Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

Eighth International Conference on Futuristic Trends in Networks and Computing Technologies (FTNCT08) held at Ghaziabad, UP, India

PROXI-NAV: Proximity-Aware Navigation with Audio-Visual Sensing Intelligence

Aswinkumar Varathakumaran[a], Akshita Jawahar[a], Sreya Mynampati[a], Sajidha S A[a,*], Sumaiya Thaseen[b]

[a]*School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India*
[b]*School of Computing Science and Informatics, De Montfort University, Dubai Internet City, Dubai, UAE*

Abstract

The visually impaired encounter significant difficulties in spatial processing and movement in complicated and constantly changing places. Traditional methods like white canes and guide dogs offer little help since they provide only local or static information and no awareness of the situation around. On the contrary, this article introduces a new navigational help system that is intended to give the user consistent auditory guidance through vision and gyroscope-based processing and spatial feedback. The integrated system uses a multi-head YOLOv8 design with monocular depth estimation to estimate spatial awareness simultaneously, providing verbiage for identified patterns, both obstacles and walkable surfaces, in the field of view. An edge device (NVIDIA Jetson Nano) processes the video input, runs a distance estimation of obstacles using a lightweight depth map, and then sends out auditory cues to provide directions such that the user ensures his safe passage through the corridor. The experiment-based testing is a strong indication that it is possible to navigate towards walkable directions that are periodically indicated with obstacles and usable surface areas and at the same time receiving low-latency audio feedback through active spatial processing. The proposed, adaptable navigation assistance system is low cost and portable, as well as scalable to ensure greater independence and enlarging situation awareness for the user compared to traditional assistive mobility technologies.

Keywords: visually impaired; YOLOv8 Multi Head; FastDepth; Object detection; Depth estimation; Audio guidance

1. Introduction

Human movement, situational awareness, and independent living are all dependent upon a visual modality. Globally, over 285 million people experience some degree of visual impairment, which includes approximately 39 million who are classified as totally blind. These individuals encounter challenges associated with independent travel that begins in environments with many obstacles and in making real-time decisions about which direction to navigate, in

* Corresponding author. Tel.: +0-000-000-0000
E-mail address: sajidha.sa@vit.ac.in

environments that continually change. These impediments do not only interfere with mobility but significantly impact social participation and overall quality of life.

Though conventional assistive options like white canes and guide dogs exist, they provide very limited navigation guidance. A white cane provides information limited to obstacles that are physically close to the ground and provides no information about higher obstacles that also may be at a distance. Guide dogs, while effective, require extensive training, come with expensive care and maintenance, and are simply not an option for many potential users due to financial and logistical constraints. Mobile applications that utilize GPS, as well as the phone camera, have added functionality but can still be inaccurate when used indoors, features limited real-time navigation feedback, and also don't give real-time information about surrounding features. A consistent need is a portable, autonomous, and real-time assistive system that provide continuous situational awareness and adaptive navigation support.

The visually impaired are made to be more independent by way of intelligent assistive technologies that use object detection, segmentation, and depth estimation. Nevertheless, state-of-the-art systems have three significant drawbacks: 1) real-time navigation is not possible due to sequential processing which causes unacceptable latency, 2) guidance choices are made without incorporating the entire spatial context, and 3) excessive power needs make it impossible to deploy the system practically in a wearable form.

Our work presents a novel multi-head YOLOv8 architecture that simultaneously detects obstacles and segments pavement to address to these gaps and thus reduces latency by 33% when compared to the sequential processing. The FastDepth monocular depth estimation that is integrated gives a real-time spatial context for the navigation decisions made. The system is made to work best for embedded deployment on NVIDIA Jetson Nano, where it gets 8 FPS along with full environmental perception. The multi-modal audio feedback tells the user where the walkable and non-walkable areas are, and what the distance of the obstacles is, thereby giving a practical, user-centered solution for real-time navigation.

2. Related Work

Devices that support and assist movement and that rely on computer vision, embedded AI, and wearable technologies are pioneering change in the area of mobility for the visually impaired. The conventional tools such as guide dogs and white canes may help their users understand the environment surrounding them but they do not enable the user to perceive the movement of objects around him/her [1, 2]. Hence, to overcome this limitation, scientists came up with the use of AI-powered technology in the form of devices to make advances in mobility and safety by integrating object detection, depth sensing, and multimodal feedback.

The systematic reviews emphasize the rapid development that has occurred in this area. The review by Casanova et al. [1] highlighted the different technologies that have contributed to human navigation, especially GPS, ultrasonic sensors, Bluetooth devices, and haptic technology, and the progress made in the technology. They noted that the majority of the contributions were centered on experimental verification, with most of the contributions either addressing a haptic interface or smartphone applications. Another review was made by Naayini et al. [2], who focused on AI-based assistive ecosystems that included wearable devices, deep learning-based perception, and cloud computing-based navigational aids, and they observed an increase in travel independence, socialization, and the ability to access education and employment.

YOLO-based networks have been the prime focus of object detection algorithms, which have now become very popular to be utilized in assistive vision systems. Zhou et al. [3] propose a multi-scale detection framework termed SMA-YOLO that incorporates Non-Semantic Sparse Attention and Bidirectional Multi-Branch Auxiliary Feature Pyramid Networks to achieve better detection precision for small or remote objects. Meanwhile, Tahir et al. [4] came up with a drone-operated algorithm for the detection of pedestrians and cars named PV-Swin-YOLOv8s, which is capable of making real-time inferences of detection accuracy and even in dynamic surroundings. Assistive devices particularly benefited from this, as Sudha et al.[5] implemented YOLOv7 in an object detection framework that can identify 86 object types through non-semantic feature extraction from pictures, then use Braille outputs to convey the recognized objects, and finally, George et al.[6] reported real-time object detection on CPU devices with accuracy as high as that of images taken from a cellphone camera. Arsalwad et al. [7] took the concept of IoT connectivity a step further by employing YOLO-based object detection to store data so that situational awareness around obstacles could be made available through a wearable assistive device.

Edge computing and embedded deployment options can enable real-time operation. Yuan et al. [8] proposed an edge computing implementation with network awareness in 5G that was especially for wearable object detection, relying on the latency, speed and field of view coverage that 5G systems can offer. Liang et al. [9] demonstrated an Edge YOLO implementation that could process the COCO2017 dataset at 26 FPS with only 8 million parameters, thereby clearly indicating that practically portable to low power systems is indeed the case. The embedded system has undergone similar development where Yousif and Al-Jammas [10] project, which combined YOLOv7 object detection and video captions in Nvidia Jetson Nano systems, was presented supporting task-oriented inference times that are ideally suited for real world navigation.

Audio and spatial feedback methods are huge plus points for user assistance. Schwartz et al. [11] came up with a mobile app called EchoSee which uses a user's position for live 3D mapping and provides spatialized audio feedback for direction. Hu et al. [12] invented StereoPilot that was using audio spatialization (in-the-ear) to provide location accuracy with high precision/reduced error. Haptic feedback and multimodal interaction have been explored as well; Kevin et al. [13] implemented stereo-vision systems incorporated into a 3D embedded system providing a head-height obstacle monitoring; Kilian et al. [14] introduced the Unfolding Space Glove that converted data from the environment into vibratory feedback.

At last, the datasets and evaluation methodologies open up the possibility of strong navigation systems. Xia et al.[15] besides this, a new Walk On The Road dataset has been released that shows annotated outdoor environments with pavements, pedestrians and street objects in different lighting which can support the standardised training and evaluation of assistive AI systems. Gupta et al.[16] and Patil et al.[17] showcase the navigation-related studies that use the YOLO-based models and exhibit the real-time detection of the walkable path along with the navigation cues for both indoor and outdoor environments.

Combined these researches showcase the development possibilities of wearable AI systems for visually impaired people when combining object detection, depth estimation as well as multi-model feedback. But still, the three studies highlight some difficulties that are connected to low latency, real-time performance, contextual understanding and user-centered interfaces. Challenges are met by the proposed system that takes advantage of a multi-head YOLO for depth estimation while offering context aware audio navigation assistance for actionable navigation thus increasing autonomy and safety.

3. Architecture

The pipeline of the proposed system consists of two key models, as illustrated in Figure 1. The first one is a multi-head YOLOv8 network that has been trained on multiple datasets for the dual tasks of curb segmentation and obstacle detection. The second model is a FastDepth, which is applied for real-time depth estimation. These two models, when combined, give the assistive navigation system the required semantic and geometric perception of the environment.

3.1. Multi-Head YOLOv8 Training and Design

The multi-head YOLOv8 architecture that is put forward broadens the standard YOLOv8 detection system by the addition of two expert output heads: one for the segmentation of curb or pavement and the other for the detection of the obstacle's bounding box. The common backbone, which consists of C2f modules and spatial pyramid pooling, is responsible for the extraction of generalized spatial and semantic features from the input frames. These generalized features are subsequently divided into dedicated heads for the tasks that learn separate objectives. The segmentation head outputs a pixel-level mask to predict the curb boundaries, whereas the detection head uses bounding boxes and class confidence scores for locating the obstacles.

Every head was trained with a unique dataset:

- The curb segmentation head was trained on labeled pavement datasets with pixel-wise curb labels.
- The obstacle detection head was trained on outdoor pedestrian and obstacle datasets with bounding box annotations.

The two heads had the same backbone parameters but were optimized in unison using a combined loss:

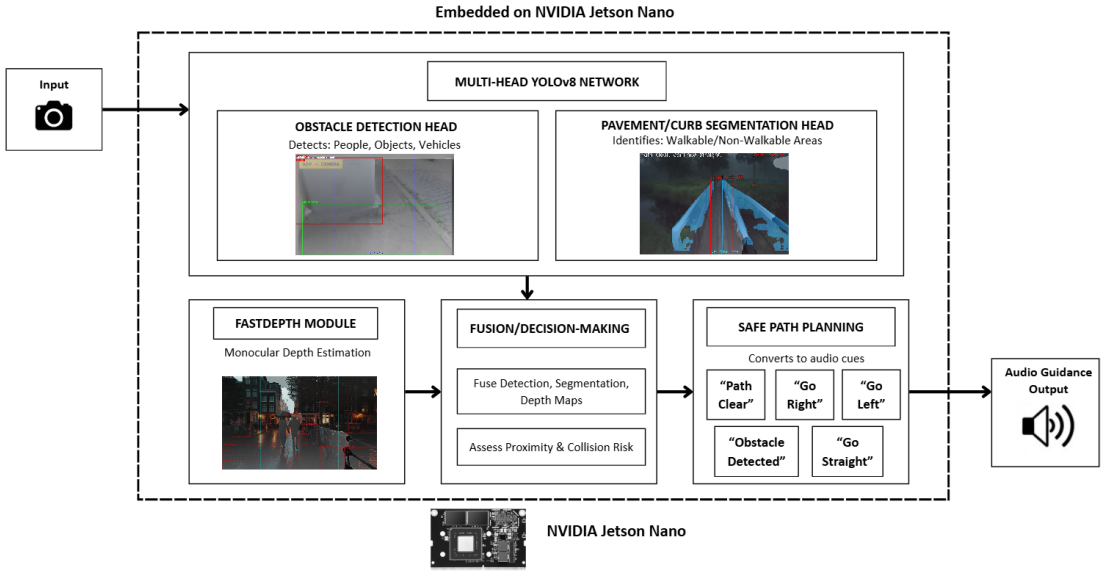


Fig. 1: System architecture diagram.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{seg} + \lambda_2 \mathcal{L}_{det}, \tag{1}$$

where $\mathcal{L}_{seg}$ is the segmentation loss (binary cross-entropy and Dice loss), and $\mathcal{L}_{det}$ is the detection loss (bounding box regression and objectness).

In this part, the utilization of the multi-task learning framework showed to be successful for the heads to share spatial representations while maintaining their separate accuracies. Finally, the final model was fine-tuned with data augmentations, which consisted of random cropping, brightness changes and application of motion blur to improve robustness against real outdoor images.

3.2. FastDepth for Depth Estimation

FastDepth, which is a lightweight monocular depth estimation network that is optimized for real-time operation and low-power embedded hardware such as the NVIDIA Jetson Nano, is used to acquire depth information. The model employs MobileNet as the encoder for efficient feature extraction and the depthwise separable convolution decoder with skip connections to reconstruct dense depth maps. This approach considerably trims the parameter's number (under 4 million) and computation cost while still delivering decent accuracy outdoors. Training was done with large-scale monocular datasets and scale-invariant loss was used to make sure the depth estimation was always consistent regardless of the distance. In the inference phase, FastDepth gives a per-pixel depth map that is aligned with the RGB frame, thus, presenting accurate distance information that enhances the results of YOLOv8 detections.

3.3. Justification for Model Selection

Real-time performance was the main consideration the two models had during their selection, and they would not lose any accuracy in the process. The multi-head YOLOv8 takes care of detecting and segmenting very well in changing surroundings, and FastDepth brings spatial depth cues that are trustworthy and have very low computational cost. The merged architecture makes it possible to have continuous operation on the Jetson Nano platform with the

support of ongoing frame processing, low latency, along with power consumption that is efficient—all of which are important factors for wearable assistive navigation systems.

4. Proposed Methodology

The solution we propose delivers immediate help in navigation through the combination of audio feedback, perception by sight, estimation of depth, and reasoning. The architecture of the system is made up of the sequential pipeline for obtaining input, which is then followed by the phases of object recognition, computation of safe zones, and the creation of directional cues. This comprehensive design is aimed at offering navigation guidance that is both dependable and usable for the individuals walking in outdoor area and facing high activity levels.

4.1. Input Acquisition and Preprocessing

A wearable device with a monocular RGB camera captures the visual data. The camera produces a frame at each moment in time t:

$$I_t \in \mathbb{R}^{H\times W\times 3}, \tag{2}$$

where H denotes the height and W represents the width of the image. Each frame undergoes resizing and normalization to prepare the input for further processing:

$$I'_t = f_{prep}(I_t) = \frac{\text{Resize}(I_t, h, w)}{255}. \tag{3}$$

Such preprocessing ensures that the frames exhibit the same scale and the same intensity distributions, which in turn, increases the reliability and accuracy of the detection and depth estimation processes in the following modules.

4.2. Object Detection and Depth Estimation

The frame that is preprocessed I'_t is directed to a multi-head YOLO model f_{yolo} that simultaneously detects pavements, pedestrians, and obstacles:

$$D = f_{yolo}(I'_t) = \{(b_k, c_k, s_k) \mid k = 1, \ldots, N\}, \tag{4}$$

where b_k is the bounding box, c_k is the class label, and s_k is the confidence score corresponding to that class. Non-Maximum Suppression (NMS) is performed to get rid of overlapping detections:

$$D^* = \text{NMS}(D, \theta_{iou}), \tag{5}$$

here, θ_{iou} is the threshold for Intersection over Union (IoU).

At the same time, a compact depth estimation model fdepth creates the depth map Z of the scene:

$$Z(i, j) = f_{depth}(I'_t), \quad Z \in \mathbb{R}^{H\times W}. \tag{6}$$

The average depth is determined for every recognized obstruction:

$$d_k = \frac{1}{|b_k|} \sum_{(i,j)\in b_k} Z(i, j), \tag{7}$$

and if $d_k \leq d_{th}$, then the obstructions are classified as actionable. The safe distance threshold is indicated by d_{th}. In this way, only the risks that are close enough can be included in the navigation decision process.

4.3. Integrated Real-Time Navigation Algorithm

To combine detection, depth, and decision-making into one single workflow, the system executes the following algorithm shown in Algorithm 1.

Algorithm 1 Real-Time Obstacle Detection and Safe Path Guidance

Require: f_{yolo}: Multi Head v8 model
Require: f_{depth}: Depth estimation model(FastDepth)
Require: f_{prep}: Image preprocessing function
Require: θ_{iou}: IoU threshold for NMS
Require: d_{th}: Maximum distance for actionable obstacles

Initialize camera with desired resolution and frame rate
while System is Active **do**
    $I_t \leftarrow$ CaptureFrame()
    $I'_t \leftarrow f_{prep}(I_t)$
    ▷ *Detection and Depth Estimation*
    $D \leftarrow f_{yolo}(I'_t)$
    $Z \leftarrow f_{depth}(I'_t)$
    $D^* \leftarrow \text{NMS}(D, \theta_{iou})$
    $O_{actionable} \leftarrow \emptyset$
    for all bounding box $b_k \in D^*$ **do**
        if b_k is an obstacle **then**
            $d_k \leftarrow \text{ComputeDepth}(b_k, Z)$
            if $d_k \leq d_{th}$ **then**
                Add b_k to $O_{actionable}$
            end if
        end if
    end for
    ▷ *Safe-Zone Analysis and Feedback*
    Partition frame into C_L, C_C, C_R
    Compute safety scores S_L, S_C, S_R based on $O_{actionable}$
    Generate directional audio feedback according to highest S_{region}
end while

4.4. Safe-Zone Analysis and Decision-Making

The frame is divided into three vertical corridors: left (C_L), center (C_C), and right (C_R). Each corridor is assigned an impaired vision score based on the ratio of obstructable to unobstructed space:

$$S_{region} = 1 - \frac{\text{Area of obstacles in region}}{\text{Total area of region}}. \tag{8}$$

The navigation decision a_t is derived from these scores:

$$a_t = \begin{cases} \texttt{No Pavement}, & \text{if no pavement detected,} \\ \texttt{Continue Straight}, & \text{if } S_C \geq \delta \wedge d_C > d_{th}, \\ \texttt{Move Left}, & \text{if } S_L > S_R, \\ \texttt{Move Right}, & \text{otherwise,} \end{cases} \tag{9}$$

where δ signifies the limitation of minimum free space for successful node traversal through the center corridor. When selected, the command will also be communicated to the user aurally through a sound cue for real-time feedback to the user.

Overall, this experience clearly illustrates the entire assistive navigation process in three subsections of: pavement detection, obstacle detection, depth-based filtering, and reasoning to identify a safe-path. The algorithm combined enables considerable real-time guidance as the user navigates, what is typically unpredictable outdoor environments that is even more intricate and dynamic (see Figure 2).

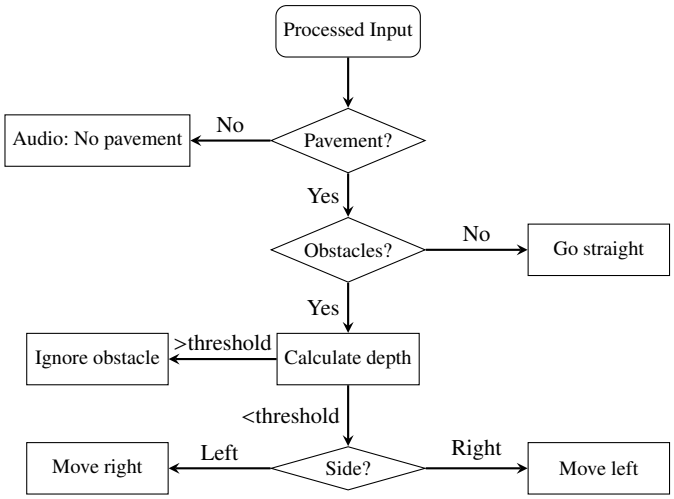


Fig. 2: Flowchart of the decision-making process for pavement detection, obstacle avoidance, and directional guidance.

5. Experimental Setup

The proposed framework is evaluated on two publicly available datasets. The first dataset is the Walk On The Road (WOTR) dataset [15] which has annotated outdoor scenarios with pavements, sidewalks, pedestrians, and urban street obstacles under varying lighting conditions, making it suitable for training the model to adapt to robust use cases. The second dataset, was a curb segmentation dataset with pixel-wise annotated masks which was leveraged for additional support. It comprised of a variety of pavement textures and road conditions, which made it suitable for the required use case. All data was resized to $H \times W$, normalized, and subsequently split into training (70%), validation (15%), and testing (15%) sets.

The pipeline was setup using two seperate environments. The training was done in the Kaggle cloud computing environment which uses a 16 GB memory NVIDIA Tesla T4 GPU and a 13 GB RAM assignment. Inference on the other hand was done on two systems: a Lenovo IdeaPad Flex 5i laptop powered by an AMD Ryzen 7 8845HS processor (8 cores, 16 threads, max 5.1 GHz), 16 GB LPDDR5X RAM, and integrated AMD Radeon 780M graphics, operating Windows 11 Home; and an NVIDIA Jetson Nano with a quad-core ARM Cortex-A57 CPU, 4 GB RAM, and a 128-core Maxwell GPU. This setup allows for a powerful computation resource for both training and inference, and at the same time, it can provide the actual measurement of per-frame latency for the embedded deployment case scenarios.

The performance of the method is evaluated based on three complementary metrics. For the obstacle detection model, the mean average precision at IoU threshold 0.5 (mAP@0.5) is used.

$$\text{mAP@0.5} = \frac{1}{N}\sum_{i=1}^{N} AP_i, \tag{10}$$

where AP_i denotes the average precision of class i across N classes. For pavement segmentation, mean Intersection-over-Union (mIoU) is used:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c} \tag{11}$$

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C} \text{IoU}_c, \tag{12}$$

where TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives for class c, and C is the number of segmentation classes. In addition, the average per-frame inference time is recorded to quantify latency:

$$\text{Latency (ms/frame)} = \frac{\text{Total Inference Time (ms)}}{\text{Number of Frames}}. \tag{13}$$

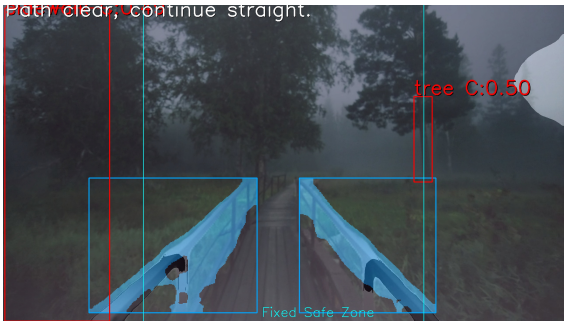
Together, these metrics capture detection accuracy, segmentation quality, and computational efficiency, providing a comprehensive basis for evaluating the proposed navigation system.

6. Results and Discussion

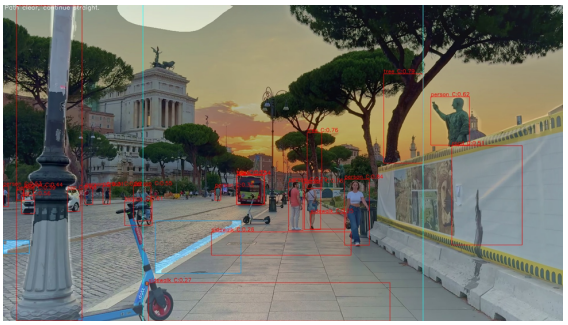
All of these metrics provide a comprehensive basis for assessment of the proposed navigation system based on detection accuracy, segmentation performance and processing efficiency.

6.1. Obstacle Detection Performance

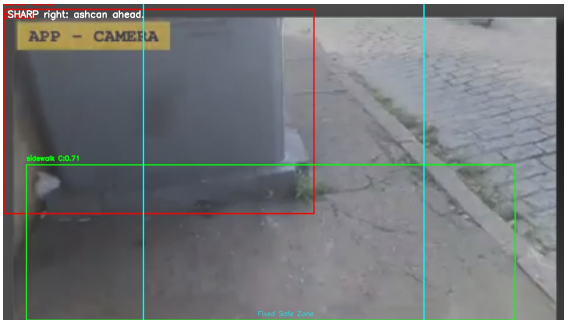
Table 1 illustrates our multi-head model's performance with respect to the most common object detection baseline methods, namely YOLOv8, Faster R-CNN, and DETR. The comparison is made using three metrics: mean Average Precision (mAP), F1 score, and Recall. The multi-head architecture gets to results that are competitive and even



(a) Curb Based Path Guidance: Continue Straight



(b) Object too far to interfere: Distance calculated by depth map



(c) Obstacle Detection with interference: Sharp right

(d) Obstacle (Person) outside safe zone

Fig. 3: Inference results with depth overlay and instructions.

slightly lower than those of single-task detectors, yet it takes advantage of the simultaneous multi-task inference, thus making the trade-off between accuracy and computational efficiency very strong.

Table 1: Obstacle detection performance comparison.

Model	**mAP (%)**	**F1 (%)**	**Recall (%)**
YOLOv8	85.1	81.9	83.2
DETR	84.8	81.7	83.5
Faster R-CNN	87.5	85.1	86.7
Proxi-NAV (Proposed)	86.2	83.0	84.3

Table 2: Pavement/curb segmentation performance comparison.

Model	**mAP (%)**	**F1 (%)**	**Recall (%)**
YOLOv8-seg	85.8	82.0	84.7
U-Net	86.7	83.5	85.8
Mask R-CNN	86.2	84.1	85.3
Proxi-NAV (Proposed)	85.9	82.4	84.6

6.2. Pavement/Curb Segmentation Performance

In order to perform the segmentation, we compare our multi-head network with top-notch segmentation models including YOLOv8-seg, U-Net, and Mask R-CNN(Table 2. The evaluated metrics consist of mAP, F1, and Recall. The multi-head network attains performance that is on par with individual segmentation models while at the same time doing obstacle detection.

6.3. Real-Time Inference on Jetson Nano

The multi-head network's performance was compared to the sequential processing of individual models on a Jetson Nano. The latency per frame and frames per second (FPS) are shown in the following table. Table 3 shows the comparision of our model's latency with the seperate obstacle detection models used for qualitative comparision

in this paper, and Table 4 shows a similar comparision with the segmentation models. The multi-head network, in comparision with the seperate YOLO models, cuts off 54% of the time during each frame while still delivering comparable performance, thus, it can be considered for real-time autonomous navigation applications.

Table 3: Latency comparison for obstacle detection models on Jetson Nano.

Model	Latency (ms)	FPS
Faster R-CNN	195	5
DETR	105	9
YOLOv8	90	11
Proxi-NAV (Proposed)	**85**	**12**

Table 4: Latency comparison for pavement/curb segmentation models on Jetson Nano.

Model	Latency (ms)	FPS
U-Net	85	12
Mask R-CNN (Seg)	130	8
YOLOv8-seg	80	12
Proxi-NAV (Proposed)	**75**	**13**

6.4. Discussion

The findings reveal that the multi-head network is capable of remaining powerful in both obstacle detection and pavement/curb segmentation tasks while still being real-time capable. The multi-head network performs detection and segmentation within 2-3% of the best single-task models while at the same time, cutting down the total inference latency, thereby allowing embedded real-time deployment. The network by dealing with all perception tasks at once presents an efficient and practicable way for autonomous navigation. These results show that multi-task learning is effective in maintaining a trade-off between accuracy and computational efficiency in the case of real-world autonomous systems.

7. Limitations

Although the suggested system provides reliable obstacle and pavement detection in a variety of scenarios, there are limitations.

First, the models may not accurately identify the asphalt/pavement or road underneath the snow. This can lead to gaps in the reliability of the guidance. An example implementing this limitation is illustrated in Figure 4a, where the pavement detection fails because of the snow-covered surface.

Second, in settings where the road and pavement are both made of the same material, such as bricks, the model will confuse one for the other, and will process them as a single class in the output. This is not ideal for spatial discrimination, which is evidenced by Figure 4b.

(a) Snow-covered pavement and road surfaces

(b) Road and pavement made of identical bricks

Fig. 4: Failure cases in pavement/road segmentation under challenging conditions.

Lastly, in crowded locations with a lot of people, the model tends to over perform and generate chaotic output. This effect can be diminished by further developing the algorithm to incorporate environment specific instruction

sets which the pipeline can refer to. This limitation provides opportunities for us to develop more improvements and research around this to improve the robustness and reliability of the system in varied conditions within the real world.

8. Conclusion

The suggested voice-assisted embedded AI system is an essential improvement in assistive navigation for the visually challenged by incorporating multi-head deep learning architecture with real-time depth estimation. It is running obstacles, pedestrians, and the ground at the same time while giving out context-aware audio guidance with directions, hence overcoming the past solutions' limitations such as lack of real-time path planning and limited and non-interactive feedback. The main contributions are low-latency edge processing on embedded hardware, simultaneously multi-task detection, and depth mapping, intuitive audio feedback, and a modular, scalable architecture for wearable devices. All these functions are creating the basis for trustworthy, instant assistive navigation in changing environments.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] E. Casanova, D. Guffanti, and L. Hidalgo, "Technological advancements in human navigation for the visually impaired: A systematic review," *Sensors*, vol. 25, no. 7, p. 2213, 2025.
[2] P. Naayini, P. K. Myakala, C. Bura, A. K. Jonnalagadda, and S. Kamatala, "AI-powered assistive technologies for visual impairment," *arXiv preprint arXiv:2503.15494*, 2025.
[3] S. Zhou, H. Zhou, and L. Qian, "A multi-scale small object detection algorithm SMA-YOLO for UAV remote sensing images," *Scientific Reports*, vol. 15, p. 9255, 2025.
[4] N. U. A. Tahir, Z. Long, Z. Zhang, M. Asim, and M. E. LAffendi, "PVSwin-YOLOv8s: UAV-based pedestrian and vehicle detection for traffic management in smart cities using improved YOLOv8," *Drones*, vol. 8, no. 3, p. 84, 2024.
[5] L. K. Sudha, R. Ajay, S. H. Manu Gowda, V. B. Poornima, and S. Vaibhav, "A deep learning-based assistive system for the visually impaired using YOLO-V7," *Revue d'Intelligence Artificielle*, vol. 37, no. 4, pp. 809–816, 2023.
[6] A. M. George, A. Ramachandran, M. C. M., M. A. T., B. A. R., and P. Subeh, "YOLO-based object recognition system for visually impaired," *Int. J. Sci. Eng. Appl. Sci.*, vol. 14, no. 1, pp. 34–42, 2025.
[7] G. Arsalwad, et al., "YOLOInsight: Artificial intelligence-powered assistive device for visually impaired using Internet of Things and real-time object detection," *Cureus*, vol. 16, no. 12, e74953, 2024.
[8] Z. Yuan, T. Azzino, Y. Hao, Y. Lyu, H. Pei, A. Boldini, M. Mezzavilla, M. Beheshti, M. Porfiri, T. E. Hudson, et al., "Network-aware 5G edge computing for object detection: Augmenting wearables to 'see' more, farther and faster," *IEEE Access*, vol. 10, pp. 29612–29632, 2022.
[9] S. Liang, M. Huang, S. Fu, W. Yang, J. Wei, and L. Zhang, "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25345–25360, 2022.
[10] A. J. Yousif and M. H. Al-Jammas, "A lightweight visual understanding system for enhanced assistance to the visually impaired using an embedded platform," *Diyala J. Eng. Sci.*, vol. 17, no. 3, pp. 146–162, 2024.
[11] B. S. Schwartz, S. King, and T. Bell, "EchoSee: An assistive mobile application for real-time 3D environment reconstruction and sonification supporting enhanced navigation for people with vision impairments," *Bioengineering*, vol. 11, no. 8, p. 831, 2024.
[12] X. Hu, A. Song, Z. Wei, and H. Zeng, "StereoPilot: A wearable target location system for blind and visually impaired using spatial audio rendering," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 1621–1630, 2022.
[13] M. Kevin, M. Chavarria, L. Ortiz, S. Sutter, K. Schönenberger, and B. Bacca-Cortes, "Embedded solution to detect and classify head level objects using stereo vision for visually impaired people with audio feedback," *Scientific Reports*, vol. 15, p. 17277, 2025.
[14] J. Kilian, A. Neugebauer, L. Scherffig, and S. Wahl, "The unfolding space glove: A wearable spatio-visual to haptic sensory substitution device for blind people," *Sensors*, vol. 22, no. 5, p. 1859, 2022.
[15] H. Xia, C. Yao, Y. Tan, and S. Song, "A dataset for the visually impaired walk on the road," *Displays*, vol. 79, p. 102486, 2023.
[16] K. Gupta, et al., "Towards walkable footpath detection for the visually impaired," *IEEE Access*, vol. 13, pp. 11234–11245, 2025.
[17] S. Patil and V. Nair, "YOLOv7 based indoor and outdoor navigation assistance for visually impaired people," *Int. J. Creative Res. Thoughts*, vol. 11, no. 7, pp. 2045–2054, 2023.