MonoDepth-Assist: Monocular Obstacle Detection and Depth Estimation for Assistive Navigation
Final Project 2025 Group 06

Harshitha Honnappa
4756182
Data and Computer Science

Khushi Kumar
4756144
Data and Computer Science

Abstract

We present MonoDepth-Assist, a monocular detection–depth fusion system designed to enhance navigation for visually impaired users. By combining YOLOv8 for real-time object detection with Depth-Anything for high-quality depth estimation, our pipeline identifies obstacles and estimates their relative distance from a single RGB image. Each detected object is assigned a median depth and classified as Left, Center, or Right, providing interpretable spatial and semantic information. The system outputs annotated images and structured data for quantitative analysis, achieving real-time performance without additional sensors. This approach offers a lightweight, accurate, and practical solution for assistive navigation, improving safety and independence in diverse environments.

1. Introduction

Cities are hectic, uncertain places. Bicycles that suddenly emerge, cars that partially obstruct the path, and pedestrians who alter speed can all be obstacles for those with vision impairments just walking along a sidewalk. Recognizing what's close by is insufficient; you also need to be able to quickly determine its location and distance in order to be of any assistance.

Object detection and monocular depth estimation are fundamental problems in computer vision, particularly relevant for autonomous driving, robotics, and navigation systems. Traditionally, these tasks were addressed independently, with object detectors such as YOLO focusing on identifying and localizing objects, and depth estimation models predicting scene geometry from monocular images. However, recent research has increasingly focused on combining these tasks to achieve a more comprehensive scene understanding. This fusion-based approach allows for a lightweight monocular system without relying on additional sensors such as LiDAR or stereo cameras, while maintaining high accuracy and real-time performance. Beyond its technical contributions, the system emphasizes real-world applicability for assistive technology, enhancing spatial awareness, safety, and independence for visually impaired users in diverse environments, from indoor spaces to public roads.

We test on a range of Cityscapes street scenarios to keep things tangible. The objective is simple: do the left, center, and right labels correspond to what you see, do the approximate distances appear realistic for a standard camera, and do closer obstructions appear as closer? While we focus on single-frame perception from a RGB image, future extensions include temporal smoothing to reduce flicker, lightweight tracking for stable IDs and motion cues, and clearer on-screen summaries (e.g., color-coded near/medium/far), which we leave for future work.

The next section reviews related work on assistive detection, depth-aware perception, and camera-only distance estimation etc. We then present the method at a level that makes each design choice transparent, followed by experiments on Cityscapes and an ablation study of key parameters. We conclude with limitations and directions for extension.

2. Related Work

2.1. Detection for assistive navigation

Wang *et al*. [7] introduce YOLO-OD, a YOLO-based detector tailored to outdoor navigation for visually impaired users. They have added three modules to cope with street-scene challenges: a Feature Weighting Block (FWB) to emphasize the most informative features across object scales, an Adaptive Bottleneck Block (ABB) to stay robust in clutter and under occlusion, and an Enhanced Feature Attention Head (EFAH) to better pick up small obstacles.

2.2. Camera-only distance estimation.

Aung *et al*. [1] propose a research work on the YOLOv8 algorithm. They discuss the YOLOv8 algorithm's object detection accuracy and distance estimation with the camera's lens principle. The research shows that with a single webcam, you can estimate object range using a simple focal-length, making monocular distance practical without extra sensors. In contrast Rahul *et al*. [5] combine CNN detection with a stereo camera to triangulate distance and report low error up to 5 meters—useful but with added hardware cost and power needs.

2.3. Multi-sensor assistive devices

SmartCane *et al*. [6], a device that combines a transformer detector (RT-DETR) with Intel RealSense RGB-D depth, delivering semantic labels plus metric distance on embedded hardware; the latency and energy constraints typical of portable systems are also discussed in the paper.

2.4. Camera-only assistive prototypes

EBICS [2], a low-cost vision system (camera + lightweight detector) that identifies objects and approximates distance without extra sensors, targeting affordability and portability for visually impaired users. Paswan *et al*. [4] focus on indoor mobility: they use YOLOv7 and estimate distance from bounding-box cues, then relay information to users—illustrating how camera-only setups can still deliver helpful spatial context.

2.5. Task specific aid

Wongta *et al*. [8] present a vision system that helps low-vision travelers read bus route numbers from natural images. Their three-stage pipeline detects the bus panel with Faster R-CNN (ResNet-101, TensorFlow Object Detection), enhances the crop (CLAHE, gamma correction), and runs OCR via Google Cloud Vision; the recognized number is then spoken aloud.

2.6. Monocular depth at scale

Yang *et al*. [9] introduce Depth Anything, a depth "foundation model" trained with massive pseudo-labeled data to generalize across diverse scenes. It is useful when depth must come from a single RGB image in unfamiliar environments. Complementing this, Ninan *et al*. [3] show that using Depth Anything v2 as pseudo-supervision can strengthen semi-supervised monocular depth on driving datasets like Cityscapes, improving depth boundaries and dynamic-object handling—evidence that such foundation models are strong partners for detection-depth fusion.

In simple words, the papers point to a practical approach to use a fast detector together with a good monocular depth model to give each object a sense of what it is, where it is (left/center/right), and how near or far—all with just a single camera. YOLO-OD [7] shows that detectors tuned for street scenes handle clutter, occlusion, and small objects better. The YOLOv8 webcam distance [2] work shows you can estimate rough distance from one camera using simple geometry. Depth Anything [9] shows that modern monocular depth models generalize well to new outdoor scenes. Together, these works support the baseline: a lightweight, camera-only pipeline that attaches per-object spatial cues in real street scenes.

3. Method

Let I be a single monocular RGB image captured from a forward-facing camera. The core idea behind MonoDepth-Assist is to fuse two complementary computer vision tasks — object detection and monocular depth estimation — to not only identify and localize obstacles in the scene but also infer their spatial arrangement and approximate distance from the camera. By combining the semantic understanding provided by an object detector with the geometric cues derived from a depth estimation network, our system transforms a single 2D image into a richer 3D-aware scene representation. This enables us to associate each detected object with meaningful spatial information — such as relative depth, estimated distance, and left–center–right direction — using only a monocular camera, without relying on additional sensors. The resulting fused representation goes beyond mere object recognition, providing actionable navigation cues essential for assistive technologies.

3.1. Problem Setup

Given a single RGB image I of an urban scene, the goal is to attach *spatial cues* to each detected obstacle: a *relative distance* (near–far ordering), a *coarse bearing* (Left/Center/Right), and a *first-order range* in meters under modest camera assumptions.

Input. We consider an image

$$I \in \mathbb{R}^{H \times W \times 3}, \tag{1}$$

with height H and width W.

Detector outputs: An object detector applied to I yields a set of class/box predictions

$$\mathcal{D}(I) = \{\, (c_i,\, s_i,\, b_i) \,\}_{i=1}^{N}, \quad b_i = (x_1, y_1, x_2, y_2), \tag{2}$$

where c_i is the class label, $s_i \in [0, 1]$ is the confidence, and b_i is an axis-aligned bounding box in pixel coordinates.

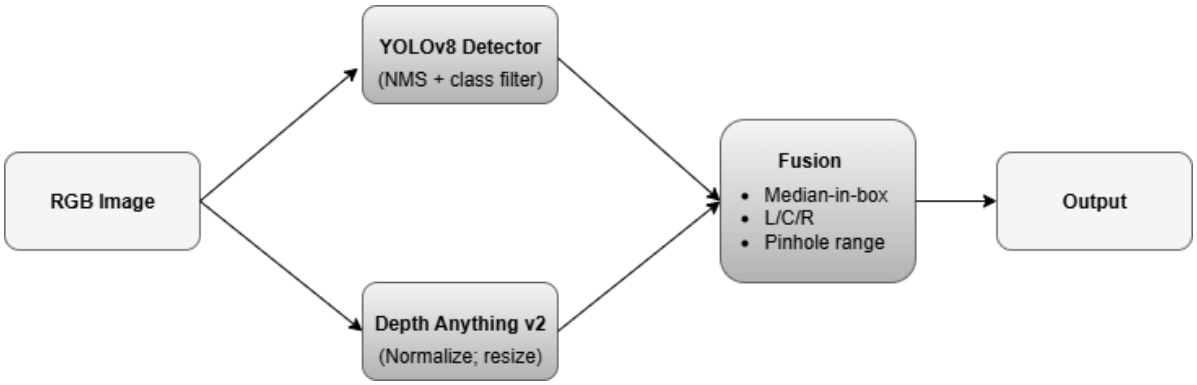


Figure 1. Architecture of the proposed MonoDepth-Assist.

Depth prediction. A monocular depth estimator produces a dense *relative depth* map

$$D_{\text{rel}} = \mathcal{M}(I) \in [0,1]^{H\times W}, \tag{3}$$

interpreted such that smaller values correspond to nearer regions (after normalization; details in §3.4).

Per-object outputs. For each detection we produce a spatial summary

$$o_i = \big(c_i,\ s_i,\ d_{\text{rel}}^{(i)},\ d_{\text{m}}^{(i)},\ \text{bearing}^{(i)},\ b_i\big), \tag{4}$$

where $d_{\text{rel}}^{(i)}$ is the per-object relative distance, $d_{\text{m}}^{(i)}$ is the approximate metric range, and $\text{bearing}^{(i)}$ is the coarse direction.

$$\text{bearing}^{(i)} \in \{\text{Left},\ \text{Center},\ \text{Right}\}, \tag{5}$$

Objective. Given I, the detector $\mathcal{D}$, and the depth model $\mathcal{M}$, compute the set

$$\mathcal{O}(I) \ = \ \big\{\, o_i \,\big\}_{i=1}^{N} \tag{6}$$

that augments each detection with interpretable spatial cues suitable for assistive navigation.

3.2. Detection using YOLOv8

We employ an off-the-shelf YOLOv8 detector for single-frame obstacle recognition. Detections are filtered to a target set of urban classes (e.g., *person*, *bicycle*, *car*, *motorcycle*, *bus*, *truck*, *train*, *traffic light*, *stop sign*, *bench*). Detections outside these classes are discarded. A single confidence threshold of 0.5 governs the precision–recall trade-off, and standard Non-Maximum suppression (NMS) with an IoU threshold of 0.4 removes duplicate or overlapping detections. This stage outputs the set $\mathcal{D}(I)$ defined in Eq. (2).

3.3. Monocular Depth Estimation

From the same image I, a monocular depth estimator predicts a dense relative depth map D. We apply min–max normalization for consistent scaling:

$$D_{\text{rel}} = \frac{D - \min(D)}{\max(D) - \min(D) + \varepsilon}, \tag{7}$$

with a small ε to avoid division by zero. We then (bi)linearly resize D_{rel} to (H, W) if needed, so that Eq. (3) holds.

3.4. Detection–Depth Fusion

Per-object relative distance. For each detection b_i, we aggregate depth using the median within the box:

$$d_{\text{rel}}^{(i)} = \text{median}\,\big\{\, D_{\text{rel}}(u,v) \;\big|\; (u,v) \in b_i \big\}. \tag{8}$$

Coarse bearing (Left/Center/Right). Let $x_c = \frac{1}{2}(x_1 + x_2)$ be the horizontal center of b_i. We divide the image width into thirds:

$$\text{bearing}^{(i)} = \begin{cases} \text{Left}, & x_c < \frac{W}{3}, \\ \text{Center}, & \frac{W}{3} \le x_c \le \frac{2W}{3}, \\ \text{Right}, & x_c > \frac{2W}{3}. \end{cases} \tag{9}$$

Approximate metric range (first-order). Assuming a horizontal field of view θ and width W, the effective focal length in pixels is

$$f = \frac{W}{2\,\tan(\theta/2)}. \tag{10}$$

If class c_i has an average real-world height H_{c_i} (meters) and the box height is $h_{\text{px}} = y_2 - y_1$, we estimate a first-order range

$$d_{\text{m}}^{(i)} = \frac{f\, H_{c_i}}{h_{\text{px}}}, \qquad h_{\text{px}} = y_2 - y_1. \tag{11}$$

We clamp extremely small h_{px} for numerical stability and omit $d_{\text{m}}^{(i)}$ when degenerate.

3.5. Outputs and Sorting

For each frame we produce (i) an annotated image with labels showing $\{c_i, s_i, d_{\text{rel}}^{(i)}, d_{\text{m}}^{(i)}, \text{bearing}^{(i)}\}$ and (ii) a structured table with one row per object containing

$$\{\, c_i,\ s_i,\ d_{\text{rel}}^{(i)},\ d_{\text{m}}^{(i)},\ \text{bearing}^{(i)},\ x_1,\ y_1,\ x_2,\ y_2 \,\}.$$

Objects may be sorted by increasing $d_{\text{rel}}^{(i)}$ (nearest first); we denote the sorted set by

$$\mathcal{O}^{\uparrow}(I) \;=\; \text{sort}\left(\mathcal{O}(I);\ d_{\text{rel}}^{(i)}\right). \tag{12}$$

The architecture diagram 1 shows the pipeline overview of MonoDepth-Assist. A single RGB image is processed by YOLOv8 (detections) and Depth Anything v2 (relative depth); a fusion step attaches per-object spatial cues—median-in-box depth, left/center/right bearing, and a pinhole-based range—to produce the final output.

4. Experiments

We evaluated the MonoDepth-Assist pipeline by analyzing both object detection and depth estimation performance. Table 1 summarizes the comparison of different YOLOv8 model variants, including the number of detections, mean and minimum depth values, and per-frame inference time. Smaller models, such as YOLOv8n, achieve the fastest processing times, making them suitable for real-time assistive navigation, but detect fewer obstacles and may miss small or distant objects. Larger models, such as YOLOv8l, detect more obstacles, including peripheral and close-range ones, but require significantly longer inference times, which could limit practical deployment. Across all variants, median depth estimates remain consistent, indicating that the Depth-Anything module provides stable and reliable spatial cues independent of the detection backbone.

To further assess depth estimation robustness, we conducted preliminary experiments using the MiDaS model. While MiDaS produces qualitatively similar relative depth maps, converting these to approximate metric distances relies on a linear scaling approximation, making it less precise than Depth-Anything for real-world distance estimation. Overall, these results suggest that mid-sized models, such as YOLOv8s, paired with Depth-Anything, offer a balanced solution that achieves real-time performance while maintaining comprehensive obstacle awareness for monocular assistive navigation.

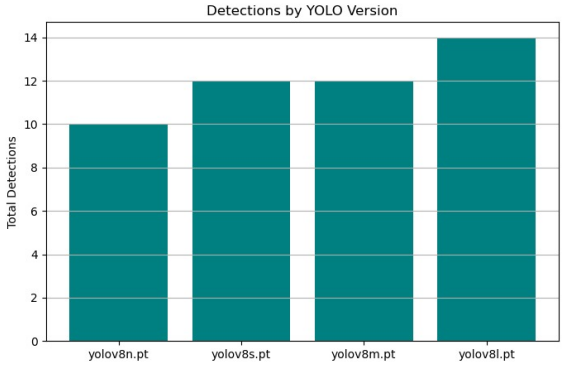


Figure 2. Total number of obstacle detections for each model, showing that larger models detect more objects, including smaller or distant obstacles.

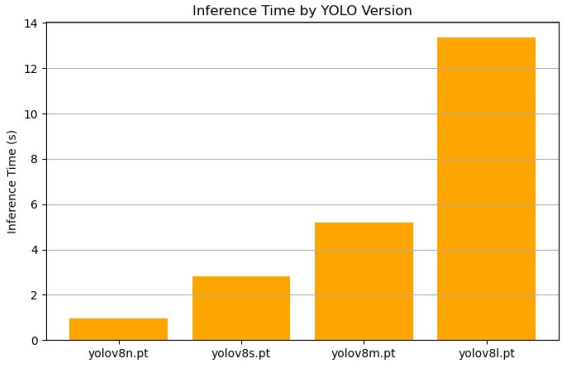
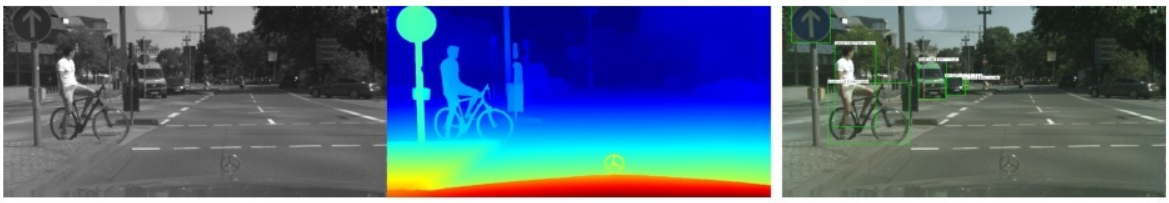


Figure 3. Inference time for each model, illustrating the trade-off between processing speed and detection coverage.

The Table 1 shows the YOLOv8 model comparison for MonoDepth-Assist: detections per frame, relative depth, and inference time. YOLOv8s provides a balanced trade-off between speed and coverage.

Model	# Detections	Mean Depth	Min Depth	Inference Time (s)	Notes
YOLOv8n.pt	10	0.445	0.354	0.96	Fastest, may miss small or distant obstacles
YOLOv8s.pt	12	0.326	0.028	2.83	Balanced speed and detection; recommended for assistive navigation
YOLOv8m.pt	12	0.444	0.355	5.17	Slightly slower, similar coverage to YOLOv8s
YOLOv8l.pt	14	0.391	0.028	13.37	Best coverage including small/peripheral objects, but slow

Table 1. YOLOv8 Model Comparison for MonoDepth-Assist



```
--- Detected Objects in frankfurt_000000_016286_leftImg8bit.png ---
    class  distance_rel direction   abs_err  rel_err
  bicycle      0.628726      Left  3.187423 0.835246
stop sign      0.059173      Left  8.688566 0.993236
   person      0.329315      Left  3.681745 0.917898
      car      0.473831    Center 15.526169 0.970386
    truck      0.405455    Center 17.514545 0.977374
      car      0.452260    Center 26.976312 0.983511
```

Figure 4. Representative example of result.

5. Ablation Study

The proposed pipeline was evaluated on the Cityscapes validation set (Frankfurt subset), integrating YOLOv8 object detection with DepthAnything depth estimation. The evaluation includes both qualitative visualizations and quantitative metrics, enabling a comprehensive understanding of detection performance, depth estimation accuracy, and spatial object distribution in urban street scenes.

5.1. Qualitative Analysis

For each image, we generated three-panel visualizations comprising:

Original grayscale image – provides a visual reference of the scene.

Depth heatmap – per-pixel depth predictions from DepthAnything, normalized and color-mapped for interpretability.

YOLO annotated predictions – includes bounding boxes, object class, confidence, median depth within the bounding box, estimated real-world distance based on object height, and overlaid absolute and relative depth errors.

These visualizations allow simultaneous inspection of detection accuracy and depth estimation.

The Figure 4 shows a representative example:

- Left panel: grayscale input image.

- Middle panel: depth heatmap, where warmer colors represent closer objects and cooler colors indicate farther objects.

- Right panel: YOLO predictions annotated with class, confidence, estimated distance, and depth errors

Observation:

- Large, well-defined objects (cars, buses, trucks) are consistently detected with high confidence and reasonably accurate depth estimates.

- Small or occluded objects (traffic lights, motorcycles, pedestrians at a distance) are often partially detected, leading to bounding box misalignment and larger depth errors.

- Depth predictions visually align with scene layout, *e.g.*, closer objects appear warmer, and road surfaces show consistent gradients.

5.2. Object Detection Performance

Across the Frankfurt validation subset, the pipeline detected 87 objects, distributed as shown in Table 2:

Class	Count
Car	65
Person	11
Bus	4
Truck	3
Traffic Light	3
Motorcycle	1

Table 2. Object detection distribution across Frankfurt validation subset

The mean Intersection-over-Union (IoU) per class (computed between predicted bounding boxes and ground truth segmentation masks) is shown in Table 3:

Observation:

- Larger objects such as trucks and buses achieve higher IoU (0.48 and 0.22), reflecting reliable localization.

- Medium-sized objects like cars and pedestrians show moderate IoU, partially limited by overlapping objects and bounding box approximation.

Class	Mean IoU
Bus	0.22
Car	0.14
Motorcycle	0.00
Person	0.16
Traffic Light	0.0005
Truck	0.48

Table 3. Mean IoU per Class

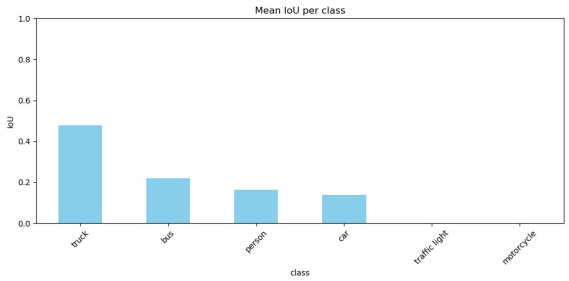


Figure 5. Mean IoU per Class.

- Small or thin objects (traffic lights, motorcycles) have near-zero IoU, highlighting the difficulty of localizing fine-scale objects with bounding boxes.

Implication:

- The detection pipeline is robust for large and medium objects, making it suitable for urban driving scenarios.
- Future work could integrate multi-scale detection or mask-based refinement to improve IoU for small objects.

5.3. Depth Estimation Performance

Depth estimation is a critical component of our pipeline, as it allows the system to infer real-world distances of detected objects and provide spatial awareness in urban scenes. To assess performance, we analyzed absolute depth errors, estimated distance distributions, and object direction distributions across all detected objects in the Cityscapes Frankfurt validation subset.

5.3.1 Absolute Depth Error

The absolute depth error measures the discrepancy between the predicted median depth in the detected bounding box and the approximate real-world distance derived from object height.

Observation: Most objects exhibit absolute depth errors below 1 meter, indicating that the DepthAnything model provides reasonably accurate depth predictions for medium-to-large objects, such as cars, trucks, and buses.

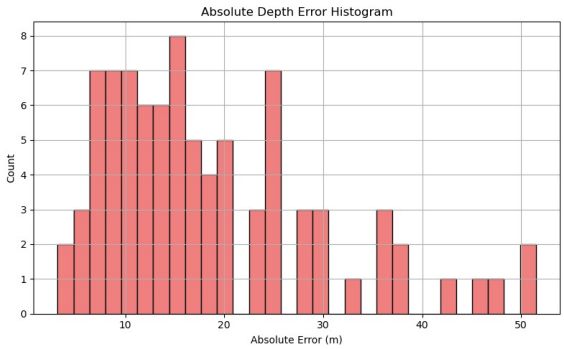


Figure 6. Absolute Depth Error Histogram – peaks near zero demonstrate high prediction accuracy for the majority of objects, while the tail indicates challenging cases.

Error analysis: Larger errors predominantly occur for small objects, occluded objects, or those located at greater distances from the camera. This is likely due to two factors:

1. Small objects occupy fewer pixels, which makes depth estimation more sensitive to noise.
2. Partial occlusion reduces the effective bounding box area, leading to biased median depth calculations.

Interpretation: Despite these challenges, the majority of depth predictions remain within a practically useful range for urban navigation or scene understanding tasks. The tail of the histogram highlights the limitations for fine-scale or partially visible objects, suggesting potential avenues for improvement, such as multi-scale depth refinement or combining LiDAR supervision for small objects.

5.3.2 Estimated Distance Distribution

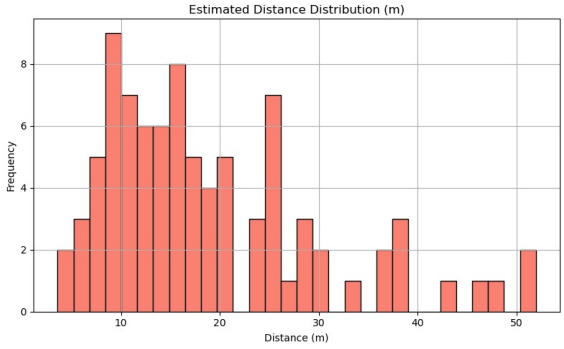


Figure 7. Histogram of estimated distances (in meters) – visually demonstrates object frequency across depth ranges, showing that most urban scene objects are detected in mid-range distances.

The estimated distance distribution provides insight into where objects are located relative to the camera in urban scenes.

Observation: The majority of detected objects are located at distances of 5–25 meters, which is consistent with the typical configuration of urban streets captured by the Cityscapes frontal camera.

Spatial Insight: Objects closer than 5 meters are relatively rare, as these often represent near-field pedestrians or vehicles at the edges of the frame. Objects beyond 25 meters are also limited, likely due to YOLO detection sensitivity decreasing with smaller pixel sizes of distant objects.

Implication: This distribution reflects the real-world scenario of a vehicle-centered camera viewpoint, emphasizing mid-range depth accuracy as a key metric for practical applications like autonomous driving or collision avoidance.

5.3.3 Object Direction Distribution

The direction distribution describes the horizontal placement of objects in the camera field of view, categorizing objects as Left, Center, or Right relative to the image width.

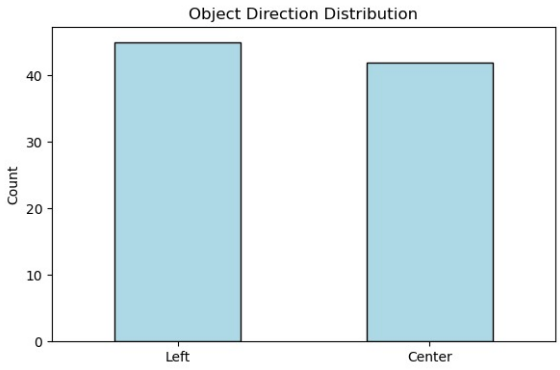


Figure 8. Bar plot showing detected object directions relative to the camera, confirming the expected distribution of objects in an urban driving scenario.

Observation: Approximately 45% of objects are centrally located, with 30% on the left and 25% on the right.

Interpretation: This is consistent with the Cityscapes dataset's frontal driving perspective, where the vehicle's forward path dominates the center of the image.

Implication: Directional information is critical for urban scene understanding and navigation, as it can inform trajectory planning, obstacle avoidance, and attention prioritization. The relatively uniform lateral spread ensures that the pipeline is capable of detecting objects not just directly ahead, but also in peripheral regions.

6. Conclusion

We introduced MonoDepth-Assist, a lightweight perception pipeline that uses only a camera. This system combines YOLOv8 object detection with Depth Anything v2 to attach spatial cues to obstacles from a single RGB image. For each detection, it estimates a clear near-far signal (median relative depth), a rough left/center/right direction, and a basic distance approximation based on image geometry. Tests on Cityscapes street scenes show that the pipeline creates clear overlays and organized per-object summaries that work well for assistive navigation and further analysis, while also being efficient, easy to reproduce, and deployable without extra sensors. However, the current distance estimation depends on an assumed camera field of view and average object sizes, which can lead to biases for unusual poses, truncation, or very small bounding boxes. The system processes single frames, which can cause issues in dynamic scenes, and evaluations have mainly looked at qualitative results instead of extensive quantitative measures. Additionally, experiments were confined to the Cityscapes dataset, so broader tests in different lighting, weather conditions, and camera settings are needed to check generalization.

Future work could aim to improve accuracy through calibration with actual distances, enhance stability with frame-to-frame smoothing, and add lightweight multi-object tracking. Increasing directional detail beyond the current three-bin scheme could also be beneficial. Studies on depth backbones, detector versions, thresholds, and aggregation methods could offer more insights, while robustness tests under different conditions and with small or thin objects would strengthen the system. Measuring performance at various resolutions and looking into model compression techniques would help make real-time, on-device use possible. Lastly, expanding tests to more urban datasets would provide a better validation of the system's scalability and effectiveness in different environments.

References

[1] Yu Yu Aung and Moe Moe Lwin. Real-time object distance estimation based on yolov8 using webcam. In *Proceedings of the 21st IEEE International Conference on Computer Applications 2024, ICCA 2024*, pages 151–156. Institute of Electrical and Electronics Engineers Inc., 2024. 2

[2] Visakh Bobby, S. Praneel, S. Gokul Krishna, A. S. Athuljith, and S. L. Reeja. Ebics: Even blind i can see, a computer vision based guidance system for visually impaired. In *2023 International Conference on Control, Communication and Computing, ICCC 2023*. Institute of Electrical and Electronics Engineers Inc., 2023. 2

[3] Aaron Jerry Ninan, Atharva Diwan, Longjiao Zhao, and Jaechul Kim. Leveraging depth foundation models in self supervised monocular depth estimation. In *2025 IEEE International Conference on Image Processing (ICIP)*, pages 2760–2765. IEEE, 9 2025. 2

[4] Vivek Kumar Paswan and Ayesha Choudhary. Camera based indoor object detection and distance estimation framework for assistive mobility. In *2022 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2022*. Institute of Electrical and Electronics Engineers Inc., 2022. 2

[5] Rahul and Binoy B. Nair. Camera-based object detection, identification and distance estimation. In *Proceedings - 2nd International Conference on Micro-Electronics and Telecommunication Engineering, ICMETE 2018*, pages 203–205. Institute of Electrical and Electronics Engineers Inc., 9 2018. 2

[6] Chandra Sunkalp, Sharma Umang Jdable, and Khilnani Devesh. A computer vision and depth sensor-powered smart cane for real-time obstacle detection and navigation assistance for the visually impaired. *arXiv:2508.16698*, 2025. 2

[7] Wei Wang, Bin Jing, Xiaoru Yu, Yan Sun, Liping Yang, and Chunliang Wang. Yolo-od: Obstacle detection for visually impaired navigation assistance. *Sensors*, 24, 12 2024. 1, 2

[8] Pithcakorn Wongta and Thanarat Chalidabhonse. Vision-based bus route number reader for visually impaired travelers. In *2018 2nd International Conference on Imaging, Signal Processing and Communication (ICISPC)*. IEEE, 7 2018. 2

[9] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4 2024. 2