REAL TIME OBSTACLE SENSING WITH PROXIMITY ALERTS

A PROJECT REPORT

Submitted by

JEYABALAN P 2022115064

GADIRAJU DINESH 2022115094

MYTREYAN JP 2022115102

submitted to the faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment

for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

NOV 2024

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"Real Time Obstacle Sensing with Proximity Alerts"** is the bonafide work of **JEYABALAN P (2022115064) , GADIRAJU DINESH (2022115094) , MYTREYAN JP (2022115102)** who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:CHENNAI
DATE:

Dr. K. VIDYA
ASSOCIATE PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Navigating public spaces independently presents significant challenges for visually impaired individuals, highlighting the need for assistive technologies that offer real-time spatial awareness. InnerVoice is a Progressive Web App (PWA) designed to enhance mobility and independence for visually impaired users by providing immediate auditory feedback on their surroundings. Upon launching the app, a voice assistant guides users through its features, including the ”Safe Street” mode, which activates a live camera feed to detect nearby objects and provides information on their distance and direction relative to the user. This information is communicated through brief voice alerts, enabling users to navigate complex environments safely. By leveraging advanced object detection (YOLOv10x) and depth estimation models, InnerVoice delivers precise and timely guidance on obstacles, helping users make immediate navigation adjustments.

The application utilizes WebSocket communication to establish a continuous, low-latency connection between the frontend and backend, ensuring smooth, real-time feedback. Frames captured from the user's device are processed by a Flask backend, which identifies objects, estimates distances, and calculates directions (left, straight, or right) based on bounding box positions. The final detection data is sent back to the frontend, where the voice assistant interprets and conveys essential navigation details. InnerVoice thus empowers visually impaired individuals to traverse their surroundings confidently, relying only on a mobile device and headphones to experience improved spatial awareness and autonomy.

ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to **Dr. K. VIDYA,** Associate Professor, Project Guide, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for her constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I am grateful to **Dr. S. SWAMYNATHAN,** Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr. M. VIJAYALAKSHMI,** Professor, **Mr. H. RIASUDHEEN,** Teaching Fellow Department of Information Science and Technology,College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

JEYABALAN P (2022115064)
GADIRAJU DINESH (2022115094)
MYTREYAN JP (2022115102)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
COCO	Common Objects in Context (dataset)
CV	Computer Vision
DA	Depth Anything
FOV	Field of View
ML	Machine Learning
POI	Point of Interest
PVI	People with Visual Impairments
PWA	Progressive Web App
UI	User Interface
UX	User Experience
YOLO	You Only Look Once (Object Detection Model)

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

In an increasingly urbanized world, navigating public spaces presents significant challenges for individuals with visual impairments. Traditional mobility aids, such as canes and guide dogs, while helpful, do not provide comprehensive situational awareness in complex environments. The advent of technology has opened new avenues for enhancing the independence and safety of visually impaired individuals. Mobile applications equipped with artificial intelligence and computer vision capabilities can significantly improve navigation by providing real-time feedback and obstacle detection. InnerVoice seeks to leverage these advancements to create a user-friendly platform that empowers visually impaired users to navigate their surroundings with confidence and ease.

1.2 OBJECTIVES

The primary objective of the InnerVoice project is to develop a mobile application that enhances the mobility and safety of visually impaired individuals by utilizing advanced object detection and auditory feedback mechanisms. The specific goals of the project are as follows:

- **Real-time Environmental Analysis:** To create a robust deep learning model capable of analyzing the surrounding environment through the device's camera, identifying obstacles, and providing timely auditory alerts to users about potential hazards.

- **User-Friendly Interaction:** To design an intuitive user interface that allows visually impaired users to easily navigate through the application, enabling them to activate voice assistance and access the ”Safe Street” feature with minimal effort.
- **Safety Enhancement:** To implement a reliable system that accurately detects and communicates the distance and direction of objects, ensuring users receive clear guidance, such as ”Watch out! A chair is detected 2 feet on your right,” thus enhancing their awareness of the environment.
- **Empowerment Through Independence:** To empower visually impaired individuals by fostering a sense of confidence and autonomy in navigating public spaces, ultimately improving their quality of life and encouraging greater engagement with their surroundings.

1.3 PROBLEM STATEMENT

Visually impaired individuals face challenges navigating public spaces independently, as traditional aids like canes and guide dogs lack real-time, comprehensive awareness of surroundings. Existing solutions often fail to provide accurate, intuitive guidance, leading to increased dependency and safety risks. InnerVoice aims to address this by developing a mobile application that offers real-time obstacle detection and auditory feedback, including distance and direction information. By enhancing spatial awareness, InnerVoice empowers users to navigate confidently, improving their safety, independence, and quality of life.

1.4 SOLUTION OVERVIEW

Our solution leverages modern web technologies to deliver a user-friendly and efficient experience for visually impaired users. The application is built as a Progressive Web App (PWA) using React.js, enabling cross-platform accessibility and ease of use. For object detection, we employ the YOLOv10 model trained on the COCO dataset, allowing the app to accurately identify common objects in the user's surroundings. To estimate distance, we integrate LiheYoung's DepthAnything model to generate depth maps, which are then converted into distance measurements in feet. Communication between the frontend and backend is streamlined using WebSockets, ensuring real-time data exchange for a seamless user experience.

In addition to detection and distance, our solution incorporates directional awareness. Using OpenCV, we implemented a custom logic to calculate the direction of each detected object relative to the user, allowing for precise guidance such as "2 feet to your right." By combining object detection, depth estimation, and directional feedback, the application provides comprehensive spatial awareness, enhancing safety and independence for visually impaired individuals.

1.5 ORGANIZATION OF THE REPORT

Chapter 1 introduces the project by providing background information, defining the main objectives, and presenting the problem statement along with an overview of the solution. ***Chapter 2*** reviews existing work in the fields of object detection, depth estimation, and navigation aids for visually impaired individuals, highlighting relevant approaches and technologies. ***Chapter 3*** outlines the technical architecture of the InnerVoice application, detailing the core components, including the Progressive Web App structure,

object detection model, depth estimation model, and real-time communication setup. ***Chapter 4*** describes the implementation process of each module in depth, explaining the logic behind the custom direction detection and the integration of YOLOv10, DepthAnything, and WebSockets to achieve real-time responsiveness. ***Chapter 5*** presents the results of testing, demonstrating the app's performance in terms of object detection accuracy, distance estimation, and reliability of directional feedback.Finally, ***Chapter 6*** discusses future improvements and potential expansions of the project, concluding with insights on how this solution contributes to enhancing the independence and safety of visually impaired users. References to all consulted works are provided at the end of the report.

The above mentioned six chapters are followed by the references which lists all the reference documents including journals and articles, used during various phases of the project.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

In developing assistive technologies for visually impaired individuals, recent advancements in computer vision, depth estimation, and real-time communication have opened new possibilities for enhancing spatial awareness and independent mobility. A thorough understanding of existing research in these areas is essential to building a robust and effective solution. This literature survey reviews key technologies and methods relevant to the InnerVoice project, examining the state-of-the-art in object detection, depth estimation, and real-time direction calculation as applied to assistive navigation tools. Additionally, it explores the design considerations for human-computer interaction tailored to visually impaired users, focusing on accessibility and ease of use. By analyzing prior work, existing applications, and various challenges, this review aims to lay the groundwork for the technical choices made in InnerVoice, highlighting both the limitations of current solutions and opportunities for innovation.

2.2 VIRTUAL NAVIGATION AID FOR BLIND

One approach to assist visually impaired individuals in navigating unfamiliar environments is through virtual navigation systems that allow users to learn routes in advance. Guerreiro et al.[1] (2020) developed a virtual navigation app that enables users to acquire route knowledge prior to physically visiting an environment. This method provides a simulation of turn-by-turn instructions along with descriptions of key landmarks and points of interest

(POIs), allowing users to form a mental map of the route. In their study, users practiced navigating specific paths virtually and then performed real-world navigation tasks, both unassisted and with the support of a guidance tool (NavCog). Results indicated that users could leverage their prior route knowledge to navigate independently in familiar environments and that this knowledge also provided limited assistance when using in-situ guidance systems like NavCog.

Unlike InnerVoice, which uses real-time object detection and distance estimation for immediate feedback in live environments, the virtual navigation approach in Guerreiro et al.'s study relies on pre-learned route knowledge. This allows users to understand spatial layouts in advance but does not adapt to dynamic obstacles or changing conditions in real time. The pre-navigation approach is particularly suited to controlled indoor environments where users can rely on memorized paths, whereas InnerVoice focuses on providing real-time, responsive feedback in unpredictable public settings. This distinction highlights the different methodologies in assistive navigation technology, with InnerVoice aiming to enhance spatial awareness and immediate decision-making without the need for pre-navigation.

2.3 OBJECT DISTANCE WITH STEREO VISION

Mustafah et al. (2012)[2] explored the use of stereo vision for real-time object distance and size measurements, demonstrating how stereo cameras can effectively process images to determine spatial information. Their research highlighted the capability of stereo vision systems to detect and recognize objects while accurately measuring their distance and size in real-time environments. This approach is particularly relevant for applications requiring precise spatial awareness, such as assistive technologies for the visually impaired.

However, stereo vision systems come with several limitations. Firstly, the cost of implementing two cameras can be prohibitive, especially for personal devices or small-scale applications. Moreover, accurate measurements depend on knowing the width of the object being measured. This is crucial because the formula for calculating distance (D) from stereo images involves the baseline distance (B), the focal length of the camera (f), and the disparity (d) between the two camera images:

$$D = \frac{d}{B \cdot f} \tag{2.1}$$

2.3.1 LIMITATIONS

- Need to know Width of an object
- Need to know angle of tilt as it heavily impacts the distance.
- This method is not cost effective as it requires two cameras and not applicable in real world where the object constantly changes with time.

2.4 DEPTH ESTIMATION

In contrast to stereo vision systems, Lihe Young's [3] 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 2.1: Depth Anything

2.5 OBSTACLE DETECTION

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.[4]

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.

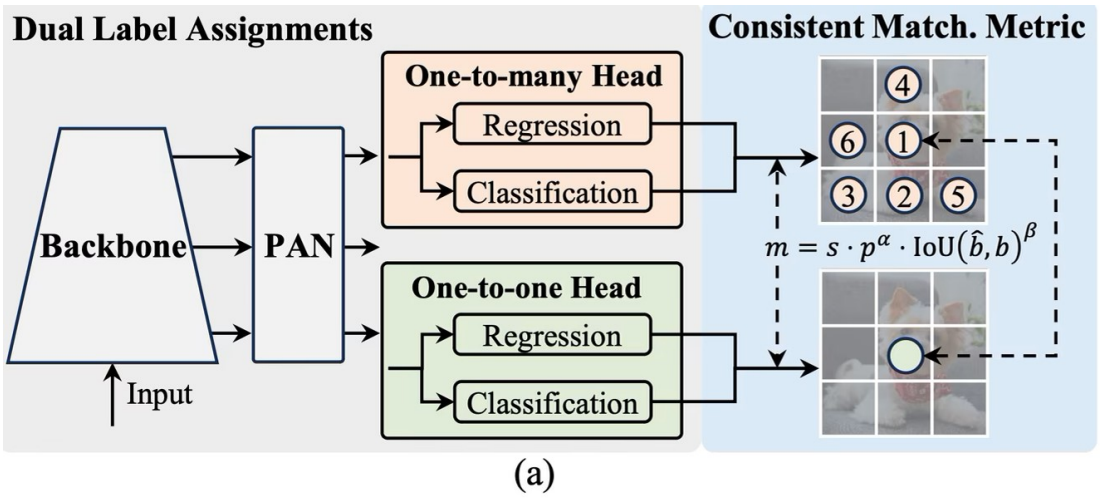

Figure 2.2: Yolov10 architecture

2.6 MOTIVATION OF THE PROPOSED WORK

Navigating urban environments presents unique challenges for visually impaired individuals, as traditional mobility aids like canes and guide dogs provide limited real-time information on surroundings. Despite advancements in navigation technology, there is still a gap in assistive applications that can effectively convey object detection, distance, and direction in a simple and accessible manner. This project proposes the InnerVoice system to address these limitations. By leveraging state-of-the-art deep learning models such as YOLO for object detection and DepthAnything for depth estimation, InnerVoice provides visually impaired users with real-time auditory feedback, empowering them with enhanced spatial awareness.

The motivation behind InnerVoice is to foster independence and confidence in visually impaired users, allowing them to navigate complex environments safely and efficiently. The InnerVoice system offers precise, real-time guidance by describing objects in the user's path, including their distance and direction, enabling users to make immediate adjustments. Future improvements could involve the integration of enhanced voice commands and

adaptive sound cues, further improving the experience. By converting complex spatial data into actionable auditory information, InnerVoice aims to improve the quality of life for visually impaired individuals, facilitating safer and more independent navigation in everyday settings.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 INTRODUCTION

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 navigation safer 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.

3.2 ARCHITECTURE OF PROPOSED WORK : INNER VOICE

Figure 3.1 illustrates the architecture of the SafeStreet system, which integrates two key components to enhance safety in street navigation: real-time object detection using YOLOv10 and a depth estimation model. The system begins by processing video frames from the user's camera with the YOLOv10 model, which identifies various objects, such as vehicles and pedestrians. This step is essential for recognizing potential obstacles and points of interest that may impact the user's navigation.

Simultaneously, in the backend, the depth estimation model analyzes the video frames to create a depth map, indicating the distance of each detected

object from the user. This model enhances spatial awareness by providing crucial distance information alongside object detection. Each object's position is determined by calculating the center x-coordinate of its bounding box, allowing the system to categorize objects as left, center, or right relative to the user's view.

To facilitate real-time communication between the frontend and backend, the SafeStreet system employs sockets and WebSockets to transmit video frames efficiently. This ensures that the processing of video data occurs seamlessly and with minimal latency. The SafeStreet system's final output includes real-time notifications informing users of detected objects, their distances, and the position of the object—whether it is on the left, right, or straight ahead—significantly improving awareness and safety while navigating the streets.

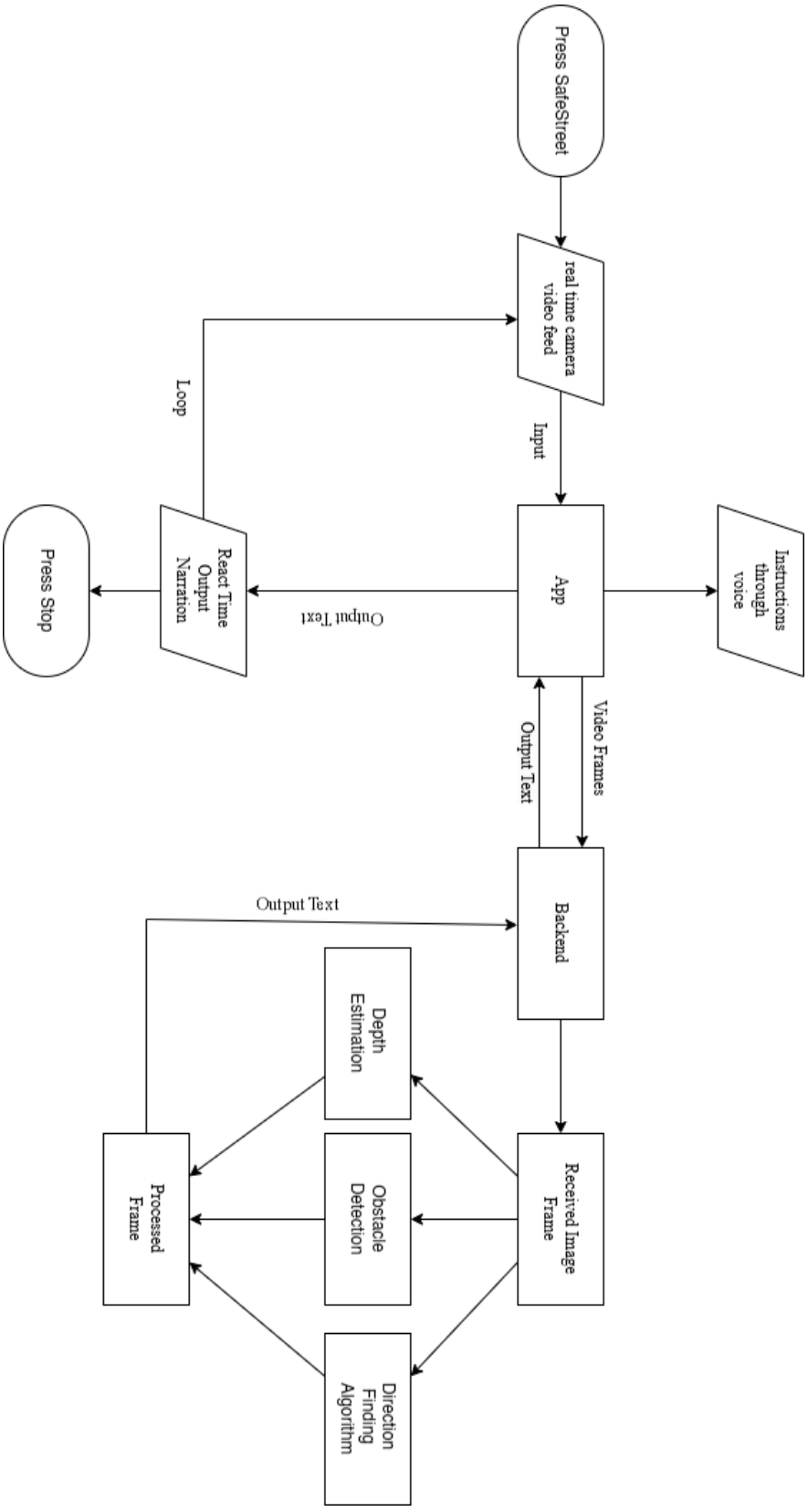

Figure 3.1: Architecture of SafeStreet

3.3 OBJECT DEPTH MODEL

The Depth Estimation Model, commonly known as ”Depth Anything,” is an advanced system designed to measure how far away objects are in a visual frame from the camera. This technology is essential for applications that need spatial awareness, such as self-driving cars, augmented reality. By using a mix of machine learning methods and image processing techniques, the model creates a depth map that shows the distance of each pixel from the camera. This ability helps users understand their surroundings better, making it easier to interact safely with different objects in real-time situations.

Figure 3.2illustrates the architecture of the Depth Estimation Model, which starts with the **Input Image**, where visual data from the surroundings is collected. Next, the image goes through **Feature Extraction**, using advanced techniques to detect and highlight essential features necessary for depth calculation. After that, the **Depth Estimation Network** employs these extracted features to create a **Depth Map**, which visually indicates how far different objects are from the camera. The model also includes a **Post-processing** stage that enhances the depth map for better accuracy and usability. This refinement involves calculating the **Mean Depth Values** from the depth map, inverting these values to convert them into usable distances, and then multiplying them by a **Scaling Factor** to adjust the measurements appropriately. This thorough process ensures that users receive precise, real-time depth information, greatly improving their spatial awareness and decision-making skills in changing environments.

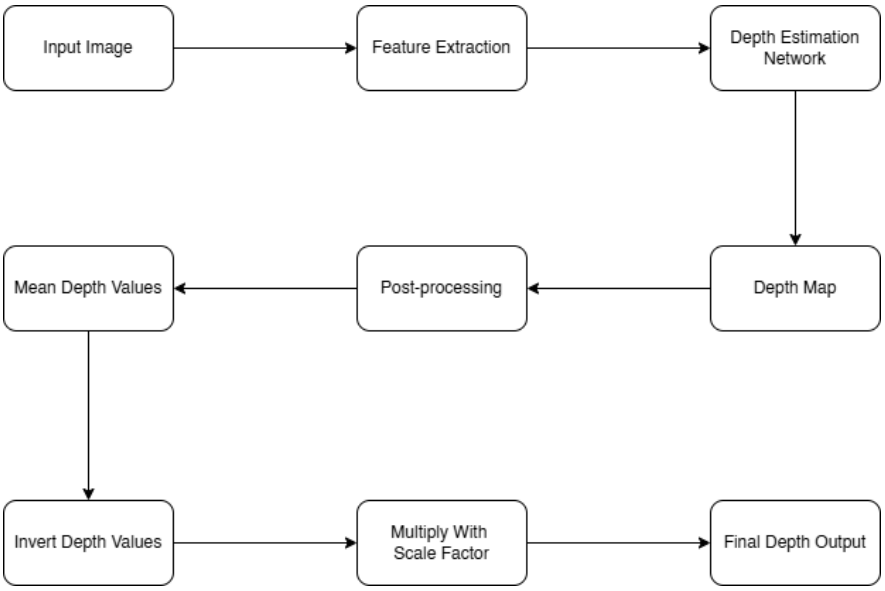

Figure 3.2: Architecture of Depth Estimation Model

3.4 OBJECT DETECTION

Object detection is an important task in computer vision that focuses on finding and identifying objects in images or video frames. This process not only identifies what objects are present but also shows where they are located using bounding boxes. Object detection is used in many fields, including security cameras, self-driving cars, robotics, and augmented reality. One well-known method for real-time object detection is the YOLO (You Only Look Once) model. YOLO stands out because it analyzes entire images at once instead of breaking them down into smaller parts. This allows it to detect objects more quickly and accurately, as it can look at multiple objects in a single image simultaneously. By training on large sets of labeled images, YOLO learns to recognize different types of objects and their features, making it effective for detecting and classifying objects in real-time situations

Figure 3.3 outlines the architecture of the YOLO model for object detection. The process begins with input images, which are passed through a

Feature Extraction stage that utilizes convolutional layers to capture essential visual features. Following this, the model divides the image into a grid and assigns bounding boxes and class probabilities for each grid cell. The output of this stage is a set of bounding boxes with associated confidence scores, indicating how likely it is that an object is present within each box. The YOLO architecture employs a single neural network to make predictions directly from the entire image, resulting in reduced computation time and fewer false positives.

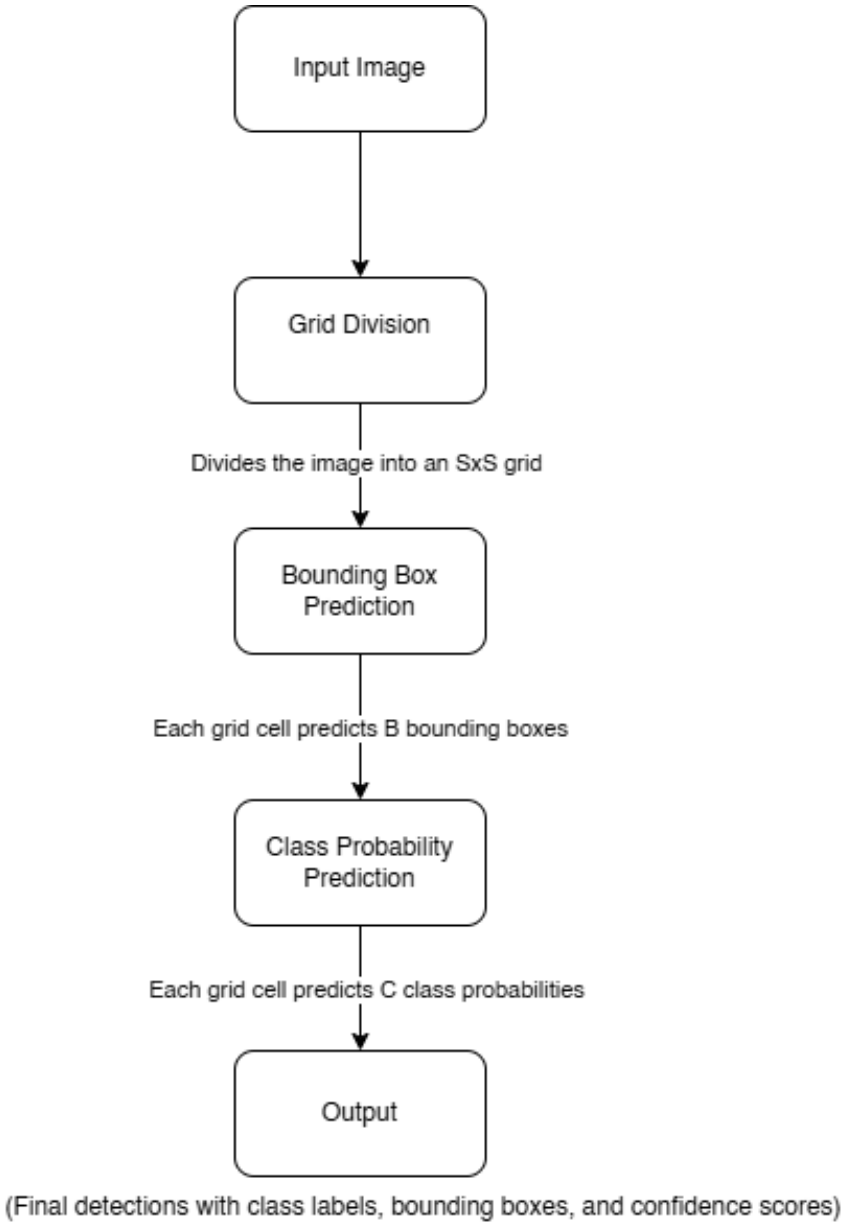

Figure 3.3: Architecture of YOLO

3.5 TECHNOLOGICAL STACK

3.5.1 ReactJS for Progressive Web App

ReactJS is used to create a responsive and dynamic PWA interface for SafeStreet. It allows for efficient user interaction and provides a smooth, app-like experience across devices without the need for installation. The component-based structure of ReactJS makes the app scalable and easy to maintain.

3.5.2 Socket.IO

Socket.IO is used to enable real-time, bidirectional communication between the frontend and backend. It ensures the seamless transmission of video frames captured by the user's camera to the backend, where they are analyzed for object detection and depth estimation. This low-latency data exchange is crucial for maintaining real-time updates in navigation guidance.

3.5.3 Flask

Flask serves as the lightweight yet powerful backend framework, responsible for handling incoming video data, processing it through YOLO and the depth estimation model, and sending analyzed results back to the frontend. Its simplicity makes Flask ideal for handling the project's RESTful requests and Socket.IO communications.

3.5.4 Vision Transformers – Hugging Face

Depth-anything was achieved by utilizing hugging face transformers which was built with DPT architecture. This model uses the Vision Transformers as backbone as mentioned in [5]

3.5.5 Ultralytics-YOLOv10

YOLOv10 by Ultralytics is the core model used for real-time object detection. Known for its accuracy and speed, YOLOv10 identifies objects. This model is essential for providing users with timely feedback on obstacles and potential hazards in their path.

3.5.6 Python Libraries

NumPy and OpenCV are fundamental libraries supporting image and data processing within the system. NumPy is used for handling array operations and numerical calculations, while OpenCV is pivotal for video frame manipulation and image preprocessing, enabling accurate depth estimation and object recognition in real-time.

CHAPTER 4

IMPLEMENTATION

This chapter focuses on the implementation details of the proposed work : Inner Voice - an Progressive Web App (PWA) .

4.1 ENVIRONMENT

InnerVoice includes a Voice Assistant which reads out the About section in 4.2 on opening the app. The assistant start off with explaining basic instructions like ,

> Welcome .. This is your innervoice... Click on the bottom
> of your screen to enter Safe Street Tab.

fig. 4.1, shows Safe Street tab . The Safe street tab is designed to guide the users as they proceed walking down the the street . This tab opens on tapping the bottom of the screen . The voice assistant now says ,

> Safe street Tab.. Tap your screen before start walking..
> Tap again to exit..

Once a person taps on the Safe street tab the users camera would be accessed and the realtime frames would be put into a full-duplex link established by the socket IO. The frames are finally received at the backend by a Flask server. The server then interprets the frame and determines the Object , its

distance as well the object's direction from the user (right / left / straight) . The final message will be sent back to the frontend react PWA , where the voice assistant narrates the message received . For example :

Watch out ,detected a door 2ft on your right!

The Visually impaired must have headphones / earbuds on to get real time voice commands . Thus by creating a PWA with react and flask connected via a socket , we were able to interpret vedio frames at real time with a delay of 1s between each frames . The visually impaired could solely rely on just their mobile phones to walk around independently . Now lets move on to the algorithms used in implementing Safe Street Mode.

4.2 OBJECT DETECTION ALGORITHM

YOLOv10 from ultralytics[4] provides the highest accuracy for object detection as for of 2024. We have discussed the models working in chapter 2 . We utilized Hugging face transformers pipeline to use this model in our backend. The algorithm for detecting objects is as follows 4.1

4.3 DEPTH ESTIMATION ALGORITHM

Once the object of interest is found using the 4.1 ,the frames are again sent to a Depth estimation algorithm , where the bounded area obtained from the 4.1 is stripped off and the corresponding depth values are taken mean . The final value is then inverted and a scaling factor is multiplied to the inverted value to convert the distance into real world distance in feet. The algorithm is as follows4.2

Algorithm 4.1 YOLOV10x.pt Object Detection Algorithm

Input: Video Frames
Output: Detected Objects and Class Names
Algorithm:
Preprocessing
Resize each frame to 416×416 pixels
Apply color adjustments (e.g., normalization)
Object Detection using YOLOV10x.pt
Feed preprocessed frames to YOLOV10x.pt model
Perform object detection to identify bounding boxes and class probabilities
Postprocessing
Extract detected class names based on COCO dataset classes
Filter detections based on confidence threshold
Return final object classes and bounding boxes

Algorithm 4.2 Depth Estimation for Distance Calculation

Input: Video Frames with Bounding Box of Object of Interest
Output: Real-World Distance (in feet)
Algorithm:
Preprocessing
Resize each frame to 416×416 pixels
Apply color adjustments (e.g., normalization)
Depth Estimation
Crop the region within the bounding box obtained from 4.1
Feed the cropped region to the depth estimation model
Obtain depth values for each pixel within the bounding box
Distance Calculation
Compute the mean of the depth values within the bounding box
Invert the mean depth value
Multiply the inverted value by a scaling factor to convert to real-world distance in feet
Return final distance measurement

Figure 4.1: Safe Street Tab

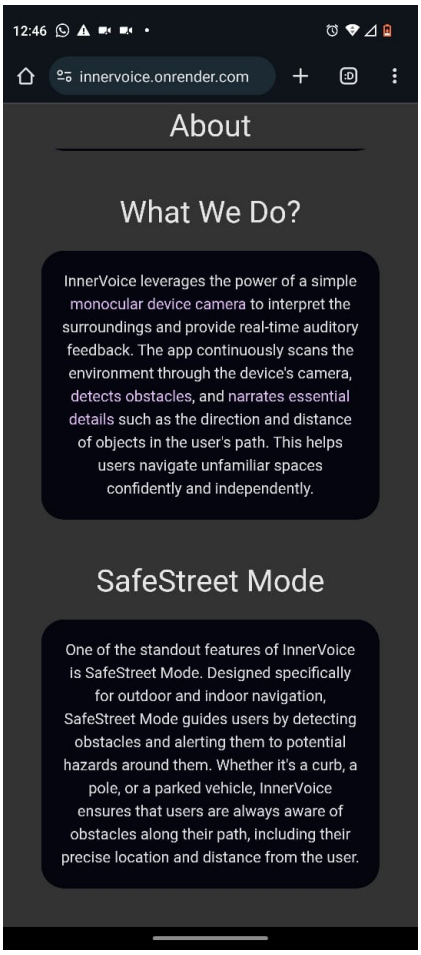

Figure 4.2: About Tab

4.4 DIRECTION FINDER ALGORITHM

The object if detected with 4.1 , then the corresponding bounding box is stripped off and sent to this algorithm where the image is splitted into three regions and based on the regions the objects bounding box leans the most is considered the region of interest example- left , right , straight. The algorithm utilizes open-cv, just have less impact on the time taken to process frames unlike other two algo which requires GPU powered backend servers. The algorithm is as follows :

Algorithm 4.3 Region of Interest Detection Using Bounding Box Position

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

4.5 COMPLETE SOCKET-IO COMMUNICATION

Once we detected object , estimated its distance , calculated distance , all that is left is to frame a final sentence for the voice assistant . The following Algorithm explains the complete socket-io communication process 4.4

Algorithm 4.4 WebSocket Communication for Safe Street Mode

Input: Video Frames
Output: Audio Feedback from Voice Assistant
1. Initialization and WebSocket Setup

User taps on Safe Street Tab.

Voice assistant says: ”Safe street Tab.. Tap your screen before start walking.. Tap again to exit.”

Establish a WebSocket connection between the frontend PWA and the backend Flask server.

Access the user’s camera, initiating real-time frame streaming to the backend.

2. Real-Time Frame Processing

For each video frame received by the backend:

2.1 Preprocess the frame (resize to 416×416, normalize).

2.2 Object Detection: Execute **Algorithm 4.1** to detect objects and bounding boxes.

2.3 Distance Calculation: Apply **Algorithm 4.2** to determine real-world distance in feet for detected objects.

2.4 Direction Determination: Use **Algorithm 4.3** to identify object direction (left, straight, or right).

3. Feedback Communication

Construct an audio message based on detected object, distance, and direction, such as:

”Watch out, detected a door 2 feet on your right.”

Send the message back through the WebSocket to the frontend.

Voice assistant narrates the message for the user.

4. Monitoring and Exit

Repeat steps 2 and 3 for each new frame (1-second delay between frames).

If the user taps the screen again, close the Safe Street Mode.

Voice assistant confirms Safe Street Mode exit and WebSocket connection closes.

End

CHAPTER 5

RESULTS AND ANALYSIS

5.0.1 INTRODUCTION

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.

5.0.2 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.

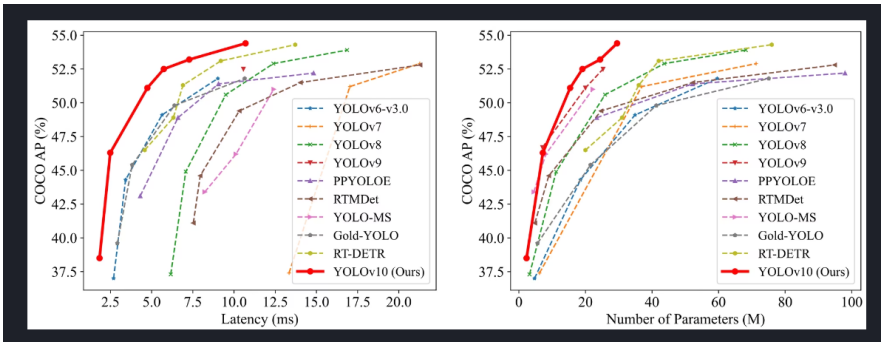

Figure 5.1: YOLOV10 when compared to other models

5.0.3 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	δ_1	AbsRel	δ_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.

5.0.4 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

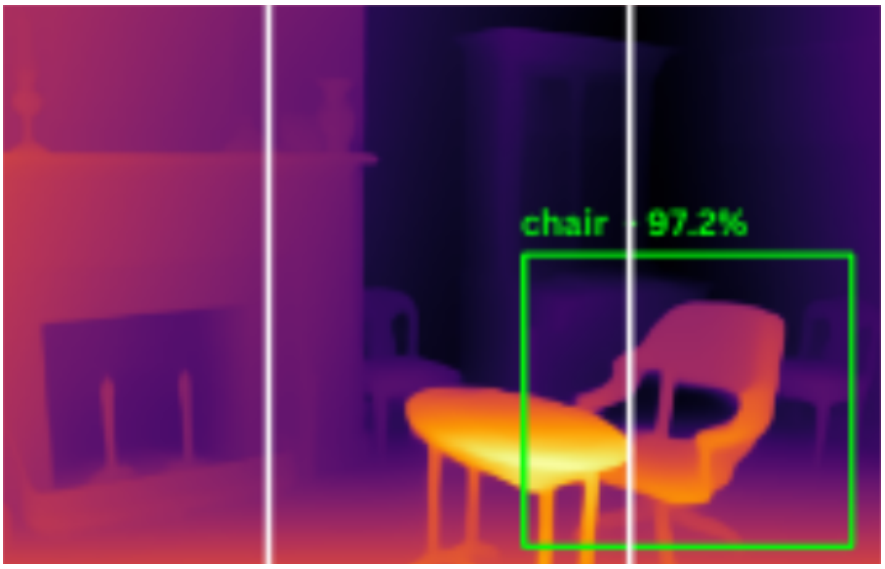

Figure 5.4: Processed Image showing Detected Objects with Bounding Box

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

```
Detected Object: bed
Position: right
Distance: nan feet

Received a frame
Action received: SafeStreet

0: 640x640 1 0, 1 26, 122.4ms
Speed: 3.0ms preprocess, 122.4ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)
Detected Object: person
Position: straight
Distance: 1.38 feet

Detected Object: handbag
Position: right
Distance: 0.18 feet

Received a frame
Action received: SafeStreet

0: 640x640 1 0, 1 26, 1 59, 124.6ms
Speed: 2.0ms preprocess, 124.6ms inference, 2.5ms postprocess per image at shape (1, 3, 640, 640)
Detected Object: person
Position: straight
Distance: 1.40 feet

Detected Object: bed
Position: right
Distance: nan feet

Detected Object: handbag
Position: right
Distance: 0.16 feet
```

Figure 5.5: Sample screenshot of Terminal which displays Detected Objects

Figure 5.5 showcases the terminal output generated at the backend. For example the model detects a Person to the Right , 1.40 feet straight ahead of them. Thus we would frame a sentence. ”A Person detected 1.40 feet straight ”

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

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.

6.2 FUTURE WORK

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.

By exploring these future upgrades, we aim to develop a powerful assistive tool that greatly enhances mobility and independence for visually impaired individuals.

REFERENCES

[1] João 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. Ao Wang, Hui Chen. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024.

[5] René 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 Müller. 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.