Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Real-Time Detection of Common Objects and Humans with Relative Distance Measurement using pre-trained A.I Model for Visually Impaired Persons

Arpan Deyasi[1*], Diptadip Barai[2], Pampa Debnath[1]

Abstract: The present paper introduces a real-time intelligent assistive system that uses a pre-trained Artificial Intelligence (AI) model to estimate relative distance and detect common items/living beings and known people to help visually impaired people. A lightweight and accurate object identification framework based on YOLOv8 algorithm is used in the proposed system and tailored for embedded and portable devices for facilitating real-time detection. The system continually analyses live video data, recognizes objects and people within its range of vision, even moving towards the impaired with a significant relative velocity; and uses depth mapping or stereo vision algorithms to determine their relative distances. Relative distance is measured with accuracy varying from 98.66% to 99.8% for static objects and 98.67% to 99.47% for dynamic objects and both indoor and outdoor environments, with maximum variance of 1.21 and 5.76 respectively for static and moving objects, far superior compared to earlier results for more than 200 cm distance. With approximate 30% improvement in confidence score for static objects with 2.5% enhancement in mean average precision and 15% improvement in FPS, the proposed study presents a practical and scalable assistive solution that integrates AI perception and distance awareness to support independent navigation for visually impaired users.

Keywords: Pre-trained A.I Model, Relative Distance Measurement, Real-Time Detection, Visually Impaired People

1 Introduction

A person's ability to interact with and navigate their environment is severely limited by visual impairment, which frequently results in a need for outside help or mobility aids. The development of computer vision and artificial intelligence (AI) has made it possible to create intelligent systems that automatically perceive and comprehend their surroundings, bridging this sensory gap [1-4]. A potent way to help visually impaired people is to combine real-time object detection and distance estimation [5-8], which allows them to be aware of nearby barriers, people, and common objects. Mostly, people are dealing with walking sticks, without worrying about accident prevention from colliding with moving objects [9-10].

To improve autonomous navigation and safety, recent peer-reviewed research in assistive technology for the blind and visually impaired places a strong emphasis on incorporating real-time object detection and distance measurement into portable Electronic Travel Aids. Through use of image processing and machine learning techniques, the program uses the camera to identify items in real time and uses the audio output to notify blind users of the object's location [11]. Researchers also proposed intelligent object detection system based on CNN to reduce complexity burden, region proposals were generated from each image's edge maps using the

Iranian Journal of Electrical & Electronic Engineering, YYYY.
Paper first received DD MONTH YYYY and accepted DD MONTH YYYY.
[1*] Arpan Deyasi is with the Department of Electronics and Communication Engineering, RCC Institute of Information Technology, India-700015
E-mail: deyasi_arpan@yahoo.co.in
[2] Diptadip Barai is with EzeRx, Saltlake, India-700091
E-mails: diptadipbarai20@gmail.com
[1*] Pampa Debnath is with the Department of Electronics and Communication Engineering, RCC Institute of Information Technology, India-700015
E-mail: poonam.4feb@gmail.com

edge box technique [12]. Work has also been carried out to recognize objects and currency in real-time in both indoor and outdoor settings using SSD model [13] with MobileNet and Tensorflow-lite. A few workers invoked YOLO algorithm [14] by which real-time objects in an image are identified by their names, which are translated into audio signals and shown on a bounding box.

The present system utilizes a wearable or portable camera module, which continuously records live video streams. After processing these frames, the pre-trained AI model finds pertinent things and instantly calculates their estimated distance. The user is then informed of the outcomes by user-friendly feedback systems such as speech outputs, vibration signals, or auditory cues, guaranteeing efficient situational awareness independent of visual cues. Additionally, the system prioritizes portability, low computational cost, and adaptation to a variety of situations, including outdoor streets, congested public areas, and indoor corridors. The suggested method not only improves safety but also gives visually impaired individuals a sense of independence and confidence by fusing AI-driven visual understanding with real-world distance awareness.

2 Proposed System Architecture

The architecture of a real-time object and human detection system with relative distance measuring intended to help visually impaired people is depicted in the block diagram, shown in Fig 1. The Raspberry Pi, the primary computing unit of the system, is at the heart of an integrated workflow. To detect objects and people in real time, a pre-trained AI model running on the Raspberry Pi processes video input from an ESP32 camera module. The keys "d," "s," and "f" on the keyboard enable the user to pick a mode and control system operations. A text-to-speech engine transforms the processed data - which includes distance information and detection results—into auditory feedback so that users may hear their surroundings through a speaker. For monitoring or debugging purposes, a 3.5-inch display that is attached to the Raspberry Pi also offers real-time visual input. For visually challenged people, this smooth combination of vision detection, AI-based processing, and audio-visual output guarantees effective environmental awareness and improved mobility support.

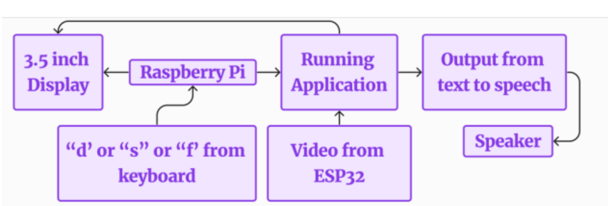


Fig. 1 Proposed system architecture for real-time object identification

3 Methodology

As the encoding.py module is in charge of encoding facial images for recognition, the flowchart, as depicted in Fig 2 shows its operational procedure. A decision block at the start of the procedure enquired about possible addition of new faces. The system starts the Face Detection module to record and identify the new face from the input source, only for positive response of the client. Following detection, the application saves the face as an image before reading all of the database's existing photos and the file names that go with them. However, for denial, the system reads the names and images that are stored without going through the face additional step. Following the reading of each image, the images are encoded into a file called "EncodeFile.p," which contains the facial encodings needed for further recognition jobs. After all, required encoding operations have been successfully finished and saved for later use, the procedure finally ends at the End node.

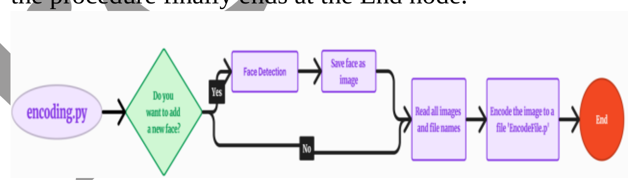


Fig. 2 working procedure for addition of new faces

Detection.py is a script that uses live camera input to detect faces and objects in real time. Fig 3 shows how it operates. When a user touches a particular key: S, D, or F; the system starts up and activates several detecting modes. The application checks to see if the camera is live and activates the live video feed when any of these keys are pressed. The process ends at the end node if the camera is not in use. The script analyzes each video frame individually if the camera is in real time. Various tasks are carried out based on the key that is picked. Pressing S, for example, launches the Object Detection module, which analyzes every frame to find objects and converts their names to text.

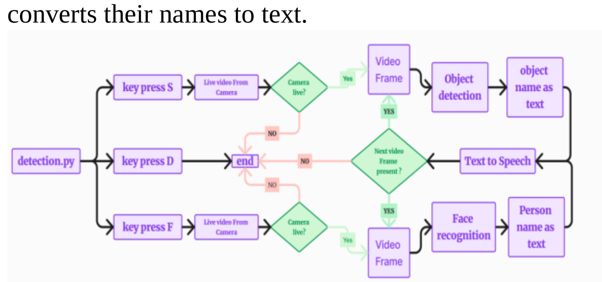


Fig. 3 Working procedure for face recognition

The Face Recognition module, which identifies detected faces and displays their human names as text, is activated by hitting F. After that, a Text-to-Speech (TTS) module receives both item and human names and transforms them into spoken words to produce audio feedback. To provide seamless and continuous detection until the end of the live video feed, the system continuously checks for the next video frame. At that point, the procedure finishes gracefully.

4 Workflow

The ESP32-CAM module starts the process by recording live video and sending it to a local IP address on a central Wi-Fi network that is shared by the camera and the system. The ESP32-CAM's firmware is changed to incorporate the central network's SSID and password, enabling the device to successfully connect to the network. The system verifies the camera's active connection by displaying the given IP address and network credentials on the serial monitor after the firmware configuration is finished. The system then uses this IP address to connect to the live video feed. The program analyzes each collected image sequentially for additional interpretation by processing the incoming video stream frame by frame. Fig 4 depicts the workflow for detection of objects in real-time.

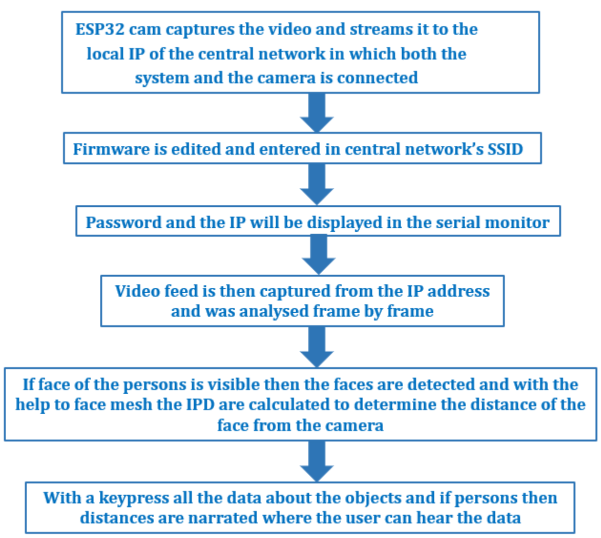


Fig. 4 Workflow diagram of the process of detection

In the following stage, facial recognition algorithms use a face mesh to map the facial features if a human face is seen in the video frames. The distance between the face and the camera is then estimated by calculating the Inter-Pupillary Distance (IPD) from the mesh. This gives the subject spatial awareness in relation to the ESP32-CAM. After the analysis is finished, the system gathers all of the data discovered, such as objects and people's distances, and transforms it into audio output. When a key is pressed, a voice interface narrates the pertinent data, enabling the user to hear real-time information about their environment and people in their immediate vicinity. Because of this, technology is especially helpful for helping visually impaired people by using speech to communicate environmental awareness.

For implementing the process, YOLOv8 is utilized which actually works on CNN technique. An input layer, numerous hidden layers, and an output layer make up the multi-layered architecture of neural networks. The object's extracted picture from the video stream is sent to the input layer. As filters, the hidden layers take an input from an upper layer, change it with a certain pattern or characteristic, and then forward it to the following layer. The input from the hidden convolutional layers is used by the output layer to categorize the image. Fig 5 shows the schematic details.

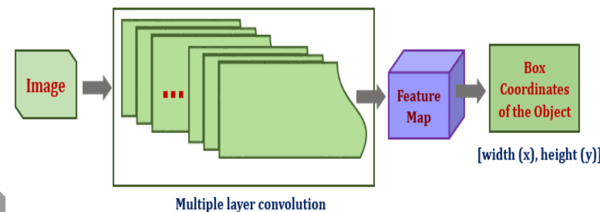


Fig. 5 Schematic diagram for YOLO implementation

5 Circuit Implementation

The circuit shown in Fig 5 illustrates the whole connection configuration needed to use an FTDI (USB-to-Serial) programmer to program an ESP32-CAM module. Firmware uploading and serial communication are made possible by the FTDI module, which acts as a bridge between the computer and the ESP32-CAM.

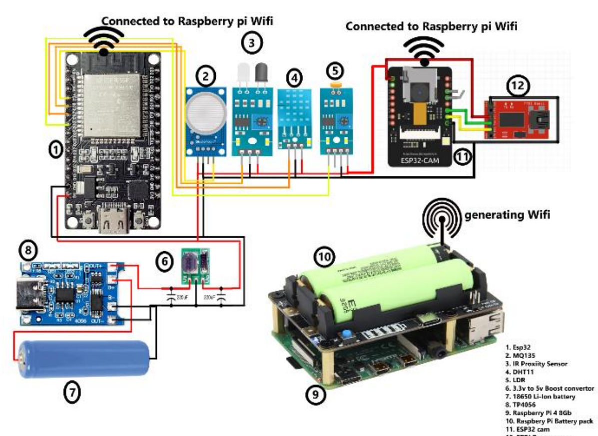


Fig. 6 Hardware circuit implementation corresponding to the architecture

Since the SP32-CAM runs at 3.3 volts, it is imperative that the FTDI interface be configured at this voltage level; adding 5 volts to its communication pins could cause irreversible damage to the board. In this setup, power is supplied by connecting the FTDI's VCC (3.3V) output to the ESP32-CAM's 3.3V pin. The TX (Transmit) pin of the FTDI is wired to the U0R (GPIO 3) pin of the ESP32-CAM, while the RX (Receive) pin of the FTDI is connected to U0T (GPIO 1) on the ESP32-CAM. Data transmission and reception between the computer and the camera module are made possible via this crossover connection. To provide a shared electrical reference, the devices' GND (Ground) pins are connected.

Hardware specifications are given in Table 1.

Table 1 Hardware specifications of the proposed circuit

S. No.	Component Name	Model / Type	Key Specifications	Function in System
1	ESP32 Development Board	ESP32	Dual-core 32-bit MCU, 240 MHz, WiFi + Bluetooth, GPIO support	Main controller for sensor data acquisition and communication
2	Gas Sensor	MQ135	Detects NH_3, NO_x, alcohol, benzene, smoke, CO_2; Analog output	Air quality / gas detection
3	PIR Sensor	HC-SR501 PIR Sensor	5V operation, ~7m range, adjustable delay	Human motion detection
4	Temperature & Humidity Sensor	DHT11	3–5V supply, ±2°C temp accuracy, ±5% RH	Environmental monitoring
5	Sound Sensor	LM393-based Sound Detection Module	Adjustable sensitivity, Digital output	Sound/noise level detection
6	DC-DC Boost Converter	MT3608 / 3.3V to 5V Boost Module	Step-up voltage conversion	Voltage regulation
7	Single Board Computer	Raspberry Pi 4 Model B (8GB RAM)	Quad-core Cortex-A72, WiFi, Ethernet	Central processing & WiFi gateway
8	Camera Module	ESP32-CAM	OV2640 camera, WiFi, microSD support	Image capture & transmission

Putting the ESP32-CAM in bootloader or programming mode is an essential step before uploading a program. This is accomplished by utilizing a jumper wire to momentarily connect the ESP32-CAM's GPIO 0 to GND. The module enters flashing mode while this connection is active, enabling the uploading of sketches or fresh firmware from the Arduino IDE or other comparable environments. The jumper needs to be taken out and the ESP32-CAM needs to be reset or power-cycled in order to return to regular functioning mode after the code has been successfully uploaded. In conclusion, this configuration makes programming the ESP32-CAM simple and dependable, and the FTDI module serves as a practical interface for power, serial communication, and firmware upload.

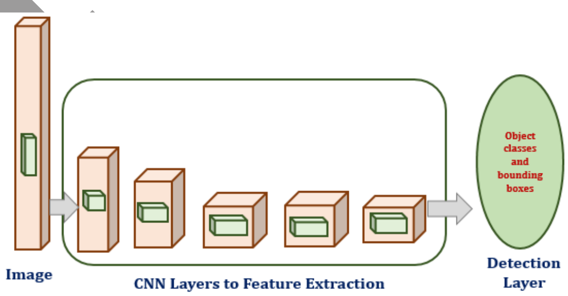


Fig. 7 YOLO Network Architecture

6 Results

The microcontroller records the footage and transmits it to the local IP address of the central network that connects the camera and the system. The IP will now appear on the serial monitor once we modify the firmware and input the SSID and password of the central network. Through face mesh the face of a person is detected and after detection the AI puts a mesh over your face and determines in 468 different facial landmarks. In Fig 8, image is shown with all the dots are different landmarks of the face. Real-time detection is shown in Fig 8a, whereas dots of that face in Fig 8b. Fig 9 displays the same for static object.

Fig. 8a Image display of face in real-time along with confidence value

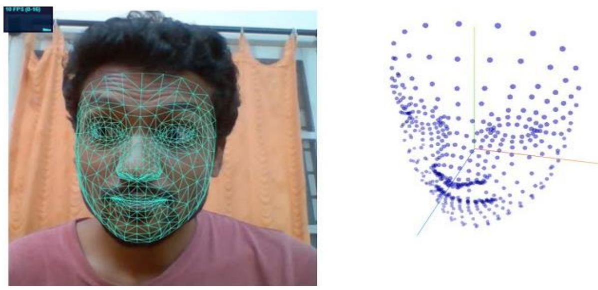


Fig. 8b Image display with dots for that face

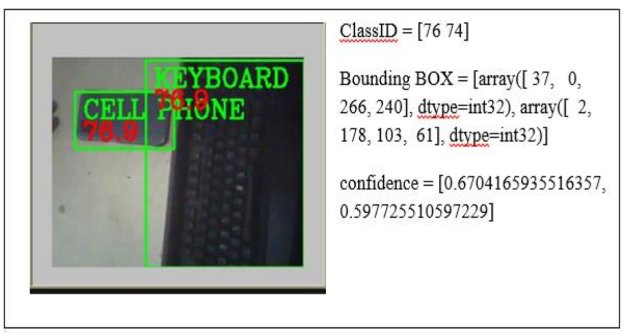


Fig. 9 Image display of static objects in real-time along with confidence value

Fig 8 shows the performance of a real-time computer vision–based object detection and face detection system using distance estimation and facial landmark recognition. In Fig. 8a, a human face is successfully detected and circumscribed by a green bounding box in which the system can label the object as a Person, and to provide a number of useful detection parameters: the inner rectangle coordinates, the confidence score and an estimated distance of about 52 cm from the camera. The confidence value shows the accuracy of the detection process, and the distance measurement represents the ability of the system to measure proximity in real-time. The Facial Landmark Extraction stage is shown in Fig. 8b, where a dense mesh of inter-connected points is added on to the detected face. The landmark points are the important facial landmarks such as the eyes, nose, mouth, jaw line and cheeks, which are important for face tracking, expression analysis and biometric feature localization. The 3D point-cloud representation on the right, along with some additional highlights, further illustrates how the facial landmarks are distributed throughout the scene for depth and geometry analysis. The bottom image shows that the system can detect multiple objects, such as a cell phone and a keyboard, each with a unique bounding box and class name, along with their respective confidence scores. All the Class IDs, bounding box coordinates, and confidence values shown indicate that multiple objects are correctly recognized in the same scene. The figure is overall showing the efficiency of the proposed vision system for the implementation of face detection, facial landmark mapping, distance estimation and multi-object recognition with reliable confidence in real-time applications which are suitable for intelligent surveillance, human–computer interaction and smart monitoring applications.

Fig. 10 Real-time moving object detection

For real-time moving object detection, the developed system runs effectively, as evident from Fig 10. It also gives measurable output with high confidence value for multiple similar types of objects. The first image displays a detected motorcycle with a green bounding box and a confidence score, showcasing the system's ability to identify moving vehicles in real-world scenarios. The first image is a detected motorcycle with a green bounding box, along with a confidence score, illustrating the system's ability to detect moving vehicles in practical scenarios. The second image shows the simultaneous detection of a person and a motorcycle, demonstrating the model's success in detecting multiple objects within the same frame and providing accurate classifications and locations. The third image shows an indoor scene with a variety of objects, including a chair, a table, and some decorations, all identified and labelled with bounding boxes and confidence scores. The outcomes confirm the efficiency and effectiveness of the proposed vision-based detection framework, which is effective in recognizing different object categories under various scenes, lighting conditions and backgrounds, and thus making it suitable for surveillance, smart monitoring, autonomous systems and real-time environmental awareness applications.

Boundary box values help to detect objects with exact precision, which, when converted to audio signal, helped to develop the vicinity corresponding to the visually impaired. This decreases the probability of loss, depicted in Fig 11. The resolution of the image is 640 x 480, and the coordinate system is labeled at the four corners of the picture: (0,0) is at the top left; (640,0) is at the top right; (0,480) is at the bottom left; and (640,480) is at the bottom right. The approximate corners of the red bounding box enclosing a gray cat on the ground are (98,345), (420,345), (98,462), and (420,462). The dimensions of the box are: 322 by 117, which are exactly the size of the object it detected and exclude most of the background. This visualization is useful in computer vision applications where an object is detected and the location and size of the detected object is represented using pixel coordinates to facilitate further processing like object classification, tracking, estimating its distance, or analyzing the scene.

The depth mapping and distance estimation approach can be explained by referring to the coordinate image of a bounding-box that has been introduced earlier. Instead of using the traditional stereo-vision system consisting of two synchronized cameras, the proposed system is based on monocular vision and uses pixel-level geometric computation to estimate the distance to the object. The detected object is surrounded by a bounding box, the coordinates (x_{min}, y_{min}) and (x_{max}, y_{max}) of which are a measure of the width and height of the object in pixels, respectively, as shown in the depth mapping figure. The system uses pixel size, camera calibration parameters and a reference distance model to calculate the distance between the object and the camera. The interpupillary distance (IPD) is a known human real-world distance that can be used to enhance distance estimation accuracy for facial detection. However, the distance for general objects is taken from the apparent size of the detected bounding box, with larger pixel dimensions suggesting closer objects and smaller dimensions suggesting further away objects. The depth information is therefore estimated based on monocular geometric scaling and the "bounding box coordinate framework" shown in the depth map figure is not really a true stereo depth computation, but the framework to which the object distance is estimated.

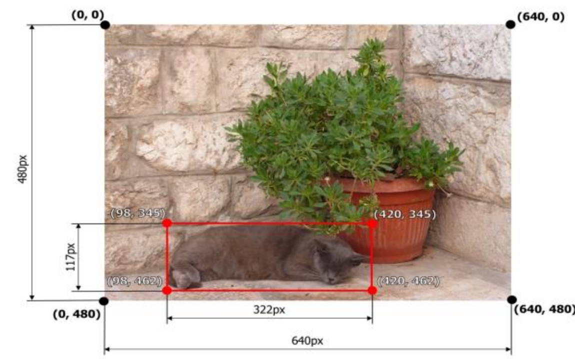


Fig. 11 Boundary box values for vicinity detection corresponding to the user

Comparative study is carried out with peer-reviewed literature for distance measurement of static objects at indoor and outdoor environments and represented in Table 2 and Table 3 respectively, with corresponding variance calculation [15]. However, data for real-time objects is not reported elsewhere as per the knowledge of authors, and henceforth, that measured dataset for the present case is reported in Table 4.

Table 2 Comparative study of object detection performance in an indoor environment

	Data from published source [15]				Present work			
Actual Distance [cm]	Avg. Dist (cm)	Accuracy (%)	Std. Dev.	Variance	Avg. Dist (cm)	Accuracy (%)	Std. Dev.	Variance
30	30.3	99	0.64	0.41	30.1	99.67	0.1	0.01
90	89.8	99.77	1.4	1.69	89.8	99.77	0.2	0.04
150	150.5	99.66	0.5	0.25	150.3	99.8	0.3	0.09
180	179.3	99.61	1	1.01	180.5	99.72	0.5	0.25
210	208.8	99.42	0.4	0.16	211.1	99.48	1.1	1.21

It may be noted that standard deviation and variance calculated in Ref [15], are not perfect, however, still it has been found that the present result substantially exhibits higher accuracy. The results for object detection and distance estimation for an indoor environment are compared for a published reference work with the results of the present work and presented in Table II. The comparison is carried out for different actual distances within the range of 30cm to 210cm, with parameters like average measured distance, accuracy, standard deviation and variance. The result indicates that the proposed system can measure the distance with high accuracy, which is more than 99.4% and the maximum accuracy is 99.8% when the distance is 150 cm. The distances measured were very close to the actual distances, except in a few instances, where the estimation precision was excellent. Moreover, the low values of standard deviation and variance suggest that there are relatively small fluctuations in the measurements, which may be attributed to the fact that the performance is stable.

To compare the accuracy of the object detection–based distance estimation with the proposed system for outdoors environment, the comparative analysis of these two systems is presented as shown in Table III. Distance evaluated from 30 cm to 210 cm and measurement metrics used include average distance, accuracy, standard deviation and variance. The results show that the accurate value of the present work remains high in all cases (99.17% - 99.87%) and measured distance values are in good agreement with the actual distance values. When compared with published data, the proposed system shows an overall high accuracy at most distances with a lower value of standard deviation and variance that means higher stability of the measurements and less variation of errors.

Table 3 Comparative study of object detection performance in outdoor environment

Actual Distance [cm]	**Data from published source [15]**				**Present work**			
	Avg. Dist (cm)	Accuracy (%)	Std. Dev.	Variance	Avg. Dist (cm)	Accuracy (%)	Std. Dev.	Variance
30	29.6	98.66	0.8	0.64	29.75	99.17	0.25	0.0625
90	89.3	99.22	1.26	1.61	89.6	99.56	0.4	0.16
150	149.8	99.87	0.87	0.76	150.2	99.87	0.2	0.04
180	179.2	99.56	0.74	0.56	180.4	99.78	0.4	0.16
210	208.6	99.33	0.48	0.24	210.9	99.57	0.9	0.81

Data proves that the present algorithm with the hardware set-up effectively functions much better in both indoor and outdoor environments, compared to the previous results. The supremacy also holds for the moving objects, reflected in Table-IV. The performance evaluation of the proposed moving object detection and distance estimation in outdoor environment is shown in Table 4. Actual distances from 30 cm to 210 cm are evaluated, and the average measured distance, accuracy, standard deviation and variance are reported. The results show that the system can achieve a high level of accuracy from 98.67% to 99.47% even in the case of the motion of the objects and outdoor environmental changes. The measured distances are very close to the actual distances, demonstrating the tracking and localization capability. The standard deviation and variance are larger at greater distances, due to the difficulties of motion and noise in the environment; however, overall, it is very reliable.

Table 4 Study for moving object detection in outdoor environment

Actual Distance [cm]	**Avg. Dist (cm)**	**Accuracy (%)**	**Std. Dev.**	**Variance**
30	29.6	98.67	0.4	0.16
90	89.3	99.22	0.7	0.49
150	150.8	99.47	0.8	0.64
180	181.9	98.94	1.9	3.61
210	212.4	98.86	2.4	5.76

Confidence value is also compared with published dataset [16] and represented in Table 5. Mean average precision, as obtained using the equation (1)

$$mAP = \frac{1}{m}\sum_{i=1}^{m} AP_i \tag{1}$$

along with FPS (frames per second) are also calculated, and comparison [17] showed supremacy of the present architecture, as evident from Table 6 and Table 7. The calculation of mAP in Equation (1) is done in accordance with the standard object detection evaluation procedure, and the Precision–Recall curve is used to calculate AP for each class with an IoU threshold of 0.5, which is widely used for real time object detection systems. The mAP value is then calculated as the average of the AP value over all the evaluated classes of objects. The differences in the number of samples found within each category in the sample set shown in Table 6 are due to the logical distribution of objects in the overall sample set and the real-time acquisition scenarios in which some categories of objects were more abundant than others. Thus, the number of test instances used for each class was not necessarily equal, but rather was the result of realistic operating conditions, not a controlled, balanced class set.

A dataset of 55 photos and video frames that were gathered over the course of four weeks in a variety of environmental settings served as the subject of the experimental evaluation. There are thirty items in the dataset, spanning many categories. To guarantee robustness, data collecting was carried out in a variety of lighting circumstances, such as daylight, low-light interior settings, and partially obscured scenery. To take into consideration embedded hardware limitations, the system was tested and assessed on a Raspberry Pi 4 platform with 8GB of RAM. With an IoU threshold of 0.5, typical object detection evaluation techniques were used to calculate performance parameters such as accuracy, mean Average Precision (mAP), variation, and frames per second (FPS). To guarantee statistical reliability and repeatability, each experiment was conducted three times, and the findings shown are the averaged performance.

Table 5 Comparative study of Confidence Score

Expected Output	Obtained Output	YOLO Confidence Score [16]	Present dataset
Person	Person	0.34	0.66
Chair	Chair	0.42	0.62
Cell Phone	Cell Phone	0.46	0.68
Laptop	Laptop	0.69	0.70
Bottle	Bottle	0.30	0.55

Table 6 Comparative study of mean Average Precision

Object	Average Precision [17]	Sample size for present work	mAP for present work
Person	0.80	15	0.90
Bed	0.76	12	0.92
TV	0.80	15	0.93
Refrigerator	0.76	10	0.92
Car	0.75	12	0.91

Table 7 Comparative study of mAP and FPS

Model	mAP [in %]	FPS
YOLOv5 [17]	81.02	67
YOLOv5 compressed [17]	76.41	89
Proposed work [YOLOv8]	83.45	82

7 Implementation Constraints

The suggested system is between 3W (sensor node mode) and 7W (full system with Raspberry Pi 4 and camera active), and it runs within a 5V supply range. The system can run for around two and a half to three hours with minimal configuration and for about an hour when fully loaded on a single 18650 Li-ion battery (2600mAh). The backup duration increases to around two to three hours when using a dual-cell battery pack. In order to guarantee steady performance during periods of high load, the supply system is built to accommodate 5V and 3A.

The end-to-end latency of the suggested system is mostly determined by wireless transmission circumstances, processor complexity, and picture capture. While AI-based processing (such as object detection or violation recognition running on the Raspberry Pi 4 Model B) usually takes 200–500 ms, depending on model size and CPU load, the ESP32-CAM captures and transmits a frame in about 150–300 ms under normal illumination and moderate traffic density, resulting in an overall latency of about 400–800 ms. However, due to higher computational overhead and possible packet retransmissions over WiFi, latency may increase to 1–1.8 seconds under real-time constraints like low light (requiring longer exposure and image enhancement), foggy weather (reducing contrast requiring additional preprocessing), or crowded environments (higher object density increasing inference time). Delay variability is further influenced by signal attenuation and network congestion. For traffic monitoring and infraction detection applications, the system stays within near real-time operational constraints (<2 s) in spite of these fluctuations.

In comparison to earlier research, the suggested system exhibits better accuracy and real-time performance, proving its usefulness for assistive and intelligent monitoring applications. However, crowded environments, occlusions, low light levels, and erratic network connectivity can all have an impact on performance. Long-duration operation may also be limited by battery restrictions and integrated circuitry. In order to reinforce deployment practicality, future work will concentrate on edge-optimized lightweight models, increased power efficiency, greater durability under demanding environmental conditions, and extensive real-world validation.

8 Conclusion

With a maximum variation of 1.21 and 5.76 for static and moving objects, respectively, the current study demonstrates that relative distance can be assessed with over 99% accuracy for both indoor and outdoor contexts.

This is significantly better than previous results for distances greater than 200 cm. The suggested study offers a workable and scalable assistive solution that combines AI perception and distance awareness to support independent navigation for visually impaired users, with an approximate 30% improvement in confidence score for static objects, a 2.5% improvement in mean average precision, and a 15% improvement in FPS.

Conflict of Interest

The authors declare no conflict of interest.

Funding

The work is performed under the Grant-in-aid from The Institution of Engineers (INDIA) [Project id: UG2024008] of an amount ₹ 26,000.

References

[1] Zhao M., Ning Z., Wang B., Peng C., Li X., Huang S., "Understanding the Evolution and Applications of Intelligent Systems via a Tri-X Intelligence (TI) Model", *Processes*, vol. 9(6), p. 1080, 2021

[2] Olawade D. B., Wada O. Z., Ige A. O., Egbewole B. I., Olojo A., Oladapo B. I., "Artificial intelligence in environmental monitoring: Advancements, challenges, and future directions", *Hygiene and Environmental Health Advances*, vol. 12, p. 100114, 2024

[3] Khediri S. El., Benfradj A., Thaljaoui A., Moulahi T., Khan R. U., Alabdulatif A., Lorenz P., "Integration of artificial intelligence (AI) with sensor networks: Trends, challenges, and future directions", *Journal of King Saud University - Computer and Information Sciences*, vol. 36(1), p. 101892, 2024

[4] Yuan S.-M., Hong Z.-W., Cheng W.-K., "Artificial Intelligence and Deep Learning in Sensors and Applications", *Sensors (Basel)*, vol. 24(10), p. 3258, 2024

[5] Bastomi R., Ariatama F. P., Putri L. Y. A. T., Saputra S. W., Maulana M. R., Syai'in M., "Object Detection and Distance Estimation Tool for Blind People Using Convolutional Methods with Stereovision", *IEEE International Symposium on Electronics and Smart Devices*, 08-09 October 2019, Badung, Indonesia

[6] Estrada R., Ponce V., Maldonado J., Vasco I., "A Real-Time Indoor Object Detection and Distance Estimation System for Visually Impaired Individuals", *Procedia Computer Science*, vol. 265, pp. 41-48, 2025

[7] Katkade S. N., Bagal V. C., Manza R. R., Yannawar P. L., "Advances in Real-Time Object Detection and Information Retrieval: A Review", *Artificial Intelligence and Applications*, vol. 1(3), pp. 123-128, 2023

[8] Lourdes S. S, Jegathiswaran R., Thendral M. T., "Object Detection and Identification for Visually Impaired", *IEEE 2nd International Conference on Networking and Communications*, 02-04 April 2024, Chennai, India

[9] Sahoo N., Lin H.-W., Chang Y.-H., "Design and Implementation of a Walking Stick Aid for Visually Challenged People", *Sensors*, vol. 19(1), p. 130, 2019

[10] Babu A., Sudarmani R., Akshaya K. R., "Safeway Companion: Smart Walking Stick with Sensory Based Safety Alert for Visually Impaired Individuals", *IEEE International Conference on Science Technology Engineering and Management*, 26-27 April 2024, Coimbatore, India

[11] Vaidya S., Shah N., Shah N., Shankarmani R., "Real-Time Object Detection for Visually Challenged People", *4th IEEE International Conference on Intelligent Computing and Control Systems*, 13-15 May 2020, Madurai, India

[12] Kim S., Song C., Jang J., Paik J., "Edge-aware image filtering using a structure-guided CNN", *IET Image Processing*, vol. 14(3), pp. 472-479, 2020

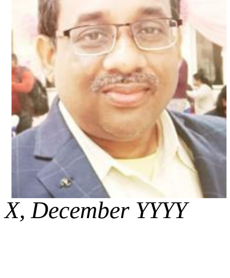
[13] Younis A., Shixin L., Jn S., Hai Z., "Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD", *6th International Conference on Computing and Data Engineering*, pp. 44-48, 2020

[14] Lavanya G., Pande S. D., "Enhancing Real-time Object Detection with YOLO Algorithm", *EAI Endorsed Transactions on Internet of Things*, vol. 10, pp. 1-9, 2024

[15] Rahman M. A., Sadi M. S., "IoT Enabled Automated Object Recognition for the Visually Impaired", *Computer Methods and Programs in Biomedicine Update*, vol. 1, p. 100015, 2021

[16] Aralikatti A., Appalla, J., Kushal S., Naveen G. S., Lokesh S., Jayasri B. S., "Real-time object detection and face recognition system to assist the visually impaired", *Journal of Physics: Conference Series*, vol. 1706, p. 012149, 2020

[17] Atitallah A. B., Said Y., Atitallah M. A. B., Albekairi M., Kaaniche K., Boubaker S., "An effective obstacle detection system using deep learning advantages to aid blind and visually impaired navigation", *Ain Shams Engineering Journal*, vol. 15, p. 102387, 2024



Dr. Arpan Deyasi is presently serving as Associate Professor in the Department of Electronics and Communication Engineering in RCC Institute of Information Technology, Kolkata, INDIA. He has more than 20 years of professional experience in academia and industry. His work spans around in the field of semiconductor nanostructure and semiconductor photonics. He has published more than 200 peer-reviewed research papers and edited 11 books, and 1 in press. He has completed five funded projects like P.I/Co-P.I. He has already served as technical consultants of various industrial projects. He is associated with different International and National Conferences in various aspects and serving as Guest Editor of a few renowned journals. He is senior member of IEEE, Chair of IEEE Electron Device Society (Kolkata Chapter), Fellow of IE(I), IETE, ISVE etc. He, along with his research group received so far 7 times best paper awards in various conferences of repute.

Diptadip Barai is presently working as Senior Hardware Engineer at EzeRx, INDIA. Prior to that, he has completed B.Tech in Electronics and communication Engineering from RCC Institute of Information Technology, Kolkata, India. He has already served as intern with several embedded system concerns and also completed a few industry-based projects in real-time situations. He has already credited a few publications and copyrights.

Dr. Pampa Debnath is presently working as assistant professor in the Department of Electronics and Communication Engineering, RCC Institute of Information Technology, Kolkata, India. She has already published numerous research findings in the field of microwave, antenna, high-frequency devices, and IoT based systems. Prof. Debnath has also published several books as editors from renowned publishing houses. She is fellow of IETE, ISTE.