# Depth Analyser for Visually Impaired Using Image Processing

## Third Eye

Mytreyan (2022115102)
Information Science and Technology
College of Engineering Guindy, Anna University
Chennai, India
mytreyan197@gmail.com

Suganth (2022115308)
Information Science and Technology
College of Engineering Guindy, Anna University
Chennai, India

Tamilarasan (2022115317)
Information Science and Technology
College of Engineering Guindy, Anna University
Chennai, India

***Abstract*— Visually impaired individuals often face significant challenges in navigating environments independently due to a lack of spatial awareness and real-time feedback regarding nearby obstacles. Traditional assistive tools like white canes and guide dogs provide limited directional and distance-based information, which restricts the user's ability to fully understand and react to dynamic surroundings. To address these limitations, this paper presents a smart, real-time assistive system titled "Depth Analyser for Visually Impaired Using Image Processing". The proposed system is designed to detect and classify objects in real-time using advanced deep learning techniques, and to estimate their spatial position and distance from the user, providing immediate and intuitive feedback through audio cues.

The system is built around YOLOv10 (You Only Look Once version 10), an efficient and accurate object detection model capable of processing live video streams with low latency. YOLOv10 identifies objects within the camera's field of view and categorizes them based on their position—left, center, or right. In addition, a depth estimation component calculates the approximate distance of each detected object from the camera, enabling the system to generate descriptive audio messages such as "The bag is 1.2 meters on the left." This information is crucial for safe and autonomous navigation in unfamiliar or cluttered environments.

The architecture consists of a React.js-based frontend for capturing the video feed using the browser's webcam API and a Python Flask backend responsible for processing frames using OpenCV, NumPy, and PyTorch. The two components communicate in real-time via Socket.IO, allowing seamless transmission of data between the client and the server. The processed output, which includes object type, direction, and distance, is then converted to speech using text-to-speech libraries, enabling the visually impaired user to receive instant and comprehensible feedback.

This solution is not only technically efficient but also socially relevant, addressing a critical accessibility need. According to the World Health Organization (WHO), over 285 million people globally are visually impaired, with nearly 39 million being completely blind. The proposed system provides an innovative and scalable approach to assistive technology, empowering users to perceive their environment more clearly and make safer movement decisions. The modular design also allows the integration of haptic feedback, edge computing, and potential voice-command features for enhanced interactivity and personalization.

Experimental evaluations demonstrate the system's capability to detect everyday objects and accurately estimate their spatial location in real-time. The architecture is optimized for low computational overhead and real-time responsiveness, making it suitable for both desktop and embedded applications. In simulated user environments, the system delivered consistent and reliable feedback that improved the user's situational awareness.

In conclusion, the proposed depth analyser enhances the mobility and autonomy of visually impaired users by merging modern computer vision with intelligent feedback mechanisms. The project showcases the potential of deep learning and real-time processing in creating inclusive and accessible technologies, contributing

meaningfully to both academic research and practical social impact.

***Keywords—****Assistive Technology, Object Detection, Depth Estimation, YOLOv10, Computer Vision, Real-Time Processing, Visually Impaired Navigation, Audio Feedback*

## I. INTRODUCTION

Visually impaired individuals encounter numerous challenges in perceiving their surroundings and navigating safely through complex environments. Traditional assistive tools such as white canes and guide dogs offer only limited spatial awareness and fail to provide real-time feedback on object types, positions, or distances. These limitations significantly affect the independence, mobility, and confidence of visually impaired individuals, especially in unfamiliar or dynamic environments.

With advancements in artificial intelligence (AI) and computer vision, there is a growing opportunity to develop intelligent systems that can enhance environmental perception for the visually impaired. Object detection models powered by deep learning have made it possible to identify and localize multiple objects in real-time with high accuracy. When combined with depth estimation techniques and intuitive feedback mechanisms such as audio narration or haptic signals, these technologies can significantly improve the user's spatial awareness and decision-making capabilities.

This paper presents a novel real-time assistive system titled *"Third Eye"* which integrates deep learning-based object detection and distance measurement to aid visually impaired individuals. The proposed system utilizes YOLOv10 for object detection and implements depth estimation methods to determine the spatial position of objects relative to the user. It provides immediate audio feedback to convey object location and distance in an intuitive and understandable format.

The system is developed using a React.js-based frontend to capture live video feeds and a Python Flask-based backend to perform image processing and inference. Communication between the two components is achieved through Socket.IO, enabling seamless real-time performance. The integration of these technologies results in a lightweight and scalable solution that can be extended for wearable or mobile applications in the future.

This introduction establishes the motivation and purpose behind the project while highlighting the significance of combining deep learning and real-time communication to support visually impaired individuals. The following sections provide a detailed description of the system architecture, implementation, and evaluation of the proposed solution.

## II. LITERATURE SURVEY

The development of assistive technologies for visually impaired individuals has been an area of active research over the years. Traditional tools like white canes and guide dogs have limitations in providing real-time, dynamic feedback about the environment.

With advancements in computer vision, deep learning, and embedded systems, modern solutions are increasingly capable of providing more intuitive and accurate assistance. This literature survey presents a review of existing research and technologies related to assistive systems for visually impaired users, focusing on object detection, depth estimation, and audio feedback mechanisms.

### *A. Existing solutions and Techniques*

White canes and guide dogs provide basic mobility but lack comprehensive spatial awareness and real-time feedback. Computer vision-based solutions, including YOLOv10, enable real-time object detection and classification, enhancing situational awareness for visually impaired individuals.

- VisionAID is a system that uses stereo cameras and depth sensors to detect obstacles in real-time, providing auditory feedback to users. This approach enhances situational awareness but requires expensive hardware and can struggle in low-light environments.
- Developed by Microsoft, Seeing AI is a mobile app that uses a smartphone camera to read text, recognize faces, and identify objects in real-time. It provides audio feedback, but its primary focus is on text and face recognition rather than spatial navigation.

- SmartCane uses ultrasonic sensors to detect obstacles and provides vibration feedback. While it offers better range than a traditional white cane, it still lacks advanced features like real-time object recognition or depth estimation.

## III. Depth Estimation

Young's approach to depth estimation offers a more cost-effective solution for measuring object distances. This method utilizes a single camera to generate depth maps from 2D images. To find the depth values of detected objects, the process typically involves the following steps: first, the depth information is extracted using a pre-trained model, which predicts depth values for each pixel in the captured image.

Once the depth values are obtained, the mean depth is calculated across the region of interest, allowing for a representative distance measurement of the object. This mean value is then inverted to convert depth into distance. The final distance measurement is computed by multiplying the inverted mean depth by a scale factor, which accounts for the camera's intrinsic parameters and any necessary calibration specific to the application.

This approach not only simplifies the hardware requirements by using a single camera but also provides real-time feedback on object distances, making it highly suitable for assistive technologies aimed at enhancing spatial awareness for visually impaired users.

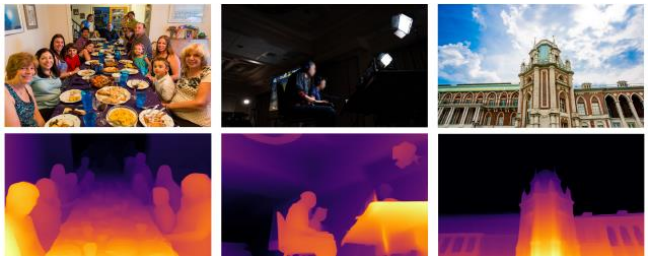

Figure 3.1 Depth Maps of various objects

### *A. YOLOv10*

The In recent years, deep learning-based object detection techniques have gained significant traction, with models like YOLO (You Only Look Once) leading the field due to their ability to perform real-time detection with impressive accuracy. The YOLOv10x model represents an advancement in this lineage, optimizing the balance between speed and precision through innovative architectural enhancements.

Unlike its predecessors, YOLOv10x incorporates a more refined feature extraction process, enabling it to detect a wider variety of objects in diverse environments while maintaining high frame rates. This capability is particularly beneficial in applications aimed at assisting visually. impaired users, where rapid object detection and distance estimation are critical for providing timely auditory feedback. By leveraging YOLOv10x in our project, we can enhance the user ' s interaction with their surroundings, thereby improving their independence and navigation in real-time.

The architecture of YOLOv10 builds upon the strengths of previous YOLO models while introducing several key innovations. The model architecture consists of the following components:

• Backbone: Responsible for feature extraction, the backbone in YOLOv10 uses an enhanced version of CSPNet (Cross Stage Partial Network) to improve gradient flow and reduce computational redundancy.

• Neck: The neck is designed to aggregate features from different scales and passes them to the head. It includes PAN (Path Aggregation Network) layers for effective multiscale feature fusion.

• One-to-Many Head: Generates multiple predictions per object during training to provide rich supervisory signals and improve learning accuracy.

• One-to-One Head: Generates a single best prediction per object during inference to eliminate the need for NMS, thereby reducing latency and improving efficiency.

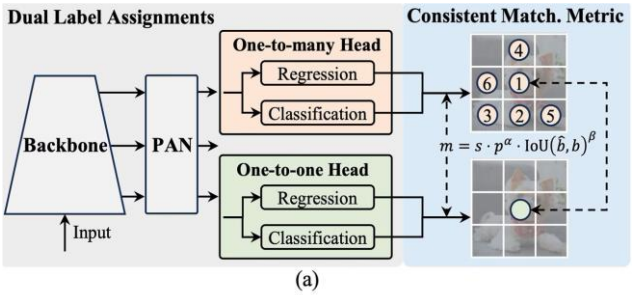


Figure 3.2 YOLOv10 Architecture

## IV. System Architecture

This chapter consists of the system design of the project with the technical architecture and various individual modules and their respective description used in this project. The primary focus of the project is to make street quick object finding by using advanced technologies for detecting objects and measuring their distance.

The system combines the YOLOv10 model for real-time object detection with a depth estimation model to help users understand their surroundings. Together, these components provide quick updates about the location and distance of possible obstacles and important spots nearby.

### *A. Video Feed acquisation*

The architecture of the proposed *Depth Analyser for Visually Impaired* system follows a modular design divided into three major phases: input, processing, and output. The system begins with the input phase, where a live video feed is captured from the environment using a webcam.

The frontend interface, built with React.js, accesses the device camera using the HTML5 Video API or WebRTC, enabling continuous and seamless frame capture from the user's environment. This live video stream is rendered directly within the browser, providing immediate visual feedback (where applicable) and ensuring that frames are captured with minimal latency. React's component-based architecture allows for modularity and responsiveness, making it well-suited for building interactive, real-time user interfaces. As video frames are captured, they are periodically extracted from the stream using JavaScript and encoded into image data formats such as base64 or binary blobs suitable for transmission.

Frontend is also responsible for rendering the received output, managing the timing of audio feedback, and optionally displaying bounding boxes or textual annotations for partially sighted users or for debugging during development.


Figure 4.1 Sample Camera input

### *B. BackEnd Processing*

In the processing phase, the backend—developed using Python Flask—receives the video frames and performs object detection using the YOLOv10 (You Only Look Once version 10) deep learning model.

Implemented using PyTorch, YOLOv10 provides fast and accurate detection of multiple objects in real time.

Once objects are identified, their spatial location within the frame is analyzed to categorize them as being on the left, center, or right relative to the user's viewpoint. The position detection of the object is done by the following algorithm.

**Input:** Video Frame with Bounding Box of Detected Object
**Output:** Direction of Object (Left, Right, or Straight)
**Algorithm:**
**Preprocessing**
Resize the frame to 416 × 416 pixels
Apply color adjustments (e.g., normalization)
**Region Segmentation**
Divide the frame into three vertical regions: Left, Straight, and Right
• Divide width of frame into three equal parts
Identify the center of the bounding box coordinates obtained from 4.1
**Direction Assignment**
Check the bounding box center position:
• If in the left region, assign direction as "Left"
• If in the center region, assign direction as "Straight"
• If in the right region, assign direction as "Right"
Return direction of the object

Figure 4.2 Finding Region of Interest Detection algorithm

Simultaneously, the system estimates the distance of each object from the camera using depth estimation techniques, which involves monocular depth estimation ,based on object size and position.

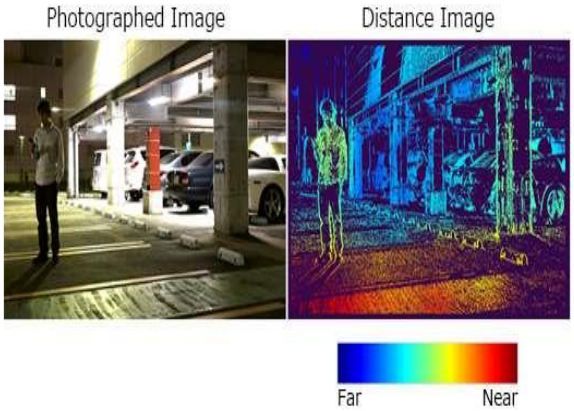


Figure 4.2 Distance from Depth Map

These three elements—object class, direction, and distance—are programmatically structured into a human-friendly sentence, such as

"The bottle is 1.5 meters to the right." The message is dynamically generated using string formatting techniques and localized units, ensuring that the language remains natural and easily interpretable by the user. The system is also designed to handle multiple detected objects within a single frame by generating separate sentences for each and prioritizing the ones closest to the user or most central in the frame.

Once the message is formulated, it is sent back to the frontend using **Socket.IO**, maintaining the same persistent WebSocket connection initially established for frame transmission.

The event-driven nature of Socket.IO ensures that the response is immediately pushed to the frontend without requiring the client to poll the server, thus minimizing delay and enabling near-instantaneous audio feedback.

On the frontend, the received passed directly to a text-to-speech module, which converts the message into audible speech. This real-time feedback loop forms the core user interaction layer, translating visual data into accessible, descriptive audio for the visually impaired user.

*C. Acquired Result*

In the final phase, the output module delivers real-time feedback to the user. The structured message is converted into speech using a text-to-speech engine , and played through headphones or a speaker.

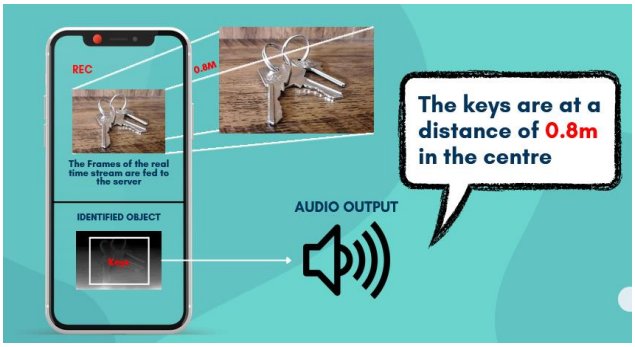


Figure 4.3 Expected Output

This audio feedback allows the visually impaired user to receive detailed, intuitive information about objects in their surroundings without relying on sight. Additionally, for testing or debugging purposes, the frontend interface also displays visual indicators such as bounding boxes, labels, and distances.


The complete system ensures real-time detection, efficient processing, and clear feedback, ultimately providing a platform for visually impaired

Also improving the spatial awareness and mobility of visually impaired users in dynamic environments provides easy identification of their required tools and accessories.

*D. Final Architecture*

As outlined in the preceding sections, the project aims to assist visually impaired individuals by translating real-time visual data into meaningful audio feedback. The system architecture is designed to efficiently handle this objective through a structured flow involving input acquisition, processing, and output delivery. The architecture leverages modern web technologies, real-time communication protocols and Deep learning based Image detection. The system Architecture is as follows

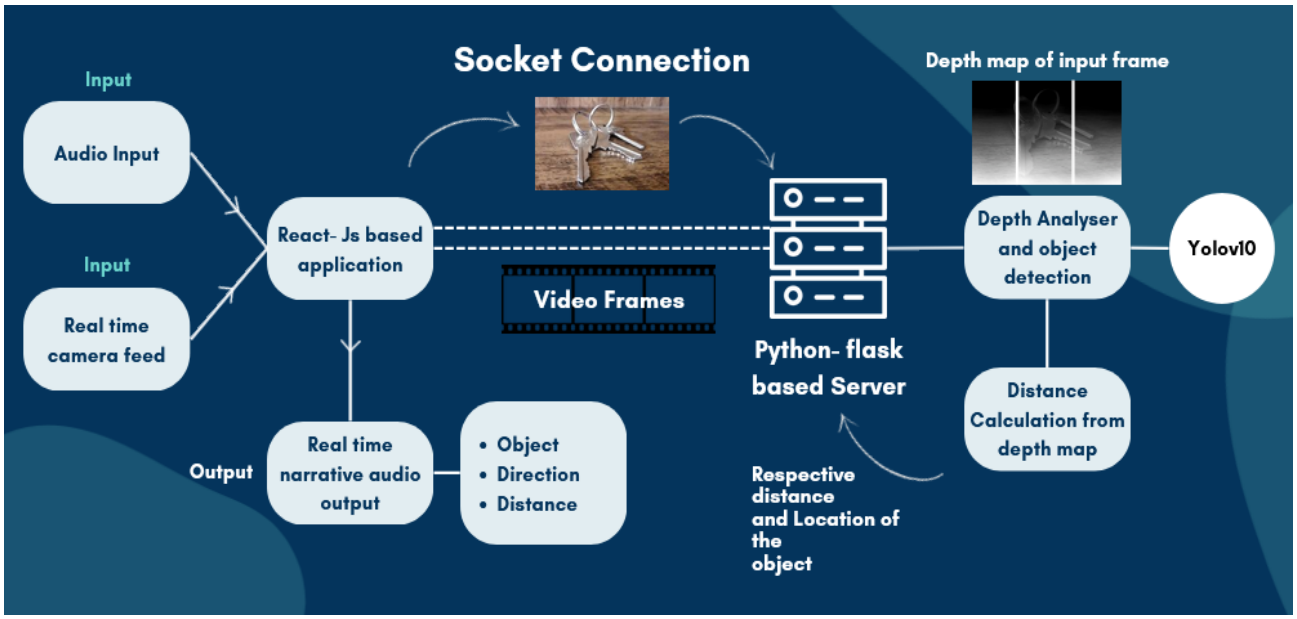


Figure 4.4 System Architecture

The system architecture of this project integrates real-time computer vision, audio output, and web technologies to create a supportive assistive tool for visually impaired users. It follows a modular and efficient design flow that encompasses three major stages: input acquisition, intelligent processing, and meaningful output delivery.

In the **input stage**, the system captures real-time video feed from the device's camera and optionally takes audio commands from the user. These inputs are processed through a React.js-based frontend, which handles the user interface and establishes a persistent socket connection with the backend server using Socket.IO. The captured video frames are transmitted frame-by-frame to the backend, enabling continuous and real-time communication.

The **processing stage** is powered by a Python Flask server. Upon receiving each frame, the backend performs two major tasks: object detection and depth estimation. The YOLOv10 model is used to detect and classify objects within the input frame, offering high-speed and accurate recognition. Alongside, a depth analyzer generates a depth map of the frame, from which the system calculates the relative distance and direction of each detected object. The results from both processes are combined to extract contextual information, including what the object is, how far it is from the user, and in which direction it lies (left, right, center, etc.).
In the **output stage**, the processed data is transformed into a clear, human-understandable audio message, such as "The chair is 2.3 meters to your right." This message is sent back to the frontend via Socket.IO, where it is played as real-time narrative feedback. This allows the user to receive continuous and accurate situational awareness without relying on visual cues.

Overall, the system architecture ensures low-latency communication, high accuracy in object detection, and seamless user interaction through real-time audio guidance. By efficiently combining computer vision with audio output and real-time communication technologies, the architecture effectively bridges the sensory gap for visually impaired individuals, enabling them to navigate their environment with greater confidence and independence. The modularity of this system also ensures easy scalability and future enhancements, such as integrating obstacle avoidance, gesture recognition, or wearable support.

## V. Result and Analysis

This chapter presents the outcomes of our object detection system, highlighting key performance metrics, model accuracy, and screenshots of detected objects within real-time video frames. The evaluation focuses on assessing the effectiveness of the YOLOv10 model integrated with the depth analyzer in terms of detection speed, precision, and distance estimation. By processing continuous frames from a live video feed, the system was able to identify and locate various objects in diverse indoor and outdoor environments with consistent accuracy.

We provide quantitative results including the mean Average Precision (mAP), frame processing time, and latency between input capture and audio output delivery. These metrics are crucial in understanding the real-time capabilities of the system.

### *A. Object Detection Accuracy*

The primary model used for object detection, YOLOv10[4], was trained on the COCO dataset to recognize various objects essential for visually impaired navigation, such as street lights, vehicles, and obstacles. YOLOv10 has been extensively tested on standard benchmarks like COCO, demonstrating superior performance and efficiency.

The model achieves state-of-the-art results across different variants, showcasing significant improvements in both latency and accuracy compared to previous versions and other contemporary detectors.

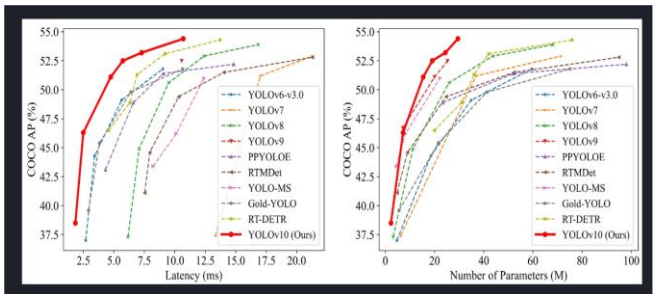


Figure 5.1: YOLOV10 when compared to other models

## B. *Depth Estimation for Spatial Awareness*

In addition to object detection, the depth estimation model, Depth Anything by Lihe Yang [3], provided highly accurate measurements of object distance, significantly enhancing spatial awareness for the user. This model outperformed the previously used MiDaS model, achieving superior results in downstream fine-tuning performance, as indicated by the following metrics:

| Method | NYUv2 | | KITTI | | Cityscapes | ADE20K |
|---|---|---|---|---|---|---|
| | AbsRel | $\delta_1$ | AbsRel | $\delta_1$ | mIoU | mIoU |
| MiDaS | 0.077 | 0.951 | 0.054 | 0.971 | 82.1 | 52.4 |
| DepthAnything | **0.056** | **0.984** | **0.046** | **0.982** | **84.8** | **59.4** |

Figure 5.2: Depth Estimation Model Accuracy Comparison

Absolute Relative Error (AbsRel): Depth Anything exhibits a lower AbsRel compared to MiDaS across datasets like NYUv2 and KITTI. This lower value indicates a reduced error in estimating object distances, making it highly reliable for real-time applications.

δ 1 Accuracy: The higher δ 1 accuracy metric in Depth Anything highlights its improved capability to accurately estimate distances within a permissible error margin, especially in challenging environments.

Mean Intersection over Union (mIoU): Depth Anything demonstrates enhanced performance in mIoU across datasets such as Cityscapes and ADE20K, validating its generalization ability and adaptability to various scene structures.

These metrics, as shown in Figure 5.2, reflect the robustness and reliability of Depth Anything over MiDaS, showcasing its effectiveness in supporting visually impaired navigation by providing precise spatial awareness.

## C. *Real-Time Object Detection and Feedback*

With the integration of real-time feedback mechanisms, the system provides audio alerts based on the detected object's proximity. This functionality is achieved through fast video frame processing and timely audio cue generation, which allows the user to be continuously updated on nearby objects.

Figure 5.3: Image captured with camera

---

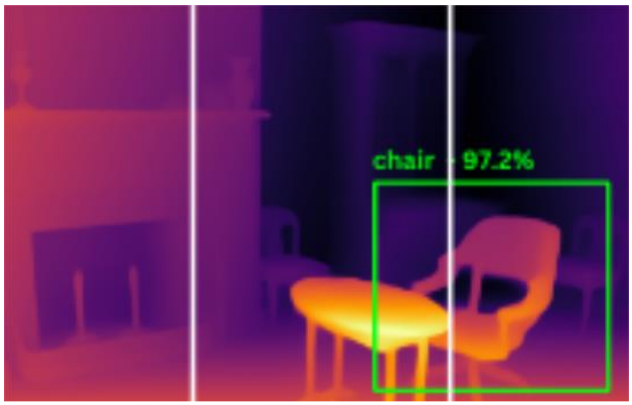


Figure 5.4: Processed Image showing Detected Objects with Bounding Box

Figure 5.4 showcases sample frames with bounding boxes around detected objects, illustrating the model’s ability to accurately outline objects within the user’s environment.

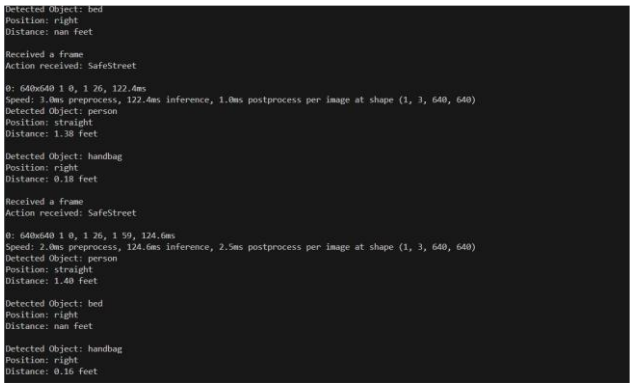


Figure 5.5: Sample screenshot of Terminal which displays Detected Objects

Figure 5.5 showcases the terminal output generated at the backend, representing the processed information derived from real-time object detection and depth analysis. When the model detects an object—such as a person—within the camera frame, it simultaneously calculates both the spatial direction and distance of the object relative to the user's viewpoint. For instance, in this particular scenario, the model identifies a "Person" located towards the right, approximately 1.40 feet straight ahead.

Once these spatial parameters are determined, the backend system dynamically constructs a meaningful and concise narrative sentence, such as: **"A person detected 1.40 feet straight."** This sentence encapsulates the object type, estimated distance, and direction, making it easier for the end-user to understand the surrounding environment. The generated message is then transmitted back to the frontend via a socket connection, where it is vocalized using a text-to-speech module. This real-time feedback ensures seamless user interaction and allows for efficient navigation support, particularly beneficial in assistive technology scenarios for the visually impaired.

The backend terminal output not only displays the detected object’s attributes but also serves as a crucial debugging and verification tool for developers, offering transparency into how the model interprets and transforms visual data into descriptive messages.

## VI. Conclusion and Future Work

In this project, we built an object detection system designed to assist visually impaired people by providing audio feedback about objects detected around them and their distances. Using the YOLOv10 model, the system can quickly and accurately recognize various objects, giving users valuable information about their surroundings.

We also added depth estimation to determine how far each detected object is, which helps users understand the layout and spacing of nearby obstacles. The system performed well in real-time tests, showing good accuracy based on precision-recall curves and average precision scores. It sends audio alerts about the location and distance of objects, making it a valuable tool for navigation and safety.

The feedback features ensure users receive timely updates, creating a more interactive and helpful experience. This project demonstrates how combining computer vision with audio feedback can help visually impaired people gain more independence and confidence in moving around.

The current system effectively demonstrates object detection and distance estimation, but there are several ways to make it even better in the future.

- Model Optimization: Future versions could focus on making the YOLOv10 model faster, so it works smoothly in real-world situations.
- Techniques like model pruning or quantization could help improve speed without losing accuracy.
- Broader Object Recognition: Training the model on a larger dataset could enable it to recognize more everyday objects, such as different vehicle types and common obstacles, making it more helpful in varied environments.
- Multi-Modal Feedback: Adding haptic feedback (like vibrations) or smartphone notifications could give users more ways to receive information, increasing their awareness and safety.
- User Testing: Testing the system with visually impaired users would provide valuable feedback on its effectiveness in real-life situations, helping to make continuous improvements based on real user needs.
- Integration with Navigation Apps: Future updates could connect this detection system with existing navigation tools, offering users guidance in both detecting objects and moving safely in their environments.

## VII. REFERENCES


[1] Jo˜ao Guerreiro, Daisuke Sato, Dragan Ahmetovic, Eshed Ohn-Bar, Kris M. Kitani, and Chieko Asakawa. Virtual navigation for blind people:
Transferring route knowledge to the real-world. International Journal of Human-Computer Studies, 135:102369, 2020.

[2] Yasir M Mustafah, Rahizall Noor, Hasbullah Hasbi, and Amelia Wong Azma. Stereo vision images processing for real-time object distance and size measurements. In 2012 International Conference on Computer and
Communication Engineering (ICCCE), pages 659 – 663, 2012.

[3] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In CVPR, 2024.

[4] Lihao Liu et al. AoWang, Hui Chen. Yolov10: Real-time end-to-end object detection. arXiv preprint arXiv:2405.14458, 2024.

[5] Ren´e Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction, 2021.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.

[7] Xiaodong Yang, Zhuang Ma, Zhiyu Ji, and Zhe Ren. Gedepth: Ground embedding for monocular depth estimation, 2023.

[8] Reiner Birkl, Diana Wofk, and Matthias M¨uller. Midas v3.1 – a model zoo for robust monocular relative depth estimation, 2023.

[9] Min Chen, Hui Lin, Deer Liu, Hongping Zhang, and Songshan Yue. An object-oriented data model built for blind navigation in outdoor space.
Applied Geography, 60:84 – 94, 2015.