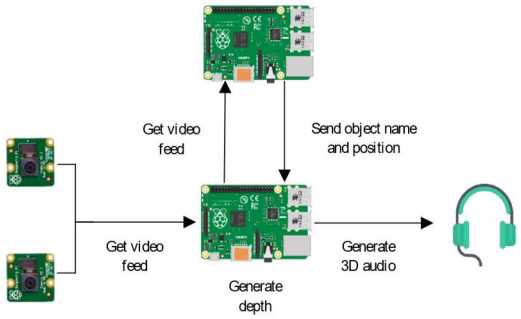

Figure 2: Current Data flow pipeline of our system

3. DEPTH ESTIMATION

Sections The perception of depth and distance concerning surrounding objects represents a crucial element in navigation, particularly for individuals with visual impairments. Spatial perception is vital to enable them to move safely and comprehend their environment.

This system utilizes the concept of stereoscopy to generate a depth map. A stereoscopic imaging sub-system was created using two Raspberry Pi Camera Module v2 units. This stereoscopic camera is positioned on a head worn helmet to continuously capture images.

The images captured by these two calibrated cameras are used as input for this module. To compute the disparity between each corresponding pixel in the two images, a semi-automatic block correspondence approach is used [20].

These determined disparities are used to construct a depth map, which is then used to create the three-dimensional audio vector. To elaborate, the 3D audio module receives a matrix comprising the depth, measured in centimeters, for each pixel. For further in-depth information on this topic, refer to [21].

4. OBJECT DETECTION

4.1 Detecting Common Objects

These We examine multiple detection systems, currently in use that have the ability to identify objects and assess them at different points in an image in order to effectively detect nearby objects. Using root filters, the Deformable Parts Model (DPM) [22] moves detection windows throughout the whole picture. Region proposal techniques are used by R-CNN [23] to produce potential bounding boxes in an image that is processed by ConvNets to categorize the content. The long test period, complex training process, and substantial storage capacity are no suppoted by our system.

The proposed regions are max-pooled using Fast R-CNN [24], which also combines the ConvNet computation for each image proposal to output features of all the regions simultaneously. After the last layer of ConvNet, Faster R-CNN [25], which is based on Fast R-CNN, inserts a region proposal network.

Both techniques increase accuracy while cutting down on computation times. These techniques still have quite complicated processes that are challenging to optimize. Given that this project requires real-time objective detection, in this project, we use the You Only Look Once (YOLO) model [19]. YOLO could efficiently provide relatively good objective detection with extremely fast speed.

YOLO stands as a cutting-edge technology in real-time object detection methodologies. The initial iterations, namely YOLOv1, YOLOv2, and YOLOv3 [26][27][28], constituted the early versions. YOLOv2 specifically aimed to

substantially enhance accuracy. The last version is YOLOv8.

Anchors for detection, a concept initially introduced in YOLOv2, were influenced by Faster R-CNN's methodology. Furthermore, YOLOv2 integrated principles from Faster R-CNN, Batch Normalization [30], and Skip connections [31] as foundational components within its design.

Evolved from its predecessors YOLOv1 and YOLOv2, YOLOv3 emerged as a leading-edge approach in object detection. This iteration adopted Darknet-53 [28] as its backbone, departing from Darknet-19 [27]. It integrated multi-scale feature extractors (FPN) [32] and replaced Softmax classification loss with binary crossentropy loss.

YOLOv4 [33] was introduced with the primary goal of enhancing the capabilities of YOLOv3. In contrast to Faster R-CNN, YOLO employs a distinctive methodology by employing a single neural network to process an entire image. This network divides the input into a S x S grid and detects within each cell, allowing bounding box predictions and associated confidences.

The confidence scores represent the model's level of confidence that an object exists within a bounding box. When the model is confident, the confidence score, as shown in Equation (1), shows the accuracy of overlap between the ground truth (GT) and the model's predictions (pred).

$$\text{Confidence} = \Pr(\text{Object}) * \text{IoU}(\text{GT}, \text{pred}) \quad (1)$$

where $\Pr(\text{Object}) \in [0,1]$.

In the YOLO model for detection, every grid cell is responsible for predicting various values associated with the detected objects, including coordinates (x, y, h, w), Confidence scores, and C class probabilities. The x and y are the coordinates of the box's center, w and h are tits width and height respectively.

We chose YOLO to detect everyday objects. However, when we examine its implementation on more limited devices such as the Raspberry Pi, we prefer the YOLOv4-tiny version due to its efficiency and optimized architecture. The Raspberry Pi 4, while a versatile and capable platform, has limitations in computational power and memory. YOLOv4, with its deeper architecture and higher parameter count, demands more processing capabilities and memory, making real-time inference challenging on this device. YOLOv4-tiny, on the other hand, addresses these constraints by offering a streamlined and lighter model without sacrificing significantly on performance. Its reduced complexity allows it to operate more efficiently on resource-limited hardware like the Raspberry Pi 4, ensuring faster inference speeds while maintaining a reasonable level of object detection accuracy. This makes YOLOv4-tiny a pragmatic choice for practical implementations on the Raspberry Pi 4, where speed and responsiveness are crucial factors alongside the available computational resources.

We developed a 37-layer CNN specifically designed for YOLOv4-tiny. Our method aligns with the YOLOv4 model specifications while notably reducing the overall weight of the final deep learning model. YOLOv4-Tiny incorporates several modifications from the original YOLOv4 architecture, optimizing it for rapid execution on inexpensive embedded systems.

Mainly, the convolutional layers within the CSP backbone undergo compression. Additionally, the YOLO layers have been condensed from three to two, accompanied by a decrease in the number of anchor boxes utilized for prediction. YOLOv4-tiny comprises three primary modules depicted in Figure 3: CSPDarknet53-Tiny, the Feature Pyramid Network (FPN), and the YOLO Head.

The CSPDarknet53-Tiny module functions as the main feature extractor, housing Convolutional blocks (Conv) and CSPBlocks. Within the Convolutional layers, batch normalization and activation functions are incorporated. Batch normalization serves to regulate the model, eliminating the necessity for dropout layers in the architecture to counter overfitting problems. Its function involves enhancing input normalization by establishing variance values.

Leaky ReLU (Rectified Linear Unit) serves as the activation function. Within the Cross Stage Partial Network (CSPNet), the CSPBlock follows a systematic approach by dividing the Base layer's model into two segments. The initial part forms a residual edge, while the subsequent segment combines with the former, culminating in the final output after a sequence of convolutional operations.

The Feature Pyramid Network (FPN) architecture is designed to amalgamate features from multiple network layers, retaining semantic content from deep networks and geometric details from low-level networks. This strategy aims to enhance the capability of feature extraction. The YOLO Head serves as the architecture's concluding module responsible for generating feature output results.

In the context of a one-stage detector, the YOLO Head serves to perform dense predictions. These predictions consist of a vector encompassing the coordinates of the predicted bounding boxes (height, width, center), the associated label, and the confidence score, as illustrated in Eq. (2). Here, Pw and Ph denote the width and height of the bounding boxes, while (Cx, Cy) represents the coordinates of the image's top-left corner.

$$bx = \sigma(t_x) + C_x$$
$$bx = \sigma(t_y) + C_y$$
$$b_w = P_w \cdot e^{tw} \quad (2)$$
$$b_h = P_h \cdot e^{th}$$

identified objects. The coordinates of the center point within these bounding boxes serve to calculate the distance on the depth map.

This distance measurement represents the user's distance to the detected object. Using this data, our system can calculate the distance between the user and the identified objects, providing critical information for PVIs navigation.

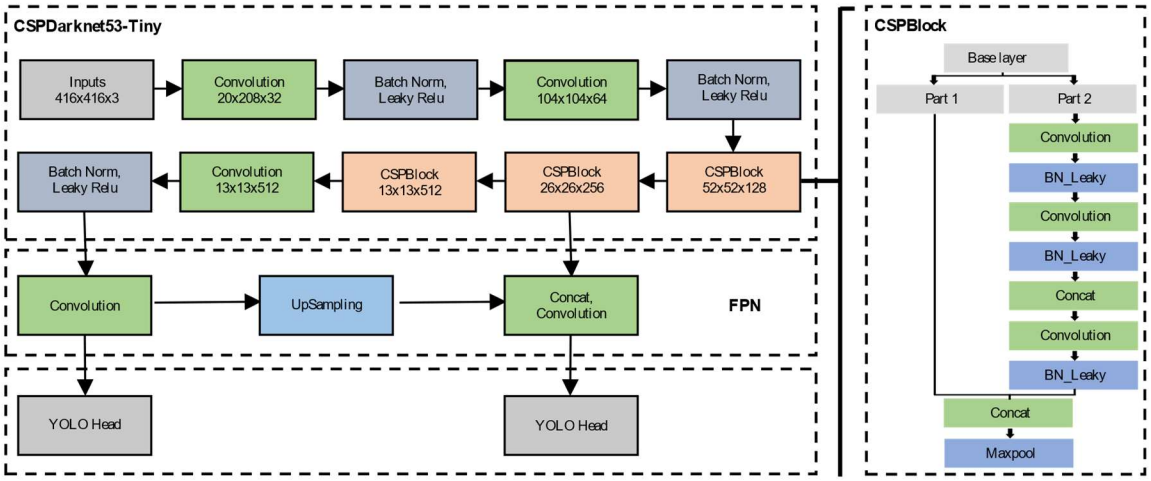

Figure 3: The architecture of YOLOv4-tiny Network

Our system's object detection methodology is based on the use of YOLOv4-tiny. Initially, we created a model using pictures from COCO data [37] , focusing specifically on the chairs and doors for our tests. However, different kinds of objects can be added to the model as needed, increasing its recognition capability.

We use one of the two images from the stereoscopic camera to do object detection. The size of the captured image is reduced to speed up the image detection and processing process. The detection process generates bounding boxes around

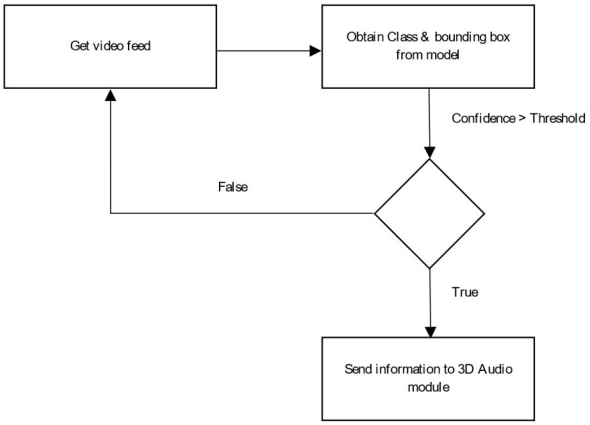

Figure 4: Object Detection Pseudocode

4.2 Detecting Uncommon Objects and Places

To detect certain objects not detectable by YOLO or to provide indications about locations for PVI, the system relies on the use of ArUco markers. These markers are placed on uncommon objects or on building facades and walls. Once the ArUco is scanned, the user of this system can know, through an audio message, the nature of the detected object or the name of the location. For example, if the system scans an ArUco placed on a pharmacy wall, it will inform the user by a vocal message that designate this situation. Several use cases are proposed, such as placing these ArUco markers next to signage panels indicating gathering points, exits, or emergency stairs. This information is highly useful, and every visually impaired person should have access to this kind of information.

ArUco markers are visual markers used in augmented reality and computer vision for precise object detection and tracking. They typically consist of black-and-white patterns with inner quadrilateral structures, enabling a camera to identify and determine their position and orientation in space.

Each ArUco marker is associated with a unique identifier. These identifiers are generally encoded as binary or decimal numbers, depending on the library or tool used for marker detection. For example, an ArUco marker can be represented by a decimal number or in binary form with a sequence bit representing a specific identifier.

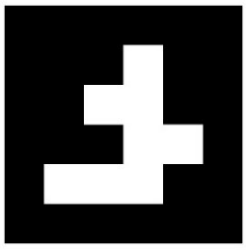

Figure 5: Exemple of an ArUco used for an emergency door

When the camera detects an ArUco marker, the corresponding identifier is interpreted by the system. For instance, an emergency exit sign could be represented by the unique identifier 13. This identifier is interpreted as an emergency exit, and a message indicating the nature of what is detected is conveyed to the PVI.

Similar to YOLO detection, the position of the ArUco marker is used on the depth map image to determine the distance between the visually impaired person and the detected ArUco marker.

Figure 6: A use case of an ArUco

4.3 User Interaction

Sections In order to optimize the system for everyday use, voice command capabilities have been integrated for controlling its operations. All program functions can be executed via voice commands.

The stereoscopic camera continuously captures images. Real-time depth map calculation occurs at the client's end. Ones images captured from the two cameras are sent to the server for object detection. As a result,'position and name in the image, for each object, are then relayed back to the client.

To activate audio feedback for object detection, users can simply say the word "scan" Upon identification of objects in the captured image, their names are audibly output. Table 2 delineates the user-issued voice commands and subsequent actions taken'.

The recognition of audio commands is implemented using SpeechRecognition a Speech recognition module for Python [34], while the audio speech output is facilitated by utilizing …

gTTS (Google Text-to-Speech) wich is a Python library and command-line interface (CLI) tool designed to interact with Google Translate's text-to-speech API. This tool enables users to convert text into spoken audio using Google's powerful text-to-speech technology [35]. Presently, only English is supported, thus limiting recognition accuracy for non-native speakers. Voice output guides users through individual steps of the program, such as informing them of a brief waiting period for object detection after taking a photo. Upon the return

of results, the system automatically announces the list of identified objects and their respective names.

Table 1 contains two columns: the user-issued voice commands, actions executed by the client:

Table 1: User, client and server interaction.

User	Client
"Scan"	Reads list of objects aloud
"Stop"	Stop reading the list of objects aloud
Says the object Name (Ex : Door)	Announces distance and location of selected object

5. 3D AUDIO FEEDBACK

5.1 Processing the Detected Object Coordinates

The 3D audio module aids in providing the virtual reality user with an intuitive feeling of the object's placement. The identified object's name and a vector denoting its position in 3D space are defined using values collected from previous modules.

The returned position consists of two coordinates, x and y, corresponding to the object's coordinates on the captured image.

The image, captured by the stereoscopic camera is divided into 6 zones, 3 horizontally and 3 vertically. Each zone is defined by coordinates two points with coordinates (x1, y1), and (x2, y2). Each of these zones also bears a name representing its spatial position relative to the user. For instance, on figure 6, Zone 1 represents "top-left".

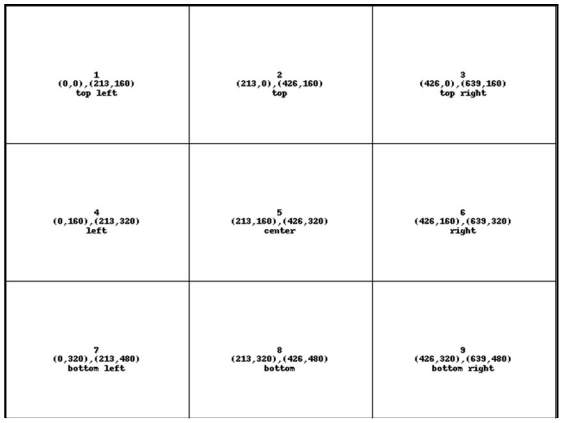

Figure 7: Division of the captured image into 6 zones

The module responsible for creating the binaural audio message checks in which zone this point lies. Determining if a point (x, y) is inside the bounding box corresponding.to one of the defined zones.

Let's say a bounding box defined by it two points: (Xmin,Ymin) for the bottom-left corner and (Xmax,Ymax) for the top-right corner. If a point (X,Y) is inside the bounding box, the following conditions must be met:

$$X_{min} \leq X \leq X_{max} \qquad (3)$$
$$Y_{min} \leq Y \leq Y_{max}$$

For example, if a bounding box with (Xmin, Ymin) = (0, 0) and (Xmax, Ymax) = (213, 160) is considered. Given a point (x, y) = (100, 110):

Xmin = 0, Xmax = 213, Ymin = 0, Ymax = 160

For the point (100, 110):

$$0 \leq 100 \leq 213 \text{ (True)}$$
$$0 \leq 110 \leq 160 \text{ (True)}$$

The audio message will then indicate to the user that the object is located in the top-left zone. The value of the distance separating the object from the user is determined on the depth map based on the same point coordinates on the captured image.

The following image shows an example of the detected ArUco marker at coordinates 100,110, located within the top-left bounding box.

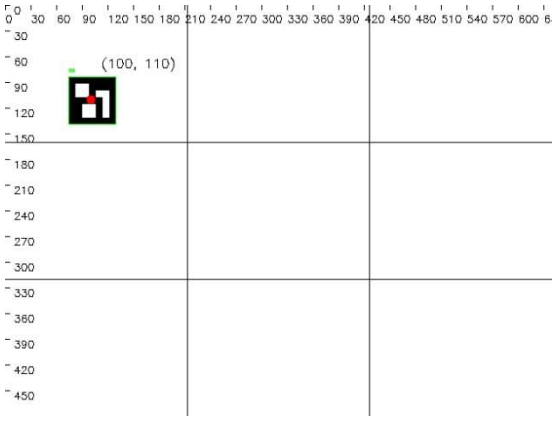

Figure 8: Detection of an ArUco marker's position

5.2 Normalization

To generate a binaural sound, the system utilizes the Python package Librosa [36] . The position of the audio vector is determined using the coordinates of a point on the image, considering to the image center as the reference point from which the audio vector's position is calculated.

In audio spatialization based on image coordinates, the goal is to simulate a sense of directionality or positionality within the audio experience. This process is akin to how one might perceive sound originating from different locations when observing an image or a scene.

Normalization is the process of scaling the image coordinates, defined in the interval [0, 640]x[0, 480] to fit within a specific range . This scaling ensures uniformity and consistency when interpreting these coordinates relative to the image dimensions.

Not considering the position of a person with a visual impairment (PVI) leads to significant issues with the 3D audio vector. When assuming the user is positioned at (0, 0), the construction of the audio vector should ensure that any sounds originating from the user's left are represented with negative X values, while those from the PVI's right are denoted by positive X values. Consequently, the X coordinate domain was adjusted from [0, 640] to [-1, 1], and similarly, the Y coordinate was transformed from [0, 480] to [-1, 1] to maintain consistency in representation.

Horizontal panning corresponds to the left-right positioning of sound sources. By using the normalized x-coordinate, the code calculates a horizontal panning factor. This factor determines how much the sound will be positioned to the left or right channels in the stereo audio.

Vertical panning relates to the up-down positioning of sound sources. However, in the audio world, the convention often inverts the vertical axis compared to the image coordinate system. Therefore, the code inverts the normalized y-coordinate to align with this convention. The resultant vertical panning factor controls the up-down placement of sound in the stereo audio.

Once the horizontal and vertical panning factors are calculated, the code adjusts the gains (loudness levels) of the left and right audio channels accordingly. For instance, a higher gain in the left channel relative to the right channel will position the sound more to the left in the stereo field, creating a sense of directionality.

By combining these gains in varying proportions based on both the horizontal and vertical factors, the code effectively spatializes the audio, creating a stereo signal that simulates a specific audio location corresponding to the image coordinates.

After manipulating the gains for each channel, the resulting signals for the left and right channels are combined to form a stereo spatialized audio signal. This stereo signal, when played through stereo speakers or headphones, recreates the illusion of the sound originating from a particular location in the listener's auditory space, corresponding to the specified image coordinates.

This spatialization technique provides a means to immerse the listener in a more realistic audio environment, aligning with the visual content observed in the image by positioning audio sources according to their relative positions within the image frame.

When a sound from the resulting audio vector is played, users will perceive the sound's origin as the object's location. This perception often prompts a natural inclination in PVIs to move towards the source of the sound. The audio message conveys the object's name, its distance in centimeters from the user, and its spatial position in front of the user, such as "top-left" as an example. Figure 8 show the visualization of the normalized vector of the point with the coordinate (100,110):

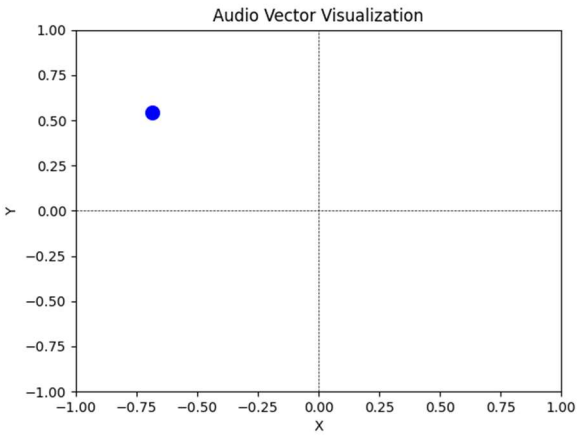

Figure 9: 3D Audio Vector Visualization

6. RESULTS AND DISCUSSION

To test this system, various experiments were conducted using YOLOv4-tiny and ArUco for

detection purposes. Figures 9, 10, and 11 display tests conducted with YOLOv4-tiny on a chair positioned to the left, in front, and to the right of the camera. Figures 12, 13, and 14 depict depth maps obtained from each of these three tests.

Additionally, experiments with ArUco involved placing markers to the left, in front, and to the right of the camera. Test results are depicted in figures 15, 16, and 17.

The results indicate that the system is robust and accurate, aligning with the predicted audio vector.

As an improvement, one can consider encapsulating the system's components in a plastic casing, reducing the device's weight by opting for a more compact battery, or even disconnecting it from the rest of the system. Another intriguing possibility would be to shrink the device in size to transform it into a pair of glasses equipped with a camera on each side.

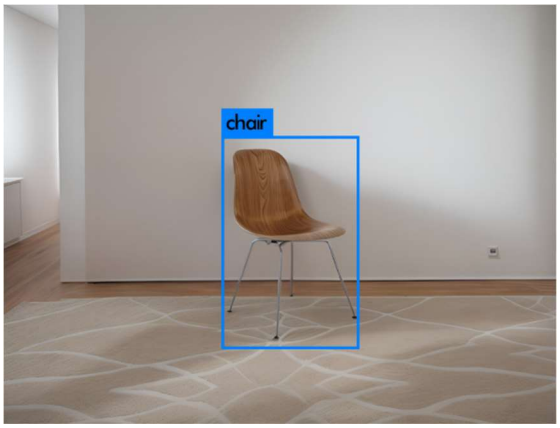

Figure 10: Object is located in front of the camera (Test 1)

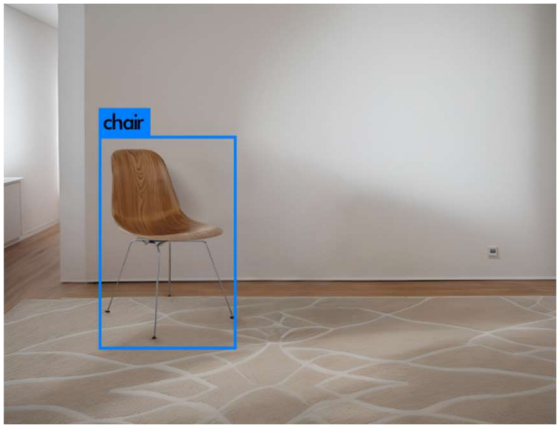

Figure 11: Object is located to the left of the camera (Test 2)

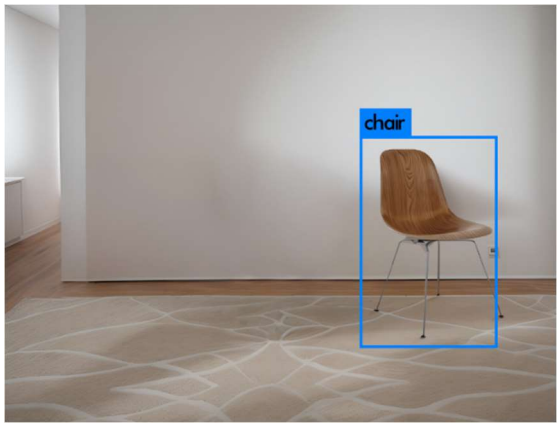

Figure 12: Object is located to the right of the camera (Test 3)

Figure 13: Depth map from test 1

Figure 14: Depth map from test 2

Journal of Theoretical and Applied Information Technology
15th October 2024. Vol.102. No. 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 15: Depth map from test 3

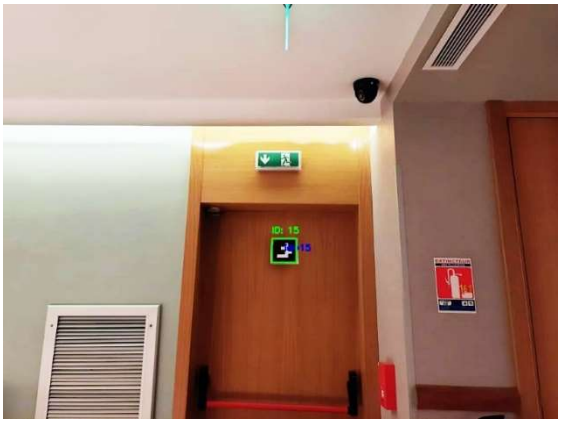

Figure 16: ArUco is located in front of the camera (Test 4)

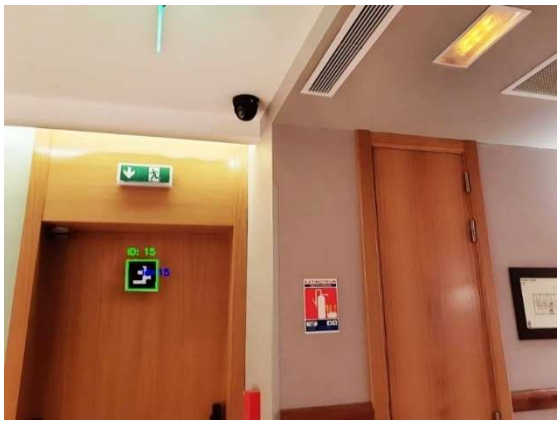

Figure 17: ArUco is located to the left of the camera (Test 5)

Figure 18: ArUco is located to the right of the camera (Test 6)

Table 2: Results from test 1, 2 and 3 using yolov4-tiny.

	Test 1	Test 2	Test 3
Average Depth	373,19 cm	369,92 cm	370,32 cm
Image Coordinates (X, Y)	(331,271)	(186,269)	(488,272)
Normalized Audio Vector (X, Y)	(0.034, -0.129)	(-0.418, -0.120)	(0.524, -0.133)
Position	Center (Zone 5)	Left (Zone 4)	Right (Zone 6)

Table 3: Results from test 4, 5 and 6 using ArUco.

	Test 4	Test 5	Test 6
Average Depth	213,48 cm	193.74 cm	175.36 cm
Image Coordinates (X, Y)	(324,286)	(159,314)	(472,256)
Normalized Audio Vector (X, Y)	(0.012, -0.191)	(-0.503, -0.308)	(0.475, -0.066)
Position	Center (Zone 5)	Left (Zone 4)	Right (Zone 6)

The graphs illustrating the 3D audio vectors from these test scenarios are presented in Figures 17 and 18. A comparison of these graphs reveals the model's proficiency in mapping the user's position in

relation to the target object for navigation, demonstrating a notably precise representation.

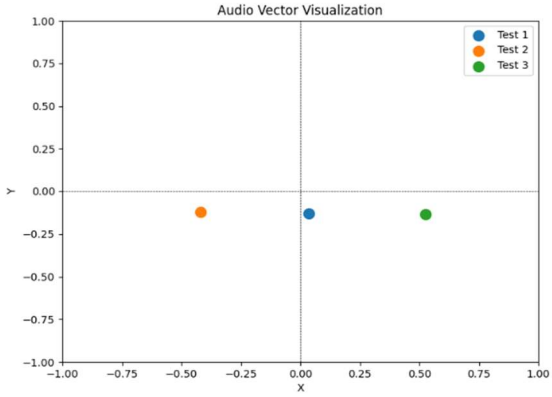

Figure 19: Visual representations for test cases 1, 2, and 3 through the utilization of YOLOv4-tiny

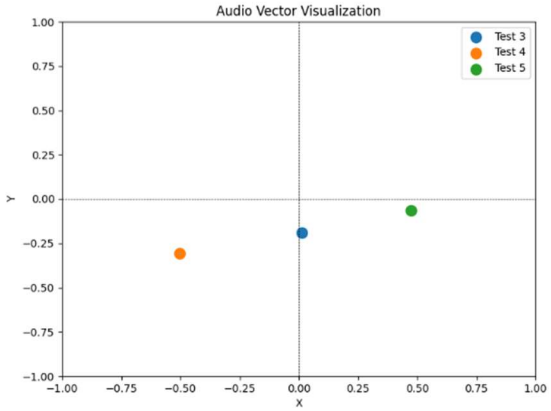

Figure 20: Visual representations for test cases 1, 2, and 3 through the utilization of ArUco.

7. CONCLUSION

This article presents a vision system integrated with a 3D auditory feedback mechanism designed to assist individuals with visual impairments in navigation. By utilizing spatial sound, the system fosters a more intuitive understanding of object locations. The combination of convolutional neural networks, ArUco Detection, and stereoscopic vision demonstrates the system's effective navigation capabilities across diverse environments.

The model's reliability is evidenced by the robust performance indicated by the 3D audio vector. While the current implementation employs YOLOv4-tiny, which offers less accuracy compared to the standard YOLO version, there remains significant potential for enhancement through the use of alternative computational platforms that could improve both speed and precision.

This research primarily focuses on chairs, which may limit the broader applicability of our findings. To address this limitation, future work will explore the integration of a trajectory planning algorithm and the diversification of the neural network's training data to include a wider array of objects, thereby enhancing overall system performance. In conclusion, our findings mark a crucial advancement toward developing effective navigation tools for visually impaired individuals, while also highlighting opportunities for future enhancements to broaden the system's impact.

REFERENCES:

[1] Fernandes, Hugo & Costa, Paulo & Filipe, Vítor & Hadjileontiadis, Leontios & Barroso, Joao. (2010). "Stereo vision in blind navigation assistance" World Automation Congress (pp. 1-6). IEEE.

[2] Chumkamon, S., Tuvaphanthaphiphat, P., & Keeratiwintakorn, P. (2008, May). "A blind navigation system using RFID for indoor environments" 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (Vol. 2, pp. 765-768). IEEE.

[3] Schinazi, Victor R., Tyler Thrash, and Daniel-Robert Chebat. "Spatial navigation by congenitally blind individuals." Wiley Interdisciplinary Reviews: Cognitive Science, 7(1), 37-58.

[4] Balakrishnan, G. N. R. Y. S., et al. "A stereo image processing system for visually impaired." International Journal of Signal Processing, 2(3), 136-145.

[5] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks", In Advances in neural information processing systems (pp. 1097-1105).

[6] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation", In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster r-cnn: Towards real-time object detection with region proposal networks.", In

Advances in neural information processing systems (pp. 91-99).

[8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). "You only look once: Unified, real-time object detection,", In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[9] Redmon, J., & Farhadi, A. (2017). "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). "Ssd: Single shot multibox detector." In European conference on computer vision (pp. 21-37). Springer, Cham.

[11] Joseph Redmon and Anelia Angelova, Real-Time Grasp Detection Using Convolutional Neural Networks (ICRA), 2015.

[12] A. Quattoni, and A.Torralba. Recognizing Indoor Scenes. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[13] Saurabh Gupta, Ross Girshick, Pablo Arbelaez and Jitendra Malik, Learning Rich Features from RGBD Images for Object Detection and Segmentation (ECCV), 2014.

[14] Tadas Naltrusaitis, Peter Robison, and Louis-Phileppe Morency, 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking (CVPR), 2012.

[15] Andrej Karpathy and Fei-Fei Li, Deep VisualSemantic Alignments for Generating Image Descriptions (CVPR), 2015.

[16] David Brown, Tom Macpherson, and Jamie Ward, Seeing with sound? exploring different characteristics of a visual-to-auditory sensory substitution device. Perception, 40(9):1120–1135, 2011.

[17] Liam Betsworth, Nitendra Rajput, Saurabh Srivastava, and Matt Jones. Audvert: Using spatial audio to gain a sense of place. In Human-Computer Interaction– INTERACT 2013, pages 455–462. Springer, 2013.

[18] Jizhong Xiao, Kevin Ramdath, Manor Iosilevish, Dharmdeo Sigh, and Anastasis Tsakas. A low cost outdoor assistive navigation system for blind people. In Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, pages 828–833. IEEE, 2013.

[19] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015.

[20] Hirschmuller, H. (2007). "Stereo processing by semiglobal matching and mutual information." IEEE Transactions on pattern analysis and machine intelligence, 30(2), 328-341.

[21] Moncef Aharchi, M'hamed Ait Kbir, "Assisting Visually Impaired People through Real-time Depth Estimation using Stereo Vision," International Journal of Engineering Trends and Technology, vol. 71, no. 11, pp. 236-246, 2023. Crossref, https://doi.org/10.14445/22315381/IJETT-V71I11P225

[22] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 32(9):1627–1645, 2010.

[23] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[24] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems, pages 91–99, 2015.

[26] Redmon, J., and Farhadi, A. (2017). "YOLO9000: Better, faster, stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (Honolulu, HI, USA: IEEE) 7263-7271. doi: 10.1109/CVPR.2017.690

[27] Redmon, J., and Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. doi: 10.48550/arXiv.1804.02767

[28] Redmon, J., Farhadi, A., Divvala, S., and Girshick, R. (2016). "You only look once: unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern recognition. (Las Vegas, NV, USA: IEEE), 779–788. doi: 10.48550/ arXiv.1506.02640

[30] Ioffe, S., and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, 448-456. doi: 10.48550/arXiv.1502.03167

[31] He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in Proceedings of the IEEE Conference on Computer vision and Pattern Recognition (Las Vegas, NV, USA, IEEE), 770–778. doi: 10.1109/CVPR.2016.90

[32] Tsung-Yi, L., Piotr, D., Ross, G., Kaiming, H., Bharath, H., and Serge, B. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on Computer vision and pattern recognition, 2117-2125. doi: 10.48550/arXiv.1612.03144

[33] Bochkovskiy, A., Wang, C. Y., and Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 doi: 10.48550/ arXiv.2004.10934

[34] SpeechRecognition documentation. Available online: https://github.com/Uberi/speech_recognition (accessed on 23 December 2023).

[35] gTTS documentation. Available online: https://gtts.readthedocs.io/en/latest/ (accessed on 23 December 2023).

[36] librosa documentation. Available online: https://librosa.org/doc/latest/index.html (accessed on 11 December 2023).

[37] COCO dataset. Available online: https://cocodataset.org/#home (accessed on 4 April 2024).

[38] E. M. Dharma and A. Trisetyarso, "Identification of Banknotes on the Visually Impaired Person through Deep Learning," in *Journal of Theoretical and Applied Information Technology*, vol. 100, no. 11, pp. 3820-3821, June 2022, [Online]. Available: http://www.jatit.org.

[39] A. Anwar and S. Aljahdali, "A Smart Stick for Assisting Visually Impaired People," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 14, pp. 4405-4406, July 2018. [Online]. Available: http://www.jatit.org.