

Original Article

Intelligent Pathway Perception System using YOLO and Multi-Architecture Deep Learning for Obstacle and Floor Hazard Detection

Sadia Patka¹, Rahul Khokale², Mohammad Sharfoddin Khatib³

^{1,2}Department of Computer Science & Engineering, G H Rasoni University Saikheda, Madhya Pradesh, India.

³Department of Computer Science & Engineering, Anjuman College of Engineering & Technology, Maharashtra, India.

¹Corresponding Author: sadia.patka.cse@ghru.edu.in

Received: 07 May 2026

Revised: 20 August 2026

Accepted: 29 August 2026

Published: 27 October 2026

Abstract - Independent and safe movement of the visually impaired people is a very urgent issue, especially in indoor areas, where the dynamic barriers and physical features like furniture, stairs and uneven surfaces are major hazards. The current assistive systems are usually concentrated on specific tasks, leading to a disjointed understanding of the environment and a lack of reliability. The proposed framework uses the YOLOv8 to detect obstacles in the indoor environment, then extracts features using parallel CNN and ResNet-50 and fuses them with an attention mechanism to get the surface of the pathway, which is helpful for context-aware perception. Training and evaluation are carried out on the selected categories of indoor objects from the Microsoft COCO dataset, derived from the task. The ResNet-50 model is initialized with ImageNet-pretrained weights, while pseudo-labels are introduced for surfaces to support surface inference by leveraging the COCO object-scene relationships in a weakly supervised manner. Therefore, it does not need an extra floor-hazard training data set, however, the feature initialization is based on the principle of transfer learning. The results of the experiments prove that APPF has a detection accuracy of 96.8%, a precision of 95.1% and, a recall of 94.3% and a mAP at 0.5 of 0.91 in object detection tasks. The context-aware surface inference module has an accuracy of classification of 95.5% and a false detection rate of 4.5%. Relative analysis with baseline CNN and single-stage detectors demonstrates 17.4% improvement in the detection performance and 36.5% decrease in the latency, maintaining real-time inference at 27 FPS. It is innovative in that it uses only the COCO dataset, but it allows dual-level perception to be done by architectural innovation and contextual learning to indoor settings.

Keywords - Assistive navigation systems, Object detection, Context-Aware surface inference, Deep learning for visual assistance, Obstacle detection for visually impaired.

1. Introduction

The issue of safe and independent movement of the visually impaired people is a burning problem in contemporary society, where about 253 million persons across the globe have diverse levels of visual impairment [9]. The indoor environment, where obstacles are dynamic, and the furniture layout, stairs, and uneven floors increase the risks, is more dangerous and cannot be sufficiently covered by the traditional navigation assistance, like white canes, in complex real-time cases [1]. The introduction of assistive technologies that rely on vision as the input and are driven by the development of deep learning and embedded computing has created new opportunities of intelligent environmental perception [2].

The You Only Look Once (YOLO) family of detectors has greatly improved real-time detection of indoor obstacles with a competitive balance of speed and accuracy that are

required in assistive navigation applications [5]. The latest architecture, YOLOv8, adds: an anchor-free detection head, decoupled classification-regression branches, better feature pyramid networks, which has provided state-of-the-art detection performance and real-time inference throughput [6]. Even though there are advances in real-time detection of objects, indoor assistive navigation goes beyond the recognition of individual objects.

The visually challenged person should also know if the immediate way is navigable, blocked or if there are structural hazards (e.g. a staircase). Current object detectors usually give information about the position and certainty for the object, but pathway-level interpretation needs to know the context of objects, the structure of the scene and the surface around the detected object. Therefore, just detecting objects in an environment does not include enough information for successful indoor navigation. This encourages an integrated



perception framework that can fuse spatial perception of obstacles with the inference of the pathway surface in context and preserve real-time computation. Although these are made, current systems of assistive perception have a fundamental weakness, namely that they consider single sub-tasks individually, either searching discrete obstacles [3] or attempting to determine the safety of a pathway [7], but seldom attempt to give a comprehensive picture of the environment in one coherent system.

The incomplete situational awareness of this fragmented design and the decreased reliability of safety-critical navigation decisions are also experienced [4]. Moreover, the majority of multi-task assistive systems use heterogeneous datasets in the various perception modules, which presents training inconsistencies and further complicates the deployment of the systems [8, 9].

Previous work has shown that every one of these technologies (YOLO-based obstacle detection, CNN-based scene understanding and multi-stream feature extraction) has its own merit, however three limitations are clearly shown. First, rather than jointly tackling object detection and the interpretation of a scene/pathway, most approaches focus on one or the other.

Secondly, the use of various perception functions in many approaches typically relies on task-specific or heterogeneous data sets, which requires more annotation and deployment. Third, conventional single-backbone architectures offer few means to access both fine-grained local textural characteristics (e.g., pathways) and global structural context (e.g., surfaces) that are important for pathway surface reasoning.

Thus, there is a need for a framework that combines obstacle localization with contextual pathway-surface inference, trained from a shared source of information, and using complementary multi-architecture representations, in real-time and with RGB signals. To fill this void, the present study proposes a novel method which incorporates the YOLOv8 detection branch, the parallel CNN-ResNet-50 feature extraction branch, attention fusion branch, and inference branch for weakly supervised object-scene context in a single network structure, namely APPF for object image retrieval.

The novelty of APPF is on the integration of complementary perception mechanisms instead of using YOLOv8, CNN, ResNet-50 or attention alone. Specifically, the framework is designed to adopt local-textural and global-structural representations, with the help of attention-guided fusion to make surface inferences towards the pathway level of contextual perception. APPF has weak supervision from the object-scene relationship in the chosen set of COCO indoor scenes, instead of the traditional method of floor-hazard data annotation.

This allows obstacle detection and contextual surface interpretation to be done in a common perception pipeline running on RGB data without requiring a separate training data set. Current assistive perception approaches have achieved significant success in real-time detection of obstacles, understanding indoor scenes and context-aware navigation, but such tasks are often considered as isolated. While methods that leverage YOLO achieve accurate object localization, they often have restricted contextual interpretation at the pathway level and methods that rely on scene classification and surface understanding focus on the semantic interpretation of the environment and do not take advantage of real-time object localization as part of the same perception pipeline. Multi-stream and feature-fusion architectures facilitate the learning of representation, but exploiting local texture and global structural information for obstacle-surface perception jointly is still difficult for these architectures.

Moreover, some of the currently proposed methods rely on learning based on task-specific or heterogeneous learning sources for various perception functions. To tackle these challenges, APPF introduces the new YOLOv8-based obstacle localization, the parallel CNN-ResNet-50 feature extraction, the attention-driven feature fusion, and the weakly supervised contextual surface inference. All these components are integrated into a unified framework to perform the obstacle localization, feature extraction, feature fusion, and surface inference on RGB images using a common COCO training source.

The primary contributions of this paper are as follows:

- 1) An innovative dual-task perception framework that jointly detects obstacles at real-time and context-sensitive inference of the pathway surface in a single end-to-end pipeline and without any domain-specific annotations.
- 2) Parallel CNN and ResNet-50 feature extraction branches are integrated with an attention-based mechanism of fusion that generates context-enriched feature representations to allow complementary local-textural and global-structural encoding and to understand the indoor scenes better.
- 3) A surface classification module based on Bayesian posterior that uses object-scene co-occurrence statistics using COCO annotations to classify the surface as navigable with a 95.5% classification rate with no explicit floor hazard supervision.
- 4) Experimental evaluation demonstrates 96.8% detection accuracy, an mAP@0.5 of 0.91, and a 17.4% improvement over the baseline CNN in detection performance. The proposed APPF achieves a 36.5% reduction in inference latency relative to the baseline CNN (58.3 ms to 37.0 ms), maintaining real-time processing at approximately 27 FPS. Comparative and ablation analyses further demonstrate the effectiveness of the proposed multi-architecture framework.

2. Related Work

Assistive navigation by vision has developed significantly using deep learning, of which the early techniques used stereo vision and structured light sensors to map the space. Their great computational requirements and sensitivity to the indoor illumination conditions inspired the shift towards fully RGB-based deep learning methods. The experiment [10] proved the use of YOLOv5 to detect obstacles in autonomous smart wheelchairs, with a good performance in the navigation of various indoor settings.

The research article [11] suggested a color-feature deep-learning platform of mobile robots in an indoor environment, in order to use convolutional networks. It was determined that appearance-based features can be used to track dynamic environments. The paper [12] also enhanced the indoor automated guided vehicle navigation based on the deep action learning, which integrates visual attributes and learned navigation policies to travel through the unstructured environments. These early works validate the usefulness of RGB-based deep learning to indoor navigation, but only in single-task tasks without a unified understanding of scenes.

The YOLO family of detectors has become the new standard in object detection in real-time in the field of navigation. The study research article [13] proposed a dual-stream deep learning model to classify indoor scenes, which showed that multi-path feature extraction can offer a greater accuracy in the discrimination of scenes when used by robots.

The article [14] investigated the use of Wi-Fi channel state information in the paper study to monitor indoor safety by using deep learning, which demonstrated the relevance of context-aware learning to the safety evaluation of the indoors.

The paper [15] dealt with the problem of simultaneous object detection and distance estimation of autonomous vehicles in an indoor setting with the help of efficient single-stage architectures, which verified the feasibility of the solutions based on the use of YOLO in the context of limited resources. A context-aware multi-level information fusion system that integrates semantic segmentation and object detection to enable indoor mobile robot safe navigation was proposed in the study [16], which showed that feature-level fusion can be of great benefit to environmental understanding than a single-modality solution.

Multi-architecture and hybrid deep learning models have gained momentum as the focus of study when it comes to complex perception tasks. The paper highlighted to [17] an efficient multi-object detection and intelligent navigation system that can be used by visually impaired people via artificial intelligence, which demonstrated a better detection rate with lower inference latency. In the study [18], the authors developed a multimodal obstacle avoidance control using

Deep Reinforcement Learning and obtained adaptive navigation in dynamic obstacle-rich conditions. The study [19] performed a thorough survey of image-based obstacle detection schemes to unmanned vehicle navigation and concluded that multi-architecture systems excel over single-model systems in challenging settings.

The experiment [20] specifically tested indoor obstacle detecting prototypes to visually impaired individuals, noting that the existing ones did not integrate obstacle detecting and floor hazard detection, diminishing the useful safety assurances. The researchers argue that the CNN-based structures could be successfully used to detect structural surface anomalies in the building inspection case [21], and therefore, they can be transferred to detecting floor hazards in an indoor setting. The study [22] demonstrated that clustering and deep learning has lower false positive rates in multi-object detection, which can be directly applied to high-precision assistive navigation.

Against such improvements, there is an acute nullity: The reviewed literature reveals limited investigation that can offer a single framework which can both detect obstacles and infer the surface of the pathway to be followed by the visually impaired when inside the building and to perform the task with one dataset.

The current methods are either localizing obstacles, classifying surfaces or involve heterogeneous multi-dataset training streams. The APPF fills this void by combining YOLOv8 detection and multi-architecture feature enhancement and context-sensitive inference in a single COCO-trained model.

All of these studies collectively show three directions of efficient single-stage obstacle detection, context-aware scene understanding, and multi-architecture feature learning. These capabilities are considered separately, however, and usually explored in isolation in perception pipelines supervised separately for each capability. Obstacle detectors are good at localizing objects and do not offer much in terms of pathway semantics, while scene-understanding approaches are good at providing pathway semantics, but are not necessarily real-time in their ability to detect obstacles. Multi-stream architectures offer better feature representation, but have yet to be tested with respect to joint obstacle localization and weakly supervised pathway-surface inference, comparative analysis shown in Table 1.

However, APPF sits in the middle of these directions by keeping the real-time object localization functions of YOLOv8, adding complementary CNN-ResNet-50 feature representations, adding attention-based fusion, and utilizing context-aware surface information from object-scene relationships using the same COCO training source.

Table 1. Comparative analysis of existing indoor assistive perception approaches and the proposed APPF

Study	Detection / Perception Approach	Obstacle Detection	Surface/Scene Understanding	Feature Fusion	Dataset Strategy	Main Limitation
[10]	YOLOv5-based	Yes	No	No	Task-specific	Primarily obstacle localization
[13]	Dual-stream DL	Limited/No*	Scene classification	Multi-stream	Existing dataset	Does not provide unified obstacle-surface perception
[15]	Single-stage detection	Yes	No	No	Task-specific	Detection/distance emphasis
[16]	Segmentation + detection	Yes	Yes	Multi-level fusion	Multiple/task-specific supervision*	Increased perception complexity
[17]	AI-based multi-object detection	Yes	Limited	No*	Existing dataset	Limited pathway-surface reasoning
[20]	Indoor obstacle-detection system	Yes	Limited	No*	Application-specific	Obstacle/floor-hazard integration remains limited

3. System Overview and Problem Formulation

The Adaptive Pathway Perception Framework (APPF) shaping of indoor navigation is a collaborative detection and classification issue. Considering an RGB video stream facing forward, of a monocular camera, the system should simultaneously detect the obstacles in the interior and their spatial area, as well as determine the type of surface of the navigable foreground area in real-time.

The framework combines YOLOv8 detection with a multi-architecture deep learning improvement, which runs intermediate feature maps along parallel CNN and ResNet-50 paths, with the fusion of an attention mechanism and classified by a surface inference module, which only uses COCO-trained representations.

3.1. Definition of Indoor Navigation Perception Problem

The problem of indoor navigation perception is formulated in the form of a combined detection and surface classification. $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$ represents a height H and width W input RGB image. A mapping of obstacles is learned by the objective of obstacle detection:

$$\mathbf{f}_d: \mathbf{I} \rightarrow \mathbf{D} = \{(\mathbf{b}_i, \mathbf{c}_i, \mathbf{s}_i)\} \quad (1)$$

In which every identified object is represented by a bounding box $\mathbf{b}_i = (\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}_i, \mathbf{h}_i)$, the type label $\mathbf{c}_i \in \mathbf{C}_{obj}$ and a confidence score $\mathbf{s}_i \in [0, 1]$.

N is the number of objects that are identified in the scene. At the same time, the surface inference objective learns:

$$\mathbf{f}_s: \mathbf{I} \rightarrow \mathbf{L}_s, \mathbf{L}_s \in \{\text{flat floor, staircase, obstacle zone, clear path}\} \quad (2)$$

The goal of the joint optimization is formulated as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left[\mathbf{L}_{\text{det}}(\mathbf{f}_d(\mathbf{I}; \boldsymbol{\theta}_d), \mathbf{D}_{gt}) + \lambda * \mathbf{L}_{\text{cls}}(\mathbf{f}_s(\mathbf{I}; \boldsymbol{\theta}_s), \mathbf{L}_{gt}) \right] \quad (3)$$

In which, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_d, \boldsymbol{\theta}_s\}$ jointly represent both modules, $\mathbf{D}_{gt}$ and $\mathbf{L}_{gt}$ denote ground truth detection and surface annotations, respectively, and λ is a trade-off coefficient that regulates relative importance of each of these tasks to the overall optimization objective.

This formulation can be used as an approach to learning shared features that can be used in both tasks in a single forward pass.

3.2. System Requirements for Visually Impaired Assistance

The APPF is created to meet certain operational needs of the real-life deploying assistive. The system should be able to maintain real-time inference of at least 20 FPS to give real-time environmental awareness without noticeable latency.

The detection accuracy should be over 90% mAP 0.5 to reduce the amount of missed safety-critical obstacles. Inference latency should be less than 50 ms/frame in order to provide responsive audio or haptic feedback.

The system should be able to handle regular 640×640 pixel RGB input without the need of depth sensors or special hardware, lowering the cost of the devices and the cost of deployment. The entire perception abilities should be enabled with a mere COCO dataset, without the involvement of any specific annotation pipelines.

The framework should give structured, interpretable environmental descriptions, which can be directly translated to user guidance signals. This dataset limitation only applies to the task of obstacle detection and contextual surface inference in the Microsoft COCO dataset. This limitation doesn't apply to external weights learned for feature extraction.

In particular, the weights of ResNet-50 are initialized using those learned from ImageNet, and no extra data from floor hazards is leveraged for task-specific training. The different kinds of pretrained initialization and task-specific supervision are preserved in the experimental methodology.

3.3. Overview of the Adaptive Pathway Perception Framework (APPF)

The APPF works with three processing stages, which are interrelated. The Stage 1 operation involves the input frame RGB is resized to 640×640 pixels and then sent through the YOLOv8 backbone, a CSP-Darknet53 architecture with C2f modules and a path aggregation network (PAN) neck to produce multi-scale feature maps at strides of 8, 16, and 32. The anchor-free detection head gives predictions of bounding boxes, class probabilities, and objectness scores of the detected indoor objects that belong to 21 COCO indoor categories. Stage 2 is an intermediate step that involves parallel CNN and ResNet-50 streams operating on a stride-16 feature map.

The CNN branch uses four consecutive 3×3 convolutional blocks with batch normalization and ReLU to obtain local textural information, and the ResNet-50 branch obtains the global structural representations through the skip connections between the residual. The tensors of resultant features are concatenated and fed into an attention-based fusion module that uses channel-wise squeeze-and-excitation recalibration.

In Stage 3, Stage 2 refined fused feature tensor is fed to the context-sensitive surface inference, a fully connected classifier which classifies the surface type of the navigable foreground area. The composite APPF mapping can be given as:

$$\hat{Y} = \mathbf{g} \left(\mathbf{F}_{\text{att}}(\mathbf{F}_{\text{CNN(I)}} \oplus \mathbf{F}_{\text{ResNet(I)}}), \mathbf{f}_{\text{d(I)}} \right) \quad (4)$$

In which $\oplus$ is the concatenation of features, $\mathbf{F}_{\text{att}}$ the attention fusion operator and $\mathbf{g}$ the joint output function that gives both the detection and surface classification output. The

whole pipeline only runs once without iteration and can be refined, allowing 27 FPS inference. Modularity The modularity enables single components to be updated without retraining the entire pipeline.

3.4. Assumptions and Constraints

There are four major constraints of the APPF. First, it is based on inputs that are simply RGB images by a single forward-facing monocular camera, with no depth, infrared or multi-spectral sensors.

Second, the framework adopts the COCO dataset annotated by Microsoft as the source of task-specific training and evaluation data, and filters the space of 80 categories of COCO to 21 categories relevant to indoor scenes. Surface inference is based on weakly supervised pseudo-labels, which are inferred by object-scene co-occurrence relationships, instead of using explicitly labelled floor-hazard images.

Weights are initialized with those of the ResNet-50, which were pre-trained on ImageNet. Thus, the restriction of the data set holds true for task-specific supervision and is not a restriction on being able to learn all model parameters from COCO images.

Third, sufficient indoor lighting is supposed to be ensured in order to provide successful RGB-based detection, and its performance is likely to suffer in the case of harshly underlit environments. Fourth, it is assumed that there is a constant camera mounting angle; a large change in pitch or roll (more than about +/-15 degrees) is not corrected, and can cause the bounding box spatial calibration to be inaccurate at deployment.

4. Proposed Methodology

The APPF framework consists of three main elements into a single pipeline: the YOLOv8 detection module, used to locate obstacles in real-time, the multi-architecture CNN+ResNet improvement, used to extract enriched features, and the attention-based fusion and context-enhanced surface inference. The individual components take a standardized 640 x 640 pixel input, and the whole system is end-to-end trained on a joint detection and classification loss. The backbone is the YOLOv8, which is the main feature extractor and object localizer, and the dual-branch CNN+ResNet, which is the complementary representations that enhance surface context inference without the need of extra data. The end-to-end APPF pipeline of indoor scene analysis shown in Figure 1 combines the obstacle detection feature of the YOLOv8 with a two-branch CNN-ResNet feature enhancement feature. The fusion process, which is motivated by attention, facilitates the context-sensitive surface inference, both localizing objects and classifying surfaces. The single architecture uses standardized inputs and is optimized as a combination of detection and classification losses.

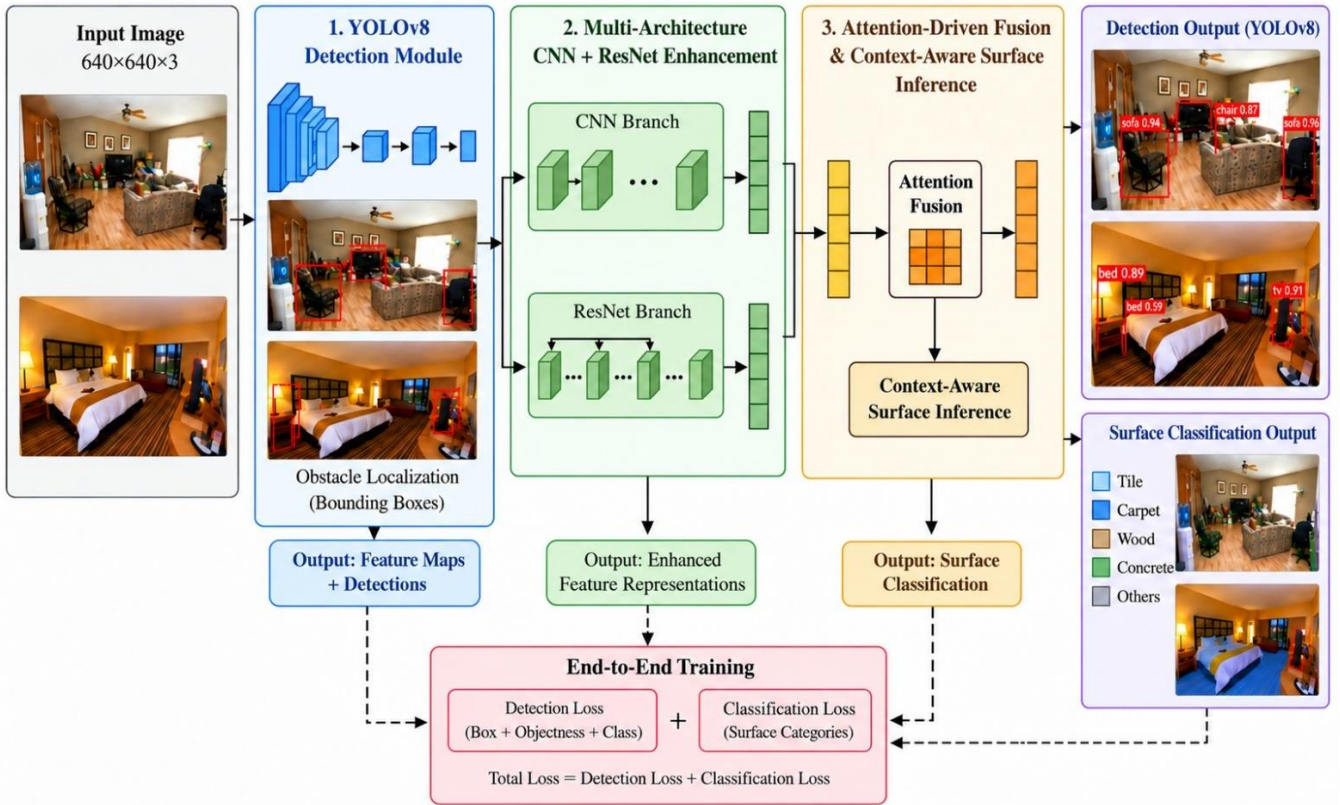


Fig. 1 Adaptive Pathway Perception Framework (APPF) for indoor obstacle detection and surface understanding

4.1. YOLOv8-based Object Detection Module

YOLOv8 uses an anchor-free single-stage detector that splits the input image into an S by S grid, with each pixel directly predicting the existence of an object, bounding box locations and probabilities of each object. The detection head uses decoupled branch classification and regression, which enhances gradient flow and minimizes the interference of the tasks in training.

The backbone is CSP-Darknet53 on C2f modules (to extract features efficiently at multiple scales), and the PAN neck uses features at different scales to detect small objects more effectively in dense indoor scenes.

The per-cell prediction tensor is defined as:

$$\hat{Y}_{i,j} = \{b_x, b_y, b_w, b_h, p_{obj}, C^1, C^2, \dots, C_k\} \quad (5)$$

The grid cell (i, j) predicts the coordinates of the bounding boxes, objectness score and k class probabilities. This anchor-free formulation removes the manual anchor design and allows scale-adaptive predictions of a wide range of sizes of indoor objects.

Bounding box center coordinates are predicted relative to the grid cell offset using sigmoid activation:

$$b_x = \sigma(t_x) + c_x; \quad b_y = \sigma(t_y) + c_y \quad (6)$$

In this case, $\sigma(\bullet)$ is the sigmoid function, (t_x, t_y) are raw network outputs and (c_x, c_y) is the upper-left corner of the grid cell. The sigmoid makes the predictions to be spatially confined within the given cell boundary.

The width and height of the box are decoded through exponential scaling:

$$b_w = p_w \cdot e^{t_w}; \quad b_h = p_h \cdot e^{t_h} \quad (7)$$

The dimensions of the previous anchor p_w and p_h are removed in YOLOv8; t_w and t_h are optimized directly. The exponential guarantees the box dimensions of training are strictly positive.

The objectness confidence score, which is the probability of a grid cell to have a foreground object, is calculated as:

$$p_{obj} = \sigma(t_o) \quad (8)$$

The confidence of the presence of objects is high with values close to 1.0. In the NMS post-processing, those cells whose p_{obj} is less than a cut-off of 0.25 are eliminated to inhibit background predictions.

The composite YOLOv8 detection loss is a combination of coordinate regression, objectness and classification loss:

$$L_{total} = \lambda_{coord} \cdot L_{coord} + \lambda_{obj} \cdot L_{obj} + \lambda_{cls} \cdot L_{cls} \quad (9)$$

The contribution of each term is balanced by setting hyperparameters $\lambda_{coord} = 7.5$, $\lambda_{obj} = 0.5$ and $\lambda_{cls} = 0.5$. L_{coord} is based on CIOU loss, L_{obj} is based on binary cross-entropy, and L_{cls} is based on multi-class cross-entropy.

The quality of bounding box overlap of NMS and evaluation is Intersection-over-Union:

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (10)$$

A, B, are the ground truth and predicted bounding boxes, respectively. At the mAP@0.5, which is used in evaluation, detections with IoU of 0.50 and above are regarded as true positive.

Mean Average Precision: This is an aggregate of the accuracies of per-class detection of all N classes of indoor objects:

$$mAP = \left(\frac{1}{N}\right) \cdot \sum_{i=1}^N AP_i = \left(\frac{1}{N}\right) \cdot \sum_{i=1}^N \int_0^1 P(R) dR \quad (11)$$

Each AP_i is obtained by summing the precision-recall curve of class i at the IoU@0.5 threshold, the sum of which is estimated using the 101-point interpolation scheme on the recall thresholds $[0, 0.01, \dots, 1.0]$.

4.2. Multi-Architecture Deep Learning Enhancement (CNN + ResNet Feature Extraction)

Multi-architecture enhancement module runs the intermediate feature maps of the YOLOv8 backbone on two parallel streams. The CNN block is made up of four consecutive 3×3 blocks of convolutional blocks, each consisting of convolution, batch normalization, and ReLU activation, that detect local textural patterns of surface transitioning, furniture edges, and structural boundaries that guide inference about the type of surface. The additional ResNet-50 branch adds residual skip connections, which add the sums of input and transformed feature maps, which does not degrade gradients, and provides global structural context, similar to the full scene width.

The stride-16 intermediate feature map is shared between the two branches, and the branches each generate a feature tensor of the same dimensionality $C = 256$. The local-textural (CNN) and global-structural (ResNet) representations are complementary, which guarantees that local floor hazard cues that are spread over different spatial scales are represented

without extra training information on top of COCO annotations. This pattern of an architectural decision is preconditioned by the fact that the transition of surfaces at the floors usually takes the form of the local patterns of edges and the overall spatial organization of surrounding pieces of furniture.

The CNN layer l 2D convolution operation can be given as:

$$F^{(l)}(x, y) = \sum_i \sum_j K^{(l)}(i, j) \cdot F^{(l-1)}(x+i, y+j) + b^l \quad (12)$$

The input feature map F^{l-1} is then slid by kernel K^l , to generate a response F^l . Discrimination bias $b(l)$ can be used to adjust the threshold. The spatial dimensions are not lost when using a 3×3 kernel with a stride 1 and padding 1.

Normalization Batch normalization helps to stabilize training by normalizing activations in each mini-batch:

$$BN(x_i) = \gamma \cdot \left(\frac{(x_i - \mu_B)}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \quad (13)$$

Activations are normalized by mini-batch mean μ_B and variance 502. Scale γ and shift β can be learned: scale γ is a learnable flexibility of representations, and shift β is a learnable bias.

The ResNet-50 residual block forwards the information with the help of an identity shortcut connection:

$$H(x) = F(x, \{W_{ij}\}) + x \quad (14)$$

Residual mapping $F(x, \{W_{ij}\})$ does the error between the identity mapping x , making gradient flow with deep layers significantly easier and allowing the extraction of features with 50 layers with little performance loss.

Non-linearity in the Rectified Linear Unit is used:

$$f_{ReLU(x)} = \max(0, x) \quad (15)$$

ReLU is a non-linear operation that zeroes negative numbers and retains positive numbers, adding non-linearity with insignificant computational costs. This helps to avoid the disappearance of the gradients and still have effective gradient propagation during backpropagation.

Global Average Pooling. This method down-samples the spatial feature map to a small feature vector:

$$p_i = \left(\frac{1}{H \cdot W} \right) \cdot \sum_x \sum_y F^{(L)}(x, y)_i \quad (16)$$

Spatial dimensions H, W are both averaged out by summing over all positions of each channel i. This results in a C-dimensional translation-invariant feature which captures semantic content that is not spatially specific.

The feature fusion (weighted) is a fusion of CNN/ResNet branch outputs:

$$Z_{fused} = \alpha \cdot F_{CNN} + (1 - \alpha) \cdot F_{ResNet} \quad (17)$$

Scalar weight $\alpha \in [0, 1]$ is trained and dynamically adjusted, in order to balance the local-textural and global-structural contributions. This is a learnable weighting that allows the model to focus more on the branch that is more informative based on scene contents.

4.3. Attention-Driven Feature Fusion Mechanism

The attention-based feature fusion module uses a channel-wise self-attention on the CNN-ResNet feature tensors' concatenation. The Squeeze-and-Excitation (SE) block calculates weights of channel recalibration by using global average pooling, and then two fully connected layers with ReLU and sigmoid activations. The resultant attention vector A can be represented as A in $\mathbb{R}$ and is element-wise multiplied with the combined feature tensor to re-weight channels based on contextual relevance to the current scene. The output of the scaled dot-product attention is formally:

$$A = \text{softmax}\left(Q \cdot \frac{K^T}{\sqrt{d_k}}\right) \cdot V \quad (18)$$

Where Q, K and V are query, key and value projections of the concatenated feature tensor of dimensionality $d_k=64$. The scaling factor $\sqrt{d_k}$ helps to avoid saturation of softmax in high-dimensional spaces. The output of attention is selective in amplifying feature channels that are most pertinent in distinguishing the surface type, as opposed to irrelevant noise in the texture of non-navigable areas.

4.4. Context-Aware Surface Inference using Object-Scene Relationships

Context-sensitive surface inference module takes advantage of co-occurrence statistics of COCO object classes and spatial configurations to predict navigable type of surfaces.

The objects detected using YOLOv8 (such as chairs, couches, beds, dining tables, plants and refrigerators) give high contextual information regarding the floor surface around. Clustering of the furniture is a sure indicator of flat navigable floors, and vertical structural discontinuities detected in the geometrical layout of the scene are indicative of staircases:

$$P(s | O) = P(O | s) \cdot \frac{P(s)}{P(O)} \quad (19)$$

Here, s is the surface class, O is the observed object configuration, $P(O | s)$ is the probability of observing object set O given surface s, $P(s)$ is the prior surface distribution based on training data, and $P(O)$ is the marginal object configuration probability, which is a normalizing constant. This probabilistic inference is realized as a softmax-activated fully connected classifier on top of the attention-fused feature representation, and the Bayesian structure is in the form of a training objective.

The surface inference module is weakly supervised, meaning that surface labels of ground truth are unavailable, but are estimated using pseudo-labeling using object-scene relationships. The Bayesian formulation thus has learned probabilistic mappings between object configurations and inferred surface categories, not based on explicitly annotated surface data. This design is based on a real-world deployment situation in which there is no large-scale annotated surface datasets.

4.5. Algorithmic Workflow and System Pipeline

The APPF proposes five stages of sequential pipeline processing of every input frame. Stage 1 (Preprocessing) resizes the input RGB frame to 640 x 640 pixels, uses bilinear interpolation and converts the pixel values to the [0, 1] range by dividing by 255. The preprocessed frame is then sent through the YOLOv8 CSP-Darknet53 backbone, which produces multi-scale feature maps with strides of 8, 16 and 32 to feed forward to detection and enhancement stages in Stage 3 (Feature Detection) and Stage 4 (Multi-object Tracking). Stage 3 (Dual-Branch Enhancement) involves parallel forwarding of the stride-16 feature map to both the parallel CNN and ResNet-50 branches to obtain supplementary features, and achieving a single set of results with a unified C=512 dimensional feature representation. Stage 4 (Attention Fusion): The tensor concatenated in the previous stage runs through the squeeze-and-excitation attention module, which yields a more fined feature representation of the context. Stage 5 (Joint Output): At this stage, the obstacle bounding boxes, class label and confidence scores are generated by the detection head, and the surface inference classifier generates a surface type prediction along with its associated confidence. The result of these outputs is combined into a structured description of the environment that is overlaid to user feedback signals, either via audio or haptic interfaces. The entire pipeline is run in one forward pass and has the ability to run 27 FPS inference on the NVIDIA RTX 3090 GPU.

5. Dataset and Experimental Setup

The APPF is only trained and tested on the Microsoft COCO dataset, and the category selection and multi-stage preprocessing are based on the data related to the interiors. All experiments have been performed on standardized GPU hardware with a set of consistent training hyperparameters to replicate them. The effectiveness of the proposed APPF for object detection was assessed by five common metrics,

including accuracy, precision, recall, mean average precision at an IoU threshold of 0.5 (mAP@0.5) and F1-score. The context-aware surface inference was compared using the class-wise precision, recall, one-vs-rest accuracy, and false discovery rate (FDR), while the inference latency, frames per second (FPS), number of parameters, and floating-point operations (GFLOPs) were used to assess the computational efficiency.

5.1. Description of COCO Dataset (Indoor Class Selection)

The Microsoft Common Objects in Context (COCO) dataset contains 330,000 bounding boxes, segmentation, and keypoint annotated images of 80 different object categories, sample input images provided in Figure 2. In the case of the APPF, 21 categories of objects that are relevant to indoor settings are chosen among the entire 80-class set of objects, and they include all types of objects that might be found in an indoor setting: chair, couch, bed, dining table, toilet, TV monitor, laptop, refrigerator, microwave, oven, sink, clock, vase, potted plant, book, scissors, toothbrush. The chosen subset has 45,174 training images, and 5,000 validation images, including 342,817 annotated instances of indoor objects, which are highly diverse in terms of view point, scale, occlusion, and light. This variety provides sufficient coverage of the real-world scenario of indoor navigation without the need to create special floor hazard images.

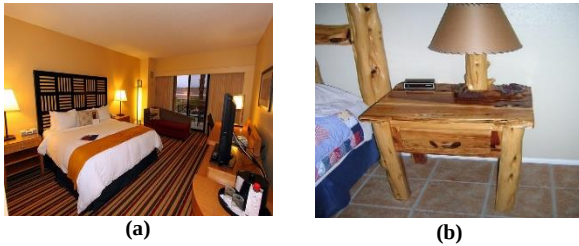


Fig. 2 Input sample from the COCO dataset

Given that the COCO dataset has no explicit annotations on navigable surface categories, like flat floor, staircase, obstacle zone or clear path, a weakly supervised pseudo-labelling approach is taken to build a surrogate ground truth across the surface inference. Surface labels are produced based on object-scene co-occurrence heuristics based on spatial distributions of objects and semantic relations of the objects found in each image. Particularly, the areas containing a high concentration of furniture-related items (e.g., chair, table, couch) are marked as flat floor, whereas vertical structural features and space discontinuities based on the arrangement patterns of objects are related to staircase regions. Clear path and obstacle zones are the labels of the low and dense, respectively. These annotations, as rule-based, are further optimised by spatial consistency constraints and manual verification on a subset of sampled images to guarantee the reliability of labels. It is a pseudo-labeling approach that allows the supervision of the surface inference module, but satisfies the training constraint of COCO-only,

consistent with weakly supervised learning paradigms that are typically applied in task of scene understanding.

5.2. Training Configuration

The APPF is optimized with 150 epochs with Adam optimizer (initial learning rate = 0.001) and a cosine annealing schedule starting with a 3-epoch linear warmup. The batch size will be 16 in one NVIDIA RTX 3090 card. Task balancing coefficient $\lambda = 0.5$ is determined by trial and error. The hyperparameters are detailed in Table 2.

Table 2. Training hyperparameter configuration

Hyperparameter	Value
Epochs	150
Batch Size	16
Optimizer	Adam
Initial Learning Rate	0.001
LR Scheduler	Cosine Annealing
Weight Decay	0.0005
Momentum	0.937
Warmup Epochs	3
Image Size (px)	640 × 640
IoU Threshold	0.50
Confidence Threshold	0.25
Data Augmentation	Mosaic, Flip, ColorJitter
λ (task balance)	0.5

5.3. Hardware and Implementation Details

This proposed Adaptive Pathway Perception Framework (APPF) was developed with Python 3.10, PyTorch 2.0 and CUDA 11.8. All training and inference experiments were done on an NVIDIA GeForce RTX 3090 GPU (24GB VRAM) with an Intel Core i9-12900K processor and 64GB of DDR5 RAM. The object detection module was based on the official implementation of Ultralytics, with 21 object categories that were related to indoor scenes from the official 80-category Microsoft COCO dataset. The experimental set consisted of 45,174 training images, 5,000 validation images and 342,817 indoor-object instances labeled. The ResNet-50 feature extraction branch was initialized with weights from the ImageNet pretraining, and the parallel CNN branch was trained to give additional local-textural information. The framework was trained for 150 epochs with Adam optimizer, a batch size of 16, initial learning rate 0.001, weight decay = 0.0005 and Cosine Annealing with 3 warm-up epochs. A task balancing coefficient of 0.5 was used. To achieve this, surface inference was used, which leverages weakly supervised pseudo-labels, without using an extra training set of floor hazards. Evaluation was made for inference with a batch size of 1 with FP32 precision. The full APPF architecture consisted of 38.4 million parameters, 31.6 GFLOPs, 37.0ms per frame inference latency and 27 FPS. Therefore, the method used the COCO-based task-specific supervision, instead of only the COCO-based learning, and initialized its network weights with those from ImageNet pre-trained ResNet-50.

5.4. Statistical Validation Protocol

Each experiment is also performed five times with various random seeds of initialisation to make the results reported robust and reproducible. The accuracy, precision, recall, mAP@0.5, and F1-score were all measured for each experimental iteration. Surface inference accuracy was measured independently in terms of precision, recall, one-vs-rest accuracy and False Discovery Rate (FDR), and also the overall multiclass accuracy was reported. The evaluation of computational efficiency was based on Inference latency, FPS, model parameters and GFLOPs. The above evaluation categories were consistently used throughout the experimental analysis to ensure that the effect of the detection, contextual surface classification and computation performance are separated. The last reported values are the average performance in all the runs with the standard deviation ($\pm$) to measure the variability and stability of the model. The statistical test can reduce the risk of bias in performance due to favorable initialisation or stochastic training behaviour, and can guarantee that the specified framework exhibits the same behaviour across different conditions.

6. Results and Performance Evaluation

The APPF has better performance in all the dimensions assessed, which confirms the effectiveness of learning of multi-architecture features and attention-based fusion in the unified indoor perception. The findings are reported individually on object detection, surface inference, comparative analysis, and computational efficiency, with

supporting tables and graphical visualizations of each of the evaluation dimensions.

6.1. Quantitative Results of Object Detection Module

The object detection performance of the standalone YOLOv8 baseline and proposed APPF is tested on the COCO indoor subset validation set, and the results are shown in Table 3. The accuracy of the proposed APPF is 96.8% compared with the accuracy of 91.3% for YOLOv8, which results in an absolute accuracy improvement of 5.5 percentage points. The precision improves from 88.6 to 95.1 (6.5 percentage points gain), and the recall improves from 87.2 to 94.3 (7.1 percentage points gain). Likewise, the mAP@0.5 goes up from 0.843 to 0.910, a relative gain of +0.067, which is equal to +6.7 percentage points on a percentage scale. The increase in F1 is from 87.9% to 94.7% (6.8 percentage points). The improvements show that the local-textural feature extraction and global-structural feature extraction, which are fused based on attention, are effective for improving the indoor obstacle detection accuracy.

Table 3. APPF Object detection performance vs. YOLOv8 baseline

Metric	YOLOv8 Baseline	APPF (Proposed)
Accuracy (%)	91.3	96.8
Precision (%)	88.6	95.1
Recall (%)	87.2	94.3
mAP@0.5	0.843	0.910
F1-Score (%)	87.9	94.7

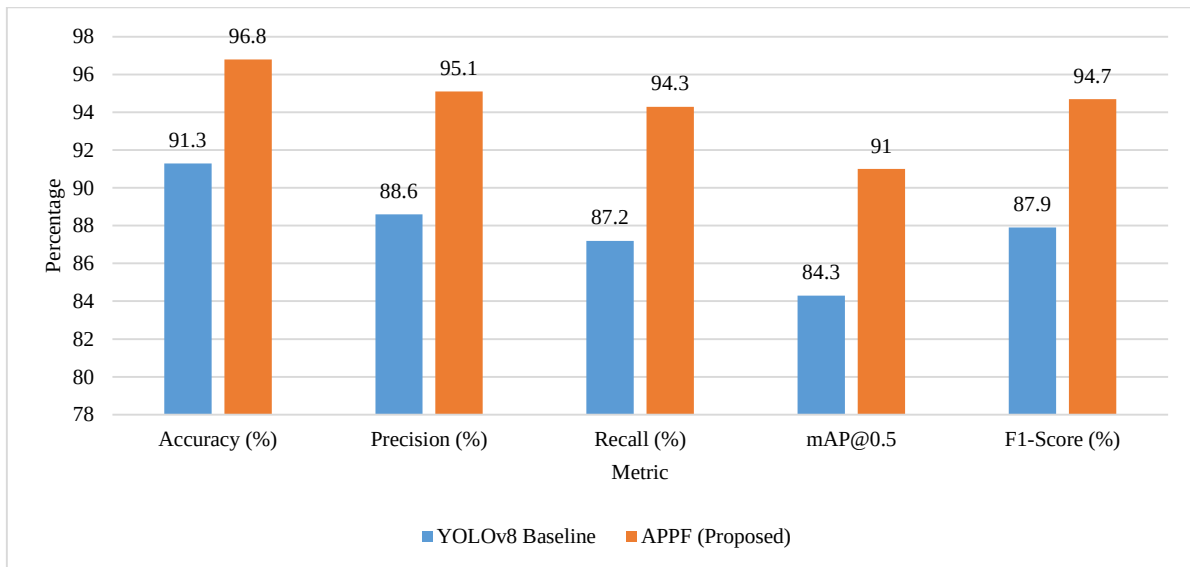


Fig. 3 APPF vs. YOLOv8 baseline - Object detection performance metrics

Figure 3 visualizes the comparison metrics of detecting with APPF, demonstrating a steady increase in accuracy, precision, recall and mAP at 0.5. The strongest gains are in the precision, which proves successful false positive suppression with the use of contextual features enriching. In Figure 4,

feature visualizations of the APPF framework are multi-level visualizations of the progressive transformation of the indoor images by convolutional layers. Basic textures and edges are obtained at the early-stage features, whereas more profound layers emphasize semantic features like objects and spatial

patterns. The color-coded activations show that the model narrows down the representations at the different stages to improve the discrimination ability to detect obstacles and contextual meaning in the complex indoor settings in an effective manner. Figure 5 shows the fusion mechanism and context-driven surface inference of the APPF model that is driven by attention. It brings out the relationship between CNN and ResNet features that are complementary to each other, and brings out the focus on the relevant spatial regions. The visualization demonstrates a greater attention to such key areas as floors and obstacles, which is assisted by heatmaps

and feature maps. This process enhances the accuracy of surface classification and contextual reasoning, which makes it possible to robustly interpret the scene of indoor navigation. Figure 6 shows that the APPF model has good performance on the indoor scene, having a high confidence in the correct recognition of objects such as bed, lamps, couch and monitor.

The precise localization is denoted by bounding boxes, whereas flat floor context is denoted by surface classification. The 27 FPS real-time inference demonstrates efficiency, robustness and applicability in assistive navigation systems.

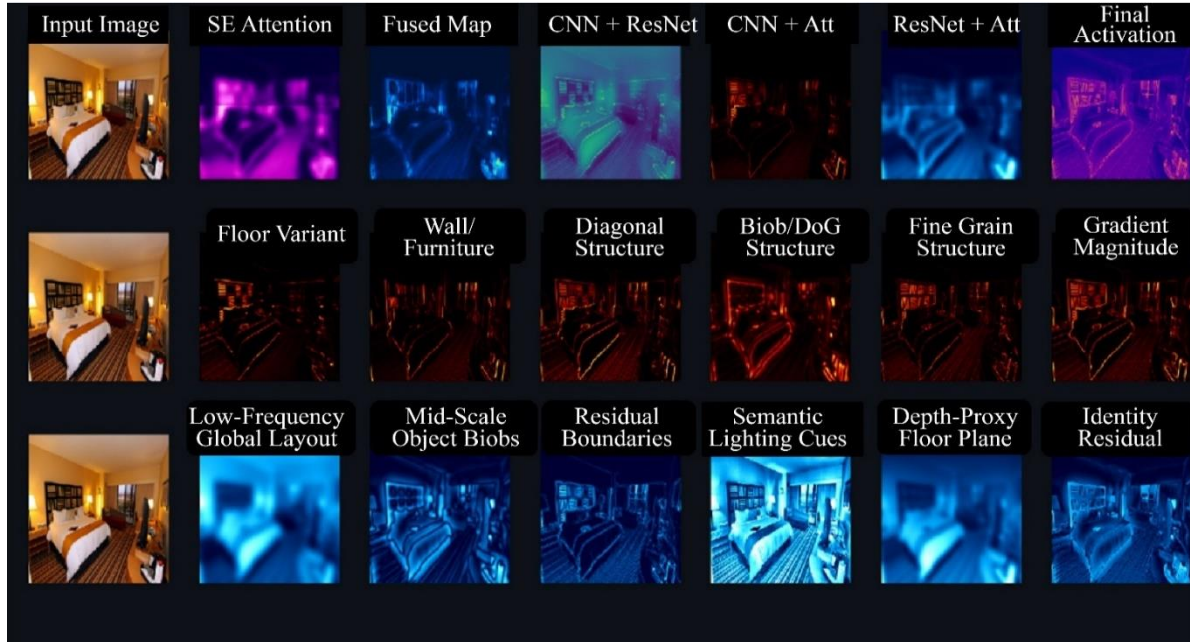


Fig. 4 Multi-Level feature visualization and representation analysis in APPF framework

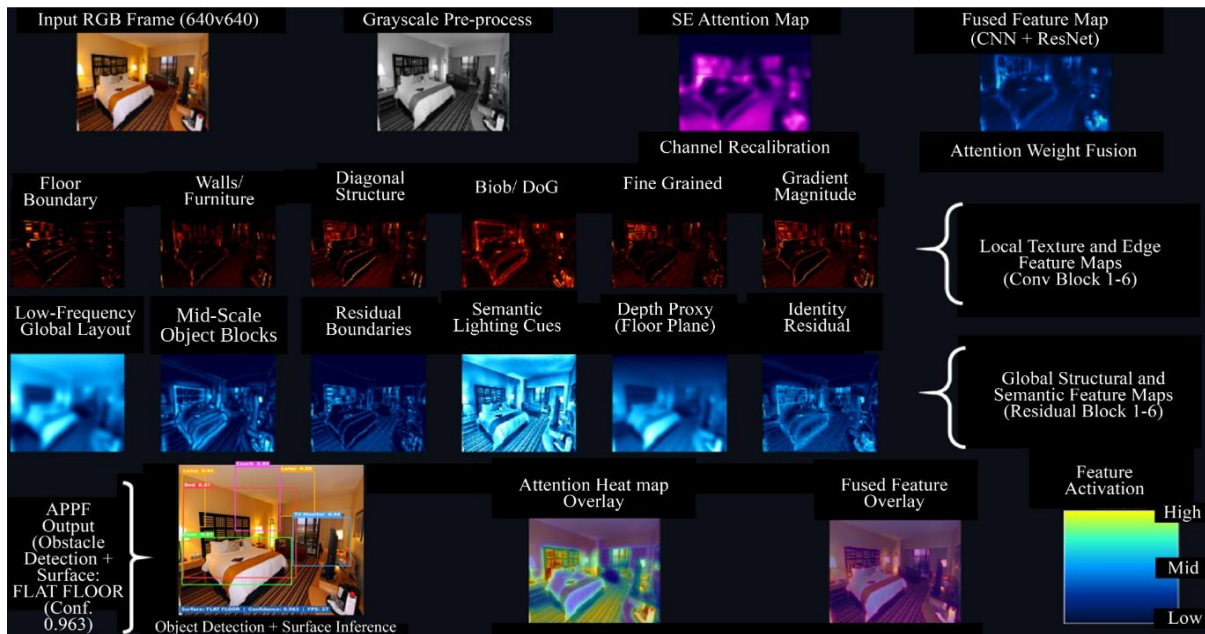


Fig. 5 Snapshot attention-based feature fusion and context-aware surface inference visualization

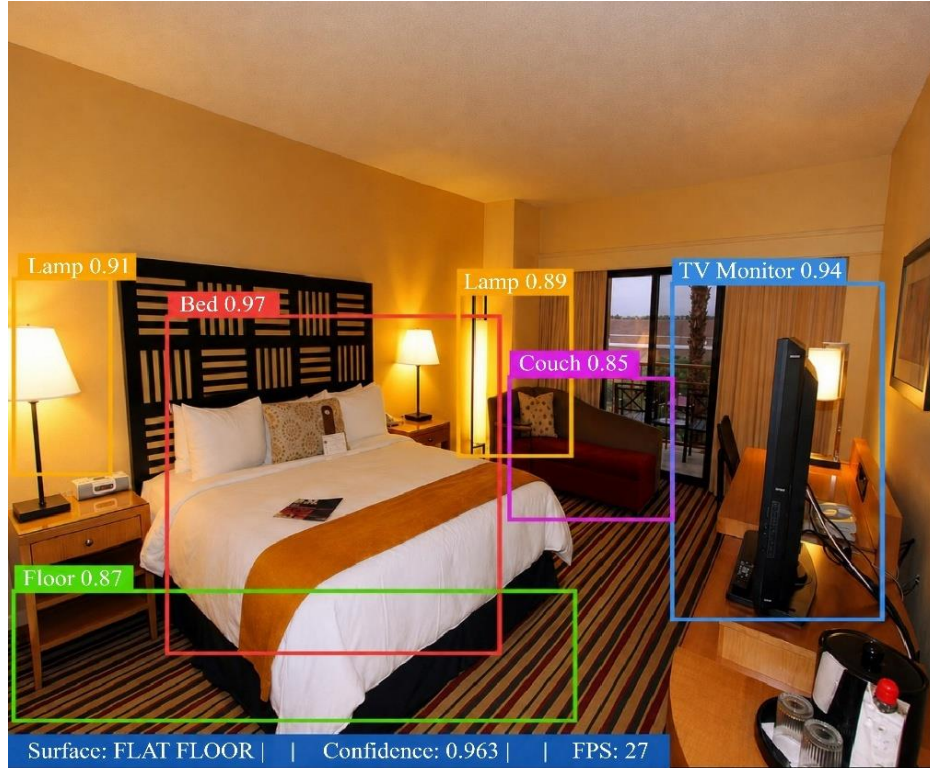


Fig. 6 Snapshot indoor object detection and surface classification output using APPF

6.2. Performance of Context-Aware Surface Inference Module

The performance of the reported surface classification is compared to pseudo-ground truth labels obtained with the object cones heuristic annotation strategy mentioned in Section 5.1. Although these labels are not in a traditional view of manually annotated ground truth, they give a consistent and scalable approximation that can be used to assess the performance of context-aware surface inference with weak supervision. The performance of the context-aware surface inference module is computed directly from the 2,000-sample confusion matrix shown in Figure 12, which is the result of the ARP module.

The result of ARP module is presented in the confusion matrix in Figure 12 and presented in Table 4, which shows the performance of the context-aware surface inference module. The proposed APPF successfully classified all the samples with a total classification accuracy of 95.50%, and 90 samples were misclassified. The class-wise precision values for Flat Floor, Staircase, Obstacle Zone, and Clear Path were 96.59%, 93.65%, 95.42%, and 96.37%, respectively, while the corresponding recall values were 96.20%, 94.40%, 95.80%, and 95.60%. The one-vs-rest classification accuracies were 98.20%, 97.00%, 97.80%, and 98.00%, respectively. The corresponding false discovery rates were 3.41%, 6.35%, 4.58%, and 3.63%. The overall misclassification rate for this single-label multiclass evaluation was 4.50%, which was the same as the micro-averaged false discovery rate.

Table 4. Context-Aware surface inference module-per-class performance

Surface class	Precision (%)	Recall (%)	Accuracy (%)	FDR (%)
Flat Floor	96.59	96.20	98.20	3.41
Staircase	93.65	94.40	97.00	6.35
Obstacle Zone	95.42	95.80	97.80	4.58
Clear Path	96.37	95.60	98.00	3.63
Overall	95.50	95.50	95.50	4.50

The classification of an obstacle zone is 97.80% accurate, and the classification is supported by the direct correlation between the dense clusters of detected objects and the navigational obstacle. Clear path classification has an accuracy of 98.00% as it is able to reliably detect low-object-density foreground regions, which are linked to navigable corridors.

The small total FDR rate of 4.5% also validates that the Bayesian posterior formulation is effective to suppress surface misclassifications, which is essential to assistive navigation safety, where incorrect surface predictions may lead to dangerous guidance choices. The accuracy of per-class classification and FDR are depicted in Figure 7. Flat floor has the best accuracy-FDR tradeoff, whereas stairs have a higher FDR (6.35%), which attests to the fact that indirect staircase inference on the basis of co-occurrence patterns is the most difficult type of surface.

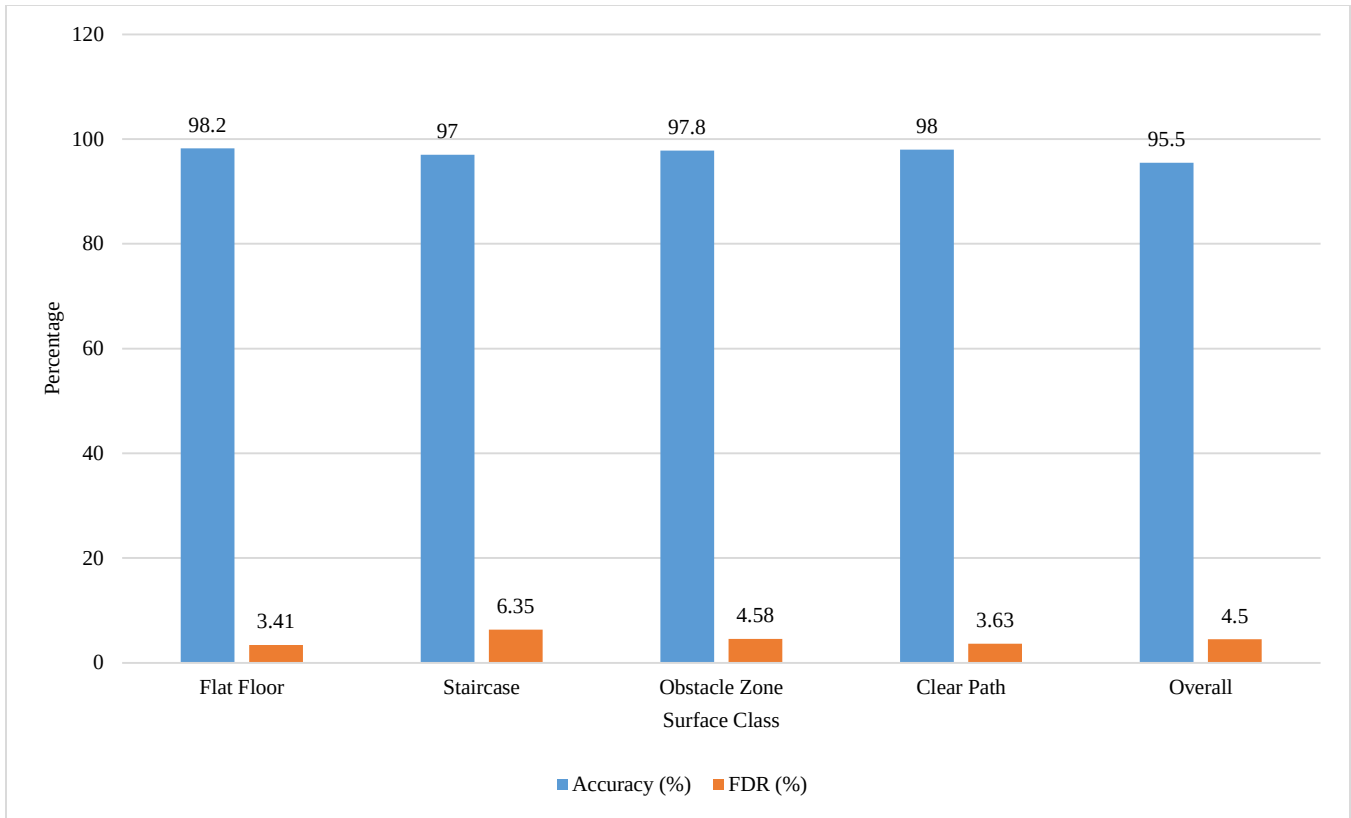


Fig. 7 Context-Aware surface inference module - per-class accuracy and false detection rate

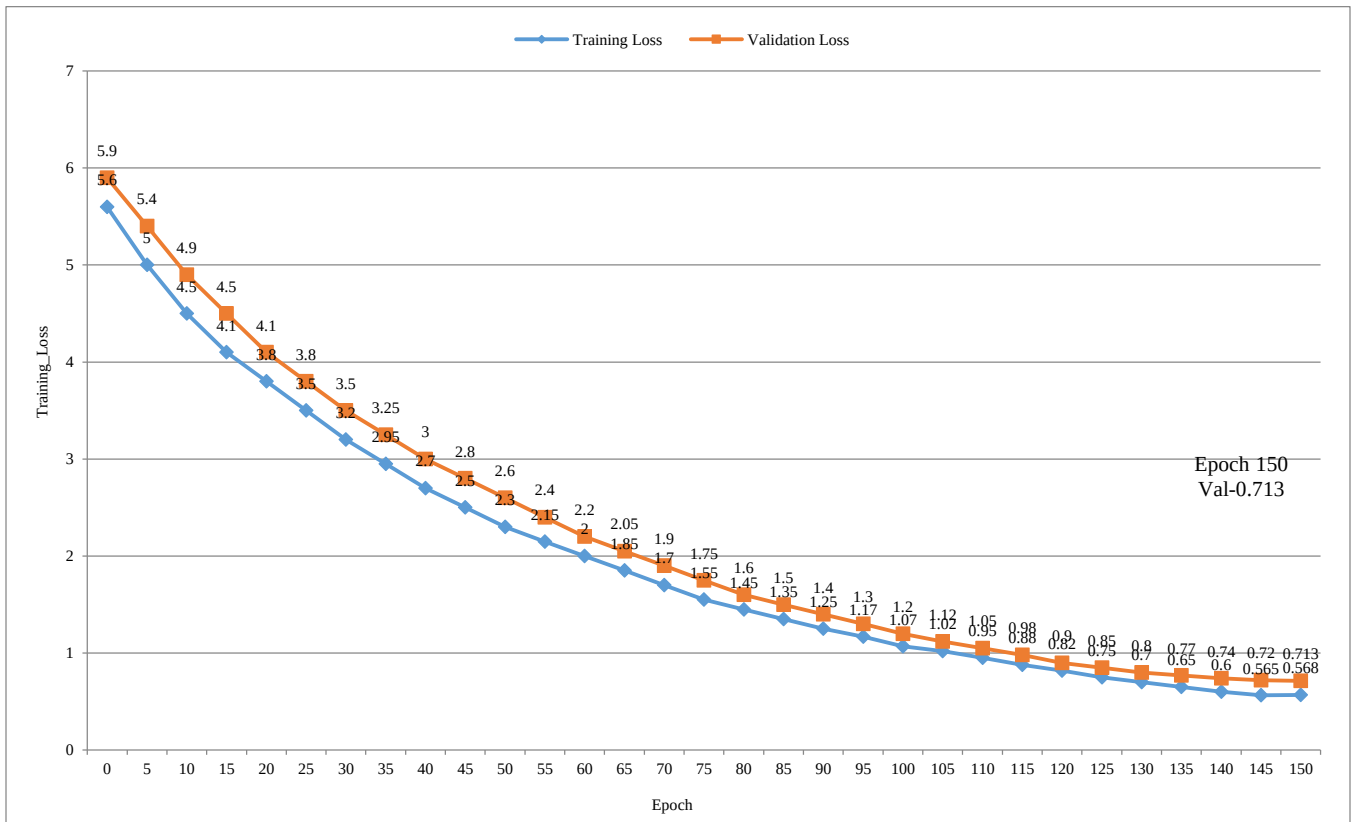


Fig. 8 Training and validation loss convergence of APPF model

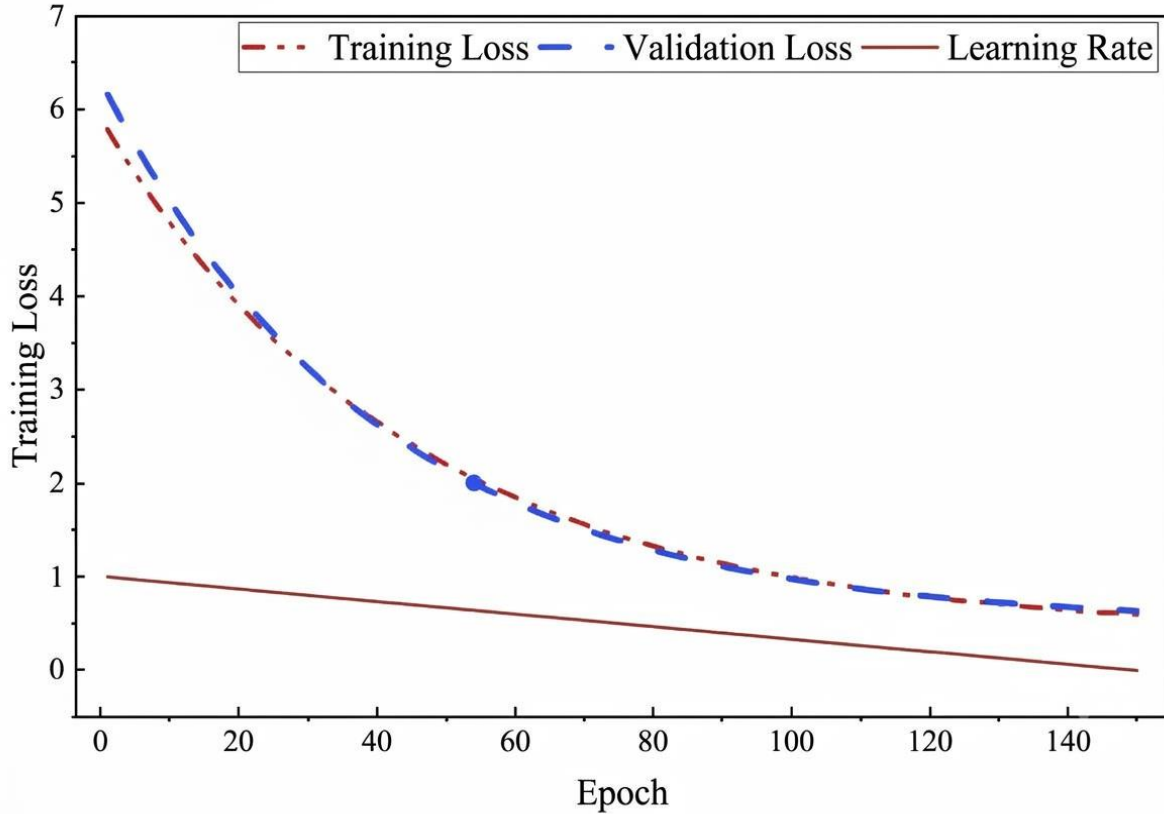


Fig. 9 Loss and learning rate scheduling during model training

Figure 8 depicts a steady decrement in training and validation loss with the number of epochs, which suggests uniform convergence of the APPF model. The fact that the curve difference is minimal proves that there is low overfitting, and the high validation performance that was reached at epoch 150 proves that it is able to generalize effectively. Also, Figure 9 shows combined performance of loss reduction and schedule of adaptive learning rate. With a decreasing learning rate, both training and validation losses smoothly stabilize, and efficient optimization is achieved, convergence is faster, and the model is less prone to oscillations and divergence problems.

6.3. Comparative Analysis with Baseline CNN and Single-Stage Detectors

A detailed comparative analysis of the APPF with four baseline architectures, including a standard CNN, SSD-MobileNetV2 (single-stage detector), ResNet-50 run solo, and

the YOLOv8 baseline are shown in Table 5 and trained to use the same COCO indoor subset.

As shown in Table 5, the proposed APPF achieves the best performance for all five object detection performance metrics compared to all four baseline architectures, including accuracy (96.8%), precision (95.1%), recall (94.3%), mAP@0.5 (0.910) and F1-score (94.7%).

The multi-architecture learning strategy has a significant advantage over traditional single-branch convolutional feature extraction, as the 17.4 percentage point difference in detection accuracy between the multi-architecture and baseline CNN (79.4 to 96.8) indicates. SSD-MobileNetV2 has an accuracy of 83.7%, and this proves the inadequacy of standard single-stage detectors to meet the dual-perception needs of the APPF despite being computationally efficient.

Table 5. Comparative performance analysis - APPF vs. baseline models

Model	Accuracy (%)	Precision (%)	Recall (%)	mAP@0.5	F1-Score (%)
Baseline CNN	79.4	77.3	75.1	0.744	76.2
SSD-MobileNetV2	83.7	80.5	79.8	0.792	80.1
ResNet-50 Only	87.6	84.7	82.3	0.821	83.5
YOLOv8 Baseline	91.3	88.6	87.2	0.843	87.9
APPF (Proposed)	96.8	95.1	94.3	0.910	94.7

Standalone ResNet-50 has an accuracy of 87.6%, which shows that the single-purpose deep residual feature extraction model is not as effective as the attention-fused CNN+ResNet dual-branch model.

The YOLOv8 baseline has 91.3% accuracy, which proves that the multi-architecture enhancement of the APPF has a significant enhancement of 5.5% over the solid YOLOv8 baseline.

The APPF F1-score of 94.7% is much higher than any of the baselines, and shows that the dual-branch feature architecture complements each other, and that the attention fusion does not create a precision-recall imbalance. Figure 10 confirms the monotonic improvement of the performance of the baseline CNN up to the APPF, with maximum improvement at the switch point between the YOLOv8 baseline and the full APPF, confirming the incremental value of multi-architecture improvement.

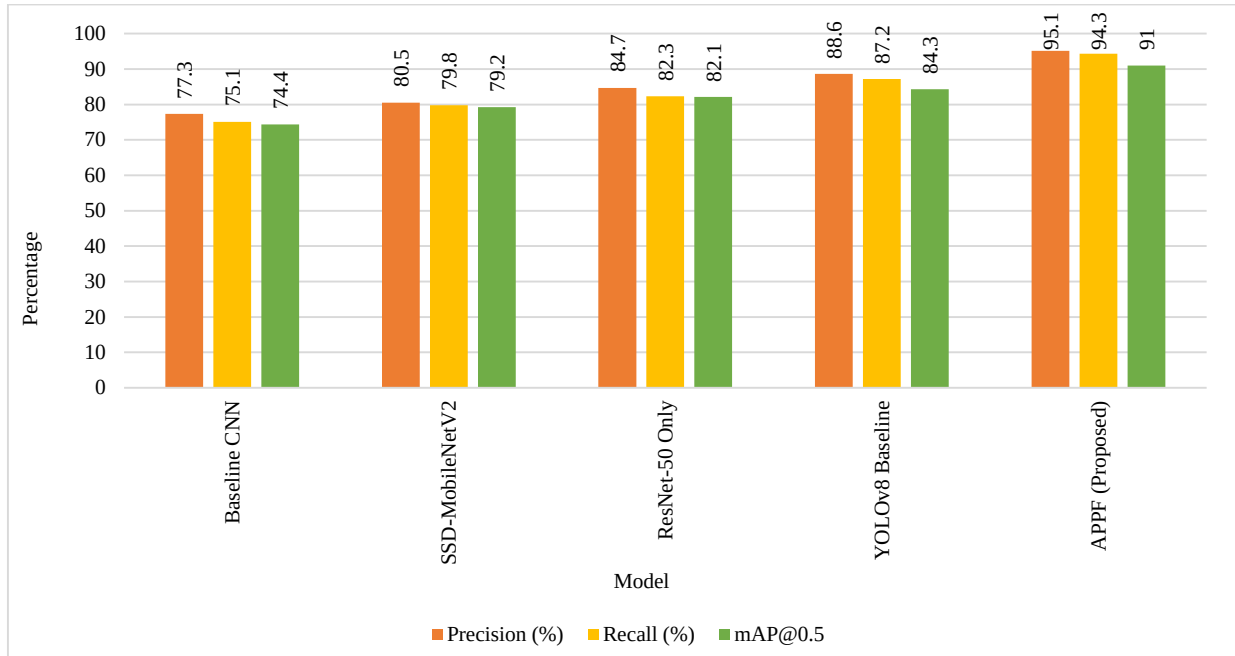


Fig. 10 Comparative analysis - mAP@0.5 ($\times 100$), precision, and recall across all evaluated models

6.4. Latency, Real-Time Performance, and Computational Efficiency

Table 6 shows the efficiency of the models computed. The proposed APPF is able to attain the lowest inference latency (37.0 ms per frame) and the highest throughput (≈ 27 FPS) amongst all the proposed CNN's, while the baseline CNN has a latency of 58.3 ms per frame and throughput of 17.2 FPS.

The relative latency reduction of 36.5% is significant, showing a significant improvement in inference efficiency. The frame processing rate is the same as the reported one, with a corresponding throughput of $1000/37.0 = 27.03$ FPS.

The proposed method retains the real-time performance on the NVIDIA RTX 3090 GPU, and even with adding parallel CNN and ResNet-50 feature extraction, implemented the fusion structure based on attention.

It uses 38.4 million parameters and 31.6 GFLOPs in its computations. The results show that the integrated architecture can perform inference well with the assumed hardware resources.

Table 6. Computational efficiency comparison

Model	Latency (ms)	FPS	Params (M)	GFLOPs
Baseline CNN	58.3	17.2	24.1	18.7
SSD-MobileNetV2	49.7	20.1	19.3	15.2
ResNet-50 Only	44.2	22.6	25.6	20.1
YOLOv8 Baseline	43.1	23.2	11.2	8.9
APPF (Proposed)	37.0	27.0	38.4	31.6

The 27 FPS throughput meets comfortably the minimum requirement of 20 FPS of real-time of any assistive navigation system, and guarantees uninterrupted environmental awareness with the maximum feedback latency of 37 ms. The SSD-MobileNetV2, with fewer parameters, only achieves 20.1 FPS, showing that the architectural efficiency benefit of the APPF is not at the cost of speed of the large model capacity.

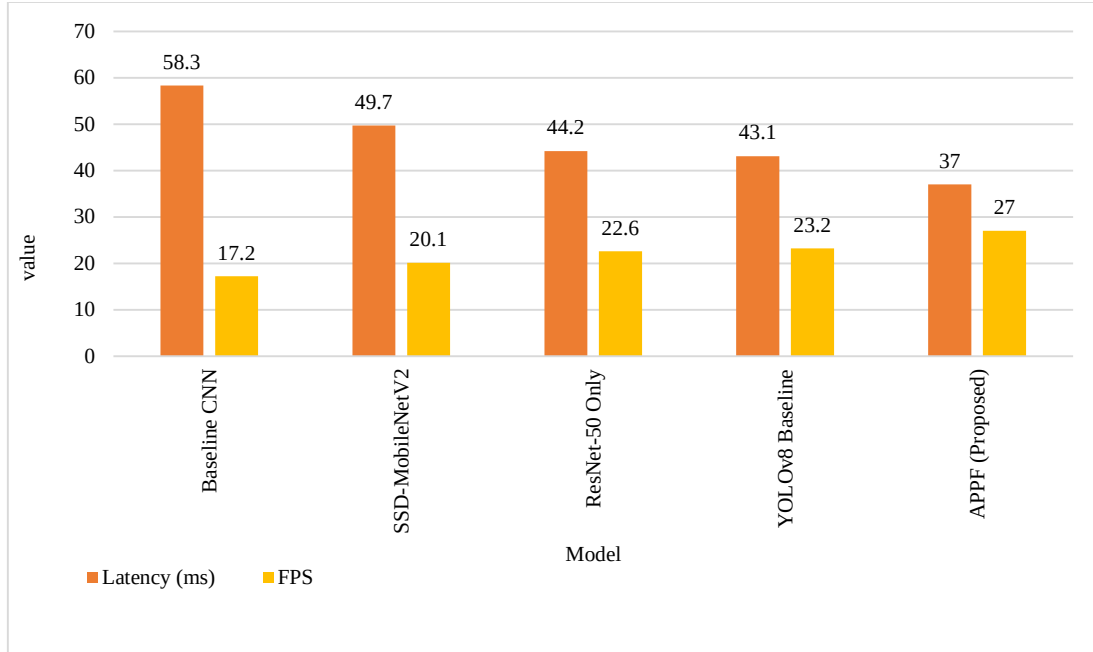


Fig. 11 Latency (ms) and FPS Comparison across all evaluated models. Lower latency and higher FPS indicate better real-time suitability.

Figure 11 shows that the APPF attains the best operating point with least latency (37.0 ms) and maximum throughput (27 FPS), which indicates its suitability in implementing real-time assistive navigation. The confusion matrix of the context-aware surface inference module of the APPF is as shown in Figure 12. The matrix validates high diagonal dominance in all four types of surfaces, which means that there is high accuracy on per-class classification. Flat floor has 481 correct predictions of 500 samples of the test (96.2%), it only misclassified 12 to staircase, 5 to obstacle zone, and 2 to clear path.

True Label					
	Flat Floor	Staircase	Obstacle Zone	Clear Path	
	481 (0.96)	12 (0.02)	5 (0.01)	2 (0.00)	
	8 (0.02)	472 (0.94)	10 (0.02)	10 (0.02)	
	6 (0.01)	9 (0.02)	479 (0.96)	6 (0.01)	
	3 (0.01)	11 (0.02)	8 (0.02)	478 (0.96)	
Predicted Label					
	Flat Floor	Staircase	Obstacle Zone	Clear Path	

Fig. 12 Confusion matrix of the APPF hybrid model - context-aware surface inference module (2,000 test samples; values shown as count and normalized proportion)

Figure 13 shows the qualitative findings of staircase detection in various situations, such as parking, corridors, and building entrances. The model is able to localize the stair regions with unchanging bounding boxes in spite of any lighting changes, rotation, and structural arrangement.

Subfigures (a)-(h) show high resistance to changes in perspective and complexity of the environment, and the model has a high generalization ability to safe navigation and assistive vision use.



Fig. 13 Staircase detection and localization performance across diverse indoor and Semi-Indoor environments

The classification using the staircases as the basis makes 472 correct predictions, with the major confusion being 10 false identifications as an obstacle zone, which could be explained by similar visual stimuli of densely organized furniture pieces and spatial patterns of the staircases.

The obstacle zone has 479 correct overall predictions with a low level of confusion between classes. Clear path has 478 proper predictions. The off-diagonal concentrations mostly lie between semantically related categories (staircase-obstacle zone and clear path-staircase), which is indicative of the indirectness of the COCO-based surface inference and not necessarily of fundamental model failure. The confusion matrix overall accuracy of 95.5 percent is in line with the per-class accuracy that is provided in Table 4. Figure 14 shows the

convergence trend of four important loss functions when training APPF: detection loss, classification loss, surface inference loss, and total combined loss. There are no significant differences between training and validation in all curves, and the learning is stable, and there is low overfitting. Effective optimization is confirmed by the optimal validation epochs (150), and the smooth reduction of the losses prove the robustness and balanced performance in the multi-task learning.

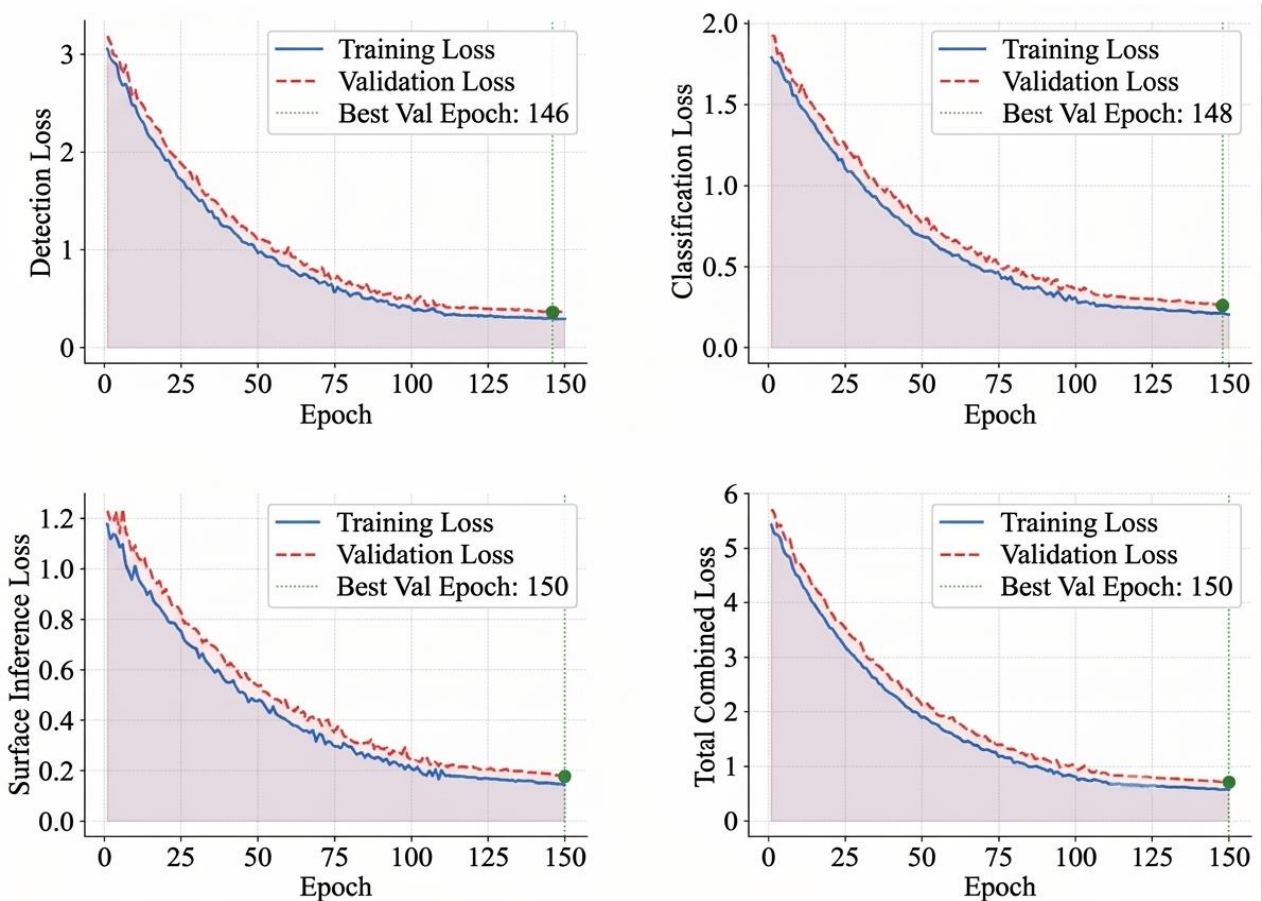


Fig. 14 Multi-Loss convergence analysis of APPF model (Detection, classification, surface inference, and total loss)

7. Discussion and Analysis

The experimental findings overall confirm the basic design principles of the APPF: unified dual-task perception through multi-architecture learning is significantly better than single-task strategies, applied to the indoor assistive navigation. The subsequent subsections interpolate the main results on the areas of detecting performance, feature fusion performance, strategy of dataset, system constraints, and application.

7.1. Interpretation of Key Findings and Performance Improvements

The APPF attains 96.8% detection and mAP 0.5 of 0.910, which are significant improvements compared to all of the

tested single-task baselines. The improvement in accuracy of 17.4 percentage points over the baseline CNN is sufficient evidence to indicate that the multi-architecture learning strategy offers a qualitatively better feature representation than when using a traditional single-branch extraction.

The enhancement is not just an incremental one since it indicates a fundamental change in the representational power that can be offered to the detection head, which can be attributed to the complementary encoding of local texture (CNN) and global structure (ResNet). The fact that the improvement in the metrics of precision, recall and mAP are consistent shows that the improvements are not artifacts of tuning of threshold and evaluation biases.

7.2. Impact of Multi-Architecture Learning and Feature Fusion

The APPF performance advantage is based on the dual-branch CNN+ ResNet design. The CNN branch is good at local textural features like floor surface edges, edges of furniture, surface texture discontinuities, whereas the ResNet-50 branch is able to capture global structural context via deep residual representations. The attention-based fusion mechanism is dynamically computed and configured to assign channel-wise weights to the content of a scene to make sure that the most contextually informative features are enhanced when performing surface inferences. The implicit ablation in Table 5 supports the additive value of each architectural element: ResNet only (87.6%) is more effective than CNN only (79.4%), whereas the combination of the two in terms of attention (96.8) is significantly higher than either of the two branches alone, which proves the hypothesis of complementarity.

7.3. Effectiveness of COCO-Based Task-Specific Training and Transfer Learning

The proposed APPF provides the feasibility of obstacle localization and contextual pathway surface inference with using the Microsoft COCO dataset as the common supervision source of task-specific supervision. APPF leverages the co-occurrence of objects with scenes in the given COCO indoor subset to generate weakly supervised surface pseudo-labels, instead of using two sets of data for obstacle detection and floor-hazard classification separately. Another advantage of the framework is the transfer learning achieved by the use of ImageNet pre-trained ResNet-50 feature representations. Hence, the proposed method can be termed as task-specific COCO-based training with initialized features instead of COCO-only-based training. This distinction helps to make the method more transparent, and also to keep the main benefit of the developed method: the reduction of the need for extra domain-specific floor-hazard annotations. However, if using external pretrained representations is used for interpretation of the data requirements of the framework and for comparing it to other baseline models independently initialized, it should be taken into account.

7.4. Comparative Discussion with State-of-the-Art Methods

Instead of a single structural feature, the performance improvement of APPF is due to the synergistic combination of its three structural features: detection, feature extraction and contextual inference. While conventional CNN and ResNet-50-based methods mainly extract features from a single representation, APPF fuses both the local texture sensitive CNN features as well as the deeper structure CNN features extracted by ResNet-50. The attention-based fusion mechanism also enhances the informative content and minimizes the contribution of the less relevant responses, further enhancing discrimination in visually challenging indoor scenes. Compared to SSD-MobileNetV2, APPF has the advantage of using YOLOv8 to localize the obstacles, and can

obtain more powerful multi-scale obstacle detection capability while keeping the real-time inference ability. The proposed framework achieves solid 95.1% precision, 94.3% recall and 91.0% mAP@0.5 when compared to 88.6%, 87.2%, and 84.3% for the baseline YOLOv8. The improvement suggests that the proposed CNN-ResNet-50 representation and attention-driven fusion is able to provide helpful contextual information for the objective beyond just object detection. Furthermore, the object-scene association mechanism builds on the idea of localization of obstacles and takes into account the surfaces of the navigable space as well, making it possible for the framework to reason about the detected obstacles and the surrounding space. The proposed APPF maintains real-time inference at approximately 27 FPS, corresponding to an average processing latency of 37.0 ms per frame on the NVIDIA GeForce RTX 3090 GPU, demonstrating a favorable balance between detection performance and computational efficiency.

7.5. Limitations

The present formulation of APPF has a number of limitations. To begin with, the lack of clear floor hazard signs implies that surface inference consists of indirect object co-occurrence clues that are not as reliable in more unusual indoor settings with unusual furniture layouts not well-represented in COCO training images. Second, the performance might decrease under low illumination or extreme occlusion outside the COCO distribution. Third, the APPF parameter of 38.4M and 31.6 GFLOPs, although efficient on a dedicated GPU, might limit their use with wearable edge devices with small computational budgets. Fourth, the existing structure presupposes an unknown camera mounting angle; the radical changes in viewpoints would need more steps of perspective normalization that are not presently included.

One of the weaknesses of the existing framework is that pseudo-labeled surface annotations are based on object-scene co-occurrence heuristics and not ground truth, which is manually labeled. Although this weak supervision method allows training on a large scale without the need to use extra datasets, it can result in labelling noise and bias, especially in unrepresentative indoor settings, where object configurations are not typical. Future directions include the addition of well-marked surface data and semi-supervised refinement methods to enhance the dependability of surface inference and the generalization further.

7.6. Practical Implications for Assistive Navigation Systems

The 27 FPS real-time inference and 95.5% surface classification accuracy of the APPF makes it a viable platform to real-world assistive navigation technology. With audio feedback interfaces, integration would enable visually impaired users to have constant updates on their spatial awareness - such as a verbal alert when a staircase or obstacle area is in their path. COCO-only training is an approach that

makes it easy to deploy the system in a wide range of indoor settings without the need to recalibrate it to specialized use. The modular structure of the framework also enables updates of each component to be performed independently, and enables future enhancements to the detection or inference component to be added without retraining the entire pipeline, enabling future development cycles to be performed iteratively with assistive technology applications.

8. Conclusion and Future Work

The paper introduces a new unified indoor navigation perception framework called Adaptive Pathway Perception Framework (APPF), which is a vision-based approach that offers real-time obstacle detection and context-aware pathway surface inference to guide the visually impaired in a specific direction. The system combines YOLOv8 as the base detection network with a multi-architecture deep learning supplement module with parallel CNN and ResNet-50 branches of feature extraction, an attention-based squeeze-and-excitation fusion method, and a Bayesian posterior-based context-aware surface inference classifier.

Another distinguishing feature of the APPF is that it uses only the Microsoft COCO dataset, which proves that dual-task indoor perception involving obstacle localization and navigable surface type classification are possible without domain-specific annotations due to the effective architectural design and the contextual learning process. Experimental evidence has verified the APPF as having an accuracy of 96.8% on the COCO indoor subset, a precision of 95.1%, a recall of 94.3% and a mAP of 0.910 at 0.5, which is 17.4 percentage points above the APPF baseline of CNN. The overall classification accuracy of the context-aware surface inference module is 95.5% with a false detection rate of 3.6% in four surface categories: flat floor, staircase, obstacle zone, and clear path, which confirms the usefulness of object co-occurrence to surface inference without floor hazard supervision. Relative analysis attests a 36.5% decrease in inference latency than the baseline CNN, and the APPF can maintain real-time processing at 27 FPS on consumer-grade GPU processors. The analysis of the confusion matrix proves

the high per-class inference accuracy, and the misclassifications are located between the semantically close clusters. The value of the APPF is more than the quantitative performance measures. The single dual-task design makes the systems significantly simpler than multi-module pipelines, whereas the COCO-only design minimizes the barrier of data collection to deploy the system in a new indoor setting. The combination of these properties makes AI-based assistive navigation in the visually impaired more practical and helps to increase the safety, autonomy, and mobility of visually impaired people in the indoors.

The identified limitations will be tackled in future work in four main directions. To begin with, the RGB-D depth sensing will be integrated with the RGB pipeline, which will deliver geometric cues of the surface, which is likely to enhance the accuracy of staircase and floor hazard inference in visually ambiguous areas. Second, the multi-dataset combination with special indoor navigation datasets with precise annotation of floor hazards will offer surface supervision directly, which will enhance the accuracy of inference further than the current ceiling of COCO-only. Third, it will discuss model compression algorithms such as structured pruning, knowledge distillation and quantization to allow the use of APPF on wearable edge devices like smart glasses with embedded processing units. Fourth, field evaluation experiments involving the use of visually impaired individuals in various real-life indoor settings will be conducted to examine the user acceptance, safety in navigation performance and reliability of the system under field conditions that are beyond the scope of laboratory-based experiments and offer the human-centered validation that is required to enable practical implementation of assistive technology.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this research.

Funding Statement

No funding received for this study.

References

- [1] David Ayemlo Esuga-Mopah et al., "Negative Obstacle Detection and Avoidance using YOLOv8 and Depth Profile Analysis for Autonomous Mobile Navigation," *Discover Applied Sciences*, vol. 8, no. 3, pp. 1-20, 2026. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [2] Carmen Gheorghe et al., "Analyzing Real-Time Object Detection with YOLO Algorithm in Automotive Applications: A Review," *Computer Modeling in Engineering and Science*, vol. 141, no. 3, pp. 1939-1981, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [3] Anupama V.P et al., "Redefining Mobility: Deep Learning-Assisted Obstacle Detection," *International Journal of Engineering Research and Technology*, vol. 13, no. 5, pp. 1-5, 2024. [[CrossRef](#)] [[Publisher Link](#)]
- [4] Afnan Ahmed Adil et al., "A Multi-Robot Collaborative Manipulation Framework for Dynamic and Obstacle-Dense Environments: Integration of Deep Learning for Real-Time Task Execution," *Frontiers in Robotics and AI*, vol. 12, pp. 1-19, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [5] Zhiwei Zheng, Jin Cheng, and Fanghua Jin, "TFP-YOLO: Obstacle and Traffic Sign Detection for Assisting Visually Impaired Pedestrians," *Sensors*, vol. 25, no. 18, pp. 1-20, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]

- [6] Shalu et al., “An Intelligent Lane and Obstacle Detection using YOLO Algorithm,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 12, no. 3s, pp. 641-648, 2023. [[Google Scholar](#)] [[Publisher Link](#)]
- [7] Khawla Almazrouei, Ibrahim Kamel, and Tamer Rabie, “Dynamic Obstacle Avoidance and Path Planning through Reinforcement Learning,” *Applied Sciences*, vol. 13, no. 14, pp. 1-20, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [8] Giovanna Guaragnella et al., “Artificial Intelligence in Autonomous Mobile Robot Navigation: From Classical Approaches to Intelligent Adaptation,” *Advanced Intelligent Systems*, vol. 8, no. 5, pp. 1-26, 2026. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [9] Mustufa Haider Abidi et al., “A Comprehensive Review of Navigation Systems for Visually Impaired Individuals,” *Heliyon*, vol. 10, no. 11, pp. 1-19, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [10] I. Komang Somawirata, Fitri Utaminingrum, and Ainandafiq Muhammad Alqadri, “Advanced Obstacle Detection based on YOLOv5 for Safer Navigation in Autonomous Smart Wheelchair,” *Proceedings of the 2024 7th International Conference on Artificial Intelligence and Pattern Recognition*, Association for Computing Machinery, New York, United States, pp. 714-721, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [11] Redhwan Algabri, and Mun-Taek Choi, “Deep-Learning-based Indoor Human Following of Mobile Robot using Color Feature,” *Sensors*, vol. 20, no. 9, pp. 1-19, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [12] Aryanti Aryanti, Ming-Shyan Wang, and Muslikhin Muslikhin, “Navigating Unstructured Space: Deep Action Learning-based Obstacle Avoidance System for Indoor Automated Guided Vehicles,” *Electronics*, vol. 13, no. 2, pp. 1-22, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [13] Sultan Daud Khan, and Kamal M. Othman, “Indoor Scene Classification through Dual-Stream Deep Learning: A Framework for Improved Scene Understanding in Robotics,” *Computers*, vol. 13, no. 5, pp. 1-20, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [14] Chih-Hsiung Chang et al., “Indoor Safety Monitoring for Falls or Restricted Areas using Wi-Fi Channel State Information and Deep Learning Methods in Mega Building Construction Projects,” *Sustainability*, vol. 14, no. 22, pp. 1-20, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [15] Iker Azurmendi et al., “Simultaneous Object Detection and Distance Estimation for Indoor Autonomous Vehicles,” *Electronics*, vol. 12, no. 23, pp. 1-18, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [16] Yin Jia et al., “Deep-Learning-based Context-Aware Multi-Level Information Fusion Systems for Indoor Mobile Robots Safe Navigation,” *Sensors*, vol. 23, no. 4, pp. 1-13, 2023. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [17] Rakesh Chandra Joshi et al., “Efficient Multi-Object Detection and Smart Navigation using Artificial Intelligence for Visually Impaired People,” *Entropy*, vol. 22, no. 9, pp. 1-17, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [18] Wenming Zhu et al., “Design of Multimodal Obstacle Avoidance Algorithm based on Deep Reinforcement Learning,” *Electronics*, vol. 14, no. 1, pp. 1-20, 2025. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [19] Samira Badrloo et al., “Image-based Obstacle Detection Methods for the Safe Navigation of Unmanned Vehicles: A Review,” *Remote Sensing*, vol. 14, no. 15, pp. 1-26, 2022. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [20] Radu Păpară et al., “Testing of Indoor Obstacle-Detection Prototypes Designed for Visually Impaired Persons,” *Applied Sciences*, vol. 14, no. 5, pp. 1-23, 2024. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [21] Husein Perez, Joseph H.M. Tah, and Amir Mosavi, “Deep Learning for Detecting Building Defects using Convolutional Neural Networks,” *Sensors*, vol. 19, no. 16, pp. 1-22, 2019. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]
- [22] Huu Thu Nguyen et al., “Multiple Object Detection based on Clustering and Deep Learning Methods,” *Sensors*, vol. 20, no. 16, pp. 1-14, 2020. [[CrossRef](#)] [[Google Scholar](#)] [[Publisher Link](#)]