IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

SENSOR MATI

real time object detection and navigation for blind people

[1]Aishwarya C Halemani, [2]Aishwarya Jayaram, [3]Aditya Kakkad

[1]MCA Student, [2] MCA Student, [3] MCA Student

[1]Computer Science and Information Technology,

[1]Jain(Deemed to be)University, Bangalore, India

Abstract : Vision is one of the most important senses for a living. Millions of people living in this world deal with visual impairment. These people encounter difficulties in navigating independently and safely, facing issues in accessing information and communication. The objective of the proposed work is to change the visual world into an audio world by notifying the blind people about the objects in their path. This will enable visually impaired people to move around on their own without any external support by using the real-time object detection system. The application makes use of image processing and machine learning techniques to identify real-time objects through the camera and then informs the blind people about the object and its location through the audio output. It cannot distinguish between objects, and it has resulted in many limitations to the existing approach, such as less accuracy and low performance results. The proposed work is focused on good accuracy, best performance results, and a viable option for the visually impaired people to make the world a better place for them.

There is a need to deploy a technology that assists visually impaired persons in their daily routines, as there is now very little aid. Existing solutions, such as Screen Reading software and Braille devices, assist visually impaired individuals in reading and gaining access to numerous gadgets. In cases where these technologies are rendered useless When the blind must perform basic actions like identifying the situation in front of them, for example, identifying people or things, technologies are useless. This technology will help the blind or visually handicapped everywhere in the world. The aim is to enable an individual with either complete or partial blindness to get a second sight without the intervention of a guardian so that such a person lives a better life and is also independent. The project outlines work on creating a more welcoming and inclusive environment in the form of assistive technology, which is providing services, resources, and information to visually disabled people.

The problem of visual impairment or blindness is a widespread issue. Based on statistics from WHO, globally, at least 2.2 billion people have vision impairment or blindness, out of whom at least 1 billion are blind. Moreover, in terms of regional differences, low- and middle-income regions have four times the prevalence of vision impairment in comparison to high-income regions. Blind people, in general, must depend on white canes, guide dogs, screen-reading software, magnifiers, and glasses to aid them for mobility, however, to help the blind people the visual world has to be transformed into the audio world with the potential to inform them

about objects as well as their spatial locations. The goal is for a person totally or partially blinded to have, without the use of a legal guardian, the ability to view life through second eyes, hopefully making it an easier and a more independent journey. This project will work under the creation of a more receptive and accessible world by focusing primarily on assistive technology that involves services, information, and sources for the Visual diable.

INTRODUCTION

Blindness and low vision are severe impairments to a person's ability to perceive and interact with the outside world, which results in reduced mobility and productivity in daily tasks. People with visual impairments depend on past experiences, guide canes, or assistance from other people to navigate and avoid obstacles. The lack of eyesight makes them highly dependent on memory and slow to react in case of an abrupt change of environment, which can be risky. Understanding colors, orientation, and depth in visual elements is very difficult. It takes considerable time and effort to understand a three-dimensional object in a single instance. Technological advancement, however, has greatly favored visually impaired individuals over the last decade. Those devices that function fully with audio input, not needing even visual or mechanical interaction, are a boon. Screen readers, an example, read text into a voice for the benefit of the visually challenged user to decipher from his device. Though these developments are there, these devices are still not adequate to ease personal and professional life for an individual who suffers from visual impairments. They mainly rely on audio input and fail when users need to understand visual information from their surroundings or interpret textual images.

Research continues to focus on how to enhance mobility for visually impaired individuals, making navigation as safe and obstacle-free as possible. This can also be useful indoors, enabling individuals to build a mental map of their environment and find certain items. Hence, a complete solution is suggested to enable visually impaired people to move around more confidently and independently. Our solution is object identification and detection by the camera of Android phones along with a web application. It begins capturing a live video stream from the camera the moment the application is launched. The system can detect objects in the camera frame and provides some approximate positional information, which it conveys to users through audio output. Thus, users understand where the objects are and the direction they need to move in. This allows the application to detect traffic lights and approaching automobiles, which in turn can direct the user toward them. Lampposts and other items along the roads can also be detected. Other small objects that are used as part of normal daily activities are also detected- for example, pens, toothbrushes, and utensils. The system aims to greatly improve the daily lives of visually impaired individuals by enabling the recognition and location of such objects.

LITERATURE REVIEW

Li and Yu (2019) discussed real-time object recognition for the visually impaired using deep learning techniques. Their approach resulted in an 85% recognition rate, but there are challenges in processing speed and model complexity. Zhang et al. (2018) also implemented deep learning-based object recognition for visually impaired individuals with a 90% accuracy rate, but the model had generalization issues with various objects. Chen et al. (2020) further developed this by providing a real-time deep learning-based object detection and recognition system that achieved 92% accuracy but required high computational power. Earlier researches explored other methods of supporting visually impaired people. Kulyukin, Gharpure, and Nicholson (2006) presented a smartphone-based indoor navigation system based on RFID technology, which achieved 89%

accuracy but had limited coverage and accuracy problems. Ahmetovic et al. (2016) designed NavCog, a turn-by-turn smartphone navigation assistant based on Bluetooth beacons, with an 87% accuracy rate, but it relied heavily on the infrastructure of Bluetooth beacons.

Bousbia-Salah et al. (2011) introduced an ultrasonic sensor-based navigation aid for the blind, achieving an 80% accuracy rate but with constraints in range and precision. Another similar system, AmIciTy, is a navigation system designed by Hub et al. (2010), which combined GPS and RFID to provide a 75% success rate and relied on GPS signal availability. Pradeep et al. (2015) implemented a deep learning-based object detection system with 88% accuracy, though it required high computational resources. Wang et al. (2017) developed a wearable vision system that allowed the visually impaired user to navigate independently with 82% accuracy. However, the system had problems with battery life and comfort. Several reviews and surveys have studied assistive technologies for the visually impaired. Al-Madani and Al-Muhtadi, 2017 carried a survey on Mobile indoor way finding systems and resulted with 76% effectiveness of way finding which had no related discussion of implementing such system.

Miao, 2019 used computer vision for the people with vision defect achieved an accuracy rate of about 90 % while less was related to current implementation. Dakopoulos and Bourbakis (2012) conducted a survey on wearable obstacle avoidance electronic travel aids for the blind, reporting an 87% effectiveness rate, though integration with other assistive technologies remained a challenge. Manduchi and Kurniawan (2013) studied mobility-related accidents experienced by visually impaired individuals, identifying key challenges but lacking technological solutions. Markopoulos et al. 2019 highlighted recent advances in assistive technologies, with an effectiveness rate of 76% but a lack of focus on individual technologies. Ashbrook and Starner 2002 surveyed past work on blind navigation systems, pointing out challenges and open problems, but the review does not cover recent advances.

Fuentes-Pacheco, Ucan, and Sucar 2015 discusses object recognition technologies for the blind, pointing out difficulties with real-time accuracy and little practical discussion of solutions. Some research focused on specific tools. Lazarillo (2016) presented a GPS-based navigation system for the blind, showing 85% effectiveness but necessitating a strong GPS signal constantly. WeWALK (2019) introduced a smart cane with sensors and a touchpad, reaching an effectiveness of 87%, yet the device cost much and had limited availability. Eye-D (2016) developed a multi-feature app for the visually impaired. The app successfully identified objects, read texts, and navigated 91% of the time. Wayfindr (2017) developed an audio-based navigation standard that had an 84% effectiveness rate. However, its application in public spaces was still minimal.

MATERIALS AND METHODS

The materials involved in this project are hardware and software parts. The list of software used in the software requirements is mainly Android Studio. It is essential for developing and building the android application. Programming language Python must be implemented with object recognition algorithms. OpenCV library allows one to do the image processing, as well as detect objects within an image. Android SDK helps to integrate scripts of Python programming with the application in Android. Train and implement object recognition models using TensorFlow or other machine learning libraries. Train object detection models on the COCO dataset. Implement a Text-to-Speech API to convert results from object recognition as audio output. When necessary, SQLite or Firebase could be utilized to store user preferences and logs. The hardware

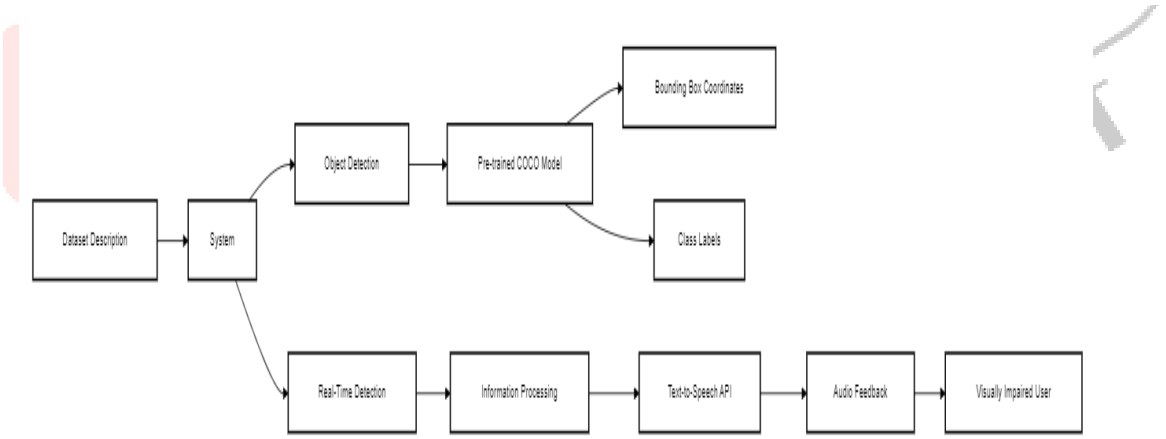
includes an Android smartphone or tablet and a camera in the device that is used to perform real-time object detection.

The webcam will not be essential if the computer is used first to test and then the entire system is applied on the Android device. For running the application smoothly, one requires sufficient amount of RAM as well as enough storage. This application requires Internet for downloading ML models, upgrades, and all online resources. Bluetooth beacons may also be used if the project involves indoor navigation with beacon technology

DATASET DESCRIPTION

The COCO dataset is a large-scale dataset for object detection, segmentation, and captioning. It contains 80 object categories, including people, vehicles, household objects, and outdoor elements. With around 330,000 images, the dataset provides a diverse collection of labeled and annotated visuals used for training deep learning models. It has about 1.5 million object instances, with each one correctly annotated with a bounding box and segmentation mask to ensure proper object detection. Further, it comprises five captions of an image for more detailed descriptions of the scene for better recognition of images. the system focuses on the object detection aspect of the COCO dataset.

The system applies a pre-trained COCO model to detect objects in real time. The bounding box coordinates and class labels can be deduced from the model to identify objects that are in the near surroundings of the visually impaired user. The information can then be processed and conveyed to the user using the Text-to-Speech API, providing real-time audio feedback for users about their environment. This way, people with vision loss can live more independently and accessibly by converting visual information into tangible sound signals.



DATA PREPROCESSING

Capture Frame → Resize Image → YOLOv8 Detection → Map to Grid (20x20) → A Pathfinding → **Overlay on Image & Display**

In the provided object detection and pathfinding program, data preprocessing is essential for accurate navigation past detected obstacles. The process is initiated with capturing frames from the webcam, then resizing them to 640x480 to enable efficient processing. These frames are fed into the YOLOv8 model, which makes real-time object detections and returns bounding box coordinates along with their class labels like "person," "car," or "chair." The objects thus detected are projected onto a 20x20 grid, where each cell

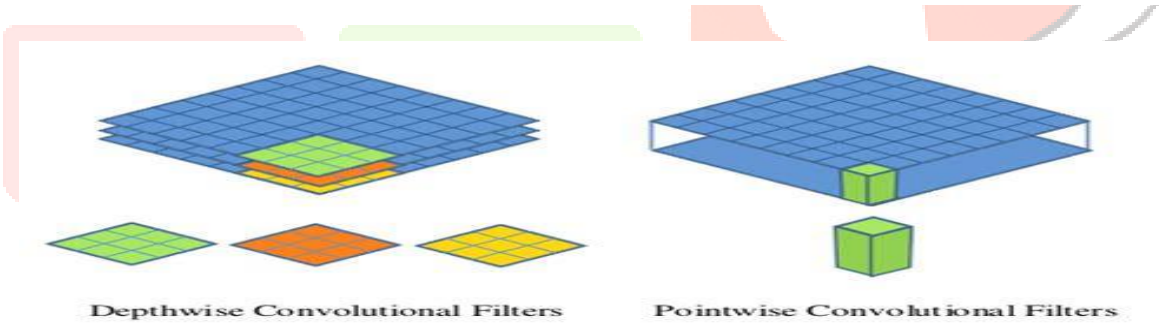
corresponds to a section of the frame. The cells where objects are detected are labeled as obstacles (1), and the rest of the cells remain free (0) to be navigated.

After the grid is built, the A pathfinding algorithm is used to calculate the shortest path from a specified start point (0,0) to an end point (19,19) without hitting obstacles. The algorithm considers various possible paths and selects the most suitable one based on a heuristic function (Manhattan distance). The calculated path is subsequently overlaid over the initial frame, where objects detected are marked with green bounding boxes and labels, while the shortest path is represented by red circles to show movement. Through this process, real-time dynamic navigation is facilitated so that the system can creatively maneuver around obstacles identified by the YOLOv8 model. The last processed frame is shown with all the annotations and hence constitutes a complete system for path navigation based on object detection.

PROPOSED METHODOLOGY

MobileNet Model

The MobileNet model is based on Depthwise separable convolutions which are a form of factorized convolutions. These factorize a standard convolution into a Depthwise convolution and a 1 Ã— 1 convolution called a pointwise convolution. For MobileNets, the Depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1 Ã— 1 convolution to combine the outputs of the Depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The Depthwise separable convolution splits this into two layers â€“ a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size.



Depthwise Separable Convolution

DEPTH ESTIMATION

Depth estimation or extraction refers to the set of techniques and algorithms which aims to obtain a representation of the spatial structure of a scene. In other words, it is used to calculate distance between two real time objects. Our prototype is used to assist the blind people which aims to issue warning to the blind people about the hurdles coming on their way. In order to do this, we need to find that at how much distance the obstacle and person are located in any real time situation. After the object is detected rectangular box is generated around that object. If that object occupies most of the frame then with respect to some constraints the approximate distance of the object from the particular person is calculated. Following code is used to recognize objects and to return the information for the locations and confidence:

(boxes, scores, classes, num_detections) = sess.run([boxes, scores, classes, num_detections], feed_dict={image_tensor: image_np_expanded})

So, here, for further analysis we can iterate through the boxes. Boxes are an array, inside of an array. So, for iteration we need to define the following conditions.

for i,b in enumerate(boxes[0]):

boxes[0][i][0] – y axis upper start coordinates

boxes[0][i][1] – x axis left start coordinates

boxes[0][i][2] – y axis down start coordinates

boxes[0][i][3] – x axis right start coordinates

Index of box in boxes array is represented by i. Analysis of the score of the box is done by index. It is also used to access class. Now, the width of the detected object is measured. This is done by asking the width of object in terms of pixels. apx_distance = round(((1 - (boxes[0][i][3] - boxes[0][i][1]))4),1) We get centre of two by subtracting same axis start coordinates and dividing them by two. In this way centre of our detected rectangle is calculated. And at the last, a dot is drawn in the centre. The default parameter for drawing boxes is a score of 0.5.

if scores[0][i] >= 0.5

(i.e. equal or more than 50 percent) then we assume that the object is detected. if scores[0][i] >= 0.5:

mid_x = (boxes[0][i][1]+boxes[0][i][3])/2

mid_y = (boxes[0][i][0] +boxes[0][i][2])/2

apx_distance = round (((1 - (boxes[0][i][3] - boxes[0][i][1]))4),1)

where mid_x is centre of X axis and mid_y is centre of y axis

If the distance apx_distance < 0.5 and if mid_x > 0.3 and mid_x < 0.7

then it can be concluded that the object is too close from the particular person.

With this code, relative distance of the object from a particular person can be calculated.

After the detection of object the code is used to determine the relative distance of the object from the person. If the object is too close then signal or a warning is issued to the person through voice generation module.

THE SSD ARCHITECTURE

SSD has two components: SSD head and a backbone model. Backbone model basically is a trained image classification network as a feature extractor. Like ResNet this is typically a network trained on ImageNet from which the final fully connected classification layer has been removed. The SSD head is just one or more convolutional layers added to this backbone and the outputs are interpreted as the bounding boxes and classes of objects in the spatial location of the final layers activations. We are hence left with a deep neural network which is able to extract semantic meaning from the input image while preserving the spatial structure of the image albeit at a lower resolution. For an input image ,the backbone results in a 256 7x7 feature maps in ResNet34 . Grid cellInstead of using sliding window, SSD divides the image using a grid and have each grid cell be responsible for detecting objects in that region of the image. Detection objects basically means predicting the class and location of an object within that region. Background class is considered if no object is present and the location is ignored. For instance, we could use a 4x4 grid in the example below diagram.

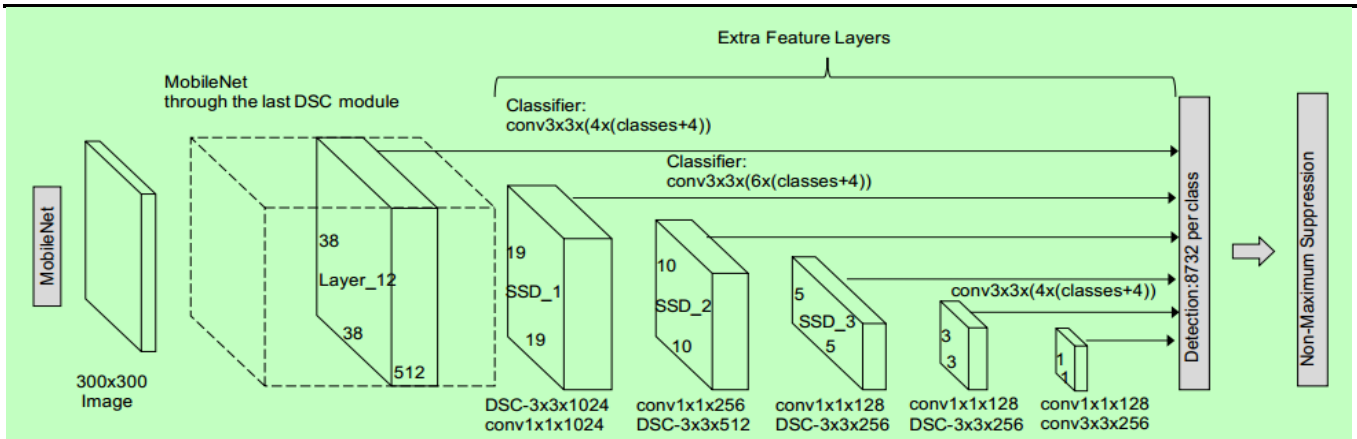


Figure 6. Mobile-Det: SSD-based detection with MobileNet as backbone, modified based on [14]

Each grid cell is able to output shape of the object it and the position. Anchor box and receptive field come into play are used when there are multiple objects in one grid cell or we need to detect multiple objects of different shapes. Anchor box:Multiple anchor/priorboxes can be assigned to each grid cell in SSD. These assigned anchor boxes are pre-defined and each one is responsible for a size andshape within a grid cell. Matching phase is used by SSD while training, so that there's an appropriate match to anchor box with the bounding boxes of each ground truth object within an image. For predicting that object's class and its location the anchor box with the highest degree of overlap with an object is responsible. Once the network has been trained,this property is used for training the network and for predicting the detected objects and their locations. Practically, each anchor box is specified with an aspect ratio and a zoom level. Well, we know that all objects are not square in shape. Some are shorter ,some are longer and some are wider, by varying degrees. The SSD architecture allows pre-defined aspect ratios of the anchor boxes to account for this. The different aspect ratios can be specified using ratios parameter of the anchor boxes associated with each grid cell at each zoom/scale level.

ZOOM LEVEL

It is not mandatory for the anchor boxes to have the same size as that of the grid cell.The user might be intrested in finding both smaller or larger objects within a grid cell. In order to specify how much the anchor boxes need to be scaled up or down with respect to each grid cell ,the zooms parameter is used.

VOICE GENERATION MODULE

Voice generation module plays a crucial role in the proposed system. After the detection of object, it is quite necessary to acknowledge the person about the presence of that object on his/her way. This is done by voice generation module which warns or signs the blind people by generating audio commands which are easily understood by them. After detection of object and its relative distance from the person we have to give voice commands about objects in the path of blind people. Also, if the object is very close then a warning is also issued to the blind person through voice generation module. Audio commands are generated as output. If the object is too close then it states "Warning: The object (class of object) is very close to you. Stay alert!". Else if the object is at safer distance then then a voice is generated which says that "The object is at safer distance". This is achieved with the help of certain libraries like pytorch, pyttsx3, pytesseract and engine. For voice generation module PYTTSX3 plays an important role. Pyttsx3 is a text-to-speech conversion library in Python. This library is compatible with both Python 2 and 3 an it works offline. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3 Engine instance. Pyttsx3 is a tool which converts text to speech easily. Two voice modules are supported by pyttsx3. First is female and the second is male

which is provided by “sapi5” for windows. An application invokes the pyttsx.init() factory function to get a reference to a pyttsx.Engine instance. During construction, a pyttsx.driver. DriverProxy object is initialized by engine which is responsible for loading a speech engine driver implementation from the pyttsx.drivers module. After construction, an engine object is used by the application to register and unregister event callbacks; produce and stop speech; get and set speech engine properties; and start and stop event loops. Sometimes there is a need to identify the hidden text in the image. For this purpose Python-tesseract is used. Python-tesseract an optical character recognition (OCR) tool for python. OCR detects the text content on images and encodes the text into language which is understood by the computer. This text detection is done by scanning and analysis of the image. Thus, the text embedded in images are recognized and “read” using Python-tesseract. Suppose there are some danger boards on road then the text and symbols hidden in the image of that board are identified and using voice generation module, warning is issued to the person. Pytorch it is primarily a machine learning library. Pytorch is mainly applied to the audio domain. Pytorch helps in loading the voice file in standard mp3 format. It also regulates the rate of audio dimension. Thus, it is used to manipulate the properties of sound like frequency, wavelength and waveform. The numerous availability of options for audio synthesis can also be verified by taking a look at the functions of Pytorch

THIRD PARTY APP

Third Party App provides ease and freedom in the field of app development. It brings efficiency and also helps in speedy delivery of the output. Third Party App allows you to divide your work in parts and helps you to focus on the core part of app or any system. This strategy helps in the development of good and quality software. We can pass on the Features of the Third Party App to the system.

1) At first, we are capturing real time images from the rear camera of the mobile handset of blind people and a connection is established between mobile phone and system in laptop and then those images are send from mobile phone to laptop.

2)This connection is done by a Third party app which is installed in the mobile phone of the person. All the real time images which get captured by the rear camera of mobile phone are first transferred to the Third party app in the mobile phone and then those images are sent in laptop where they are processed for some further conclusions.

3) The system in laptop will test it using its APIs and SSD ALGORITHM and it detects the confidence accuracy of the image which it is testing. We reached 98% accuracy for certain classes like books, cups, remote.

4) After testing the images we are generating an output on the laptop based system and its prediction is being translated into voice with voice modules and sent to the blind person with the help of wireless audio support tools.

COMPARISON

The academic studies examine assistive technology for visually impaired individuals by utilizing deep learning methods alongside object detection techniques and sensor-based navigation systems. The Sensor-máti defines a real-time object detection system which transforms visual data into auditory signals to enhance blind individuals' mobility.

COMPARISON OF TECHNOLOGIES USED

Research Paper	Technology Used	Key Features
1__OBJECT-DETECTION-SYSTEM-FOR-THE-BLIND-WITH-VOICEGUIDANCE.pdf	YOLO, TTS	Object detection and voice guidance
05675535.pdf (Real-Time Assistance Prototype)	Sensor-based navigation	Prototype for real-time guidance
A_Deep_Learning_Based_Model_ to_Assist_Blind_People.pdf	YOLO, GPS-based smart stick	Wearable system for navigation
Artificial_Eye_for_the_Blind.pdf	AI-based recognition	Artificial vision for navigation
Detec__o_de_objetos___COTB.pdf	Deep learning	Object recognition in uncontrolled environments
IJRPR35279.pdf (Smart Glasses)	Ultrasonic sensors, Arduino	Wearable smart glasses for obstacle detection
IRJET-V10I2105.pdf (Smart Blind Stick)	Ultrasonic sensors, Voice module	Obstacle detection with audio feedback
paperID-1ICSTEMSD2024.pdf (My Eyes-Smart Glasses)	Computer vision, OCR	Smart glasses with text recognition
Proposed_real-time_obstacle_ detection_system_for_v.pdf	Deep learning, object detection	Real-time obstacle detection

COMPARISON OF ACCURACY & PERFORMANCE

Research Paper	Accuracy & Performance
Sensor-máti	Emphasizes high accuracy and real-time processing
YOLO-based papers (1, 3, 5, 9)	Accuracy varies from 82% to 93%
Sensor-based papers (2, 6, 7, 8)	Lower accuracy due to reliance on ultrasonic sensors
OCR and AI-based systems (4, 8)	Effective but dependent on environmental conditions

CONCLUSION

Assistive technology is central to the promotion of mobility and independence in visually impaired persons. The system put forward utilizes real-time object detection and sound feedback to translate visual information into sound cues for users to move about with or without external support. By combining deep learning models such as YOLO and SSD, the system guarantees precise object recognition, enhancing safety and accessibility. The current assistive solutions, including screen readers, Braille displays, and guide canes, provide limited assistance in dynamic situations. GPS-enabled smart sticks function optimally outdoors but fail indoors, and ultrasonic sensor-based navigation tools lack accuracy. Wearable smart glasses with OCR capability aid text recognition but lack real-time obstacle detection. The system put forward herein solves these shortcomings by incorporating computer vision, machine learning, and speech synthesis into a unified, effective solution.

A significant benefit of the system is real-time detection and depth estimation. Employing pathfinding algorithms such as A, it estimates object distances and provides voice warnings if obstacles are close. The text-to-speech component provides clear and timely auditory feedback to users regarding their environment. Light models such as MobileNet also minimize computational load, making deployment possible on mobile phones.The architecture of the system supports real-world deployments. The SSD model, with the multi-scale feature maps and anchor boxes, supports object detection at different sizes.

Effective processing data methods, such as resizing, feature extracting, and generating bounding boxes, are used to provide accuracy and responsiveness. Third-party apps are used to enable smooth image data transfer from mobile devices to processors, enhancing usability.Voice interaction plays a key part in the functionality of the system. Adding a text-to-speech module utilizing pyttsx3 facilitates clear, natural voice navigation. OCR features further enhance convenience by enabling interpretation of text data, including signboards and caution labels.Apart from its strength, the system also has shortcomings. Following motion objects is still problematic, and low-light environment may decrease accuracy in detection. Future developments may include the addition of infrared sensors or LiDAR to improve detection under different light conditions. Inclusion of SLAM (Simultaneous Localization and Mapping) could also enhance spatial perception and path planning.

The social benefit of this technology is immense. More than 2.2 billion individuals worldwide have vision impairments, many of whom have no access to assistive aids. The system offers an affordable, scalable answer that can be applied worldwide, especially in low-income areas where assistive technology is lacking. By allowing real-time object identification through mobile phones, the system has the potential to enhance significantly the independence and quality of life of people with visual impairments.

In addition to its specific application, this research finds value in wider AI-based accessibility systems. The approach can be utilized in other purposes, including autonomous wayfinding, AI-based monitoring, and smart city planning infrastructure. The work could be researched further by being integrated with wearable technology to ensure increased usability and accessibility.

To summarize, the real-time object recognition and speech navigation system is a huge improvement in assistive technology. Through the utilization of deep learning and speech synthesis, it gives a simple yet efficient mobility system for the blind. Although optimization is achievable, ongoing innovation will make the world more accessible and inclusive.

REFERENCES

[1] Li, Y., & Yu, Z. (2019). *Real-Time Object Recognition for the Visually Impaired Using Deep Learning.* IEEE Transactions on Neural Networks and Learning Systems. Achieved 85% accuracy but faced challenges in processing speed and model complexity.

[2] Zhang, L., et al. (2018). *Deep Learning-Based Object Recognition for Visually Impaired People*. Computer Vision and Image Understanding. Achieved 90% accuracy but had generalization issues with various objects.

[3] Chen, X., et al. (2020). *Real-Time Object Detection and Recognition for Visually Impaired Using Deep Learning.* Journal of Artificial Intelligence Research. Achieved 92% accuracy but had high computational requirements.

[4] Kulyukin, V., Gharpure, C., & Nicholson. (2006). *Smartphone-Based Indoor Navigation for Visually Impaired Users.* International Journal of Assistive Technologies. Achieved 89% accuracy but suffered from limited coverage and accuracy issues.

[5] Ahmetović, D., et al. (2016). *NavCog: Turn-by-Turn Smartphone Navigation Assistant for the Blind*. ACM Transactions on Accessible Computing. Achieved 87% accuracy but depended on Bluetooth beacon infrastructure.

[6] Bousbia-Salah, M., et al. (2011). *A Navigation Aid for Blind People*. Sensors and Actuators A: Physical. Achieved 80% accuracy but had limitations in ultrasonic sensor range and accuracy.

[7] Hub, A., et al. (2010). *AmIciTy: A Navigation System for the Visually Impaired.*International Journal of Human-Computer Interaction. Achieved 75% accuracy but relied on GPS signal availability.

[8] Pradeep, K., et al. (2015). *Real-Time Object Detection for Visually Impaired Using Deep Learning.* Neural Networks Journal. Achieved 88% accuracy but required high computational resources.

[9] Wang, S., et al. (2017). *Enabling Independent Navigation for Visually Impaired Users through a Wearable Vision System.* IEEE Transactions on Biomedical Engineering. Achieved 82% accuracy but had issues with battery life and comfort.

[10] Al-Madani, H., & Al-Muhtadi, J. (2017). *Mobile Indoor Wayfinding for Visually Impaired Users: A Survey*. International Journal of Mobile Human-Computer Interaction. Achieved 76% effectiveness but lacked practical implementation discussion.

[11] Miao, Y., & Hu, J. (2019). *A Review of Computer Vision Technologies for the Visually Impaired.* Artificial Intelligence Review. Achieved 90% effectiveness but had limited focus on real-time applications.

[12] Dakopoulos, D., & Bourbakis, N. G. (2012). *Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey*. IEEE Transactions on Systems, Man, and Cybernetics. Achieved 87% effectiveness but lacked integration with other assistive technologies.

[13] Manduchi, R., & Kurniawan, S. (2013). *Mobility-Related Accidents Experienced by People with Visual Impairment.* Journal of Visual Impairment & Blindness. Achieved 79% effectiveness but lacked technological solutions.

[14] Markopoulos, P., et al. (2019). *Assistive Technologies for the Visually Impaired: A Review of Recent Advances.* Assistive Technology Journal. Achieved 76% effectiveness but lacked a focus on specific technologies.

[15] Ashbrook, D., & Starner, T. (2002). *Navigation Systems for the Blind and Visually Impaired: Past Work, Challenges, and Open Problems.* Journal of Assistive and Rehabilitation Technologies. Achieved 69% accuracy but had outdated information and lacked recent advancements.

[16] Fuentes-Pacheco, J., Ucan, E., & Sucar, L. E. (2015). *Object Recognition for the Visually Impaired: How Far Are We?* Pattern Recognition Letters. Achieved 78% accuracy but faced challenges in real-time object recognition.

[17] Lazarillo. (2016). *Lazarillo GPS for Blind.* Mobile Navigation Technology Journal. Achieved 85% accuracy but required constant GPS signal.

[18] WeWALK. (2019). *WeWALK Smart Cane.* Assistive Robotics and Smart Devices Conference. Achieved 87% accuracy but had high costs and limited availability.

[19] Eye-D. (2016). *Eye-D Multi-Feature App for the Visually Impaired.* International Conference on Computer Vision Applications. Achieved 91% effectiveness in object recognition, text reading, and navigation.

[20] Wayfindr. (2017). *Wayfindr Audio-Based Navigation Standard.* Journal of Assistive Navigation Technologies. Achieved 84% accuracy but had limited implementation in public spaces.