sensors

MDPI

Article

Scene-Aware Perception Routing: An Efficient On-Device Framework for Walking Assistance of the Visually Impaired

Hyojeong Kwon, Taein Kim, Shihyun Lim and Hyung Seok Kim *

Department of Artificial Intelligence and Robotics, Sejong University, Seoul 05006, Republic of Korea; kwhj1106@naver.com (H.K.); zero0164@naver.com (T.K.); shihyunlim1217@naver.com (S.L.)
* Correspondence: hyungkim@sejong.ac.kr

Highlights

What are the main findings?

- A temporally stabilized scene classifier that gates semantic segmentation and selects scene-specific detectors reduced mean end-to-end latency by 21.6% (classifier every fifth frame, 30.7% indoor frames) and steady-state energy per frame by 15.9% (classifier on every frame) relative to a fixed perception pipeline on an INT8 neural processing unit.
- Scene-specific detectors improved FP32 mAP@50:95 by 3.7 and 9.6 percentage points outdoors and indoors, respectively, without measurable latency cost under routing.

What are the implications of the main findings?

- Pipeline-level routing can complement model-level compression for wearable assistive perception on edge devices.
- Routed perception cues supported local voice guidance in controlled walking trials, with 93.23% of decisions matching the reference command and 54 of 55 obstacles avoided.

Abstract

Vision-based walking assistance for visually impaired individuals requires efficient perception across indoor and outdoor environments. However, fixed perception pipelines may incur unnecessary computational overhead by executing modules that are not required in the current scene. This paper presents Scene-Aware Perception Routing (SAPER), an on-device framework that dynamically configures perception modules according to environmental context. A temporally stabilized scene classifier selects scene-specific object detectors and controls the execution of semantic segmentation, while monocular depth estimation remains active in both environments. The framework was deployed on an edge neural processing unit (NPU) using INT8 quantization and evaluated through controlled benchmarks and real-world system tests. The deployed scene classifier achieved 95.61% accuracy on a balanced test set of 1800 images. For comparison, the fixed perception pipeline used in existing walking-assistance systems was implemented on the same NPU and evaluated on the same data. With the scene classifier executed every fifth frame, SAPER reduced the mean end-to-end latency by 21.6% relative to this baseline on a benchmark containing 30.7% indoor frames and increased the sustained throughput by 11.7%. With the classifier executed on every frame, steady-state energy per frame decreased by 15.9%. In ten indoor–outdoor transition runs, unsafe outdoor routing exposure was 3.2% and 5.4% for indoor-to-outdoor and outdoor-to-indoor transitions, respectively. In a supplementary navigation-decision evaluation, SAPER achieved 96.0% command

Academic Editor: Hwan-Sik Yoon

Received: 20 October 2026
Revised: 21 November 2026
Accepted: 27 November 2026
Published: 29 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

accuracy, compared with 92.8% for the fixed-pipeline baseline. These results demonstrate the computational benefits of scene-aware perception routing without an observed degradation in navigation-decision accuracy under the evaluated conditions.

Keywords: scene-aware perception routing; on-device artificial intelligence; neural processing unit; walking assistance; visual impairment; object detection; semantic segmentation; monocular depth estimation

1. Introduction

Visual impairment can restrict access to environmental information needed for obstacle avoidance and navigation, particularly when moving between indoor and outdoor environments. The World Health Organization estimates that at least 2.2 billion people worldwide have near or distance vision impairment [1]. Conventional mobility aids, including white canes, guide dogs, and tactile paving, remain essential for independent travel but provide limited information about the categories and spatial arrangements of surrounding objects [2]. Vision-based wearable systems can complement these aids by providing information about obstacles and environmental conditions [3,4]. Smart glasses are particularly suitable for this application because they offer a forward-facing camera perspective and hands-free operation [5,6].

Recent walking-assistance systems have incorporated object detection, semantic segmentation, and monocular depth estimation to represent different aspects of the surrounding environment. Object detection identifies discrete objects, such as pedestrians, vehicles, furniture, and doors, whereas monocular depth estimation provides relative proximity information from RGB images. Semantic segmentation complements these outputs by identifying continuous regions, including sidewalks, roads, crosswalks, and other walkable surfaces. Combining these tasks enables the system to represent both object-level hazards and the spatial structure of the environment.

However, executing multiple perception modules continuously can impose substantial computational overhead on resource-constrained wearable devices. Semantic segmentation, in particular, requires dense pixel-level predictions and can introduce considerable latency and memory demands [7]. These costs are important for battery-powered edge systems that must process camera input continuously while maintaining responsive operation [8,9]. A fixed perception pipeline that executes the same modules in every scene may therefore perform computations that are unnecessary for the current environment.

Previous studies have addressed the computational constraints of wearable perception through lightweight neural architectures and model optimization [3,10]. More broadly, adaptive inference methods, including conditional computation, dynamic neural networks, early-exit architectures, and mixture-of-experts models, adjust computational execution according to the input [11–16]. Such approaches demonstrate the benefits of selective computation, but they commonly focus on computational paths within a model or expert selection for a shared prediction task. Walking-assistance systems present a related but distinct requirement: object detection, semantic segmentation, and depth estimation produce heterogeneous outputs that may not be equally necessary in every environment. Consequently, reducing the cost of individual models alone does not fully address the selection of perception tasks at the pipeline level.

The information required for walking assistance also varies with environmental context. Outdoor scenes often require region-level information about sidewalks, roads, crosswalks, and walkable areas, for which semantic segmentation provides useful spatial cues [17,18]. Indoor environments, meanwhile, frequently involve discrete obstacles and

structures, including chairs, doors, tables, and stairs, that can be represented through object detection and relative depth estimation [19–21].

Existing walking-assistance systems combine multiple perception tasks, but the potential benefits of using scene information to control module execution remain insufficiently evaluated [17–19,22]. In particular, the effects of selecting scene-specific detectors and conditionally executing semantic segmentation on perception accuracy and computational efficiency require evaluation on resource-constrained edge hardware.

To address this problem, this study proposes Scene-Aware Perception Routing (SAPER), an on-device framework that configures perception modules according to the indoor–outdoor context. A lightweight scene classifier, combined with temporal stabilization, controls the selection of scene-specific object detectors and the execution of semantic segmentation. In outdoor mode, SAPER executes object detection, semantic segmentation, and monocular depth estimation to provide object-level, surface-region, and relative-proximity cues. In indoor mode, it executes object detection and monocular depth estimation while omitting the segmentation branch to reduce computational overhead. The proposed approach adapts the composition of the perception pipeline without modifying the underlying prediction architectures.

SAPER is implemented on an edge neural processing unit (NPU) using 8-bit integer (INT8) quantization. The evaluation examines perception accuracy, the effects of scene-specialized detection and selective execution, routing stability, and on-device computational efficiency. Real-world system operation is assessed using continuous environmental input, while a supplementary navigation-decision evaluation examines command accuracy and potentially unsafe guidance outputs.

The main contributions of this study are as follows:

- Scene-aware perception routing: A temporally stabilized indoor–outdoor scene classifier determines the active scene-specific object detectors and whether semantic segmentation is performed, with segmentation enabled in OUTDOOR_MODE and BOTH_MODE. On an INT8 NPU, SAPER reduced the mean latency by 21.6% (classifier executed every fifth frame) and the steady-state energy per frame by 15.9% (classifier executed on every frame) compared with the fixed pipeline.
- Scene-specific detection with negligible overhead: Compared with a unified detector, scene-specific detectors improved mAP@50:95 by 3.7 and 9.6 percentage points outdoors and indoors, respectively. Because SAPER typically executes only the detector for the current scene under stable routing conditions, its latency remained nearly identical to that of using a single unified detector (220.81 ms vs. 220.05 ms), whereas executing both detectors in a fixed pipeline increased latency by 17.6%.
- Navigation performance under scene-aware routing: These efficiency gains were obtained without loss of navigation-decision accuracy. SAPER achieved 96.0% command accuracy, compared with 92.8–94.0% for the fixed pipelines, and in real-world walking trials, 93.23% of its commands matched the reference decisions, with 54 of 55 obstacles avoided.

The remainder of this paper is organized as follows. Section 2 reviews related work on walking-assistance perception, on-device inference, and adaptive computation. Section 3 describes the proposed SAPER framework and its implementation. Section 4 presents the datasets, training procedures, and experimental protocols. Section 5 reports the experimental results. Section 6 discusses the findings and limitations, and Section 7 concludes the paper.

2. Related Work

2.1. Object Detection and Depth Estimation for Walking Assistance

Object detection is widely used in vision-based walking-assistance systems to identify obstacles and provide object-level information about the surrounding environment. Ashiq et al. [23] presented a convolutional neural network-based framework for recognizing and tracking potentially hazardous objects. Wang et al. [3] developed YOLO-OD to improve obstacle detection for visually impaired navigation, while Kumari and Hammady [5] investigated YOLOv8 variants for object recognition in smart-glasses-based assistance. These studies illustrate the role of object detection in identifying pedestrians, vehicles, and other objects relevant to walking assistance.

Object detection has also been combined with distance estimation to provide spatial information about detected obstacles. Davanthapuram et al. [24] integrated YOLO-based object recognition, monocular depth estimation, and binaural audio feedback for indoor navigation. Seo et al. [25] combined YOLOv11 with MiDaS-based depth estimation in a smart-vest system. Other studies have investigated indoor object detection and distance estimation [20], spatial awareness for daily activities [21], and distance-based obstacle alerts with auditory feedback [26].

These approaches provide complementary object-category and proximity information, making them useful for object-centered obstacle recognition. However, object detection and depth estimation alone may not adequately represent continuous spatial regions, such as sidewalks, crosswalks, and walkable areas. Semantic segmentation can complement these methods by providing region-level information, particularly in outdoor environments.

2.2. Semantic Segmentation for Spatial Understanding

Object detection represents discrete obstacles but does not directly describe the full spatial extent of continuous surfaces. Semantic segmentation addresses this limitation by assigning semantic labels to image regions, allowing walking-assistance systems to distinguish sidewalks, roads, crosswalks, and other walkable areas.

Hsieh et al. [17] combined semantic segmentation and depth information in a wearable system for outdoor walking assistance. Chen et al. [27] investigated semantic visual SLAM for wearable navigation, integrating semantic information with spatial mapping. Mao et al. [28] developed a perception approach based on panoptic segmentation to represent both object instances and background regions. These studies highlight the value of region-level representations for understanding walking environments.

However, dense pixel-level prediction introduces additional computational demands. Lightweight segmentation architectures have therefore been explored to improve the balance between accuracy and inference efficiency. Lei et al. [10] proposed a compact semantic segmentation network for assistive navigation, while broader studies have examined the accuracy–efficiency trade-offs of real-time segmentation models [7]. Such approaches reduce the computational cost of segmentation itself. In contrast, SAPER employs scene-dependent execution rules to selectively activate semantic segmentation within a multi-module perception pipeline.

2.3. Wearable and On-Device Visual Perception Systems

Wearable visual-assistance systems increasingly integrate multiple perception functions to provide environmental information without relying on continuous cloud connectivity. Mukhiddinov and Cho [6] developed a smart-glasses-based system incorporating object recognition and other visual-assistance functions. Kumari and Hammady [5] investigated object detection, distance estimation, optical character recognition, and audio

feedback on smart glasses. Kan et al. [29] introduced eLabrador, a wearable navigation system for visually impaired users.

Edge-based implementations have also been studied to support local inference. Mahendran et al. [4] presented a visual-assistance system based on mobile edge artificial intelligence. Hsieh et al. [17] and Zhang et al. [18] investigated embedded implementations of perception functions for wearable assistance. These studies demonstrate the feasibility of incorporating deep learning into portable assistive platforms.

Nevertheless, integrating multiple functions does not necessarily establish how their execution should be scheduled under limited computational resources. Model-level optimization and embedded implementation address important deployment constraints, whereas scene-dependent module selection introduces an additional design decision. The present study examines this decision through a controlled comparison of fixed and scene-aware perception configurations on an INT8 NPU platform.

2.4. Adaptive Inference and Dynamic Computation

Adaptive inference aims to reduce unnecessary computation by adjusting model execution according to input characteristics. Conditional computation activates selected network components [11], while dynamic neural networks employ input-dependent execution paths [12]. Representative methods include SkipNet [13] and BlockDrop [14], which selectively execute network blocks, and BranchyNet [15], which enables inference to terminate at intermediate exits. Mixture-of-experts architectures use gating mechanisms to select subsets of specialized components [16].

These approaches establish the general principle that not all computations need to be performed for every input. Their adaptation mechanisms, however, commonly operate within a model or among experts associated with a shared prediction objective.

Adaptation has also been studied at the system level. Video analytics systems such as Chameleon [30] adjust configurations, including frame rate, input resolution, and detection model, to balance accuracy against resource use, and LiteReconfig [31] reconfigures object-detection pipelines on mobile GPUs according to content characteristics and execution cost. These systems tune the accuracy–cost operating point of a given task.

Multi-task perception for walking assistance presents a complementary problem because object detection, semantic segmentation, and depth estimation produce different types of information. SAPER therefore applies selective computation at the perception-pipeline level. Rather than modifying the internal architecture of an individual model or tuning the operating point of a single task, it uses temporally stabilized scene classification to select scene-specific object detectors, thereby changing the object-class vocabulary, and to control semantic segmentation, while maintaining monocular depth estimation in both environments. The routing signal reflects which navigation information the current environment requires, and because misrouting has asymmetric safety consequences in assistive navigation, uncertain predictions are resolved toward the fuller perception configuration (BOTH_MODE).

2.5. Multi-Task Perception and Comparative Analysis

Multi-task walking-assistance systems combine complementary perception outputs to represent obstacles, spatial structure, and environmental context. Hsieh et al. [17] integrated semantic segmentation with depth information, while Zhang et al. [18] combined real-time semantic segmentation and scene recognition in a wearable perception framework. Liu et al. [19] investigated holistic indoor understanding using wearable LiDAR and semantic instance segmentation. Ou et al. [22] integrated RGB-D sensing, dynamic SLAM, and panoptic segmentation for indoor navigation assistance.

These approaches differ in sensing modalities, target environments, and perception outputs. Some emphasize rich spatial representations, whereas others prioritize wearable implementation or computationally efficient individual models. SAPER addresses a complementary system-level question: how can environmental context determine which perception tasks and scene-specific detectors should be executed on a resource-constrained edge device?

Table 1 compares representative studies in terms of perception tasks, scene-information usage, pipeline configuration, and deployment platform. These systems execute a predefined set of perception modules regardless of the scene, whereas SAPER uses scene classification to determine which modules are executed. Because these systems were developed with different sensors, datasets, and hardware platforms, their reported results cannot be compared directly. Therefore, their common fixed-pipeline design was implemented on the same NPU and evaluated on the same data to serve as the baseline for quantitative comparison (Section 4.5). Accordingly, the contribution of this study lies in the scene-aware selection of perception modules rather than in the development of a new perception model.

Table 1. Comparison of existing assistive perception systems and the proposed framework.

Reference	Perception Tasks	Scene Use	Pipeline	Platform
Kumari et al. [5]	Det. + Dist.	–	Fixed	Smart glasses
Hsieh et al. [17]	Seg. + Dep.	–	Fixed	Wearable/Jetson
Lei et al. [10]	Seg.	–	Fixed	Mobile
Davanthapuram et al. [24]	Det. + Mono-Dep.	–	Fixed	Indoor
Zhang et al. [18]	Seg. + Scene Rec.	Output	Fixed	Wearable
Ou et al. [22]	Panoptic Seg. + SLAM + RGB-D	–	Fixed	RGB-D
Chen et al. [27]	Semantic visual SLAM	Semantic map	Fixed	Wearable
Liu et al. [19]	3D Inst. Seg.	–	Fixed	LiDAR
Proposed	Outdoor: Det. + Seg. + Dep. Indoor: Det. + Dep.	Routing	Adaptive	NPU edge

Det., Seg., Dep., Mono-Dep., Scene Rec., and Inst. Seg. denote object detection, semantic segmentation, depth estimation, monocular depth estimation, scene recognition, and instance segmentation, respectively. "Output" indicates the use of scene information as a perception result, whereas "Routing" indicates its use to control perception-module execution. "Fixed" and "Adaptive" refer to predefined and scene-dependent pipeline configurations, respectively.

3. Materials and Methods

3.1. Overall System Architecture

The proposed SAPER framework dynamically configures perception modules according to the indoor–outdoor context. As illustrated in Figure 1, the framework comprises an RGB image acquisition module, a scene classification and routing module, scene-specific object detectors, a semantic segmentation module, a monocular depth estimation module, and a navigation cue construction module.

An input RGB frame is processed by a lightweight scene classifier, whose output is used to determine the perception configuration through temporally stabilized routing. In outdoor mode, SAPER activates the outdoor object detector, semantic segmentation, and monocular depth estimation. In indoor mode, it activates the indoor object detector and monocular depth estimation while disabling semantic segmentation. When scene classification is uncertain, BOTH_MODE activates both scene-specific object detectors, semantic segmentation, and monocular depth estimation. The detection results are combined using class-wise NMS to remove duplicate detections. Thus, the routing mechanism controls

both the selection of scene-specific object detectors and the execution of the segmentation branch.

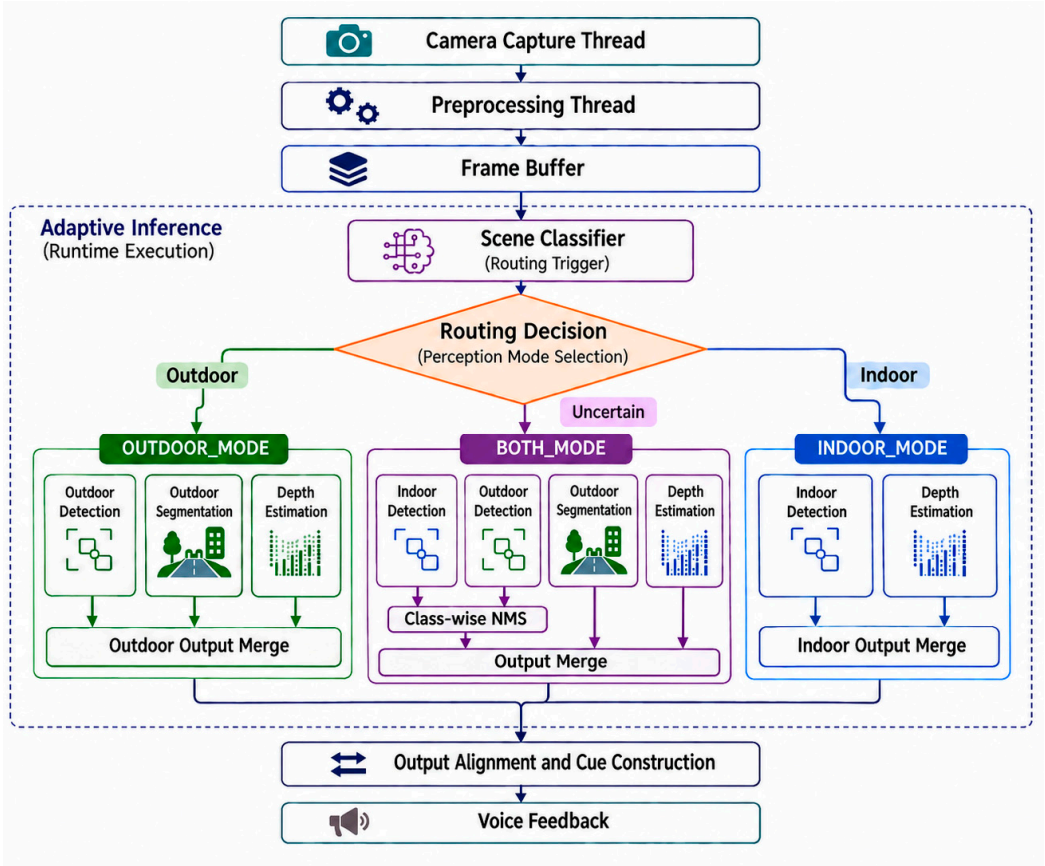


Figure 1. Overall architecture and workflow of the proposed scene-aware perception routing (SA-PER) framework. Green, purple, and blue branches represent OUTDOOR_MODE, BOTH_MODE, and INDOOR_MODE, respectively. Arrows indicate the processing flow.

The outputs of the activated modules are subsequently aligned and integrated into scene-dependent navigation cues. Object detection provides class labels, bounding boxes, and confidence scores, while monocular depth estimation provides relative proximity information. In outdoor mode, semantic segmentation additionally supplies surface-region information. The complete framework is deployed on an INT8 NPU platform to enable on-device perception without cloud-based inference.

3.2. Scene-Aware Routing and Pipeline Selection

The scene classification module uses MobileNetV3-Small to estimate indoor and outdoor probabilities from RGB frames. The classifier is executed on every fifth frame captured by the 30 fps camera, independently of the perception processing rate, so the routing state is updated at approximately 6 Hz. Because the classifier requires only 1.42 ms of NPU inference time (Section 5.5.4), this update rate adds little computational overhead. To reduce fluctuations in routing decisions, an exponential moving average (EMA) with a time constant of 1.5 s is applied to the classification probabilities. All routing decisions are based on the smoothed probabilities.

The routing mechanism incorporates confidence-based fallback, hysteresis, temporal consistency, and cooldown constraints. When the maximum smoothed probability falls below 0.60, the system immediately activates BOTH_MODE, which executes both indoor and outdoor object detectors, semantic segmentation, and monocular depth estimation. The detection results from both detectors are combined, and class-wise non-maximum

suppression (NMS) is applied to remove duplicate detections. This configuration maintains object detection coverage for both environments during uncertain scene classification, reducing the risk of missing scene-specific obstacles due to incorrect detector selection. Otherwise, a transition between stable indoor and outdoor modes requires the opposing probability to exceed 0.75 for three consecutive routing updates. After a transition, a cooldown of five routing updates prevents additional stable-mode switching. The routing parameters (EMA time constant, confidence threshold, hysteresis threshold, temporal-consistency window, and cooldown) were selected using separate walking videos recorded by the authors on the Sejong University campus, which were not used in any of the evaluations reported in Section 5.

Accordingly, SAPER operates in three routing states: INDOOR_MODE, OUTDOOR_MODE, and BOTH_MODE. Their corresponding perception configurations are summarized in Table 2, and the routing procedure is presented in Algorithm 1.

Table 2. Scene-dependent perception module selection in the proposed framework.

Mode	Modules	Main Navigation Cue
OUTDOOR_MODE	Outdoor Det. + Seg. + Dep.	Hazardous objects, walkable areas, spatial structures, and relative distance
INDOOR_MODE	Indoor Det. + Dep.	Object-centered obstacles and object-wise relative distance
BOTH_MODE	Indoor Det. + Outdoor Det. + Seg. + Dep.	Indoor and outdoor object cues, walkable areas, and relative distance

Abbreviations are defined in Table 1. The BOTH_MODE state, activated when scene classification is uncertain, executes both indoor and outdoor object detectors, together with semantic segmentation and monocular depth estimation.

Algorithm 1. Environment-adaptive perception routing

Input: RGB frame I
Output: Navigation cue C

$(p_{indoor}, p_{outdoor}) \leftarrow$ SceneClassifier(I)
$(p_{indoor}, p_{outdoor}) \leftarrow$ EMA($p_{indoor}, p_{outdoor}, \tau$ = 1.5 s)
mode $\leftarrow$ StabilizedRouting($p_{indoor}, p_{outdoor}$)
$D \leftarrow$ MonocularDepthEstimation(I)5: **if** mode = BOTH_MODE **then**
    $B_{indoor} \leftarrow$ IndoorObjectDetection(I)
    $B_{outdoor} \leftarrow$ OutdoorObjectDetection(I)
    $B \leftarrow$ ClassWiseNMS ($B_{indoor} \cup B_{outdoor}$)
    $S \leftarrow$ SemanticSegmentation(I)
else if mode = OUTDOOR_MODE **then**
    $B \leftarrow$ OutdoorObjectDetection(I)
    $S \leftarrow$ SemanticSegmentation(I)
else
    $B \leftarrow$ IndoorObjectDetection(I)
    $S \leftarrow \emptyset$
end if
for each bounding box $b_i \in B$ **do**
    $r_i \leftarrow$ MapToDepthCoordinates(b_i)
    $d_i \leftarrow$ median$\{D(x, y) \mid (x, y) \in r_i\}$
end for
$C \leftarrow \{B, \{d_i\}\}$

22: **if** $S \neq \emptyset$ **then**
23: $C \leftarrow C \cup \{S\}$
24: **end if**
25: **return** C

In Algorithm 1, I, B_{indoor}, $B_{outdoor}$, B, D, and S denote the input RGB frame, indoor object detection results, outdoor object detection results, final detection results, relative depth map, and segmentation map, respectively. In BOTH_MODE, B is obtained by combining the two detectors' outputs and applying class-wise NMS. The variable b_i represents the i-th bounding box in B, while r_i denotes its corresponding region in the depth map. The object-wise relative depth cue d_i is computed from the corresponding region of D.

3.3. Perception Module Configuration

The SAPER framework comprises four perception components: scene-specific object detection, semantic segmentation, monocular depth estimation, and scene classification. Each component performs a distinct perception task, and the routing mechanism described in Section 3.2 determines the active perception configuration. Table 3 summarizes the model architectures, input resolutions, outputs, and functional roles of the perception modules.

Table 3. Specifications and roles of perception models in the proposed system.

Module	Model/Architecture	Input Size	Output	Role
Scene cls.	MobileNetV3-Small	224 × 224	Indoor/Outdoor	Routing trigger
Outdoor det.	YOLOv8m	640 × 640	5-class boxes	Outdoor object cue
Indoor det.	YOLOv8m	640 × 640	6-class boxes	Indoor object cue
Seg.	ResNet-34 + FPN	640 × 640	7-class semantic map	Outdoor surface cue
Depth est.	MiDaS v2.1 Small	256 × 256	Relative depth map	Relative proximity cue

cls., det., seg., and est. denote classification, detection, segmentation, and estimation, respectively.

3.3.1. Scene-Specific Object Detection

SAPER employs two YOLOv8m-based object detectors specialized for indoor and outdoor environments. The outdoor detector recognizes five object classes: Car, Person, Pole, Tree, and Bollard. The indoor detector recognizes six classes: Chair, Door, Stairs, Table, Column, and Person.

The detection classes were selected based on their relevance to pedestrian safety and their occurrence in everyday indoor and outdoor environments. Outdoor classes prioritize potentially hazardous objects, including vehicles and pedestrians, as well as static obstacles such as poles, trees, and bollards that may obstruct walking paths. Indoor classes focus on commonly encountered furniture and architectural structures, including chairs, doors, stairs, tables, and columns. Person was included in both detectors because pedestrians may be encountered in either environment.

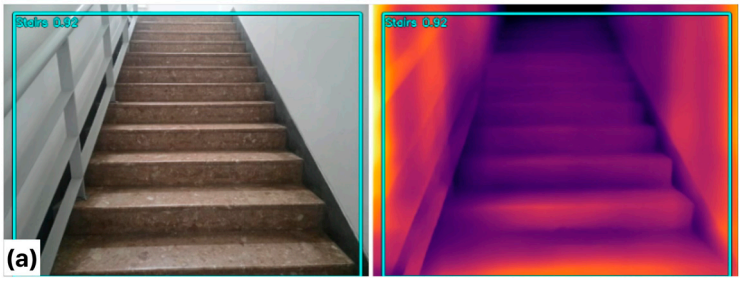
YOLOv8m was selected on the basis of its detection performance, compatibility with the target NPU, and deployment consistency. Using the same YOLOv8m architecture for both scene-specific detectors also enables controlled comparisons while maintaining a consistent model conversion and deployment procedure. Although recent YOLOv10-based studies have improved small-object detection through multi-scale feature enhancement [32,33], this study focuses on scene-aware routing rather than detector architecture optimization. Such multi-scale enhancements could address the relatively low recall observed for small indoor objects and narrow outdoor obstacles, such as bollards and poles

(Section 5.2). However, they may increase computational cost or introduce operators that require additional validation for INT8 conversion on the target NPU, and their benefits in walking-assistance scenes were not evaluated in this study. Integrating such detectors into the scene-specific branches of SAPER is therefore left for future work.

3.3.2. Semantic Segmentation

The semantic segmentation module uses a ResNet-34-based encoder and a feature pyramid network (FPN) to generate pixel-wise semantic predictions [34,35]. The network processes RGB images at a resolution of 640 × 640 pixels and provides region-level information about outdoor spatial structures, including roads, sidewalks, and walkable areas. Multi-scale feature fusion enables the model to represent regions with different spatial extents. The model predicts seven outdoor semantic classes: Background, Sidewalk, Braille Guiding Block, Roadway/Alley, Crosswalk, Bike Lane, and Caution Zone. The resulting pixel-wise segmentation map provides continuous surface-region information that complements object-level detection results in the outdoor perception pipeline.

Semantic segmentation is activated in OUTDOOR_MODE and BOTH_MODE, where safe walking depends on distinguishing continuous surface regions such as sidewalks, roadways, and crosswalks. Indoor floors, in contrast, typically form continuous walkable surfaces without such region distinctions, and the main indoor hazards are discrete objects and structures. These hazards are addressed by the indoor detector and relative depth estimation. Large elevation changes, such as staircases, are detected as a dedicated Stairs class. As with other detected objects, a staircase that lies close ahead within the central assessment region is treated as a frontal obstacle by the decision rules in Section 3.4, which then generate an avoidance or STOP command. As shown in Figure 2a,b, staircases were detected with high confidence from both frontal and oblique viewpoints, and the relative depth maps indicate that they lie directly ahead of the user. Smaller elevation changes are also reflected in the depth representation: as shown in Figure 2c, a low step in a corridor, which is not represented by any detection class, appears as a distinct discontinuity in relative depth across the walking path. Although elevation changes may appear as discontinuities in the relative depth map (Figure 2c), the current navigation algorithm does not explicitly analyze these discontinuities. Indoor semantic segmentation is omitted to reduce computational overhead, as the proposed framework primarily focuses on object-level obstacle avoidance. Lightweight segmentation architectures reduce the cost of each segmentation inference, whereas SAPER avoids executing segmentation when it is unnecessary; thus, the two approaches are complementary.



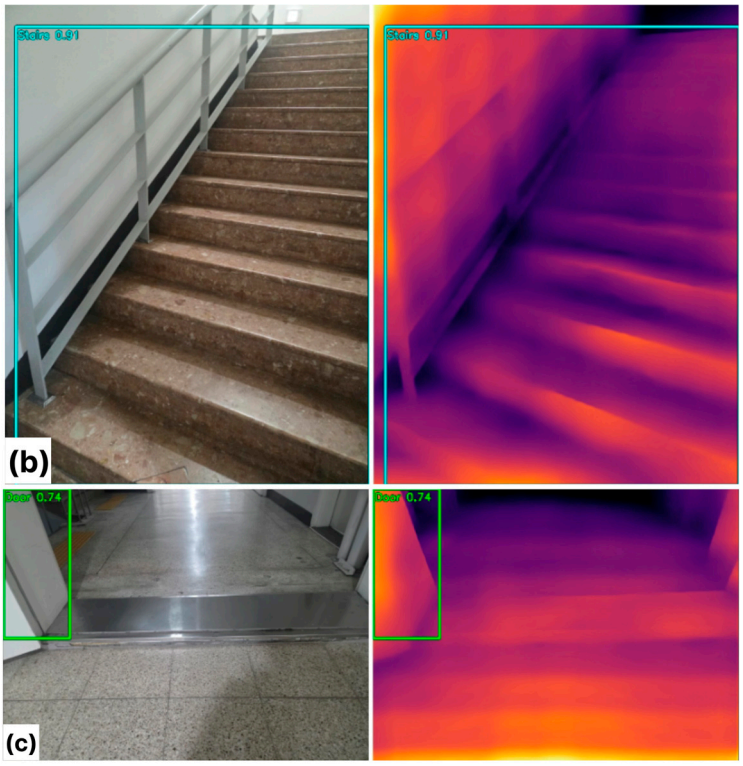


Figure 2. Examples of indoor elevation changes captured by object detection and relative depth estimation without semantic segmentation: (**a**) frontal view and (**b**) oblique view of staircases, and (**c**) a low step in a corridor. Each panel shows the detection result (left) and the corresponding relative depth map (right), in which brighter colors indicate closer regions and darker colors indicate farther regions.

3.3.3. Monocular Depth Estimation

SAPER employs MiDaS v2.1 Small to estimate relative depth from RGB images [36]. The model processes 256 × 256 inputs and produces a depth representation used to compare the relative proximity of detected objects. Unlike metric depth estimation, the resulting values are not interpreted as physical distances in meters.

The depth estimation module remains active in all three routing states. Its output is spatially aligned with object detection results, and representative depth values are extracted from the corresponding bounding-box regions. MiDaS v2.1 Small was selected for its deployment compatibility and low computational cost, allowing relative depth estimation to remain active across both perception configurations.

3.3.4. Scene Classification

The scene classification module employs MobileNetV3-Small with an input resolution of 224 × 224 pixels to distinguish indoor and outdoor environments [37]. The classifier outputs scene probabilities that are temporally smoothed and used by the routing mechanism described in Section 3.2.

MobileNetV3-Small was selected to limit the computational overhead introduced by scene classification. Unlike systems that provide scene recognition as an independent perception output, SAPER uses the classifier probabilities to control scene-specific detector selection and semantic segmentation execution.

3.4. Navigation Cue Construction

The outputs of the perception modules are post-processed and integrated to construct scene-dependent navigation cues. These cues represent intermediate environmental

information, including detected objects, their relative proximity, and outdoor surface regions, rather than final path-planning or navigation commands.

For object detection, bounding-box coordinates are obtained from the YOLOv8m outputs through distribution focal loss (DFL) decoding. Confidence filtering and non-maximum suppression (NMS) are subsequently applied to remove low-confidence and overlapping detections. Each retained detection contains an object class, a bounding box, and a confidence score.

The detection results are then aligned with the monocular depth map. Since object detection and depth estimation operate at different input resolutions, each detection bounding box is mapped to the depth-map coordinate system. The object-wise relative depth cue is computed as the median depth value within the corresponding region:

$$d_i = \text{median}\{D(x,y) \mid (x,y) \in r_i\}, \quad (1)$$

where D denotes the relative inverse-depth map, r_i represents the i-th bounding box mapped to the depth-map coordinates, and d_i is the corresponding object-wise relative depth cue. The median operation reduces sensitivity to local depth variations and outliers within the detected region. The resulting values are used to compare relative object proximity rather than to estimate absolute distances in meters.

The navigation cues are constructed according to the selected perception mode. In indoor mode, the cue combines object detection results and their relative depth values:

$$C_{\text{indoor}} = \{B, \{d_i\}\}, \quad (2)$$

whereas outdoor mode additionally incorporates the semantic segmentation map:

$$C_{\text{outdoor}} = \{B, \{d_i\}, S\}, \quad (3)$$

where B denotes the final detection results obtained according to the selected routing mode, and S denotes the semantic segmentation map. In BOTH_MODE, the indoor and outdoor detection results are combined using class-wise NMS to remove duplicate detections before extracting object-wise relative depth cues.

$$C_{\text{both}} = \{B, \{d_i\}, S\}, \quad (4)$$

These mode-specific outputs provide a unified representation of the available perception information for subsequent guidance processing.

Navigation Command Generation: A lightweight rule-based decision layer converts the navigation cues into four commands: FORWARD, LEFT, RIGHT, and STOP. The image is divided into left, center, and right regions. The central obstacle assessment region spans 34–66% of the image width and 45–90% of its height. To assess obstacle proximity, the inverse-depth map is normalized to [0, 1] using the minimum and maximum values within each frame:

$$D_n(x, y) = \frac{D(x, y) - D_{\min}}{D_{\max} - D_{\min}}, \quad (5)$$

where $D_{\min}$ and $D_{\max}$ denote the minimum and maximum inverse-depth values within the current frame, respectively. The larger normalized values indicate closer objects. A detected object is considered near when its normalized inverse-depth value reaches 0.60. The center region is considered obstructed when a near object with a detection score of at least 0.25 is identified within the central assessment region.

A lateral region is considered obstructed when its median inverse-depth value is at least 0.8 times the inverse-depth value of the frontal obstacle; in outdoor mode, a candidate direction must also contain at least 15% walkable area and no more than 25% hazardous area in the segmentation map. The complete decision procedure is summarized in Algorithm 2, and the selected command is delivered as voice guidance.

https://doi.org/10.3390/s26237548

Algorithm 2. Navigation command generation

Input: Detections B with confidence scores and object-wise depth cues; normalized inverse-depth map; segmentation map S

Output: command $\in$ {FORWARD, LEFT, RIGHT, STOP}

$F \leftarrow \{b_i \in B \mid \text{within the central region, score}(b_i) \geq 0.25, \text{depth}(b_i) \geq 0.60\}$

if $F = \emptyset$ **then** return FORWARD

$d_{max} \leftarrow \max\{\text{depth}(b_i) \mid b_i \in F\}$

for $k \in$ {LEFT, RIGHT}: $\text{free}_k \leftarrow \text{median}(D_n \text{ in region } k) < 0.8 \cdot d_{max}$, and, **if** $S \neq \emptyset$, $\text{walkable}_k \geq 0.15$ and $\text{hazardous}_k \leq 0.25$

if $\text{free}_{\text{LEFT}}$ and $\text{free}_{\text{RIGHT}}$ **then** return the direction with greater clearance

else if $\text{free}_{\text{LEFT}}$ **then** return LEFT

else if $\text{free}_{\text{RIGHT}}$ **then** return RIGHT

else return STOP

3.5. NPU-Based On-Device Implementation

The SAPER framework was implemented on an edge NPU platform to perform on-device inference without relying on cloud-based processing. The perception models were exported from PyTorch to the Open Neural Network Exchange (ONNX, v1.19.1) format [38] and subsequently converted into NPU-compatible MXQ models using the vendor-provided software development kit (Mobilint Compiler SDK v0.12.0.0). Hardware-specific optimizations and INT8 quantization were applied during deployment [39].

The inference pipeline employs a multithreaded architecture to coordinate image acquisition, preprocessing, scene classification, object detection, semantic segmentation, and depth estimation. The camera acquisition thread captures RGB frames, while preprocessing prepares the inputs according to the resolution and normalization requirements of each model. The scene classification results are processed by the routing mechanism described in Section 3.2 to determine the active perception configuration.

Based on the routing decision, the system executes the indoor detector in INDOOR_MODE, the outdoor detector in OUTDOOR_MODE, or both detectors in BOTH_MODE. In BOTH_MODE, the detection results are combined using class-wise NMS. Monocular depth estimation remains active in all three routing states, whereas semantic segmentation is executed only in OUTDOOR_MODE and BOTH_MODE. The multithreaded architecture separates image acquisition and CPU-side processing from NPU inference.

Following inference, the model outputs are post-processed and integrated into the navigation cues defined in Section 3.4. Detection results are converted into object classes, bounding boxes, and confidence scores, while relative depth values are extracted from the corresponding regions of the depth map. In outdoor mode, the segmentation output is additionally incorporated. The resulting perception information is then made available for subsequent guidance processing.

4. Experimental Setup and Evaluation Methods

4.1. Experimental Setup

Model training and FP32 evaluation were conducted on a GPU workstation equipped with an Intel Core i7-14700KF (Intel, Santa Clara, CA, USA) processor, an NVIDIA RTX 4070 SUPER (NVIDIA, Santa Clara, CA, USA) GPU with 12 GB of memory, and 32 GB of RAM. The workstation operated on Ubuntu 22.04 with PyTorch 2.9.1 and CUDA 12.8.

On-device experiments were performed using a Mobilint Regulus (Mobilint, Seoul, Republic of Korea) NPU platform running Linux kernel 5.15.125-1. The trained models

were converted into NPU-compatible executable formats using the vendor-provided SDK, as described in Section 3.5. The deployed framework was evaluated using INT8 models to measure perception performance, routing behavior, and computational efficiency under hardware execution.

A wearable prototype was assembled to examine system operation in real-world environments, as shown in Figure 3. The camera was mounted on the temple of a pair of eyeglasses and connected to the NPU board carried in a sling bag. Perception inference and routing were performed entirely on the NPU-based platform. A speaker delivered spoken navigation prompts based on the generated guidance commands.

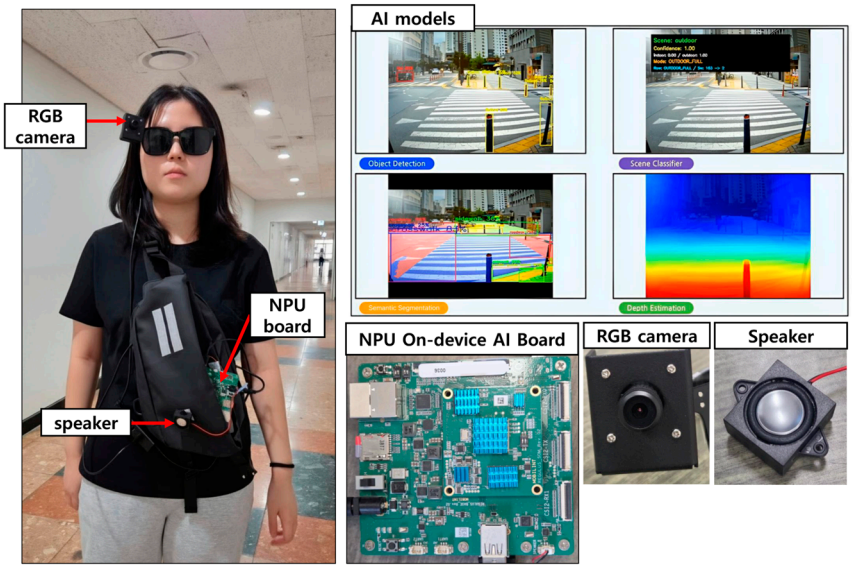


Figure 3. Wearable prototype and representative perception outputs. The prototype consists of an RGB camera mounted on the eyeglass temple, a speaker attached to the sling bag, and an NPU board carried inside the bag. Representative outputs include object detection, scene classification, semantic segmentation, and depth estimation. Walking commands are delivered to the user through spoken prompts. Red arrows indicate the locations of the hardware components. Warmer colors in the depth map indicate closer regions, whereas cooler colors indicate farther regions.

4.2. Datasets

4.2.1. Object Detection and Segmentation Datasets

The outdoor object-detection dataset was primarily constructed from the AI Hub Sidewalk Walking Dataset [40], supplemented with a publicly available bollard dataset from Roboflow Universe [41]. The final dataset contained 46,513 training, 5416 validation, and 5831 test images, with five object classes: Car, Person, Pole, Tree, and Bollard.

The indoor object-detection dataset combined Person annotations from COCO [42] with 14 publicly available datasets from Roboflow Universe [41], comprising two chair, five table, two stair, three door, and two column datasets. The source annotations were converted into a unified YOLO format, and class labels were mapped to Chair, Door, Stairs, Table, Column, and Person. The COCO Person samples were retained without indoor-background filtering to introduce variation in pedestrian appearance, pose, occlusion, and background, as the indoor detector was designed to recognize indoor-relevant object classes rather than exclusively indoor scenes.

For the two sources containing instance-segmentation annotations (“table 4” and “column2” datasets), polygons were converted into enclosing axis-aligned bounding boxes. The integrated annotations were manually reviewed to correct missing or invalid

labels, including unlabeled target objects and incorrectly assigned bounding boxes. The final indoor dataset contained 32,102 training, 3275 validation, and 1109 test images.

The outdoor semantic-segmentation dataset was constructed from the AI Hub Sidewalk Walking Dataset [40]. It contained 37,084 training, 4636 validation, and 4636 test images. Seven semantic classes were retained: Background, Sidewalk, Braille Guiding Block, Roadway/Alley, Crosswalk, Bike Lane, and Caution Zone.

Source-video-level separation was maintained where source identifiers were available. For datasets without predefined test splits, selected images were manually reviewed and assigned to the test set during integration. Cross-split duplicate screening was conducted using SHA-256 and perceptual hashing (pHash), with a Hamming distance threshold of ≤3. No exact or visually near-duplicate image pairs were identified under these screening criteria. However, source-video-level independence could not be fully verified for all publicly sourced images because some datasets lacked acquisition identifiers. The dataset sources and available partitions are summarized in Appendix A.

The class-wise instance distributions of the outdoor and indoor object-detection datasets are shown in Figures 4 and 5, respectively. The distributions exhibit class imbalance, with fewer instances of Stairs, Table, and Column than Chair and Person in the indoor dataset.

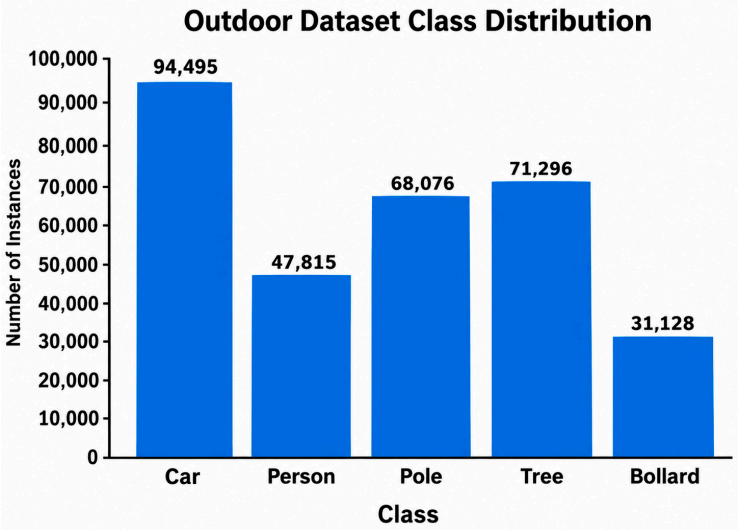


Figure 4. Class-wise instance distribution for outdoor object detection.

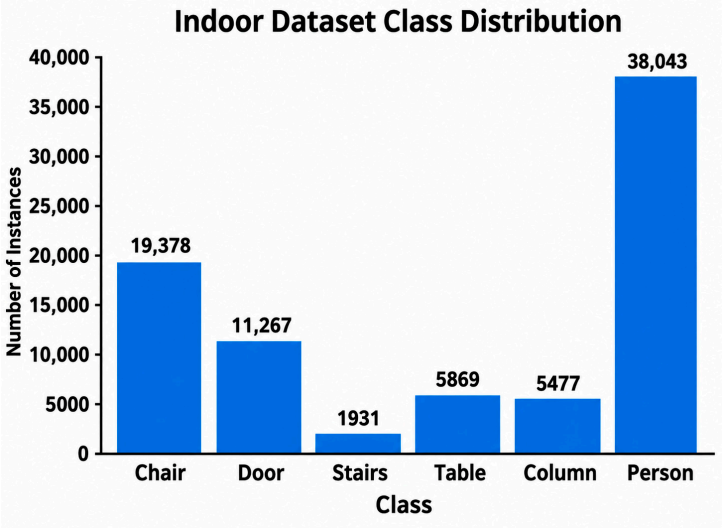


Figure 5. Class-wise instance distribution for indoor object detection.

4.2.2. Scene Classification Dataset

The scene-classification dataset incorporated indoor and outdoor images from Places365 [43], AI Hub [40,44], and eight publicly available first-person walking videos [45–52], including four recorded in Korea and four in other countries. These sources were selected to include diverse scene appearances, walking viewpoints, illumination conditions, and indoor–outdoor transitions. The walking videos covered stations, underground malls, city streets, an outlet mall, nighttime conditions, and rain (Appendix B).

The final held-out scene-classification test set comprised 1800 images, equally divided between indoor and outdoor environments. The same test images were used to compare FP32 and deployed INT8 classification performance. Quantization calibration samples were selected from training data and excluded from the held-out test set.

Continuous walking-video sequences were evaluated separately from the static classification test set to assess temporal routing behavior. Four external sequences were used for stable-scene routing evaluation: Berlin [53], Kuala Lumpur [54], Seoul-Night [55], and Seoul-Rain [56]. Three additional sequences were used for transition-response analysis: an indoor-to-outdoor mall sequence from Taipei [57], an outdoor-to-indoor mall sequence from Livingston, New Jersey [58], and an author-recorded sequence captured on the Sejong University campus. These sequences included indoor and outdoor environments, transition regions, and challenging conditions such as nighttime and rain. Ground-truth scene labels and transition intervals were manually annotated for evaluation. None of the evaluation sequences were used for scene-classifier training.

Cross-partition screening using SHA-256 and perceptual hashing (Hamming distance ≤ 3) found no exact or visually near-duplicate pairs between the scene-classifier training pool and the held-out test set.

4.2.3. Relative Depth Evaluation Dataset

The monocular depth estimation module was evaluated using the official SUN RGB-D validation split [59], comprising 4872 aligned RGB-depth image pairs. MiDaS v2.1 Small was evaluated without additional fine-tuning.

Because the model generates relative rather than metric depth, the evaluation focused on near–far ordering instead of absolute distance error. Ground-truth depth maps and object bounding boxes were used to assess whether the predicted depth representation preserved relative proximity relationships.

4.3. Training and Model Conversion

The perception models were trained or prepared according to their respective tasks and deployment requirements. The MobileNetV3-Small scene classifier was fine-tuned for binary indoor–outdoor classification using the training dataset described in Section 4.2.2. The indoor and outdoor object detectors were trained separately using the same YOLOv8m architecture but with environment-specific class sets. The semantic segmentation model was trained to predict seven outdoor surface-region classes, whereas the pretrained MiDaS v2.1 Small model was used without additional fine-tuning.

For the controlled comparison between unified and scene-specific object detection, a unified YOLOv8m detector was trained using the combined indoor and outdoor training data with a common 10-class label space. The unified and scene-specific detectors shared the same network architecture and were evaluated on identical environment-specific test images. This design enables a comparison of scene-specific detector specialization under a consistent evaluation setting.

The unified and scene-specific YOLOv8m detectors were trained using the common configuration summarized in Table 4, with 300 epochs, a 640 × 640 input resolution, and otherwise default Ultralytics (v8.3.234) training settings [60].

https://doi.org/10.3390/s26237548

Table 4. Common training configuration for unified and scene-specific YOLOv8m detectors.

Parameter	Configuration
Model architecture	YOLOv8m
Training epochs	300
Input resolution	640 × 640
Batch size/optimizer/learning rate	Ultralytics defaults
Default augmentation	Ultralytics default (no additional custom augmentation)

The trained and prepared models were exported from PyTorch to ONNX and subsequently compiled into NPU-compatible MXQ models using the vendor-provided SDK. INT8 quantization was applied during model conversion. Quantization calibration was performed using images selected from the training data, with 100 images allocated to each module. For object detection, calibration-set sizes of 50, 100, and 200 images were examined, and 100 images were selected as a common calibration setting based on the observed performance. The scene-classifier calibration set was balanced across the training data sources.

To ensure a consistent comparison between FP32 and INT8 scene classification, both implementations were evaluated using the same 1800-image held-out test set with equivalent RGB conversion, resizing to 224 × 224 pixels, ImageNet normalization, and argmax decision rules. The input tensor layout was adjusted to meet the requirements of each execution environment.

4.4. Evaluation Metrics

The proposed framework was evaluated in terms of perception accuracy, routing behavior, computational efficiency, and navigation-decision performance. Evaluation metrics were selected according to the characteristics of each perception module and the corresponding system-level requirements.

Scene classification was evaluated using overall accuracy and class-wise accuracy for indoor and outdoor environments. The FP32 and INT8 models were evaluated on the same held-out test set to assess the effects of quantization. For continuous walking sequences, routing performance was evaluated separately using the agreement between the selected perception mode and the ground-truth environment. Routing stability was further assessed through mode transitions, switching offsets, and unsafe routing exposure. Unsafe routing exposure was defined as the proportion of outdoor-ground-truth frames for which the router selected INDOOR_MODE, thereby omitting the outdoor segmentation branch. BOTH_MODE was not counted as unsafe outdoor routing because it executes the outdoor perception configuration.

Object detection performance was evaluated using precision, recall, AP@50, AP@75, and AP@50:95. The unified and scene-specific detectors were evaluated on identical environment-specific test images. Additional analyses examined missed detections, object-scale performance, and occlusion-related errors. For the error analysis, ground-truth objects were matched one-to-one with predictions of the same class using a confidence threshold of 0.25 and an intersection-over-union (IoU) threshold of 0.5. Object-scale categories were defined using bounding-box areas of 1024 and 9216 pixels on the 640 × 640 evaluation input.

Semantic segmentation was evaluated using mean IoU (mIoU) and class-wise IoU. The FP32 and deployed INT8 versions were evaluated to assess the effects of model conversion and quantization. Monocular depth estimation was evaluated using Spearman rank correlation and pairwise relative-depth ordering accuracy, reflecting the model's ability to preserve near–far relationships rather than estimate absolute metric distances. Object-level depth ordering was additionally evaluated using median depth cues

extracted from ground-truth bounding-box regions. An additional controlled evaluation compared relative-depth ordering using ground-truth and predicted bounding boxes under FP32 and INT8 inference.

On-device computational efficiency was evaluated using processing latency, summarized as the mean, median, and 95th-percentile (p95) values; sustained throughput; energy per processed frame; and peak host-process memory. The measurement protocols are described in Section 4.5.

Navigation-decision performance was evaluated using command accuracy, Unsafe FORWARD rate, and False STOP rate. Command accuracy represents the proportion of generated commands matching the reference decisions. Unsafe FORWARD rate was defined as the proportion of hazardous frames in which the system incorrectly issued a FORWARD command, while False STOP rate was defined as the proportion of non-hazardous frames in which the system unnecessarily issued a STOP command.

4.5. Controlled Pipeline Evaluation

To evaluate the computational effects of scene-aware routing and detector specialization, four perception configurations were compared using a 2 × 2 experimental design. The two factors were scene-dependent routing (disabled or enabled) and detector specialization (unified or scene-specific). The evaluated configurations were Scene-Agnostic Monolithic (SAM), Scene-Agnostic Dual (SAD), SAPER-Monolithic (SAPER-M), and SAPER-Decoupled (SAPER-D), as summarized in Table 5.

SAM and SAD represent the fixed-pipeline approach used in existing systems (Table 1), in which all perception modules are executed on every frame regardless of the scene. SAM is the conventional fixed pipeline, which executes a unified object detector, semantic segmentation, and monocular depth estimation. SAD executes both scene-specific object detectors together with semantic segmentation and depth estimation, representing a fixed pipeline that employs scene-specific detectors without routing. SAPER-D is the proposed configuration, which adaptively selects scene-specific object detectors and controls semantic segmentation according to the routing state. SAPER-M is an ablation of SAPER that uses the unified detector while activating semantic segmentation according to the routing state. Monocular depth estimation remains active in all four configurations.

This experimental design separates the effects of segmentation gating and detector specialization. Comparing SAM with SAPER-M evaluates the computational benefit of scene-dependent segmentation execution while retaining the unified detector. Comparing SAPER-M with SAPER-D examines the effect of detector specialization under routing. Comparing SAD with SAPER-D evaluates the difference between continuously executing both scene-specific detectors and adaptively selecting detector execution according to the routing state.

All configurations were evaluated using the same deterministic 1000-frame input set sampled from three walking sequences containing indoor–outdoor transitions. The sequences comprised an author-recorded campus video and two publicly available mall-transition videos described in Section 4.2. All configurations used identical input frames and the deployed INT8 perception models. For the routed configurations (SAPER-M and SAPER-D), identical precomputed routing decisions were used to maintain consistent module-selection conditions.

Computational performance was assessed using separate latency, throughput, and energy measurement protocols. End-to-end processing latency was measured from frame decoding to navigation-cue construction using a sequential frame-processing benchmark. Sustained throughput was measured independently as the completed-output rate of the multithreaded pipeline under continuous frame input. Energy consumption was evaluated using system-level power measurements during a separate sequential processing

experiment, with energy per processed frame calculated from measured power and execution time. Each configuration was evaluated in three independent runs, with 50 warm-up iterations performed before measurement.

The controlled 2 × 2 latency ablation and energy evaluation executed scene classification on every frame. The deployment-schedule evaluations used a classifier interval of five frames: in the sustained-throughput evaluation, frames were supplied at 30 fps and the classifier was executed on every fifth input frame, matching the deployed 6 Hz routing update rate, whereas in the sequential latency benchmark, which processes one frame at a time, the classifier was executed on every fifth benchmark frame. These experiments were conducted under separate measurement protocols.

Module-wise inference latency was additionally measured for scene classification, object detection, semantic segmentation, and monocular depth estimation. This analysis was used to identify the computational contributions of individual perception modules and interpret the efficiency improvements achieved through selective execution. An additional measurement was conducted to characterize camera-to-audio guidance latency on the deployed system. End-to-end guidance latency was defined as the elapsed time between camera frame capture and the corresponding audible output from the speaker. Intermediate timing measurements included the intervals from camera capture to the PLAYING state recorded in the playback log, from the audio request to audible output, and from the PLAYING state to audible output. The measurement was conducted separately in indoor mode (INDOOR_MODE) and outdoor mode (OUTDOOR_MODE). For prompts with two audible outputs, both outputs were required to achieve a normalized cross-correlation (NCC) score of at least 0.30, with an inter-output interval within ±2 ms of the expected 507.6 ms. For prompts interrupted after the first output, the first output was required to achieve an NCC score of at least 0.30. In indoor mode, 21 of 38 detected audio-output events met these criteria (9 with two outputs and 12 interrupted); in outdoor mode, 30 of 42 events met the criteria (13 with two outputs and 17 interrupted). Latency statistics were reported separately for the validated events and all detected events in each mode.

Table 5. Configurations used in the controlled pipeline evaluation.

Configuration	Scene Routing	Object Detection	Segmentation	Depth
Scene-Agnostic Monolithic (SAM)	Off	Unified	Always	Always
Scene-Agnostic Dual (SAD)	Off	Both Specialized	Always	Always
SAPER-Monolithic (SAPER-M)	On	Unified	Conditional	Always
SAPER-Decoupled (SAPER-D) (proposed)	On	Scene-specific	Conditional	Always

Conditional segmentation is enabled in OUTDOOR_MODE and BOTH_MODE and disabled in INDOOR_MODE. Monocular depth estimation remains active in all configurations.

4.6. Routing Stability and Transition Evaluation

Routing stability was evaluated using continuous walking-video sequences containing indoor, outdoor, and transition environments. Unlike static scene-classification accuracy, this evaluation examined the temporal behavior of the routing mechanism, including mode consistency, switching responsiveness, and routing errors under changing environmental conditions.

For stable-scene evaluation, the routing decisions were compared with manually annotated indoor and outdoor ground-truth labels. Exact-route accuracy was calculated as the proportion of frames assigned to the correct environment-specific stable mode (INDOOR_MODE or OUTDOOR_MODE). BOTH_MODE activations were analyzed separately to distinguish uncertainty-based fallback from incorrect stable-mode selection. Module-level accuracy additionally counted BOTH_MODE as correct for both indoor and

outdoor frames because it executes both scene-specific object detectors, semantic segmentation, and monocular depth estimation, thereby retaining the required perception modules for either environment. Additional evaluations used nighttime and rainy walking sequences to assess routing behavior under challenging visual conditions.

Transition responsiveness was assessed using sequences containing labeled indoor-to-outdoor and outdoor-to-indoor transitions. Switching offset was measured as the time from the beginning of the annotated target-scene interval to the router's selection of the corresponding module configuration, with negative values indicating selection before the annotated transition. The analysis also examined transient mode changes and the duration of incorrect routing during transition intervals.

Ten on-device indoor–outdoor transition runs were conducted to evaluate switching responsiveness and unsafe routing exposure. In both the recorded-video analysis and the on-device trials, the router was updated on every fifth camera frame, corresponding to approximately 6 Hz. Because the hysteresis, cooldown, and temporal-consistency windows are defined in routing updates, using the same update rate ensures that their temporal behavior is identical in both evaluations.

4.7. Navigation-Decision Evaluation

Navigation performance was evaluated through real-world walking trials and a supplementary controlled navigation-decision comparison. The walking trials examined the operation of the deployed SAPER framework along predefined routes, while the controlled comparison assessed whether scene-dependent module execution affected navigation-decision accuracy relative to fixed-pipeline configurations.

4.7.1. Real-World Walking Trials

Four predefined walking routes were used, comprising two outdoor routes (R1 and R2) and two indoor routes (R3 and R4). Each route was evaluated in five trials, resulting in 20 trials in total. The routes included both repeated and varied obstacle arrangements to examine the consistency of navigation decisions under different conditions.

Command accuracy (Section 4.4) was calculated for each trial, and variability across trials was summarized using the mean and sample standard deviation. During each trial, the device carrier, the observer, and the evaluator were different individuals. The observer monitored the system outputs using a laptop and recorded the generated navigation commands and obstacle-avoidance outcomes. The evaluator, who did not view the system outputs, independently assigned reference commands based on the observed obstacle configuration and the predefined decision criteria described below. The recorded commands were then compared with the reference commands to determine decision correctness.

The authors served as device carriers because the trials were intended to evaluate the operation of the navigation system rather than the mobility performance of visually impaired users. Reference commands were assigned according to the observed obstacle configurations: FORWARD when the forward path was clear, LEFT or RIGHT when forward movement was obstructed but a feasible lateral direction remained, and STOP when no feasible direction was available.

Route completion, navigation-decision correctness, and obstacle avoidance were recorded. An obstacle encounter was considered successfully avoided when an appropriate avoidance command was issued and the walker passed the obstacle without contact.

4.7.2. Controlled Navigation-Decision Comparison

A supplementary evaluation compared the navigation decisions produced by SAM, SAD, SAPER-M, and SAPER-D using the same evaluation data. Each configuration

generated perception outputs according to its module-execution rules, and the resulting information was assessed using a common navigation-decision procedure.

The evaluation data consisted of 774 frames, comprising 117 hazardous and 657 non-hazardous frames, which served as the denominators for the Unsafe FORWARD and False STOP rates, respectively. The frames were sampled from indoor and outdoor walking videos that were recorded separately by the authors for this comparison. Reference commands were annotated by the authors based on the observed walking conditions and obstacle configurations, following the criteria described in Section 4.7.1, without access to the outputs of the four configurations.

Command accuracy, Unsafe FORWARD rate, and False STOP rate (Section 4.4) were compared across the four configurations to examine whether the computational benefits of scene-aware routing were accompanied by changes in navigation-decision performance.

5. Results

5.1. Scene Classification Results

Table 6 presents the scene-classification performance of the FP32 GPU and INT8 NPU models on the same balanced held-out test set of 1800 images. The FP32 model achieved an overall accuracy of 96.22%, with indoor and outdoor accuracies of 95.89% and 96.56%, respectively. The deployed INT8 model achieved an overall accuracy of 95.61%, with indoor and outdoor accuracies of 97.22% and 94.00%, respectively. INT8 deployment therefore resulted in a 0.61 percentage-point reduction in overall accuracy.

Both models were evaluated using identical test images and equivalent preprocessing procedures, as described in Section 4.3. The INT8 model showed higher indoor accuracy but lower outdoor accuracy than the FP32 model, indicating a change in class-specific prediction behavior following deployment.

The results demonstrate that the scene classifier largely preserved its overall classification performance after INT8 deployment on the target NPU. Its temporal routing performance, including stable-scene accuracy, fallback behavior, and transition responsiveness, is evaluated separately in Section 5.6.

Table 6. Scene classifier accuracy before and after INT8 NPU deployment.

Deployment	Overall	Indoor	Outdoor	Balanced
GPU FP32	96.22%	95.89%	96.56%	96.22%
NPU INT8 (MXQ)	95.61%	97.22%	94.00%	95.61%

Both models were evaluated on the same 1800-image held-out set comprising 900 indoor and 900 outdoor images.

5.2. Scene-Specific Object Detection Results

The indoor and outdoor scene-specific YOLOv8m detectors were evaluated individually before comparison with the unified detector. Table 7 summarizes their overall FP32 test performance. The outdoor detector achieved an mAP@50 of 0.9608 and an mAP@50:95 of 0.7514, while the indoor detector achieved 0.9251 and 0.8217, respectively. Class-wise performance is presented in Table 8.

Table 7. Overall performance of the scene-specific object detectors.

Detector	Precision	Recall	mAP@50	mAP@50:95
Outdoor	0.924	0.896	0.961	0.751
Indoor	0.919	0.858	0.925	0.822

Values are FP32 GPU results for the scene-specific detectors.

Additional error analysis examined detection reliability according to object size and occlusion. Small-object recall was 72.48% for the outdoor detector and 45.07% for the indoor detector, corresponding to false-negative rates of 27.52% and 54.93%, respectively. In the manually reviewed occlusion subsets, recall under heavy occlusion was 90.91% outdoors and 85.71% indoors. These results indicate that small indoor objects presented a greater detection challenge, while heavily occluded objects remained a source of missed detections in both environments.

At the fixed evaluation operating point, the highest class-wise false-negative rates were observed for Bollard (10.55%), Tree (7.86%), and Pole (7.08%) outdoors, and Chair (13.77%), Person (12.95%), and Table (11.19%) indoors. These findings identify object categories for which missed detections remain relatively frequent.

From a walking-assistance perspective, missed detections of small or narrow obstacles may reduce the reliability of obstacle-aware navigation decisions. The relatively low small-object recall, particularly in indoor environments, indicates that small nearby obstacles may be missed by the current detectors. Outdoors, missed detections of bollards and poles are also relevant because these structures may occupy a limited image area while obstructing a walking path. However, these detection metrics alone do not fully characterize how missed obstacles affect navigation decisions during walking.

To further investigate obstacle detection failures relevant to walking assistance, a region-wise and failure-type analysis was conducted for bollards and poles. The frontal evaluation region included objects whose bounding-box centers were horizontally located between 34% and 66% of the image width and whose bottom edges were at or below 45% of the image height, measured from the top. Evaluation used a confidence threshold of 0.25, an IoU threshold of 0.5, and class-matched one-to-one matching.

In the frontal region, recall was 88.3% for bollards and 92.8% for poles, with similar performance across image regions. Nevertheless, 144 bollards and 184 poles were missed in the frontal region. Among all false negatives, complete non-detection was the most frequent failure type for bollards (160/346, 46.2%), whereas low-confidence predictions were the most frequent failure type for poles (255/504, 50.6%). These findings highlight the remaining detection limitations for narrow outdoor obstacles.

Table 8. Class-wise performance of the scene-specific object detectors.

Detector	Class	Precision	Recall	AP@50	AP@50:95
Outdoor	Car	0.941	0.943	0.984	0.871
	Person	0.945	0.923	0.974	0.776
	Pole	0.915	0.881	0.960	0.749
	Tree	0.916	0.874	0.949	0.656
	Bollard	0.903	0.858	0.936	0.706
	Overall	0.924	0.896	0.961	0.751
Indoor	Chair	0.937	0.839	0.929	0.853
	Door	0.946	0.897	0.941	0.844
	Stairs	0.953	0.892	0.955	0.867
	Table	0.864	0.838	0.910	0.849
	Column	0.902	0.887	0.938	0.828
	Person	0.912	0.795	0.877	0.690
	Overall	0.919	0.858	0.925	0.822

5.2.1. Controlled Comparison with the Unified Detector

To evaluate the effect of detector specialization, the unified YOLOv8m detector was compared with the scene-specific detectors on identical environment-specific evaluation images. Both approaches used the same YOLOv8m architecture, while the unified model

combined the indoor and outdoor classes into a common label space. Class-wise AP@50 results are presented in Table 9, and overall performance is summarized in Table 10.

Table 9. Class-wise controlled comparison (AP@50, FP32 GPU).

Environment	Class	Unified AP@50	Scene-Specific AP@50
Outdoor	Car	0.976	0.984
	Person	0.969	0.974
	Pole	0.939	0.960
	Tree	0.934	0.949
	Bollard	0.890	0.936
Indoor	Chair	0.913	0.929
	Door	0.928	0.941
	Stairs	0.932	0.955
	Table	0.868	0.910
	Column	0.756	0.938
	Person	0.811	0.877

The scene-specific detectors achieved higher AP@50 than the unified detector across all reported classes. The largest improvement was observed for indoor Column, increasing from 0.7556 to 0.9380 (+18.24 percentage points). Improvements were also observed for indoor Person and Table, indicating that the benefits of specialization varied across object categories.

For outdoor detection, mAP@50 increased from 0.9417 to 0.9608 (+1.91 percentage points), while mAP@50:95 increased from 0.7149 to 0.7514 (+3.65 percentage points). For indoor detection, mAP@50 increased from 0.8680 to 0.9251 (+5.71 percentage points), and mAP@50:95 increased from 0.7253 to 0.8217 (+9.64 percentage points).

Table 10. Overall controlled performance of unified and scene-specific YOLOv8m detectors.

Environment	Metric	Unified	Scene-Specific
Outdoor	Precision	0.899	0.924
	Recall	0.871	0.896
	mAP@50	0.942	0.961
	mAP@75	0.813	0.851
	mAP@50:95	0.715	0.751
Indoor	Precision	0.842	0.919
	Recall	0.796	0.858
	mAP@50	0.868	0.925
	mAP@75	0.779	0.869
	mAP@50:95	0.725	0.822

Unified and scene-specific models were compared using identical environment-specific evaluation images.

These results support the use of scene-specific object detectors to improve detection accuracy under the evaluated conditions. The corresponding computational effects of detector specialization and routing are examined separately in Section 5.5.

5.2.2. INT8 Deployment Performance

To evaluate the effects of model conversion and quantization, the scene-specific detectors were compared across PyTorch FP32, ONNX FP32, and NPU INT8 (MXQ) deployment formats. Table 11 summarizes the detection performance across these formats.

For the indoor detector, mAP@50 decreased from 0.9251 in PyTorch FP32 to 0.9080 in NPU INT8, while mAP@50:95 decreased from 0.8217 to 0.7767. For the outdoor detector, mAP@50 decreased from 0.9608 to 0.9544, and mAP@50:95 decreased from 0.7514 to 0.7172.

The reductions in mAP@50 were 1.71 percentage points indoors and 0.64 percentage points outdoors. Larger reductions were observed in mAP@50:95, indicating that deployment affected bounding-box localization performance more strongly than detection at the less stringent IoU threshold.

These results indicate that the deployed INT8 detectors largely preserved their FP32 detection performance, although localization accuracy decreased after conversion and quantization. The computational effects of using these deployed detectors within the scene-aware pipeline are evaluated in Section 5.5.

Table 11. Detection performance across model deployment formats.

Detector	Format	mAP@50	mAP@75	mAP@50:95
Indoor	PyTorch FP32	0.925	0.869	0.822
	ONNX FP32	0.913	0.851	0.801
	NPU INT8 (MXQ)	0.908	0.839	0.777
Outdoor	PyTorch FP32	0.961	0.851	0.751
	ONNX FP32	0.959	0.848	0.746
	NPU INT8 (MXQ)	0.954	0.822	0.717

Results represent the same detector model lineage across PyTorch FP32, ONNX FP32, and MXQ INT8 formats.

5.3. Semantic Segmentation Results

The semantic segmentation module was evaluated on the outdoor segmentation test set to assess the effects of model conversion and INT8 deployment. The PyTorch FP32 model achieved an mIoU of 0.7957, while the ONNX FP32 model achieved 0.7955. Direct evaluation of the deployed INT8 MXQ model yielded an mIoU of 0.7705, representing a reduction of 2.52 percentage points relative to the PyTorch FP32 model.

Table 12 presents the class-wise IoU results for the FP32 and INT8 models. Performance decreased across all seven classes following INT8 deployment, without a pronounced degradation concentrated in a single class. Roadway/Alley achieved the highest IoU in the FP32 model (0.8801), whereas Bike Lane and Caution Zone showed the lowest IoU values in both models.

These results indicate that the deployed INT8 segmentation model largely preserved its FP32 performance, although moderate accuracy reductions were observed across the evaluated surface-region classes. The computational contribution of selectively executing this module is examined in Section 5.5.

Table 12. Class-wise IoU of the outdoor semantic segmentation model before and after INT8 NPU deployment.

Class	PyTorch FP32	NPU INT8 (MXQ)
Background	0.853	0.825
Sidewalk	0.861	0.839
Braille Guiding Block	0.851	0.832
Roadway/Alley	0.880	0.843
Crosswalk	0.825	0.801
Bike Lane	0.640	0.622
Caution Zone	0.660	0.632
mIoU	0.796	0.771

The FP32 and INT8 models were evaluated on the same outdoor segmentation test set.

https://doi.org/10.3390/s26237548

5.4. Relative-Depth Estimation Results

The pretrained MiDaS v2.1 Small model was evaluated on all 4872 RGB-depth image pairs in the official SUN RGB-D validation split. At the pixel level, the model achieved a Spearman rank correlation of 0.852 and a pairwise relative-depth ordering accuracy of 91.43%.

At the object level, 4017 images containing valid ground-truth object pairs were evaluated. The model achieved a Spearman rank correlation of 0.835 and an image-wise object-pair ordering accuracy of 93.70%. Across 56,353 valid object pairs, the pooled ordering accuracy was 93.77%.

These results indicate that MiDaS v2.1 Small generally preserved relative near–far relationships at both pixel and object levels. This supports its use for object-wise relative proximity estimation in SAPER, although the predicted depth values do not represent absolute distances in meters.

To evaluate relative-depth reliability under deployment-oriented conditions, an additional controlled comparison was performed using ground-truth and predicted bounding boxes with FP32 and INT8 inference, as summarized in Table 13. The evaluation used a common set of 1404 valid object pairs from 598 SUN RGB-D images, with object-level Spearman correlation evaluated on 238 eligible images. Using ground-truth boxes, FP32 and INT8 depth estimation achieved pairwise ordering accuracies of 96.34% and 96.55%, respectively. With predicted bounding boxes, the FP32 detector–depth configuration achieved 96.48%, whereas the fully INT8 detector–depth configuration achieved 96.99%, with a Spearman correlation of 0.883. These results indicate that the deployed INT8 perception configuration preserved high relative-depth ordering accuracy on the commonly evaluated object pairs. The reported comparisons are restricted to matched objects and do not independently measure detection failures.

Table 13. Controlled relative-depth ordering comparison using ground-truth and predicted bounding boxes.

Configuration	Object Spearman	Pairwise Accuracy	Valid Pairs
GT Box + FP32	0.886	96.34%	1404
GT Box + INT8	0.886	96.55%	1404
Predicted Box + FP32	0.899	96.48%	1404
Predicted Box + INT8	0.883	96.99%	1404

All configurations were evaluated using the same 1404 matched object pairs from 598 images. Object-level Spearman correlation was calculated on 238 eligible images. Predicted bounding boxes were generated using the corresponding FP32 or INT8 detector.

5.5. Computational Efficiency Results

5.5.1. Controlled Pipeline Ablation

The computational efficiency of the four pipeline configurations was evaluated using the controlled 1000-frame NPU benchmark described in Section 4.5. Table 14 presents the results of the 2 × 2 ablation with scene classification executed on every frame. Each configuration was evaluated in three independent runs.

Table 14. Controlled pipeline ablation on the NPU (classifier on every frame).

Configuration	Mean (ms)	SD (ms)	Median (ms)	p95 (ms)
Scene-Agnostic Monolithic (SAM)	272.50	1.82	273.51	278.69
Scene-Agnostic Dual (SAD)	320.35	2.29	320.47	327.47
SAPER-Monolithic (SAPER-M)	220.05	1.02	280.62	286.27
SAPER-Decoupled (SAPER-D) (proposed)	220.81	1.31	281.65	287.89

Each configuration was evaluated in three independent runs. SD denotes the standard deviation across run means; median and p95 are pooled frame-level statistics.

The proposed SAPER-D configuration reduced mean processing latency from 272.50 ms for SAM to 220.81 ms, corresponding to an 18.97% reduction. SAPER-M achieved a similar mean latency of 220.05 ms, indicating that selective execution of semantic segmentation was the primary source of the improvement. In contrast, SAD increased mean latency to 320.35 ms because both scene-specific detectors were executed without routing.

Comparing SAPER-M and SAPER-D shows that scene-specific detector selection introduced only a 0.76 ms difference in mean latency under routing. This difference was smaller than the observed run-to-run variation. Thus, detector specialization provided the accuracy benefits reported in Section 5.2 without a substantial increase in processing latency under the routed configuration.

The latency reduction depended on the selected perception route. As shown in Table 15, SAPER-D achieved a mean latency of 79.60 ms for INDOOR_MODE frames and 283.36 ms for full-route frames, compared with 273.20 ms and 272.19 ms for SAM, respectively. The lower overall mean latency therefore resulted primarily from avoiding semantic segmentation in indoor mode, while full-route execution incurred a modest routing overhead.

Table 15. Route-separated mean latency (classifier on every frame).

Configuration	Overall (ms)	Light-Route (ms)	Full-Route (ms)
SAM	272.50	273.20	272.19
SAPER-M	220.05	80.59	281.84
SAPER-D	220.81	79.60	283.36

Light route refers to INDOOR_MODE, whereas full-route execution includes OUTDOOR_MODE and BOTH_MODE. Unlike OUTDOOR_MODE, BOTH_MODE additionally executes the indoor object detector.

Although BOTH_MODE executes both scene-specific object detectors, it was activated in only 3.0% of the evaluated frames. Its relatively low activation frequency suggests that the additional computational overhead associated with BOTH_MODE had a limited impact on the overall mean latency under the evaluated routing distribution.

5.5.2. Deployment-Schedule Latency and Throughput

A separate evaluation was conducted using the deployed scene-classification interval of five frames. Under this condition, SAPER-D reduced mean latency from 271.80 ms for SAM to 213.23 ms, representing a 21.55% reduction.

The pooled median latency remained similar between the configurations, with 272.03 ms for SAM and 272.66 ms for SAPER-D. However, the p95 latency increased from 278.41 ms to 284.93 ms. These results indicate that scene-aware routing reduced average processing cost through indoor lightweight execution, while individual full-route frames could incur slightly higher latency.

To examine the sensitivity of computational efficiency to scene composition, latency was analyzed under different indoor-frame proportions ranging from 25% to 100%. As summarized in Table 16, SAPER-D reduced mean latency relative to SAM by 17.3%, 36.1%, 54.9%, and 73.7% at indoor proportions of 25%, 50%, 75%, and 100%, respectively. The original benchmark, containing 30.7% indoor frames, yielded a mean latency reduction of 21.6%. Although mean latency decreased as the indoor proportion increased, median and p95 latencies did not improve uniformly because outdoor frames continued to invoke the full perception pipeline. These results demonstrate that the computational benefit of

scene-aware routing depends on workload composition, with greater improvements in indoor-dominated environments.

Table 16. Latency sensitivity to indoor–outdoor frame proportions.

Indoor Ratio	SAM Mean	SAPER-D Mean	Reduction	Median (SAM/SAPER-D)	p95 (SAM/SAPER-D)
25%	271.8	224.9	17.3%	272.1/274.2	278.4/285.1
30.7%	271.8	213.2	21.6%	272.0/272.7	278.4/284.9
50%	271.8	173.7	36.1%	271.9/173.1	278.3/283.7
75%	271.7	122.5	54.9%	271.6/70.1	278.2/279.8
100%	271.6	71.3	73.7%	271.3/69.1	278.1/79.5

Sustained throughput was measured independently using the multithreaded pipeline. At the deployed classifier interval, throughput increased from 6.594 outputs/s for SAM to 7.362 outputs/s for SAPER-D, corresponding to an 11.65% improvement.

Because the multithreaded pipeline overlaps image acquisition, preprocessing, NPU inference, and post-processing across consecutive frames, its output rate is limited by the most heavily loaded stage rather than by the sum of per-frame processing times. Skipping semantic segmentation in INDOOR_MODE frames removes both its NPU inference and its dense post-processing, which reduces the load on this bottleneck and increases the completed-output rate. The smaller relative gain in throughput (11.65%) than in mean sequential latency (21.55%) suggests that stages unaffected by routing, such as image acquisition, preprocessing, and detection and depth inference, also constrain the pipeline output rate.

5.5.3. Energy Consumption and Memory Usage

Energy consumption was evaluated separately with scene classification executed on every frame. As shown in Table 17, SAPER-D reduced steady-state energy consumption from 2.123 J/frame for SAM to 1.785 J/frame, corresponding to a 15.92% reduction. Whole-run energy consumption, including model initialization and warm-up, decreased from 2.271 to 2.000 J/frame, representing an 11.94% reduction.

Table 17. System-level energy consumption (classifier on every frame).

Configuration	Mean Power (W)	Steady Energy (J/Frame)	Whole-Run Energy (J/Frame)
SAM	5.908	2.123	2.271
SAD	5.939	2.408	2.591
SAPER-M	5.860	1.783	1.980
SAPER-D	5.832	1.785	2.000

Steady-state energy excludes initialization overhead, whereas whole-run energy includes model loading and warm-up.

The average system power remained relatively similar across configurations during maximum-speed processing. Consequently, the energy reduction was primarily associated with the shorter processing time per frame rather than a substantial decrease in instantaneous system power.

Peak host-process memory usage was 115.4 MiB for SAM and 119.2 MiB for SAPER-D, an increase of 3.8 MiB. This increase reflects the additional model residency required to retain both scene-specific detectors for runtime selection.

5.5.4. Module-Wise Latency Analysis

Module-wise NPU inference latency was measured to identify the computational contribution of individual perception components. As shown in Table 18, semantic segmentation exhibited the highest latency at 93.128 ms, followed by outdoor object detection (32.322 ms) and indoor object detection (29.962 ms). Scene classification and monocular depth estimation required 1.423 ms and 6.585 ms, respectively.

These findings support the pipeline-level ablation results, indicating that selectively disabling semantic segmentation was the primary source of the mean latency reduction. Scene-specific detector selection mainly contributed to detection accuracy while introducing limited additional computational cost under routing.

Overall, SAPER reduced mean processing latency and energy consumption while improving sustained throughput under the evaluated conditions. These improvements were primarily achieved through selective segmentation execution, complemented by scene-specific object detection.

Table 18. Module-wise inference latency on the NPU.

Perception Module	Latency (ms)
Scene Classification	1.423
Indoor Object Detection	29.962
Outdoor Object Detection	32.322
Semantic Segmentation	93.128
Monocular Depth Estimation	6.585

Table 18 reports NPU inference time only. The summed NPU inference times are approximately 38 ms for INDOOR_MODE, 133 ms for OUTDOOR_MODE, and 163 ms for BOTH_MODE. These values are lower than the corresponding end-to-end pipeline latencies, which additionally include CPU-side processing such as input preprocessing, detection decoding and non-maximum suppression, resizing and alignment of segmentation and depth outputs, and navigation-cue construction. The full-route mean latency of 283.36 ms in Table 15 includes both OUTDOOR_MODE and BOTH_MODE. Because semantic segmentation requires dense post-processing at 640 × 640 resolution, disabling the segmentation branch reduces both NPU inference time and its associated post-processing cost.

5.5.5. Camera-to-Audio Guidance Latency

To evaluate spoken navigation guidance latency, an additional on-device measurement was conducted from camera frame capture to the corresponding audible speaker output in indoor and outdoor modes. As summarized in Table 19, the median end-to-end latency of the validated events was 189.3 ms (p95: 291.8 ms) in indoor mode and 328.6 ms (p95: 438.2 ms) in outdoor mode. The higher latency in outdoor mode is consistent with the additional semantic segmentation processing in the full perception configuration (Table 15).

The median delay between the logged PLAYING state and actual speaker output for validated events was 9.8 ms and 9.9 ms in indoor and outdoor modes, respectively. These results extend the perception-processing latency evaluation by incorporating the actual audio-output stage.

To examine the relationship between guidance latency and walking speed, travel distances were estimated using the 95th-percentile latency of the validated events in outdoor mode (438.2 ms), the slower of the two modes, and assumed walking speeds of 0.5–1.2 m/s. The estimated distances ranged from 0.219 to 0.526 m, corresponding to approximately 0.351 m at 0.8 m/s and 0.438 m at 1.0 m/s; in indoor mode, the corresponding range

was 0.146–0.350 m. These estimates do not include human reaction time or stopping distance.

Table 19. End-to-end camera-to-audio guidance latency on the deployed system in indoor and outdoor modes.

Mode	Evaluation Set	*n*	Median (ms)	p05 (ms)	p95 (ms)
Indoor (INDOOR_MODE)	Validated events	21	189.3	173.8	291.8
	All detected events	38	236.2	173.4	292.0
Outdoor (OUTDOOR_MODE)	Validated events	30	328.6	287.4	438.2
	All detected events	42	337.8	284.9	462.5

p05 and p95 denote the 5th and 95th percentiles, respectively.

5.6. Routing Stability and Transition Results

5.6.1. Routing Stability on External Walking Sequences

The temporal stability of the deployed routing mechanism was evaluated using four external walking sequences: Berlin, Kuala Lumpur, Seoul-Night, and Seoul-Rain. Table 20 summarizes module-level routing accuracy, unsafe routing exposure, BOTH_MODE usage, module switches, and module-level flicker. Module-level flicker refers to a light- or full-route interval shorter than 2 s and bounded by the opposite module configuration.

Table 20. Routing stability on four external walking sequences.

Sequence	Module-Level Accuracy	Unsafe Exposure	BOTH_MODE	Module Switches	Module Flicker
Seoul-Night	100.00%	0.00% (0/3597)	7.23%	0	0
Seoul-Rain	100.00%	0.00% (0/7200)	0.28%	0	0
Berlin	96.33%	n/a	7.23%	17	11
Kuala Lumpur	96.78%	16.67% (90/540)	7.19%	3	1

Labeled transition frames are excluded from stable-scene accuracy. BOTH_MODE is treated as full-module execution. Berlin contains no outdoor ground-truth frames in the evaluated segment; therefore, its unsafe exposure ratio is undefined.

On the Seoul-Night and Seoul-Rain sequences, the router maintained full-module execution throughout the evaluated outdoor footage, with no unsafe INDOOR_MODE assignments or module-level flicker. Seoul-Night used OUTDOOR_MODE for 92.77% of frames and BOTH_MODE for 7.23%, while Seoul-Rain used BOTH_MODE for only 0.28% of frames.

The Berlin sequence achieved 96.33% module-level routing accuracy, with incorrect routing decisions occurring primarily in a glass-roofed arcade. These results highlight the difficulty of scene classification in visually ambiguous environments. In contrast, the Kuala Lumpur sequence achieved 96.78% module-level routing accuracy but exhibited an unsafe routing exposure of 16.67% (90/540 outdoor frames). These unsafe assignments occurred within a continuous 3.0 s interval on a covered elevated walkway.

Temporal stabilization substantially reduced unnecessary module switching. Compared with the routing mechanism without EMA smoothing, module-level flicker decreased from 14 to 0 on Seoul-Night, from 49 to 11 on Berlin, and from 25 to 1 on Kuala Lumpur. The corresponding module-switch counts decreased from 16, 59, and 29 to 0, 17, and 3, respectively. Nevertheless, the Kuala Lumpur result indicates that temporally stable but incorrect scene predictions can still produce unsafe routing exposure.

5.6.2. Indoor–Outdoor Transition Responsiveness

Transition responsiveness was examined using three annotated indoor–outdoor transitions. Table 21 compares switching offsets across successive routing-stabilization configurations. Switching offset was measured relative to the beginning of the annotated target-scene interval, with negative values indicating that the corresponding module configuration was activated before the labeled transition.

Table 21. Transition switching offsets across routing configurations.

Variant	Campus (out → in)	Mall (in → out)	Mall (out → in)	Mean	Median	Missed
V0 argmax only	−3.500	−4.850	−3.954	−4.101	−3.954	0
V1 + confidence fallback	−3.500	−3.183	−2.953	−3.212	−3.183	0
V2 + hysteresis	−3.334	−3.183	−2.786	−3.101	−3.183	0
V3 + cooldown	−3.334	−3.183	−2.786	−3.101	−3.183	0
V4 + temporal consistency	−2.667	−2.350	−1.952	−2.323	−2.350	0
V5 + EMA (deployed)	+3.000	−0.850	−0.284	+0.622	−0.284	0

The arrow (→) indicates the direction of the scene transition.

The final EMA-based router switched to the correct configuration at all three transitions. The recorded switching offsets were +3.000 s for the campus outdoor-to-indoor transition, −0.850 s for the mall indoor-to-outdoor transition, and −0.284 s for the mall outdoor-to-indoor transition.

In these cases, early activation of the full perception configuration before outdoor entry and delayed deactivation following indoor entry retained outdoor perception modules around the transition boundaries. Across the three annotated transition intervals, outdoor exposure to INDOOR_MODE decreased from 3.0 s without EMA smoothing to 0 s with the deployed stabilization mechanism.

In addition to the recorded-video analysis, ten live indoor–outdoor transition trials were conducted on the deployed NPU system, comprising five trials in each direction. The results are summarized in Table 22.

Table 22. On-device indoor–outdoor transition results.

Metric	Indoor → Outdoor	Outdoor → Indoor
Number of trials	5	5
Unsafe routing exposure	3.2%	5.4%
Maximum continuous unsafe duration	2.12 s	3.20 s
Mean switching offset	+1.11 s	−2.28 s
BOTH_MODE usage	6.3%	7.3%
Mean module switches	3.0	4.0

Switching offsets are measured relative to the annotated environmental transition, with negative values indicating mode activation before the transition. Unsafe routing exposure denotes the proportion of outdoor-ground-truth time assigned to INDOOR_MODE. BOTH_MODE usage denotes the proportion of evaluated time during which both scene-specific object detectors were activated due to routing uncertainty. The arrow (→) indicates the direction of the scene transition.

The live trials revealed routing behavior that was not observed in the recorded-video analysis. In indoor-to-outdoor transitions, the full perception configuration was restored on average 1.11 s after the annotated transition, whereas in outdoor-to-indoor transitions, INDOOR_MODE was selected on average 2.28 s before the annotated transition, while the walker was still approaching the entrance from outside. These delayed and premature switches account for the observed unsafe exposure of 3.2% and 5.4%, with maximum

continuous unsafe durations of 2.12 s and 3.20 s, respectively. Building entrances, canopies, and covered approaches can resemble indoor scenes, as also observed on the covered walkway in the Kuala Lumpur sequence. Therefore, the absence of unsafe exposure in the three recorded transitions should not be interpreted as evidence that unsafe routing cannot occur near entrances; under live conditions, unsafe intervals of up to approximately 3 s were observed.

To examine navigation-command safety during incorrect routing, we analyzed the commands generated within unsafe-routing intervals across 10 live indoor–outdoor transition trials. As shown in Table 23, the cumulative unsafe-routing duration was 24.5 s, during which 36 navigation commands were generated, including 22 FORWARD commands. No unsafe FORWARD commands were observed in either transition direction (0/22 FORWARD commands). Furthermore, no unsafe LEFT, RIGHT, or STOP commands were observed among the remaining 14 navigation commands.

Table 23. Navigation-command safety during live indoor–outdoor transitions.

Transition Direction	Unsafe-Routing Duration (s)	Navigation Commands	FORWARD Commands	Unsafe FORWARD
Indoor → Outdoor (5 trials)	10.5	16	10	0
Outdoor → Indoor (5 trials)	14.0	20	12	0
Overall (10 trials)	24.5	36	22	0

5.6.3. Routing Accuracy on the Controlled Benchmark

Routing accuracy was also assessed on the 1000-frame benchmark employed in the computational evaluation. The routing trace used for that evaluation comprised 307 INDOOR_MODE, 663 OUTDOOR_MODE, and 30 BOTH_MODE decisions. Table 24 compares these decisions with the ground-truth scene labels.

Table 24. Routing decisions versus ground-truth scene labels on the 1000-frame benchmark.

Ground Truth	INDOOR_MODE	OUTDOOR_MODE	BOTH_MODE	Total
INDOOR	285	1	14	300
OUTDOOR	0	650	10	660
Transition	22	12	6	40
Total	307 (30.7%)	663 (66.3%)	30 (3.0%)	1000

Transition frames are excluded from stable-scene accuracy but remain available to the routing state machine.

After excluding the 40 annotated transition frames, the router achieved an exact-route accuracy of 97.40% (935/960) across the 960 stable-scene frames. The module-level accuracy was 99.90% (959/960), counting BOTH_MODE as correct for both indoor and outdoor frames because it retains the required perception modules for both environments. No outdoor frame was routed to INDOOR_MODE (0/660). Among the 15 indoor frames not assigned to INDOOR_MODE, 14 were routed to BOTH_MODE, which retained the indoor object detector at the cost of additional computation. Only one indoor frame was assigned to OUTDOOR_MODE, resulting in the omission of the indoor object detector.

These results indicate that the stabilized router generally selected the intended perception configuration on the controlled benchmark. However, routing accuracy and unsafe routing exposure represent different aspects of performance, and high overall accuracy does not eliminate the possibility of unsafe module selection in challenging scenes.

To evaluate the effectiveness of BOTH_MODE under uncertain routing conditions, object detection recall was compared between scene-specific configurations and BOTH_MODE. In BOTH_MODE, both indoor and outdoor object detectors were executed,

and their detection results were combined using class-wise NMS before recall evaluation. As shown in Table 25, indoor recall increased from 88.0% under INDOOR_MODE to 95.3% under BOTH_MODE, while outdoor recall increased from 94.4% under OUTDOOR_MODE to 95.5% under BOTH_MODE. These results indicate that BOTH_MODE maintained object detection coverage in both environments during uncertain routing conditions.

Table 25. Obstacle detection recall under scene-specific and BOTH_MODE configurations.

Scene	Normal Recall (%)	BOTH_MODE Recall (%)	Difference
Indoor	88.0	95.3	+7.3 pp
Outdoor	94.4	95.5	+1.1 pp

5.7. Navigation-Decision Results

5.7.1. Real-World Walking Trial Results

The proposed system was evaluated in controlled walking trials involving four routes, consisting of two outdoor routes (R1 and R2) and two indoor routes (R3 and R4). Each route was repeated five times, resulting in 20 trials. Table 26 summarizes route completion, navigation-decision accuracy, and obstacle avoidance performance.

All 20 trials were completed successfully. Across the four routes, the system generated 443 navigation decisions, of which 413 matched the reference commands, yielding an overall decision accuracy of 93.23%. Per-route accuracy ranged from 92.23% on R1 to 94.59% on R3.

Among 55 obstacle encounters, 54 were successfully avoided according to the predefined evaluation criterion. The remaining encounter occurred on indoor route R3, where an avoidance command was issued later than the reference timing. Nevertheless, the obstacle was passed without contact, and the route was completed.

These findings demonstrate that the perception outputs generated by the scene-dependent configurations can be translated into actionable local walking commands under the evaluated conditions.

Table 26. Controlled walking-trial results across four routes.

Route	Environment	Completion	Decisions	Accuracy	Obstacle Avoidance
R1	Outdoor	5/5	103	92.23%	12/12
R2	Outdoor	5/5	104	93.27%	14/14
R3	Indoor	5/5	111	94.59%	15/16
R4	Indoor	5/5	125	92.80%	13/13
Overall	–	20/20	443	93.23%	54/55

Trial-level navigation results are provided in Supplementary Table S1. Across the 20 trials, the mean trial-level command accuracy was 93.24% with a standard deviation of 2.20 percentage points. The aggregate accuracy across all 443 navigation decisions was 93.23%.

5.7.2. Controlled Navigation-Decision Comparison Results

A separate controlled evaluation compared navigation-decision performance across the four perception configurations defined in Section 4.5: SAM, SAD, SAPER-M, and SAPER-D. All configurations were assessed using the same evaluation samples and decision criteria. Table 27 reports overall command accuracy, unsafe FORWARD decisions, and false STOP rates.

SAPER-D achieved the highest navigation-decision accuracy at 96.0%, compared with 92.8% for SAM, 94.0% for SAD, and 94.5% for SAPER-M. This corresponds to a 3.2 percentage-point improvement over the fixed unified-detector baseline (SAM).

Unsafe FORWARD decisions occurred in 7 of 117 hazardous cases for SAM (6.0%), 4 for SAD (3.4%), 5 for SAPER-M (4.3%), and 3 for SAPER-D (2.6%). Accordingly, SAPER-D achieved the lowest unsafe FORWARD rate, reducing the number of such decisions from seven to three relative to SAM.

False STOP rates were 0.91% for SAM and SAD, 0.76% for SAPER-M, and 0.61% for SAPER-D. The proposed configuration therefore achieved the highest overall command accuracy and the lowest observed rates of unsafe FORWARD and false STOP decisions among the evaluated configurations.

Together, the walking trials and controlled navigation-decision comparison provide complementary evidence: the walking trials demonstrate the feasibility of generating executable guidance along controlled routes, while the comparative evaluation characterizes differences in navigation-decision accuracy and safety-related errors between perception configurations.

Table 27. Controlled navigation-decision performance across four configurations.

Configuration	Command Accuracy	Unsafe FORWARD	False STOP
SAM	92.8%	6.0% (7/117)	0.91% (6/657)
SAD	94.0%	3.4% (4/117)	0.91% (6/657)
SAPER-M	94.5%	4.3% (5/117)	0.76% (5/657)
SAPER-D	96.0%	2.6% (3/117)	0.61% (4/657)

Unsafe FORWARD and False STOP rates were calculated over 117 hazardous and 657 non-hazardous frames, respectively.

5.8. Qualitative Results

Figures 6 and 7 present qualitative perception results generated by the proposed SAPER framework in indoor and outdoor environments, respectively.

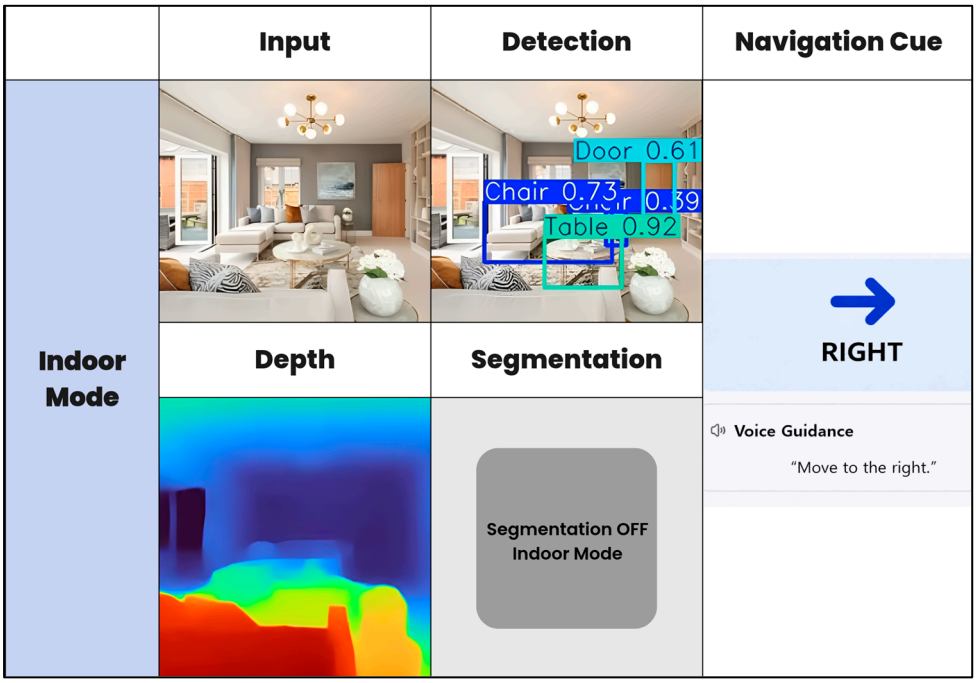


Figure 6. Qualitative results of the proposed SAPER framework in indoor mode (object detection and relative depth cues). Warmer colors in the depth map indicate closer regions, whereas cooler colors indicate farther regions.

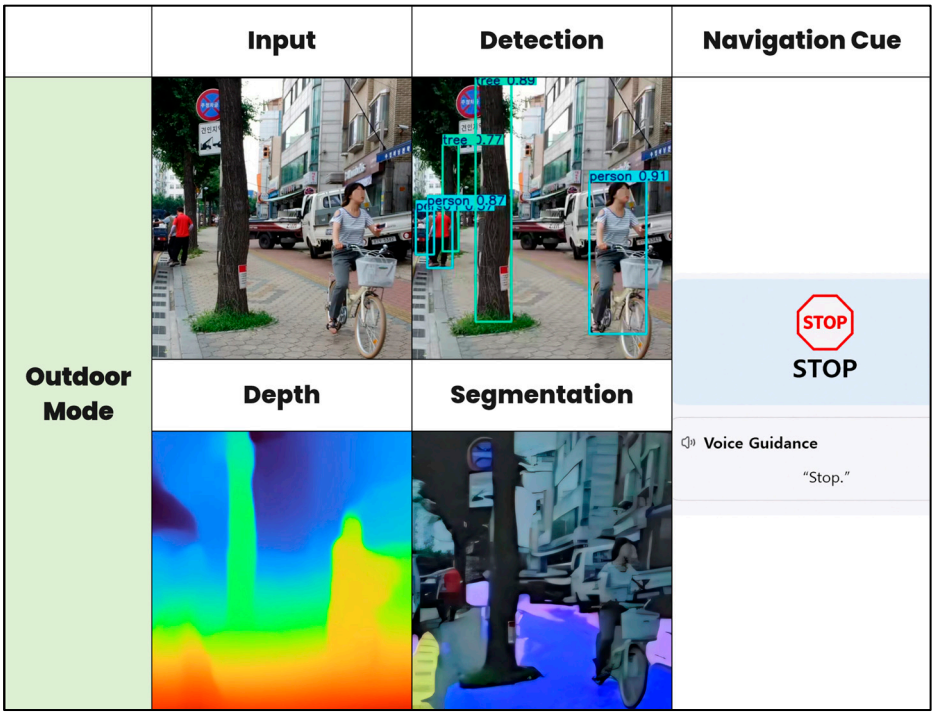


Figure 7. Qualitative results of the proposed SAPER framework in outdoor mode (object detection, semantic segmentation, and relative depth cues). Warmer colors in the depth map indicate closer regions, whereas cooler colors indicate farther regions. Different colors in the segmentation map represent different semantic surface classes.

Figure 6 illustrates the indoor configuration, in which navigation cues are constructed using scene-specific object detection and relative depth estimation without semantic segmentation. Figure 7 illustrates the outdoor configuration, which integrates object detection, semantic segmentation, and relative depth estimation to provide both obstacle-related and surface-region information.

These examples demonstrate how SAPER adapts the composition of perception outputs to the surrounding environment. Indoor mode emphasizes object-level information, while outdoor mode additionally incorporates surface-region cues for local walking guidance.

6. Discussion

6.1. Effects of Environment-Adaptive Perception Routing

The controlled 2 × 2 ablation demonstrates that the two routing mechanisms serve complementary purposes. Scene-dependent segmentation gating primarily reduces computational cost, whereas scene-specific detector selection improves detection accuracy. Compared with the unified detector, the scene-specific detectors improved mAP@50:95 by 3.65 percentage points outdoors and 9.64 percentage points indoors. Meanwhile, SAPER-M and SAPER-D exhibited nearly identical processing latency, indicating that detector specialization introduced limited additional computational cost when combined with routing.

The computational benefit primarily arises from avoiding semantic segmentation in indoor scenes. At the deployed classifier interval of five frames, SAPER-D reduced mean

processing latency by 21.55% and increased sustained throughput by 11.65% compared with SAM. In a separate evaluation with scene classification executed on every frame, steady-state energy consumption decreased by 15.92%. These results demonstrate that selectively executing perception modules can improve the efficiency of heterogeneous on-device perception without requiring a reduction in the input resolution or complexity of individual models.

However, the efficiency gain depends on the distribution of indoor and outdoor scenes. INDOOR_MODE avoids the computationally expensive segmentation branch, whereas OUTDOOR_MODE retains the outdoor perception modules. BOTH_MODE executes both scene-specific detectors together with semantic segmentation and depth estimation when scene classification is uncertain. Consequently, the overall benefit varies according to the proportion of frames assigned to each routing state.

6.2. Significance of Navigation Cue Construction

The proposed framework combines scene-specific perception outputs into navigation cues suitable for local walking assistance. Indoor guidance relies on object detection and relative depth estimation, whereas outdoor guidance additionally incorporates semantic segmentation to identify surface regions relevant to walking. A lightweight decision layer converts these outputs into four navigation commands: FORWARD, LEFT, RIGHT, and STOP.

The controlled walking trials demonstrated the feasibility of this approach, with all 20 trials completed, 93.23% agreement between generated and reference commands, and 54 of 55 obstacle encounters successfully avoided according to the predefined criteria. In the separate navigation-decision comparison, SAPER-D achieved the highest command accuracy of 96.0% and the lowest unsafe FORWARD rate of 2.6% among the four evaluated configurations, compared with 92.8% accuracy and 6.0% unsafe FORWARD for SAM.

These findings indicate that, in addition to improving computational efficiency, scene-aware perception did not degrade and slightly improved navigation-decision accuracy in the evaluated samples.

6.3. Limitations

Several limitations remain. First, the binary scene classifier may encounter difficulties in ambiguous environments, such as covered walkways and semi-outdoor spaces. Although temporal stabilization improves routing consistency, occasional misclassifications may affect the selection of perception modules. Furthermore, sustained incorrect scene predictions in visually ambiguous environments may cause the router to remain in an inappropriate scene-specific mode, potentially omitting perception modules required for the actual environment. Future work will investigate more robust scene classification and routing strategies to mitigate these limitations.

Second, the primary focus of this study is scene-aware perception routing and object-level obstacle avoidance rather than explicit floor-level hazard detection. Semantic segmentation is not executed in INDOOR_MODE because the main indoor hazards are discrete objects and structures, which are addressed by object detection and relative depth estimation. Although low steps may appear as discontinuities in the relative depth map (Figure 2c), these discontinuities are not yet used directly by the navigation decision rules. Consequently, floor-level hazards that are not detected by the object detector may remain unrecognized, potentially resulting in inappropriate FORWARD commands. Reliable detection of such hazards is a challenging research problem in its own right, as it requires robustness to depth estimation noise and diverse environmental factors, including varying illumination, reflective or wet floor surfaces, low-contrast step edges, and textureless

surfaces. Future work will investigate explicit detection of floor-level depth discontinuities and their integration into navigation decision-making.

Third, small or narrow obstacles and heavily occluded objects remain challenging, potentially affecting navigation reliability. Although relative-depth estimation preserves near–far relationships, reliable detection and avoidance of small hazards under varying walking conditions remain insufficiently validated. Future work will focus on improving small-obstacle detection and evaluating distance-dependent warning performance. In addition, SAPER will be combined with lightweight segmentation models on the target NPU, and relative-depth accuracy will be evaluated for outdoor scenes and thin objects such as poles, trees, and bollards.

Finally, the controlled walking trials did not include visually impaired participants. Future work will extend on-device evaluations to more diverse environments and investigate the system's usability and effectiveness through user-centered studies.

7. Conclusions

This study proposed SAPER, a scene-aware perception-routing framework that supports walking assistance for visually impaired individuals on an edge NPU. SAPER dynamically configures scene-specific object detectors and controls semantic segmentation according to the routing state, while maintaining relative depth estimation in all three modes. The proposed design combines segmentation gating for computational efficiency with scene-specific detection for improved perception accuracy. The experimental results demonstrated that the deployed INT8 models largely preserved their pre-deployment accuracy and that scene-specific detectors improved mAP@50:95 by 3.65 and 9.64 percentage points in outdoor and indoor environments, respectively, compared with a unified detector. At the deployed scene-classification interval, SAPER-D reduced mean processing latency by 21.55% and increased sustained throughput by 11.65% compared with the fixed pipeline (SAM), while a separate evaluation demonstrated a 15.92% reduction in steady-state energy consumption. In live indoor–outdoor transition trials, unsafe outdoor routing exposure was 3.2% and 5.4% for indoor-to-outdoor and outdoor-to-indoor transitions, respectively. In the navigation-decision comparison, SAPER-D achieved 96.0% command accuracy and a 2.6% unsafe FORWARD rate, while the controlled walking trials achieved 100% route completion and 93.23% command agreement. These findings demonstrate the potential of scene-aware perception routing to improve computational efficiency and navigation-decision performance on resource-constrained edge devices. Future work will focus on more robust routing, improved detection of floor-level hazards, and user-centered validation with visually impaired participants.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/s26237548/s1, Table S1: Trial-Level Navigation Decision Accuracy and Obstacle Avoidance Results.

Author Contributions: Conceptualization, H.K., T.K. and H.S.K.; methodology, H.K. and H.S.K.; software, H.K. and S.L.; validation, H.K. and S.L.; formal analysis, H.K.; investigation, H.K. and S.L.; data curation, H.K. and S.L.; writing—original draft preparation, H.K.; writing—review and editing, H.K., T.K. and H.S.K.; visualization, H.K.; supervision, T.K. and H.S.K.; project administration, H.S.K.; funding acquisition, H.S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Trade, Industry and Energy (MOTIE), grant number RS-2025-02311228; partly by the Institute of Information & Communications Technology Planning & Evaluation (IITP), grant number RS-2026-25516382; and by the National Research Foundation of Korea (NRF) grant funded by the Korean government, grant number RS-2025-19722970.

Institutional Review Board Statement: The route-navigation experiment was conducted by the four authors, who voluntarily acted as device carriers with full knowledge of the experimental protocol. The purpose of the trials was to evaluate system functionality rather than the mobility performance of visually impaired users. An experimenter accompanied the device carrier throughout each trial to ensure safety. Camera frames were processed in memory on the device and were not stored; only the generated navigation commands were logged for analysis, and no personal or health-related information was collected. Formal institutional ethical review was not sought for these trials.

Informed Consent Statement: All participants were authors of this study and consented to participate. The person shown in Figure 3 is one of the authors and consented to the publication of the image.

Data Availability Statement: The publicly available datasets and videos used in this study are cited in the manuscript and listed in Appendices A and B. The processed datasets generated during this study are not publicly available due to their large size and technical limitations associated with data distribution.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AP	Average precision
COCO	Common Objects in Context
CPU	Central processing unit
DFL	Distribution focal loss
EMA	Exponential moving average
FP32	32-bit floating point
fps	Frames per second
FPN	Feature pyramid network
GPU	Graphics processing unit
INT8	8-bit integer
IoU	Intersection over union
LiDAR	Light detection and ranging
mAP	Mean average precision
mIoU	Mean intersection over union
NCC	Normalized cross-correlation
NMS	Non-maximum suppression
NPU	Neural processing unit
ONNX	Open Neural Network Exchange
p95	95th percentile
RGB	Red, green, and blue
RGB-D	RGB-depth
SAD	Scene-Agnostic Dual
SAM	Scene-Agnostic Monolithic
SAPER	Scene-Aware Perception Routing
SAPER-D	SAPER-Decoupled
SAPER-M	SAPER-Monolithic
SD	Standard deviation
SDK	Software development kit
SLAM	Simultaneous localization and mapping
ToF	Time-of-flight
YOLO	You Only Look Once

Appendix A. Source Provenance of the Roboflow Universe Datasets

Table A1 lists the Roboflow Universe source datasets used in this study, identified by workspace, project, and version, together with the object class retained from each source and the size of the source version at download time. The first fourteen entries were used for the indoor object detection dataset, and the last entry provided the additional outdoor bollard images described in Section 4.2.

Table A1. Source provenance of the Roboflow Universe datasets used in this study.

Alias	Workspace/Project (Version)	Class	Annot.	Split (Train/Valid/Test)
chair1	test-0wunv/chair-zjdo1 v1	Chair	Det.	163/20/20 (203)
chair2	aldri/chair-b3wqb v14	Chair	Det.	11,946/1534/0 (13,480)
table1	aifull/table-person v9	Table	Det.	1320/111/111 (1542)
table3	object-detect-zjtnj/table-gclfn v3	Table	Det.	694/199/99 (992)
table4	mfa/dinning-table-design-style v2	Table	Inst. seg.	1783/30/0 (1813)
table5	w-uk7zo/table-dnymk v1	Table	Det.	429/54/53 (536)
table6	robocup2022p2/table-dsqvh v1	Table	Det.	1041/100/50 (1191)
stair3	class-obuqr/stair-x1rgs v1	Stairs	Det.	270/78/39 (387)
stair5	kaist-9sscg/stairs-segmentation v1	Stairs	Det.	788/190/0 (978)
door1	coins-j8vwp/door-p28ku v1	Door	Det.	856/123/59 (1038)
door2	door-ca23l/door-3oaz9 v2	Door	Det.	393/6/15 (414)
door4	company-qhpep/doors-h4epw v4	Door	Det.	6852/345/154 (7351)
column1	sd-ug5qc/column-2i5mg v2	Column	Det.	850/0/0 (850)
column2	hyo-rrxbt/column-r4bd5-gq6an v4	Column	Inst. seg.	747/50/51 (848)
bollard	project-m4zhi/bollardnavermap v5	Bollard	Det.	156/44/23 (223)

Annot., annotation type; Det. and Inst. seg. are defined in Table 1. The listed counts are the sizes of each source version at download time and do not indicate how many images from each source survived integration. The fourteen indoor source versions total 31,623 images, whereas the final indoor dataset contains 31,167 Roboflow-derived images after class filtering, format conversion, and duplicate removal. Among the 14 indoor sources, "table 4" and "column2" were released as instance segmentation datasets; their polygon annotations were converted to axis-aligned bounding boxes using the minimum and maximum x- and y-coordinates of the polygon vertices, as described in Section 4.2. The remaining 12 indoor sources and the outdoor bollard source were released as object detection datasets.

Appendix B. Walking Videos Used for Scene Classifier Training and Routing Evaluation

Table A2 lists the publicly available first-person walking videos used in this study. The first eight were used only as sources of scene classifier training frames, and the last six were used only for routing evaluation (Section 4.2); no video appears in both roles. Frames were sampled at fixed intervals from the listed segments, and the videos were used solely as input; no frames or derived images were redistributed.

Table A2. Walking videos used for scene classifier training and routing evaluation.

Alias	Location	Setting/Condition	Role	Ref.
SeoulStation	Seoul, Republic of Korea	Railway station interior and exterior, daytime	Training	[45]
BusanStation	Busan, Republic of Korea	Railway station surroundings, daytime	Training	[46]
Seomyeon	Busan, Republic of Korea	Underground shopping street	Training	[47]
JamsilStation	Seoul, Republic of Korea	Station surroundings, night, rain	Training	[48]
Montreal	Montreal, QC, Canada	Underground city and downtown streets	Training	[49]
NewYork	New York, NY, USA	Grand Central Terminal and streets	Training	[50]

Tokyo	Tokyo, Japan	City streets, day and night, rain	Training	[51]
Mitsui	Kisarazu, Japan	Outlet mall, outdoor and covered areas	Training	[52]
Livingston	Livingston, MT, USA	Mall, outdoor-to-indoor transition	Evaluation	[58]
Taipei	Taipei, Taiwan	Mall to street, indoor-to-outdoor transition	Evaluation	[57]
Berlin	Berlin, Germany	Mall entrance and glass-roofed arcade	Evaluation	[53]
Kuala Lumpur	Kuala Lumpur, Malaysia	Mall and covered walkways	Evaluation	[54]
Seoul-Rain	Seoul, Republic of Korea	City streets in daytime rain	Evaluation	[56]
Seoul-Night	Seoul, Republic of Korea	Gangnam and Jongno streets at night	Evaluation	[55]

The author-recorded campus transition sequence is not a public video and is therefore not listed; it is described in Section 4.2.

References

1. World Health Organization. Blindness and Vision Impairment. Available online: https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed on 19 October 2026).
2. Abidi, M.H.; Siddiquee, A.N.; Alkhalefah, H.; Srivastava, V. A comprehensive review of navigation systems for visually impaired individuals. *Heliyon* **2024**, *10*, e31825. https://doi.org/10.1016/j.heliyon.2024.e31825.
3. Wang, W.; Jing, B.; Yu, X.; Sun, Y.; Yang, L.; Wang, C. YOLO-OD: Obstacle detection for visually impaired navigation assistance. *Sensors* **2024**, *24*, 7621. https://doi.org/10.3390/s24237621.
4. Mahendran, J.K.; Barry, D.T.; Nivedha, A.K.; Bhandarkar, S.M. Computer vision-based assistance system for the visually impaired using mobile edge artificial intelligence. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, 19–25 June 2021; pp. 2418–2427. https://doi.org/10.1109/CVPRW53098.2021.00274.
5. Kumari, P.; Hammady, R. Assisting blind people with AI and audio using smart glasses: System design with YOLOv8 variants comparisons. *Multimed. Syst.* **2026**, *32*, 73. https://doi.org/10.1007/s00530-025-02139-z.
6. Mukhiddinov, M.; Cho, J. Smart glass system using deep learning for the blind and visually impaired. *Electronics* **2021**, *10*, 2756. https://doi.org/10.3390/electronics10222756.
7. Elhassan, M.A.M.; Zhou, C.; Khan, A.; Benabid, A.; Adam, A.B.M.; Mehmood, A.; Wambugu, N. Real-time semantic segmentation for autonomous driving: A review of CNNs, transformers, and beyond. *J. King Saud Univ.-Comput. Inf. Sci.* **2024**, *36*, 102226. https://doi.org/10.1016/j.jksuci.2024.102226.
8. Singh, R.; Gill, S.S. Edge AI: A survey. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 71–92. https://doi.org/10.1016/j.iotcps.2023.02.004.
9. Wang, T.; Guo, J.; Zhang, B.; Yang, G.; Li, D. Deploying AI on edge: Advancement and challenges in edge intelligence. *Mathematics* **2025**, *13*, 1878. https://doi.org/10.3390/math13111878.
10. Lei, Y.; Phung, S.L.; Di, Y.; Bouzerdoum, A. CSN: A compact semantic segmentation network for visual scene perception in assistive navigation. *Comput. Vis. Image Underst.* **2026**, *264*, 104665. https://doi.org/10.1016/j.cviu.2026.104665.
11. Scardapane, S.; Baiocchi, A.; Devoto, A.; Marsocci, V.; Minervini, P.; Pomponi, J. Conditional computation in neural networks: Principles and research trends. *Intell. Artif.* **2024**, *18*, 175–190. https://doi.org/10.3233/IA-240035.
12. Han, Y.; Huang, G.; Song, S.; Yang, L.; Wang, H.; Wang, Y. Dynamic neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 7436–7456. https://doi.org/10.1109/TPAMI.2021.3117837.
13. Wang, X.; Yu, F.; Dou, Z.-Y.; Darrell, T.; Gonzalez, J.E. SkipNet: Learning dynamic routing in convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 420–436. https://doi.org/10.1007/978-3-030-01261-8_25.
14. Wu, Z.; Nagarajan, T.; Kumar, A.; Rennie, S.; Davis, L.S.; Grauman, K.; Feris, R. BlockDrop: Dynamic inference paths in residual networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8817–8826. https://doi.org/10.1109/CVPR.2018.00919.
15. Teerapittayanon, S.; McDanel, B.; Kung, H.T. BranchyNet: Fast inference via early exiting from deep neural networks. In Proceedings of the International Conference on Pattern Recognition (ICPR), Cancún, Mexico, 4–8 December 2016; pp. 2464–2469. https://doi.org/10.1109/ICPR.2016.7900006.
16. Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.

17. Hsieh, I.-H.; Cheng, H.-C.; Ke, H.-H.; Chen, H.-C.; Wang, W.-J. A CNN-based wearable assistive system for visually impaired people walking outdoors. *Appl. Sci.* **2021**, *11*, 10026. https://doi.org/10.3390/app112110026.
18. Zhang, Y.; Chen, H.; Yang, K.; Zhang, J.; Stiefelhagen, R. Perception framework through real-time semantic segmentation and scene recognition on a wearable system for the visually impaired. In Proceedings of the IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 863–868. https://doi.org/10.1109/RCAR52367.2021.9517086.
19. Liu, H.; Liu, R.; Yang, K.; Zhang, J.; Peng, K.; Stiefelhagen, R. HIDA: Towards holistic indoor understanding for the visually impaired via semantic instance segmentation with a wearable solid-state LiDAR sensor. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, QC, Canada, 11–17 October 2021; pp. 1780–1790. https://doi.org/10.1109/ICCVW54120.2021.00204.
20. Estrada, R.; Ponce, V.; Maldonado, J.; Vasco, I. A real-time indoor object detection and distance estimation system for visually impaired individuals. *Procedia Comput. Sci.* **2025**, *265*, 41–48. https://doi.org/10.1016/j.procs.2025.07.154.
21. Yu, X.; Saniie, J. Visual impairment spatial awareness system for indoor navigation and daily activities. *J. Imaging* **2025**, *11*, 9. https://doi.org/10.3390/jimaging11010009.
22. Ou, W.; Zhang, J.; Peng, K.; Yang, K.; Jaworek, G.; Müller, K.; Stiefelhagen, R. Indoor navigation assistance for visually impaired people via dynamic SLAM and panoptic segmentation with an RGB-D sensor. In Proceedings of the Joint International Conference on Digital Inclusion, Assistive Technology and Accessibility (ICCHP-AAATE), Lecco, Italy, 11–15 July 2022; pp. 160–168. https://doi.org/10.1007/978-3-031-08648-9_19.
23. Ashiq, F.; Asif, M.; Ahmad, M.B.; Zafar, S.; Masood, K.; Mahmood, T.; Mahmood, M.T.; Lee, I.H. CNN-based object recognition and tracking system to assist visually impaired people. *IEEE Access* **2022**, *10*, 14819–14834. https://doi.org/10.1109/ACCESS.2022.3148036.
24. Davanthapuram, S.; Yu, X.; Saniie, J. Visually impaired indoor navigation using YOLO based object recognition, monocular depth estimation and binaural sounds. In Proceedings of the IEEE International Conference on Electro/Information Technology (EIT), Mt. Pleasant, MI, USA, 14–15 May 2021; pp. 173–177. https://doi.org/10.1109/EIT51626.2021.9491913.
25. Seo, J.-Y.; Sin, B.-J.; Lee, H.-S.; Choi, S.-M.; Lee, B.-J. Smart vest for the visually impaired with YOLOv11 and MiDaS. *J. Korea Inst. Electron. Commun. Sci.* **2025**, *20*, 899–908. https://doi.org/10.13067/JKIECS.2025.20.4.899.
26. Joshi, R.C.; Singh, N.; Sharma, A.K.; Burget, R.; Dutta, M.K. AI-SenseVision: A low-cost artificial-intelligence-based robust and real-time assistance for visually impaired people. *IEEE Trans. Hum.-Mach. Syst.* **2024**, *54*, 325–336. https://doi.org/10.1109/THMS.2024.3375655.
27. Chen, Z.; Liu, X.; Kojima, M.; Huang, Q.; Arai, T. A wearable navigation device for visually impaired people based on the real-time semantic visual SLAM system. *Sensors* **2021**, *21*, 1536. https://doi.org/10.3390/s21041536.
28. Mao, W.; Zhang, J.; Yang, K.; Stiefelhagen, R. Panoptic Lintention Network: Towards efficient navigational perception for the visually impaired. In Proceedings of the IEEE International Conference on Real-Time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 857–862. https://doi.org/10.1109/RCAR52367.2021.9517615.
29. Kan, M.; Zhang, L.; Liang, H.; Zhang, B.; Fang, M.; Liu, D.; Shan, S.; Chen, X. eLabrador: A wearable navigation system for visually impaired individuals. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 12228–12244. https://doi.org/10.1109/TASE.2025.3541055.
30. Jiang, J.; Ananthanarayanan, G.; Bodik, P.; Sen, S.; Stoica, I. Chameleon: Scalable adaptation of video analytics. In Proceedings of the ACM SIGCOMM Conference, Budapest, Hungary, 20–25 August 2018; pp. 253–266. https://doi.org/10.1145/3230543.3230574.
31. Xu, R.; Lee, J.; Wang, P.; Bagchi, S.; Li, Y.; Chaterji, S. LiteReconfig: Cost and content aware reconfiguration of video object detection systems for mobile GPUs. In Proceedings of the 17th European Conference on Computer Systems (EuroSys), Rennes, France, 5–8 April 2022; pp. 334–351. https://doi.org/10.1145/3492321.3519577.
32. Wang, S.; Kim, H.; Yeo, J.; Kim, J. YOLOv10-based multi-scale variant object detection for multi-category PPE non-compliance monitoring on construction sites. *Sci. Rep.* **2026**, *16*, 24591. https://doi.org/10.1038/s41598-026-53052-y.
33. Wang, S.; Zhang, E.-L.; Hong, R.-L.; Kim, J. An enhanced YOLOv10 framework for small-object safety helmet detection on construction sites. *Sci. Rep.* **2026**, *16*, 31411. https://doi.org/10.1038/s41598-026-60485-y.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. https://doi.org/10.1109/CVPR.2016.90.

35. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944. https://doi.org/10.1109/CVPR.2017.106.
36. Ranftl, R.; Lasinger, K.; Hafner, D.; Schindler, K.; Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 1623–1637. https://doi.org/10.1109/TPAMI.2020.3019967.
37. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324. https://doi.org/10.1109/ICCV.2019.00140.
38. ONNX. Open Neural Network Exchange. Available online: https://onnx.ai/ (accessed on 3 June 2026).
39. Wei, L.; Ma, Z.; Yang, C.; Yao, Q. Advances in the neural network quantization: A comprehensive review. *Appl. Sci.* **2024**, *14*, 7445. https://doi.org/10.3390/app14177445.
40. AI Hub. Sidewalk Walking Video Dataset. Available online: https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=189 (accessed on 3 June 2026).
41. Roboflow. Roboflow Universe: Open Source Computer Vision Datasets. Available online: https://universe.roboflow.com/ (accessed on 3 June 2026).
42. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
43. Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1452–1464. https://doi.org/10.1109/TPAMI.2017.2723009.
44. AI Hub. Virtual Indoor Space 3D Synthetic Data. Available online: https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=data&dataSetSn=71765 (accessed on 12 August 2026).
45. Dingo Walk. [4K] Seoul Station and Seoullo Relax Walking Tour. YouTube. Available online: https://youtu.be/VUSCRC4aLQ8 (accessed on 16 September 2026).
46. Walkerscape. [4K] Relaxing Walk around Busan Station on a Saturday Afternoon. YouTube. Available online: https://youtu.be/tInHpWVey8c (accessed on 16 September 2026).
47. MT Walk. [4K] Korea Walking Tour 10: Seomyeon Underground Shopping Street, Busan. YouTube. Available online: https://youtu.be/w9y38Q2Tr3M (accessed on 16 September 2026).
48. Hello Korea. Walking in the Rain around Jamsil, Seoul (Rainy Night). YouTube. Available online: https://youtu.be/P9gPiww0Qgc (accessed on 16 September 2026).
49. Elena Walks. Montreal Underground City Tour: Downtown Centre-Ville Walking Experience [4K]. YouTube. Available online: https://youtu.be/1jBTIFC3vYY (accessed on 16 September 2026).
50. Julien In NYC. New York City Walking Tour [4K]: Grand Central Terminal. YouTube. Available online: https://youtu.be/Kdg7lCiAoy4 (accessed on 16 September 2026).
51. World Wanderings. Tokyo, Japan Walking Tour in the Rain: Walk the Streets of Japan Day & Night, 4K HDR 60fps. YouTube. Available online: https://youtu.be/wm27ElpSxbM (accessed on 16 September 2026).
52. Walking Tokyo. Walk in Mitsui Outlet Park Kisarazu, 4K HDR Japan. YouTube. Available online: https://youtu.be/5UV42-yUhd4 (accessed on 16 September 2026).
53. Walking OZ. Mall of Berlin and Potsdamer Platz, 4K Shopping Walking Tour. YouTube. Available online: https://www.youtube.com/watch?v=TV0MWsaA6Eg (accessed on 12 August 2026).
54. Ten Degrees Warmer. Kuala Lumpur Walking Tour: Pavilion Mall to Suria KLCC by Covered Walkway. YouTube. Available online: https://www.youtube.com/watch?v=zatNqIEgOD0 (accessed on 12 August 2026).
55. ZEROBIG Walking Tour. [Korea] Seoul 4K Walking Tour: A Dazzling Summer Night in Gangnam and Jongno. YouTube. Available online: https://youtu.be/tI42I3eweUM (accessed on 16 September 2026).
56. Seoul_4K. A Rainy-Day Walk in Downtown Seoul: Seoul City Hall, Deoksugung Stone-Wall Road, Gyeongbokgung. YouTube. Available online: https://www.youtube.com/watch?v=G0C5-2kOU44 (accessed on 16 September 2026).
57. Raksantara Everywhere. Walking Tour: From Taipei City Hall Mall to Iconic Taipei 101, 4K. YouTube. Available online: https://www.youtube.com/watch?v=uamNPBE37Tk (accessed on 12 August 2026).
58. Walking & Driving Incognito. 4K60 Tour of Livingston Mall, Livingston, New Jersey. YouTube. Available online: https://www.youtube.com/watch?v=KiTEMtRId40 (accessed on 12 August 2026).

59. Song, S.; Lichtenberg, S.P.; Xiao, J. SUN RGB-D: A RGB-D scene understanding benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 567–576. https://doi.org/10.1109/CVPR.2015.7298655.
60. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8, Version 8.0.0. 2023. Available online: https://github.com/ultralytics/ultralytics (accessed on 19 October 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.