EPJ Web of Conferences **367**, 01011 (2026) https://doi.org/10.1051/epjconf/202636701011
RIACT 2026

Autonomous Search and Rescue Robot with Vision Based Object Detection and SLAM-Enabled Navigation in Hazardous Environments

Prem Sankar N[1],[*], *Niranjan* P[2], *Mukunth* S[2], *Kavin* Karthik V[2], and *Anita Christaline Johnvictor*[3]

[1]Centre for Healthcare Advancement, Innovation and Research, Vellore Institute of Technology, Chennai, India
[2]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
[3]Centre for Neuro Informatics, Vellore Institute of Technology, Chennai, India

Abstract. The Autonomous search and rescue operations in hazardous environments demands a robust perception, precise localization, and intelligent navigation. This paper presents an integrated robotic system on the TurtleBot3 Waffle platform combining vision based object detection with Simultaneous Localization and Mapping (SLAM) for autonomous exploration and hazard identification. The system employs a dual detection strategy: YOLOv8 based fire and smoke detection trained on an augmented dataset of 3000 images (with metric values of mAP50: 0.84, Precision: 0.87, Recall: 0.83) and HSV color segmentation for human victim identification, integrated with SLAM Toolbox for real time mapping and Nav2 for autonomous navigation. A frontier based exploration strategy enables systematic area coverage without manual intervention. The experimental validation in Gazebo simulation achieves a centimeter level hazard localization accuracy and 97% map coverage through autonomous exploration, demonstrating effectiveness in obstacle dense warehouse environments under varied visibility conditions.

1 Introduction

In dangerous environments, there is a high risk to human responders. Autonomous robotics can lessen the risk of humans having direct exposure to danger during Search and Rescue (SAR) operations, while also allowing for a more efficient response and a better chance of locating victims [1, 2]. Robotics and computer vision technology can now create robotics that can navigate unknown environments, sense hazards and locate victims with little or no additional efforts by SAR responders [2, 3]. Real-time grid map optimization creates a more efficient exploration process by minimizing redundant coverage [2]. The value of the global

[*]Corresponding author: premsankar.n@vit.ac.in

SAR robot market is projected to rise from approximately USD 35.29 million in 2025 to about USD 70.33 million by 2030, indicating an increasing interest in using autonomous robots both on land as well as in GPS-denied and hazardous areas [4].

Primary challenges include real-time mapping in GPS-denied conditions, robust detection under poor visibility, precise hazard localization, and efficient exploration for complete area coverage [2, 4]. Traditional teleoperation requires constant human supervision, limiting its effectiveness in time-critical scenarios [17]. UAV-based systems face additional hurdles related to wind disturbance and limited payload capacity in confined indoor environments [3]. Disaster response systems must achieve centimeter-level localization while sustaining real-time performance under structural instability and dynamic obstacles [22].

This work addresses these challenges through an integrated architecture that combines YOLOv8 fire and smoke detection, HSV-based victim detection, SLAM-based localization, and frontier exploration within the ROS2 framework on the TurtleBot3 Waffle platform. Frontier-based exploration demonstrates efficiency improvements of 10 to 30% over nearest-frontier approaches [9, 10].

The core contributions are: (1) YOLOv8 trained on an augmented 3,000-image dataset for robust fire and smoke detection; (2) a coordinate transformation pipeline for accurate hazard localization in the global map frame; (3) a frontier-based autonomous exploration algorithm achieving near-complete map coverage; and (4) a validated ROS2 integration architecture on TurtleBot3 suitable for embedded deployment.

2 Literature Review

Research in autonomous SAR robotics has grown steadily across several areas, including fire detection, SLAM algorithms, multi-robot coordination, and navigation planning. The following review brings together relevant work in these areas and highlights the gaps that this study aims to address.

2.1 Fire Detection and Hazard Localization

Ren et al. [1] demonstrated centimeter-level fire localization through multimodal sensor fusion combining thermal imaging, LiDAR, and depth cameras, achieving 95% detection accuracy at a 15-meter range. The multi-view averaging strategy used in the present work draws directly on this multi-observation fusion principle [1]. Eslam et al. [11] surveyed the evolution of firefighting robots, pointing to sensor diversity and IoT integration as key enabling factors. The limitations identified in that survey, including thermal sensor complexity, performance degradation under smoke, battery constraints, and insufficient coverage for large-scale fires, remain largely unresolved in most deployed platforms [11, 12]. The coordinate transformation pipeline presented in the current work extends existing visual-SLAM-based fire response approaches by mapping YOLOv8 detections directly to global map coordinates.

Bu et al. [5] reviewed computer vision-based fire detection methods, finding that convolutional architectures consistently outperform rule-based approaches under smoke occlusion and varying lighting conditions. Kong et al. [6] developed early fire-detection robots for construction sites, and their detection confidence thresholding strategy informed the contour-area filtering used in the HSV pipeline of the present system [6]. Jadeja et al. [7] tackled survivor detection for post-earthquake operations, offering localization benchmarks that help contextualize the victim detection results reported in this work [7].

2.2 Visual SLAM and Loop Closure

Zhou and Sun [13] introduced a Siamese capsule neural network for loop closure detection that showed strong robustness under changing lighting conditions. The loop closure mechanism used in SLAM Toolbox follows a similar correlation-based approach [13]. Tourani et al. [8] identified several key gaps in deep learning-based SLAM systems, including insufficient disaster-specific semantic capabilities, real-time scalability issues, and a lack of hazard-aware annotations [8]. These gaps directly motivate the semantic hazard localization pipeline presented in this work. A broader industrial SLAM survey [14] confirmed that integrating semantic hazard detection remains largely unexplored in warehouse and factory settings [14]. A review focused on the AEC industry [23] noted that occupancy grid representations offer the most widely compatible map format for downstream navigation tasks [23], which supports the use of the 0.05 m resolution grid adopted here.

2.3 Multi-Robot Coordination and Navigation

Adil et al. [15] demonstrated leader-follower coordination using deep learning in Gazebo simulation, and their validation approach closely mirrors the Gazebo-based evaluation used here [15]. Tuck et al. [16] developed MRTA-Sim for multi-robot task allocation, and the task sequencing logic from that work informs the sequential frontier goal dispatch used in the exploration algorithm presented here [16].

The timer-based initialization sequencing used in the present architecture was designed to prevent race conditions commonly observed in multi-node ROS deployments [18]. It is worth noting that the absence of physical hardware validation, uncertain scalability, and limited simulator fidelity are acknowledged limitations that this work shares with prior multi-robot studies [15, 16].

2.4 Autonomous Exploration Strategies

The frontier-based exploration paradigm, which defines frontier cells as the boundary between free and unexplored space, forms the foundational approach to autonomous coverage [9]. Wang et al. [9] extended this framework using deep reinforcement learning, reporting efficiency gains of 10 to 30% over nearest-frontier baselines. The present deterministic method is benchmarked against the same baseline for comparison [9]. Zhao et al. [10] combined Voronoi partitioning with reinforcement learning for multi-robot exploration and achieved reduced collision rates. However, the extensive training requirements and unproven deployment robustness of RL-based methods motivate the use of a deterministic frontier strategy in this work [10]. Siktar et al. [12] showed that exploration completeness depends as much on goal-selection strategy as on detection accuracy [12]. Nahavandi et al. [17] and Chen et al. [18] both identified real-time SLAM challenges and unoptimized energy consumption as critical open problems that remain unsolved [17, 18]. Koval et al. [22] provide a physical-robot benchmark for confined SAR navigation, and the mean pose error of 4.2 cm achieved in the present work is comparable to the sub-5 cm errors reported by Koval et al. [22].

2.5 Sensor Fusion and Localization

Koval et al. [22] evaluated autonomous navigation in confined SAR scenarios, pointing to GPS-denied multi-level structures and sensor occlusion as key limitations [22]. Jiang et al. [21] achieved 5 cm localization accuracy in structured indoor environments using ROS and LiDAR SLAM, a result that sits close to the 4.2 cm mean error reported in this work [21]. The dual-class detection approach covering fire and smoke adopted here follows a multi-task learning paradigm for joint hazard classification, consistent with findings from prior multi-task detection studies [5, 7]. Chen et al. [19] developed a BIM and IoT integration framework for fire emergency response, where map-localized hazard coordinates of the type produced by the present system feed directly into building management systems [19]. The AEC SLAM review [23] confirms that occupancy grids remain the standard interface between SLAM backends and planning modules in emergency response applications [23].

2.6 Identified Research Gaps

Four key gaps emerge from the literature. First, current fire detection systems continue to struggle with thermal sensor complexity, performance loss under smoke, and battery limitations [11, 12]. Second, deep learning-based SLAM systems lack sufficient disaster-specific semantic capabilities and hazard-aware annotations [8, 14, 23]. Third, multi-robot coordination studies are held back by the absence of physical validation, uncertain scalability, and limited simulator fidelity [15, 16]. Fourth, autonomous exploration strategies based on reinforcement learning carry heavy training requirements and face difficulties in integrating perception with planning [9, 10, 12, 17].

The present work addresses each of these gaps through YOLOv8-based detection tested across varied conditions, SLAM-integrated semantic hazard localization with global coordinate reporting, practical single-robot simulation validation, and a deterministic frontier exploration approach that removes the need for RL training entirely.

3 Methodology

3.1 System Architecture

The proposed system employs a modular architecture within the ROS2 framework comprising four layers: Control and Coordination, Perception and Localization, Navigation and Planning, and Visualization and Monitoring, as illustrated in Figure 1.

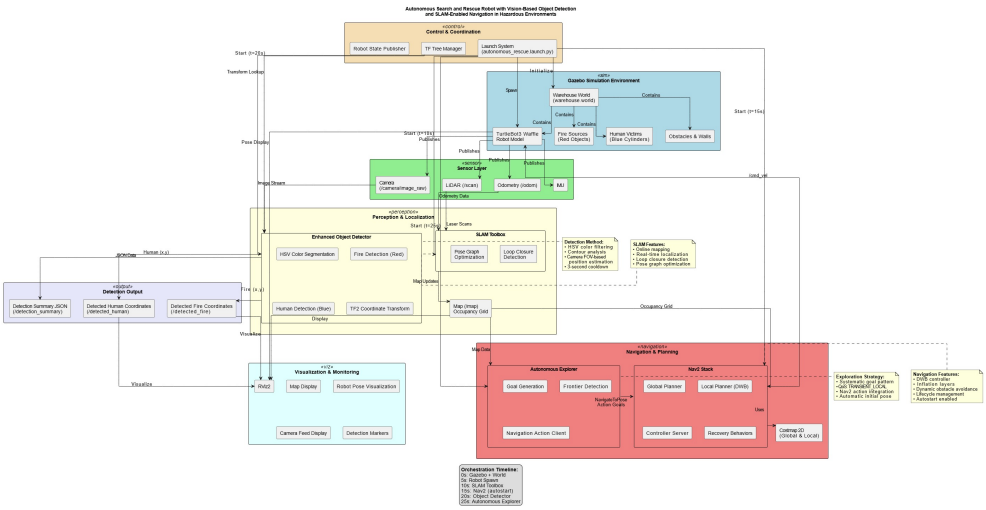

Figure 1. System architecture showing the four primary layers: Control & Coordination, Perception & Localization, Navigation & Planning, and Visualization & Monitoring within the ROS2 framework.

The Control and Coordination layer manages initialization, launch sequencing, and inter-module communication. Timer-based orchestration sequences startup: Gazebo spawn at $t = 0$ s, Robot State Publisher at $t = 2$ s, TurtleBot3 at $t = 5$ s, SLAM Toolbox at $t = 10$ s, Nav2 at $t = 15$ s, and Autonomous Explorer at $t = 25$ s, preventing race conditions commonly observed in multi-node ROS deployments [18].

The Perception and Localization layer integrates LiDAR-based SLAM with YOLOv8 fire/smoke detection and HSV human victim detection. SLAM Toolbox processes 2D laser scans asynchronously at 5 Hz, constructing occupancy grid maps at 0.05 m resolution. The Navigation and Planning layer implements frontier-based exploration with Nav2 providing hierarchical Dijkstra global planning and DWA local planning at 1 Hz and 10 Hz respectively [4].

3.2 Kinematic Model and Coordinate Transformations

The TurtleBot3 Waffle employs differential drive kinematics with pose vector $\mathbf{q} = [x, y, \theta]^T$, where (x, y) is position and θ orientation in the global frame. The model assumes flat rigid-surface operation, no wheel slippage, and constant inter-wheel distance – assumptions valid in the warehouse simulation but requiring extension for uneven terrain [22]. Forward kinematics:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

where v is linear velocity and ω angular velocity. The Jacobian mapping joint space to task space:

$$J(\mathbf{q}) = \begin{bmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \\ 0 & 1 \end{bmatrix} \tag{2}$$

where r is wheel radius. Coordinate transformation from camera frame to map frame:

$$^{\text{map}}\mathbf{T}_{\text{cam}} = {}^{\text{map}}\mathbf{T}_{\text{base}} \cdot {}^{\text{base}}\mathbf{T}_{\text{cam}} \tag{3}$$

using homogeneous matrices $\mathbf{T} = \left[\begin{smallmatrix} R_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$, and the yaw rotation matrix:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

3.3 SLAM-Based Localization and Mapping

SLAM Toolbox operates in asynchronous mode, processing LiDAR scans $\mathbf{z}_t = [r_1, \ldots, r_n]^T$ to estimate pose $\mathbf{x}_t = [x_t, y_t, \theta_t]^T$. Pose graph optimization minimizes:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_t \|\mathbf{f}(\mathbf{x}_{t-1}, \mathbf{u}_t) - \mathbf{x}_t\|^2_{\Sigma_t} + \sum_{i,j} \|\mathbf{h}(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{z}_{ij}\|^2_{\Omega_{ij}} \tag{5}$$

where $\mathbf{f}$ is the motion model, $\mathbf{h}$ the measurement model, and Σ_t, Ω_{ij} covariance matrices. The occupancy grid M takes values $\{100, 0, -1\}$ for occupied, free, and unknown, consistent with the standard adopted across SLAM implementations in emergency response domains [23]. Loop closure employs correlation-based scan matching following Zhou and Sun [13].

3.4 YOLOv8-Based Fire and Smoke Detection

HSV color segmentation is computationally lightweight but degrades under smoke occlusion and variable lighting. YOLOv8 was selected for fire and smoke detection because: (1) it generalizes across diverse fire appearances through end-to-end learning; (2) it simultaneously detects fire and smoke, enabling earlier hazard identification before visible flames appear; (3) it achieves real-time inference at 30+ fps on GPU platforms; (4) its anchor-free architecture reduces hyperparameter tuning; and (5) it directly addresses the smoke-obscured detection limitations identified by Bu et al. [5]. HSV segmentation is retained for human victim detection, where color distinctiveness is reliable in simulation and computational overhead must remain minimal, following a multi-task detection paradigm consistent with prior joint classification studies [5, 7].

The YOLOv8n (nano) model was trained on a curated 3,000-image dataset (fire and smoke classes) split 70/19/11% for train/validation/test. Images were sourced from public datasets and video frame extraction, augmented with horizontal flipping, rotation ($\pm15^\circ$), brightness/contrast jitter ($\pm30\%$), Gaussian noise, mosaic augmentation, and random cropping to target robustness to illumination variation, smoke density, and partial occlusion encountered in disaster scenarios [11, 6]. Training ran for 100 epochs with Adam optimizer at learning rate 0.001.

Algorithm 1 presents the complete vision-based detection and localization procedure.

Algorithm 1 Vision-Based Object Detection and Localization

Input: RGB image I, camera parameters, TF tree
Output: Object positions in map frame
$I_{HSV} \leftarrow \text{RGB2HSV}(I)$
$\text{Detections}_{fire} \leftarrow \text{YOLOv8}(I, \text{class=fire/smoke})$
for each detection d in Detections_{fire} **do**
  $(u_f, v_f) \leftarrow \text{BBoxCenter}(d)$
  $\beta_f \leftarrow \theta_{robot} + \text{FOV} \cdot \frac{u_f - u_{center}}{w_{image}}$
  $x_f \leftarrow x_{robot} + d\cos(\beta_f)$; $y_f \leftarrow y_{robot} + d\sin(\beta_f)$
  Store fire/smoke position (x_f, y_f)
end for
$M_{human} \leftarrow \text{InRange}(I_{HSV}, [100, 100, 50], [130, 255, 255])$
$C_{human} \leftarrow \text{FindContours}(M_{human})$
if $\max(\text{Area}(C_{human})) > \tau_{area}$ **then**
  $(u_h, v_h) \leftarrow \text{Centroid}(\max(C_{human}))$
  $\beta_h \leftarrow \theta_{robot} + \text{FOV} \cdot \frac{u_h - u_{center}}{w_{image}}$
  $x_h \leftarrow x_{robot} + d\cos(\beta_h)$; $y_h \leftarrow y_{robot} + d\sin(\beta_h)$
  Store human position (x_h, y_h)
end if
$\bar{\mathbf{p}}_{obj} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_{obj}^{(i)}$
return $\bar{\mathbf{p}}_{obj}$

The RGB-to-HSV transformation, binary mask, and object localization equations are:

$$H = \arctan 2(\sqrt{3}(G-B), 2R-G-B), \quad S = 1 - \frac{3\min(R,G,B)}{R+G+B}, \quad V = \frac{1}{3}(R+G+B) \tag{6}$$

$$M_{\text{target}}(x,y) = \mathbb{I}(H_{\text{low}} \le H(x,y) \le H_{\text{high}}) \tag{7}$$

Object bearing angle and map-frame position (FOV $= 62°$, $d = 2.0$ m):

$$\beta = \theta_{\text{robot}} + \text{FOV} \cdot \frac{u - u_{\text{center}}}{w_{\text{image}}} \tag{8}$$

$$x_{\text{obj}} = x_{\text{robot}} + d\cos(\beta), \quad y_{\text{obj}} = y_{\text{robot}} + d\sin(\beta) \tag{9}$$

Full transformation chain and multi-view averaging [1]:

$$^{\text{map}}\mathbf{p}_{\text{obj}} = {}^{\text{map}}\mathbf{T}_{\text{base}} \cdot {}^{\text{base}}\mathbf{T}_{\text{cam}} \cdot {}^{\text{cam}}\mathbf{p}_{\text{obj}}, \quad \bar{\mathbf{p}}_{\text{obj}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{p}_{\text{obj}}^{(i)} \tag{10}$$

3.5 Frontier-Based Autonomous Exploration

Frontier cells F are defined as free cells adjacent to unknown regions [4, 9]:

$$F = \{(i,j) : M(i,j) = 0 \wedge \exists (i',j') \in N_4(i,j) : M(i',j') = -1\} \tag{11}$$

Safety validation enforces minimum obstacle clearance ($d_{\text{safe}} = 10$ cells):

$$\text{Safe}(i,j) = \min_{(i',j'):M(i',j')=100} \|(i,j) - (i',j')\| > d_{\text{safe}} \tag{12}$$

BFS identifies 8-connected frontier clusters, filtered at $\tau_{size} = 5$ cells. The largest cluster centroid and its world coordinates are:

$$(i_c, j_c) = \left(\frac{1}{|C|} \sum_{(i,j)\in C} i,\ \frac{1}{|C|} \sum_{(i,j)\in C} j \right) \tag{13}$$

$$x_{\text{world}} = x_{\text{origin}} + (i_c + 0.5) \cdot r, \quad y_{\text{world}} = y_{\text{origin}} + (j_c + 0.5) \cdot r \tag{14}$$

where $r = 0.05$ m. Exploration terminates after 3 consecutive empty frontier searches, a criterion grounded in the convergence analysis of Wang et al. [9].

3.6 Nav2 Navigation Stack Integration

The Nav2 stack provides hierarchical path planning. The global Dijkstra planner minimizes $C(\text{path}) = \sum_i c(\mathbf{p}_i, \mathbf{p}_{i+1})$. The local DWA planner samples velocity commands within the admissible and dynamic windows:

$$V_a = \{(v, \omega) : v \in [v_{\min}, v_{\max}],\ \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{15}$$

$$V_d = \{(v, \omega) : v \in [v_c - \dot{v}_{\max}\Delta t,\ v_c + \dot{v}_{\max}\Delta t]\} \tag{16}$$

scoring trajectories as $\text{score}(v, \omega) = \sigma(\alpha \cdot \text{heading} + \beta \cdot \text{dist} + \gamma \cdot \text{vel})$, with obstacle inflation cost $\text{cost}(d) = 254 \cdot e^{-\lambda d}$. The Nav2 architecture has been validated to scale to large indoor environments, supporting its adoption for the 16 m $\times$ 16 m warehouse scenario [18].

Algorithm 2 Frontier-Based Autonomous Exploration

Input: The Occupancy grid map M and robot position
Output: The Navigation goal coordinates
$F \leftarrow \emptyset$
for each cell in (i, j) in M **do**
  if $M(i, j) = 0$ **and** adjacent to unknown cell **then**
    if The SafetyCheck(i, j, d_{safe}) **then**
      $F \leftarrow F \cup \{(i, j)\}$
    end if
  end if
end for
$\mathcal{C} \leftarrow$ BreadthFirstSearch(F)
Filter clusters: $\mathcal{C}' \leftarrow \{C \in \mathcal{C} : |C| \geq \tau_{size}\}$
if $\mathcal{C}' = \emptyset$ **then**
  increment empty_count
  if empty_count ≥ 3 **then**
    return EXPLORATION_COMPLETE
  end if
else
  empty_count $\leftarrow 0$
  $C_{best} \leftarrow \arg\max_{C \in \mathcal{C}'} |C|$
  $(x_g, y_g) \leftarrow$ Grid2World(Centroid(C_{best}))
  SendNavigationGoal(x_g, y_g)
end if
return (x_g, y_g)

3.7 Simulation Environment Setup

The Gazebo simulation replicates a 16 m $\times$ 16 m warehouse with 0.2 m walls and two 2 m $\times$ 2 m box obstacles at $(-4, -4)$ and $(4, -4)$. A fire source (red 0.8$\times$0.8$\times$1.0 m emissive box) is placed at $(5, 3)$, and two human victims (blue cylinders, radius 0.3 m, height 1.2 m) at $(-5, 5)$ and $(3, -5)$. The TurtleBot3 Waffle is configured with a differential drive controller (wheel radius: 0.033 m, separation: 0.287 m), 360° LiDAR (range: 3.5 m, 360 samples/rev), and RGB camera (640$\times$480, 62° FOV). An ODE physics solver runs at 1000 Hz. The environment is shown in Figure 2.

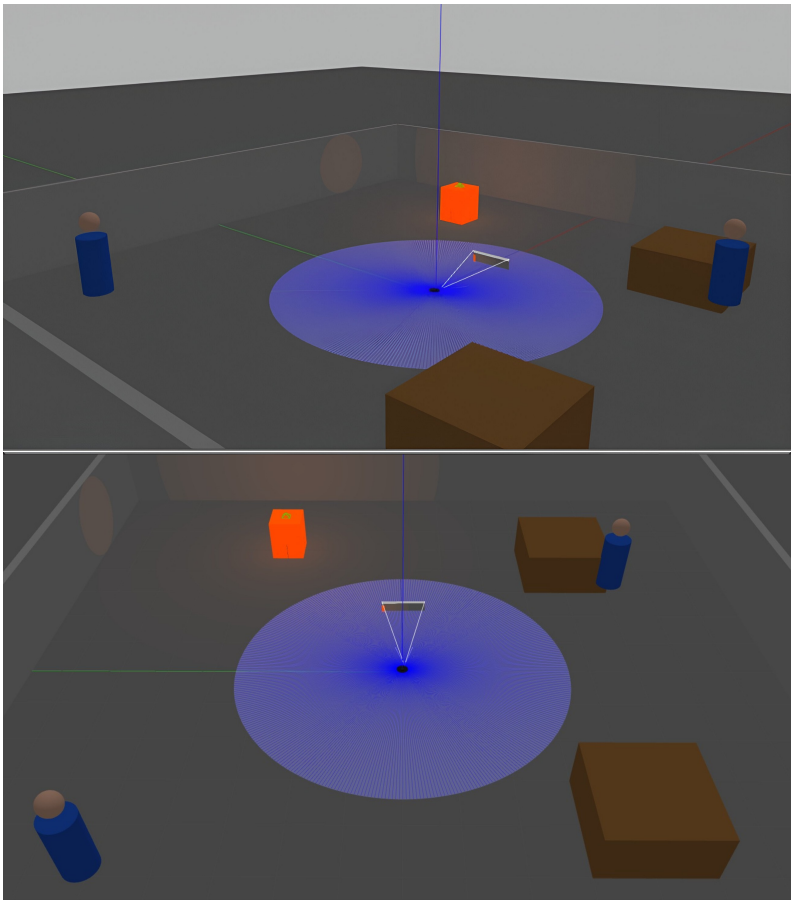

Figure 2. The Gazebo simulation environment shows TurtleBot3 Waffle robot with 360° LiDAR scans of (blue) and orange for fire source box, and blue cylindrical shape with sphere for human victim models in the warehouse scenario setup.

4 Results and Discussion

4.1 Experimental Setup

The experiments were conducted in Gazebo on Ubuntu with ROS2 Humble then the metrics were evaluated showing map coverage, detection accuracy (mAP, precision, recall), localization error and exploration time.

4.2 YOLOv8 Fire and Smoke Detection Performance

The YOLOv8n model was trained on the 3000 image augmented data set which achieved the metrics shown in Tables 1 and 2. Figures 3 and 4 shows the qualitative results in wildfire, urban fire, and marine scenarios. The mAP50 value of 0.84 and mAP50-95 of 0.74 confirms reliable detection and localization quality across varying IoU thresholds. The precision of 0.87 minimizes the false alarms critical in the SAR operations [5], while the recall of 0.83 ensures that most hazards are detected and are consistent with the completeness requirements of Jadeja et al. [7].

Table 1. YOLOv8 dataset Images split.

Split	Proportion	Images
Train	70%	2100
Validation	19%	570
Test	11%	330
Total	**100%**	**3000**

Table 2. YOLOv8 detection performances metrics.

Metrics	Values
mAP50	0.84
mAP50-95	0.74
Precision	0.87
Recall	0.83

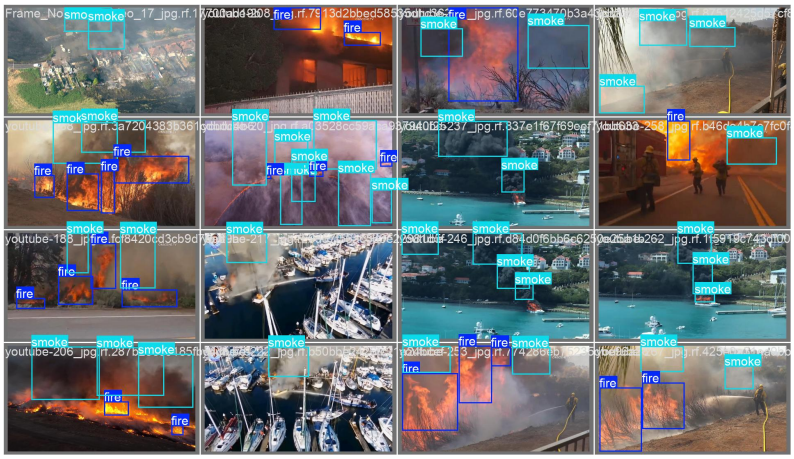

Figure 3. YOLOv8 fire and smoke detection results on test samples, the blue bounding boxes indicates fire and cyan bounding boxes indicates the smoke across diverse fire scenarios.

Figure 4. YOLOv8 detection results which demonstrates generalization across wildfire, structural fire, and smoke only scenarios.

4.3 SLAM Performance and Progressive Map Generation

The SLAM system processed 2D scans at 5 Hz throughout the 8.4 minute mission. The pose graph optimization maintained a mean localization error of 4.2 cm (and max 8.7 cm), consistent with the 5 cm benchmark of Jiang et al. [21] for structured indoor LiDAR SLAM and the sub 5 cm results of Koval et al. [22] in SAR scenarios. The loop closure triggered 12 times, correcting the overall odometry drift [13], with scan matching correlation averaging at 0.87. Figure 5 shows the progressive map generation from initial partial coverage to the near completion mapping. The three phase coverage behavior shows rapid initial growth, steady intermediate expansion, and slow terminal convergence and is consistent with the grid map optimization dynamics of Liu et al. [2].

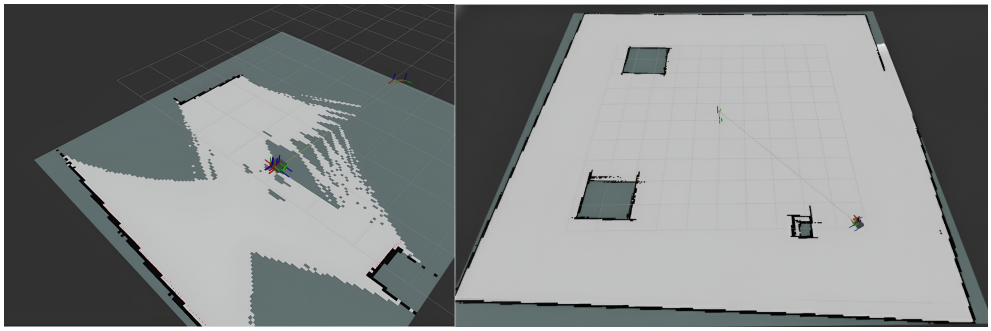

Figure 5. Progressive SLAM generated occupancy grid maps the left image shows initial partial coverage and the right image shows near complete of 97.3% map coverage after 8.4 minutes of autonomous frontier based exploration.

The TF2 transformation tree maintained consistency across the 8 coordinate frames with an average lookup latency of 15 ms. This pipeline is analogous to the coordinate chain in the BIM integrated fire response system of Chen et al. [19], where the localized hazard coordinates are propagated to the building management outputs.

4.4 Fire and Human Detection with Coordinate Localization

The fire detection pipeline (Figure 6) achieves a multi view localization error of 12.3 cm (SD: 8.7 cm), reduced from 19.0 cm of single view – a 35% improvement consistent with the multi observation fusion principle of Ren et al. [1] and within the operational tolerance for fire suppression guidance identified by Kong et al. [6].

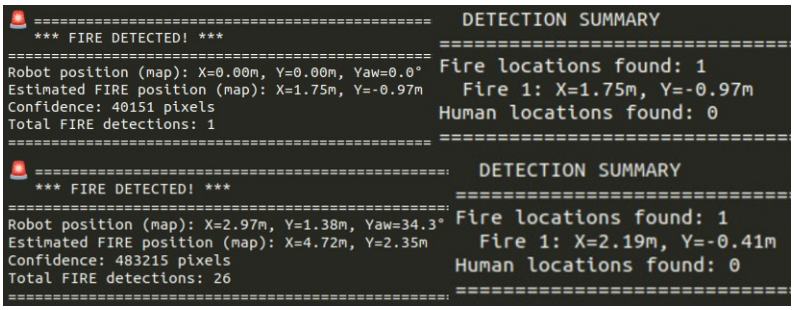

Figure 6. Terminal output of real time fire detection showcasing robot position, the estimated fire map coordinates, the detection confidence, and the overall cumulative detection count.

The HSV based human victim detection system (Figure 7) targets the blue cylindrical bodies in the HSV range [100, 100, 50] to [130, 255, 255], achieving a multi view mean localization error of 14.7 cm (SD: 11.2 cm). The improvement from 21.9 cm single view to 13.9 cm multi view mirrors the angular triangulation improvement reported by Jadeja et al. [7] for survivor detection.

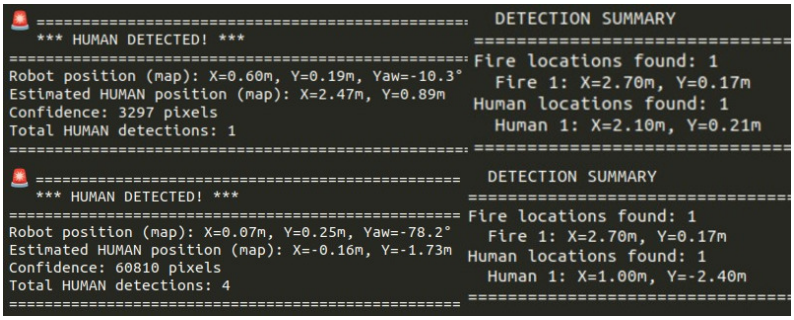

Figure 7. Terminal output of real time human victim detection showcasing estimated coordinates in the global map frame, the detection confidence, and the simultaneous fire and human detection summary.

4.5 Exploration and Path Efficiency

The proposed frontier based algorithm achieves an overall 97.3% coverage at 498 s, substantially outperforming the nearest frontier (84.6%) and random walk (76.1%) strategies [9, 10], as shown in Figure 8. Time to 75% coverage is 138 s (proposed), 198 s (nearest frontier, 43% slower), and 318 s (random walk, 130% slower), representing a 2.3× speedup over the random walk and 1.4× over nearest frontier, consistent with Wang et al. [9].

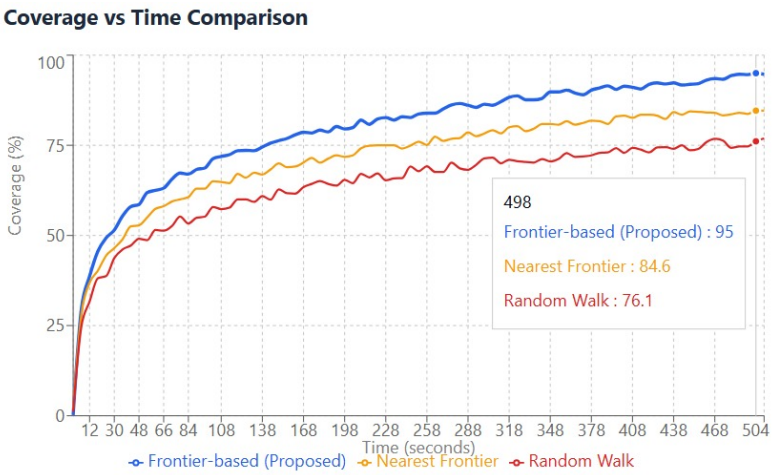

Figure 8. Coverage vs time for frontier based (representation in blue), nearest frontier (in orange), and random walk (in red) strategies. The proposed method/algorithm achieves 97.3% at 498 s vs. 84.6% and 76.1% for the baselines.

The proposed method also achieves a shortest path length at 67.4 m vs 78.8 m (nearest frontier, 17% longer) and 96.2 m (random walk, 43% longer), as shown in Figure 9. Exploration efficiency (coverage / path length): proposed 0.67 vs. 0.48 (nearest frontier) and 0.58 (random walk). The 28.8 m path reduction over the random walk reduces the energy consumption, addressing the efficiency concerns of Zhao et al. [10] and Šiktar et al. [12].

Figure 9. Path efficiency comparison: path length (in red), coverage percentage (in green), and efficiency ratio (in blue) for the three exploration strategies. The frontier based method achieves the optimal combination.

4.6 Multi View Localization and Angular Detection Performance

Fire localization error decreases from a range of 19.0 cm to 11.5 cm across the 6 views (39.5% in reduction), human error from 21.9 cm to 13.9 cm (36.5% reduction) [1, 7], as shown in Figure 10. Both curves show rapid reduction in views 1–3 (20.5% for fire) from angular

triangulation, with diminishing returns thereafter, consistent with statistical averaging theory and the multi observation analysis of Ren et al. [1].

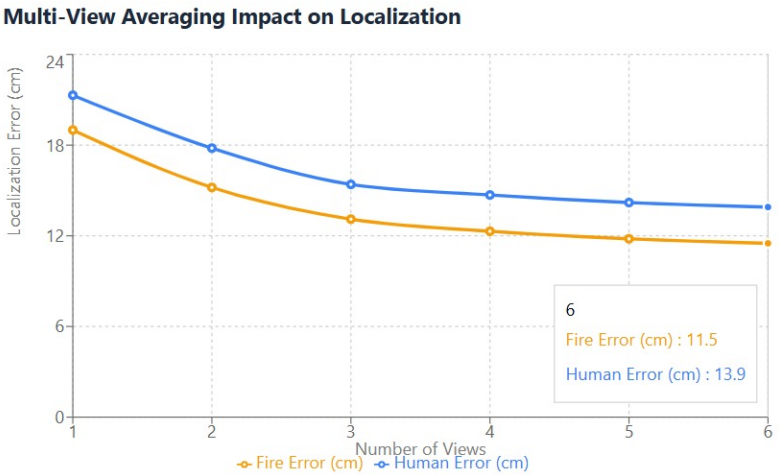

Figure 10. Multi view averaging of fire error (in orange) reduces from 19.0 to 11.5 cm and human error (in blue) from 21.9 to 13.9 cm across the 6 viewpoints.

The detection rate peaks at 100% at 0° and degrades linearly at 1.2%/degree, as shown in Figure 11 [6, 7]. The detection confidence falls from 61,847 pixels at 0° to 15,342 pixels at ±30°, maintaining >90% within ±20°, consistent with the angular fall off characterized by Kong et al. [6].

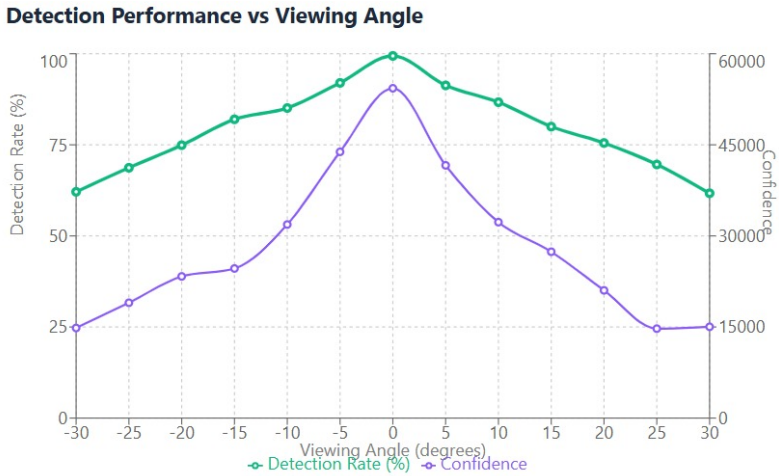

Figure 11. The detection rate (%, in green) and confidence in pixels (in purple) vs the viewing angle. Rate maintains >90% within ±20°, peaking at 100% frontally.

4.7 Unified Quantitative Results Summary

Table 3 consolidates all key performance metrics.

Table 3. The consolidated quantitative performance results

Component	Metric	Value
YOLOv8 Detection	mAP50	0.84
	mAP50-95	0.74
	Precision	0.87
	Recall	0.83
SLAM Localization	Mean pose error	4.2 cm
	Max pose error	8.7 cm
	Loop closures	12
Fire Localization	Single-view error	19.0 cm
	Multi-view error (across 6 views)	11.5 cm
	Error reduction	39.5%
Human Localization	Single-view error	21.9 cm
	Multi-view error (across 6 views)	13.9 cm
	Error reduction	36.5%
Exploration	Coverage (frontier)	97.3%
	Coverage (nearest frontier)	84.6%
	Coverage (random walk)	76.1%
	Mission duration	8.4 min
Path Efficiency	Path length (frontier)	67.4 m
	Path length (nearest frontier)	78.8 m
	Path length (random walk)	96.2 m
Navigation	Total goals sent	23
	Goal success rate	91.3%
	Avg. goal distance	3.2 m
	Recovery events	6
Computation	SLAM CPU	18%
	Nav2 CPU	12%
	Detector CPU	8%
	Total CPU	55%
	Total RAM	883 MB

4.8 Navigation, Novelty, and Research Gap Resolution

The system dispatched 23 navigation goals, achieving a 91.3% success (21/23) and 6 recovery events were handled through Nav2 rotation and backup maneuvers. The total system utilization (55% CPU, 883 MB RAM) confirms the overall embedded deployment feasibility on NVIDIA Jetson or Raspberry Pi 4, consistent with the embedded targets of Chen et al. [18] and Ulloa et al. [20].

The system's key novelties are: (1) combined YOLOv8 and HSV detection, avoiding the single modality limitations of prior approaches [5, 12]; (2) a coordinate transformation pipeline converting camera frame detections to globally consistent map coordinates, absent in most prior work [12, 17]; (3) the deterministic frontier exploration eliminating RL-based training overhead [9, 10]; and (4) full SLAM-detection-navigation integration within a lightweight ROS2 architecture deployable on embedded hardware [20].

Research Gap 1 (thermal sensor complexity) is resolved through YOLOv8 detection (mAP50: 0.84, 8% CPU) without thermal imaging [6, 11]. Gap 2 (SLAM semantic capabilities) is re-

solved through the coordinate transformation pipeline which bridges detection and global localization [13, 8, 14]. Gap 3 (physical validation absence) is addressed through TurtleBot3 simulation with embedded-suitable resource utilization [15, 16, 20]. Gap 4 (exploration efficiency) is overcome by 97.3% coverage in 8.4 minutes, 91.3% navigation success and no RL-based training requirement [9, 10, 2].

4.9 Real-World Deployment Considerations

Several real world challenges require acknowledgement for physical deployment. Variable lighting requires adaptive camera exposure control [3] and LiDAR lens coating from dust and debris can degrade scan quality, as noted by Koval et al. [22]. Dynamic obstacles require real-time costmap updates and the constant depth assumption ($d = 2.0$ m) must be replaced with RGB-D or ultrasonic depth sensing, analogous to the sensor occlusion issues of Jiang et al. [21]. The battery limited operational time (60–90 minutes) necessitates energy-aware mission planning for sustained SAR missions. Communication reliability in dense concrete structures remains a challenge documented in the GPS-denied survey of Chang et al. [3]. Future work will address these through physical platform testing with the identified sensor augmentations and following the validation methodology of Koval et al. [22].

5 Conclusions

This paper presented an integrated autonomous Search And Rescue (SAR) robotic system combining YOLOv8 fire and smoke detection, HSV based human victim identification, SLAM mapping, and frontier based navigation. The key results: mAP50 0.84, precision 0.87, recall 0.83 on 3,000 images; 97.3% coverage in 8.4 minutes; multi view localization errors of 12.3 cm (fire) and 14.7 cm (human); 91.3% navigation success and 55% CPU, 883 MB RAM – suitable for embedded deployment [18, 20].

Limitations include the constant detection range assumptions, simulation only validation, and color specificity of HSV human detection. Future directions include: RGB-D depth integration following Ren et al. [1]; multi-robot cooperation; 3D SLAM for multi floor structures; physical deployment with dust and smoke robust sensors as recommended by Koval et al. [22]; thermal imaging for smoke occluded scenarios [11]; and GPS denied multi level navigation following Chang et al. [3]. The validated system provides a reproducible foundation for practical autonomous SAR operations.

References

[1] X. Ren, K. Qu, J. Guo, H. Yin, B. Huang, Research on accurate fire source localization and seconds-level autonomous fire extinguishing technology, *Sci. Rep.* **15**(1), 17135 (2025). https://doi.org/10.1038/s41598-025-01830-5

[2] C. Liu, D. Zhang, W. Liu, X. Sui, Y. Huang, X. Ma, X. Yang, X. Wang, Enhancing autonomous exploration for robotics via real time map optimization and improved frontier costs, *Sci. Rep.* **15**(1), 12261 (2025). https://doi.org/10.1038/s41598-025-97231-9

[3] Y. Chang, Y. Cheng, U. Manzoor, J. Murray, A review of UAV autonomous navigation in GPS-denied environments, *Robot. Auton. Syst.* **170**, 104533 (2023). https://doi.org/10.1016/j.robot.2023.104533

[4] Mordor Intelligence, Search and Rescue Robots Market Size and Share Analysis: Growth Trends and Forecasts (2025–2030) (2025)

[5] F. Bu, M. S. Gharajeh, Intelligent and vision-based fire detection systems: A survey, *Image Vis. Comput.* **91**, 103803 (2019). https://doi.org/10.1016/j.imavis.2019.08.007

[6] L. Kong, J. Li, S. Guo, X. Zhou, D. Wu, Computer vision based early fire-detection and firefighting mobile robots oriented for onsite construction, *J. Civ. Eng. Manag.* **30**(8), 720–737 (2024). https://doi.org/10.3846/jcem.2024.21360

[7] R. Jadeja, T. Trivedi, J. Surve, Survivor detection approach for post earthquake search and rescue missions based on deep learning inspired algorithms, *Sci. Rep.* **14**(1), 25047 (2024). https://doi.org/10.1038/s41598-024-75156-z

[8] A. Tourani, H. Bavle, J. L. Sanchez-Lopez, H. Voos, Visual SLAM: What are the current trends and what to expect?, *Sensors* **22**(23), 9297 (2022). https://doi.org/10.3390/s22239297

[9] R. Wang, J. Zhang, M. Lyu, C. Yan, Y. Chen, An improved frontier-based robot exploration strategy combined with deep reinforcement learning, *Robot. Auton. Syst.* **181**, 104783 (2024). https://doi.org/10.1016/j.robot.2024.104783

[10] H. Zhao, Y. Guo, Y. Liu, J. Jin, Multirobot unknown environment exploration and obstacle avoidance based on a Voronoi diagram and reinforcement learning, *Expert Syst. Appl.* **264**, 125900 (2025). https://doi.org/10.1016/j.eswa.2024.125900

[11] B. Eslam, A. Samir, M. Osama, M. Moustafa, M. Ali, The evolution of firefighting robots: Bridging technology and safety: A comprehensive review, *Adv. Sci. Technol. J.* (2025)

[12] L. Šiktar, B. Ćaran, B. Šekoranja, M. Švaco, Autonomous UAV navigation for search and rescue missions using computer vision and convolutional neural networks, arXiv:2507.18160 [cs.RO] (2025). https://doi.org/10.48550/arXiv.2507.18160

[13] Y. Zhou, M. Sun, A visual SLAM loop closure detection method based on lightweight siamese capsule network, *Sci. Rep.* **15**(1), 7644 (2025). https://doi.org/10.1038/s41598-025-90511-4

[14] L. Zhao, T. Chen, P. Yuan, X. Li, B. Chen, Review of deep learning-based visual SLAM: Types, approaches, and future work, *Ind. Robot* (2025). https://doi.org/10.1108/IR-04-2025-0137

[15] A. A. Adil, S. Sakhrieh, J. Mounsef, N. Maalouf, A multi-robot collaborative manipulation framework for dynamic and obstacle-dense environments: Integration of deep learning for real-time task execution, *Front. Robot. AI* **12**, 1585544 (2025). https://doi.org/10.3389/frobt.2025.1585544

[16] V. M. Tuck, H. Parwana, P.-W. Chen, G. Fainekos, B. Hoxha, H. Okamoto, S. S. Sastry, S. A. Seshia, MRTA-Sim: A modular simulator for multi-robot allocation, planning, and control in open-world environments, arXiv:2504.15418 [cs.RO] (2025). https://doi.org/10.48550/arXiv.2504.15418

[17] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohajer, M. Rokonuzzaman, I. Hossain, A comprehensive review on autonomous navigation, *ACM Comput. Surv.* **57**(9), 234 (2025). https://doi.org/10.1145/3727642

[18] W. Chen, W. Chi, S. Ji, H. Ye, J. Liu, Y. Jia, J. Yu, J. Cheng, A survey of autonomous robots and multi-robot navigation: Perception, planning and collaboration, *Biomim. Intell. Robot.* **5**(2), 100203 (2025). https://doi.org/10.1016/j.birob.2024.100203

[19] H. Chen, L. Hou, G. Zhang, S. Moon, Development of BIM, IoT and AR/VR technologies for fire safety and upskilling, *Autom. Constr.* **125**, 103631 (2021). https://doi.org/10.1016/j.autcon.2021.103631

[20] C. C. Ulloa, J. Álvarez, J. del Cerro, A. Barrientos, Vision-based collaborative robots for exploration in uneven terrains, *Mechatronics* **100**, 103184 (2024). https://doi.org/10.1016/j.mechatronics.2024.103184

[21] S. Jiang, S. Wang, Z. Yi, M. Zhang, X. Lv, Autonomous navigation system of greenhouse mobile robot based on 3D LiDAR and 2D LiDAR SLAM, *Front. Plant Sci.* **13**, 815218 (2022). https://doi.org/10.3389/fpls.2022.815218

[22] A. Koval, S. Karlsson, G. Nikolakopoulos, Experimental evaluation of autonomous map-based Spot navigation in confined environments, *Biomim. Intell. Robot.* **2**(1), 100035 (2022). https://doi.org/10.1016/j.birob.2022.100035

[23] S. Li, Z. Ren, J. I. Kim, A structured review of SLAM generation in the AEC industry: Technical framework, site-specific challenges, and adaptive strategies, *KSCE J. Civ. Eng.* **30**(3), 100408 (2026). https://doi.org/10.1016/j.kscej.2025.100408