

## Enhancing Home Safety through Multisensory Robotic Detection of Fallen Persons and Fall Hazard Warning

<sup>1</sup> David FEILACHER, <sup>1</sup> Andreas JAKL, <sup>1</sup> Bernhard GIRSULE,  
<sup>1</sup> Sabine LEHNER, <sup>2</sup> Henrik SMITH, <sup>2</sup> Sven DOMINKA <sup>1</sup> Mario HELLER  
<sup>1</sup> University of Applied Sciences St. Pölten, Austria  
<sup>2</sup> Bosch Engineering Robert Bosch AG, Austria

Received: 29 August 2025 / Revised: 11 Nov. 2025 / Accepted: 15 Dec. 2025 / Published: 30 Dec. 2025

---

**Abstract:** Falls are a major risk for independently living elderly people, often resulting in hospitalization or loss of independence if help is delayed. Reliable, unobtrusive fall detection is important – but existing systems often rely on a single modality, require manual activation, or constantly surveil the user, limiting user acceptance. We present Smart Companion, an audio-visual interactive lying person detection system embedded in a robot vacuum cleaner, augmented with a voice assistant. It operates as a daily-use appliance with the added functionality of lying person detection and fall hazard notification. The system utilizes multi-view image analysis with a convolutional neural network for lying people detection and initiates verbal interaction to further assess the situation. With this combination, the system reliably triggered emergency services and caused only one false alarm during six months of a prototype field study in the private apartments of elderly users. The dialog and system behavior were developed through a user-centered design process, contributing to high user acceptance.

**Keywords:** Fall detection, Lying person detection, Fall hazard notification, Human-robot interaction, Convolutional neural network, Voice assistant, Multimodal sensing.

---

### 1. Introduction

With an aging population and rising healthcare costs, independently living elderly people are increasingly common [1]. Ensuring both their safety and autonomy requires the reliable detection of emergency situations with minimal intrusion into personal privacy. Falls, in particular, can lead to prolonged hospitalizations or permanent injury if left undetected [2].

#### 1.1. Lying Person Detection

While a variety of fall detection systems exist [3], many either require manual activation, rely on a single sensing modality, or continuously surveil the user – often leading to poor user acceptance [1]. Manual call buttons are often not worn or activated during an

emergency [4]. Many fall detection systems focus on the fall event itself rather than the state of a person lying on the ground. However, many falls lack the clear impact patterns that these fall detection systems are designed to recognize [5]. While in general the brief nature of the fall event itself is a better indicator for the time of occurrence, it can also be missed more easily or result in false positives because of confusion with other movements [6]. If the event is missed, people can lie on the ground for a long time, with severe effects for their health [7]. An analysis of 1417 observed falls among older adults [5] shows that while walking is a frequent cause of falls, a substantial proportion occur in contexts such as getting out of bed, standing up, sitting, or during the use of a wheelchair. Fall detection systems like smartwatches struggle with these scenarios, leading to very low sensitivity [8]. A person lying on the floor also represents a more general potential emergency scenario, whether induced by a fall or for other reasons.

In contrast, our detection system (Smart Companion) is designed to reliably detect lying individuals, avoiding constant surveillance. It also provides an additional everyday utility, promoting user acceptance.

## 1.2. Assistive Scenarios

The system builds on the results our prior exploratory study [9], identifying four scenarios in which an assistive mobile robot with voice assistance would ideally support its user. This covers not only lying person detection but also fall prevention and general emergency support (see Fig. 1). Lying person detection is covered by scenarios **a** and **b**:

- a) Detection of an **unresponsive** lying person;
- b) Detection of a **responsive** lying person.

As the core functionality of our system, they aim to reduce the time until emergency support arrives after a fall, if the person is unable to receive assistance via other means. They are both based on the robot actively scanning for a lying person, while performing its function as a vacuum cleaner. Cleaning runs are supplemented by faster search runs to increase the search frequency according to the users situation and wishes. Once a lying person is detected, the robot starts its emergency Verbal Interaction (VIA), during which the user can verbally confirm or refuse their need for assistance. In case of a non-responsive user, the emergency service is notified automatically.

To provide more generalized emergency support, we added scenario **c**:

- c) Proactive call for help;

In this use case, the user can call for assistance for any reason, independent of lying person detection. The implementation of the verbal communication is discussed in Section 3.4.

According to [10, 11] around 10% of all falls among older adults are caused by trip hazards in the form of objects in their environment. To prevent falls from occurring in the first place, potential hazards can be detected and the user notified accordingly. Therefore, we added the scenario:

- d) Fall prevention via fall hazard detection and notification.

It is based on the robot actively scanning for potential fall hazards during a cleaning run. If detected, the robot communicates the type and position of the fall hazard, motivating the user to avoid or remove it. While this measure is proposed at a conceptual level by works such as [12], we demonstrate and test its feasibility in practice. The implementation specifics are discussed in Section 3.5.

When the system is not triggered proactively, detection speed depends on run-scheduling as the system is not designed for immediate emergency response. A high-frequency search schedule could improve detection speed, but would likely decrease user acceptance. Therefore, its ideal use case is a combination with quick, non-invasive systems, to provide detection speed and a reliable backup in

situations where the user is non-responsive and other measures are not triggered.

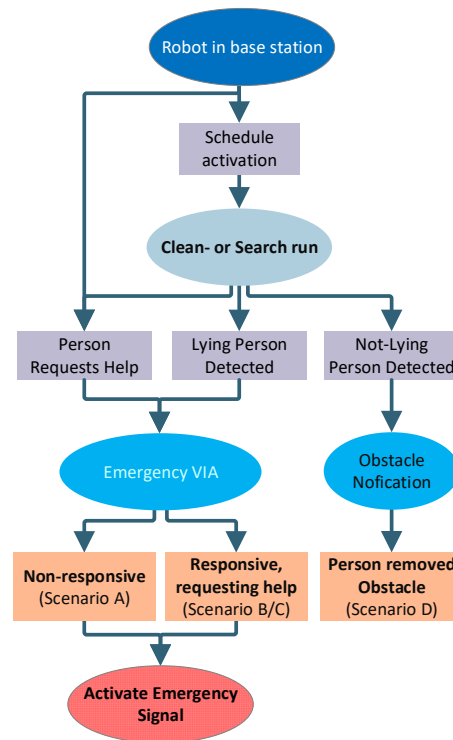


Fig. 1. Flowchart of assistive scenarios.

## 2. Related Work

Overviews of fall detection systems are given in [3] and [13]: Wearable sensors such as accelerometers and gyroscopes are portable but depend on user compliance. Permanent imaging systems provide detailed monitoring but raise privacy concerns. Environmental sensors like smart carpets are unobtrusive but lack portability. Imaging systems are dominated by RGB video, with depth or infrared sensing being the exception [14] though these would reduce privacy issues. Recent developments are Wi-Fi and radar-based technologies, which, while privacy-preserving, often require multiple sensors for full coverage, similar to stationary RGB cameras, with Wi-Fi being less precise and radar more complex and costly ([3, 15]).

### 2.1. Fall Detection Systems

Prior work has also examined multimodal systems and mobile robots equipped with fall- and lying person detection capabilities, offering a closer comparison to our work.

As early as 2014, [16] proposed a concept for an assistance system for lying individuals, combining a stationary visual fall detection setup with a robot vacuum cleaner equipped with basic voice communication. Once triggered by the fall detection

system, the robot would navigate to the location and attempt verbal interaction. If unanswered, it would notify a caregiver. However, as a conceptual study, it lacked implementation details and a field study.

The authors of [17] developed a fallen person detection system using an autonomous assistive robot on a rollator platform equipped with a camera. Images were analyzed using a YOLOv3 Convolutional Neural Network (CNN) combined with a Support Vector Machine (SVM). The system achieved high recall and precision on their custom dataset, however no real-world field tests were conducted, and the dataset seems skewed towards images of fallen persons, leaving its robustness under realistic conditions uncertain. In addition, the robot's large size and lack of secondary functionality may further limit user acceptance.

Showing promise for unobtrusive integration into daily-use robots, [18] proposes a human activity detector integrated in a robot vacuum cleaner. It uses 2D-LiDAR (Light Detection and Ranging) scans and analyzes with a Conv-LSTM (Long Short-Term Memory) model, reaching a very high F1 score of 0.99 for lying body detection. Testing was however limited to simulations.

[19] presents a fall monitoring system combining a dual camera setup with YOLOv11-Pose detection and a simple voice confirmation step using Google Speech-to-Text to reduce false alarms. The system achieved an F1-score of about 0.94 on 200 staged sequences, rising to 0.98 with voice confirmation to cut false alarms. However, testing was limited to a fixed setup.

Using the NAO humanoid robot for fall prevention in smart homes, [20] reported that it was unable to navigate cluttered household environments, limiting its practical deployment.

## 2.2. Fall Hazard Notification

Although fall detection has been widely studied, less work addresses fall hazard detection and notification as discussed in [13]. In particular, robotic and environmental sensing technologies have the potential to enable proactive identification and mitigation of fall hazards.

Already in 2014, [21] demonstrated the advantages of a robot's floor-level perspective for identifying environmental hazards relevant to fall prevention, via remotely operated robots as a feasibility study.

In [22], an augmented reality-based home assessment system is presented, in which a caregiver uses a head-mounted device to scan and visualize potential trip hazards. While effective for manual assessment, it relies on a separate operator and lacks autonomous detection or user notification.

A probabilistic framework for context-aware risk estimation in home environments could enable robots to infer potential fall hazards from object relationships [23]. In their publication, they achieved about 75 % accuracy, showing strong potential for semantic risk

reasoning, though not yet validated in a field test and lacking the capability to notify users directly.

## 2.3. Conclusion

As this overview shows, existing work for fall detection often relies on fixed installations or specialized robotic platforms with little practical testing, facing challenges in privacy, utility, or user acceptance. Robot vacuum-based concepts have so far remained at the simulation or concept stage.

By comparison, our system demonstrates the integration of lying-person detection, voice interaction, and emergency connectivity into a commercially available robot vacuum, combining everyday functionality with assistive capability. A long-term field evaluation shows reliable operation and high user acceptance under real-world conditions.

Furthermore, it detects trip hazard objects and notifies the user directly, enabling timely avoidance or removal during daily activities. To the best of our knowledge, such user-oriented fall-hazard warning has not previously been realized in a home robotic platform.

## 3. Implementation

This section introduces the overall system architecture, summarizing how the robot vacuum cleaner integrates fall detection and user interaction. Further subsections give more background on the major components and obstacle detection for fall prevention.

Fig. 2 shows a technical system overview of the core functionality handling scenarios **a** and **b**: During a search or cleaning run, the robot periodically checks the user's living area (1). The schedule for both is set by the user and the frequency can be increased in case of a higher fall probability. During these runs, the robot records images (2). If its image analysis detects a lying person, it activates its voice assistant to communicate verbally with the person, to ascertain if assistance is actually required (3). This preserves the user's control and prevents false alarms. If the user asks for help or does not respond, an emergency service is notified, receiving a picture of the lying person for further assessing the situation (4).

Verbal communication also enables users to actively call for help at any point, not having to wait until the robot finds them (scenario **c**). The voice interface additionally enables users to operate the robot's regular functions.

### 3.1. System Components

For the robot to achieve its tasks in all four scenarios, the system relies on three primary sensors: LiDAR for navigation, an RGB camera for visual

detection of lying individuals, and a directional microphone for voice-based interaction, as shown in Fig. 2 and detailed in Sections 3.2 to 3.4.

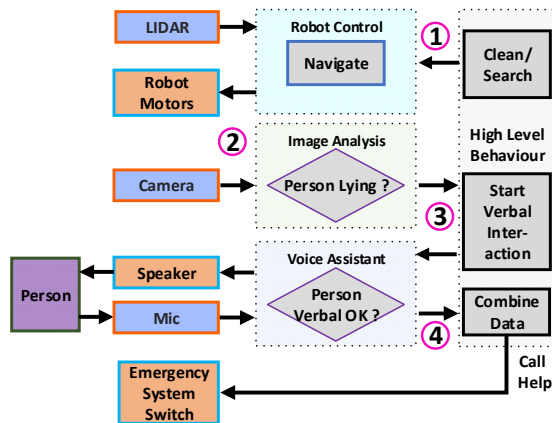


Fig. 2. System components and interactions during a search or cleaning run in scenarios a and b. The robot finds a lying person and communicates verbally to deduce if they need help.

While the image analysis is structured as a feed-forward processing pipeline, the voice assistant and overall system behavior are achieved with cascaded state machines (see Fig. 4). The voice assistant's actions depend on verbal user input following a dialog model controlled by a dialog state machine (DSM). The overall system behavior depends on feedback from the DSM and all other system components and is controlled by the behavior state machine (BSM). Both state machines are custom designs, developed for robustness and reliability. The core of the image analysis is the YOLOv8 CNN, fine-tuned specifically for the detection of lying people.

The stock robot is augmented with a voice assistant, an emergency service switch and a main control unit (MCU, Raspberry Pi 4B). The MCU runs the BSM. The emergency service switch is integrated with the emergency service of the ASB. The system's voice assistant is an Amazon Echo Dot smart speaker that utilizes the Amazon Voice Service (AVS) for speech recognition and response handling. These components are housed in a 3D printed mount and draw power from the robot. The augmented robot can be seen in Fig. 3.

The image processing hardware (NVIDIA Jetson Orin Nano<sup>1</sup>) is located at the robot's base station to conserve energy and space on the robot. The MCU communicates with the Jetson and robot via Wi-Fi. While the MCU, Jetson and robot process data locally, the voice assistant code is executed in the AVS Cloud. To ensure robust communication, the system performs continuous connection checks with error handling and

retries, and all inter-component signals require acknowledgements. Robot actions and voice assistant interactions are automatically retried if they fail, while the BSM supervises component behavior and adapts when deviations from expected operation are detected.

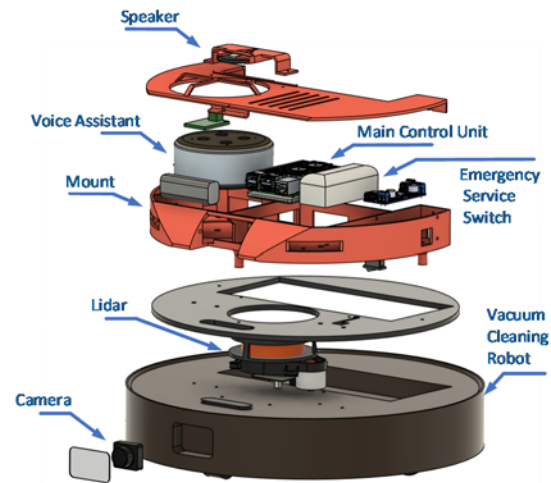


Fig. 3. Augmented robot: 3D printed mount with Alexa voice assistant, MCU<sup>2</sup>, ASB emergency service switch.

### 3.2. Robot Interaction

The Dreame D10s Plus<sup>3</sup> robot vacuum cleaner was chosen due to its affordable price of around 400 € (in 2025), its onboard camera and its auto-empty base station for long-term use without manual intervention. To interact with the robot, the open-source server Valetudo<sup>4</sup> is installed, as the original proprietary firmware offers no open control interface. Valetudo enables reading the robot status and sending commands via an API and prevents the robot from sending data to the manufacturer, ensuring data privacy.

The robot's LiDAR records distance measurements and its firmware facilitate navigation with its proprietary Simultaneous Localization and Mapping (SLAM). Once an area is mapped, sections can be set off-limits for privacy- or other reasons. The BSM relays mapping and navigation commands to the robot through Valetudo. The stock robot is also equipped with lift and jamming sensors and onboard control buttons. The buttons enable manual control and can override the system's behavior. When the robot is lifted, a jam is detected, or other mechanical errors occur, the voice assistant provides verbal notification to the user.

When a scheduled search or cleaning run is started (Fig. 4, top-left), the BSM switches to its search or cleaning state and executes the respective movement (Fig. 4, bottom-right). While cleaning, the robot

<sup>1</sup> <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/>

<sup>2</sup> 3<sup>rd</sup> party 3D model: <https://grabcad.com/library/raspberry-pi-4-heatsink-case-with-fan-1> (Badea Andrei)

<sup>3</sup> <https://global.dreametech.com/products/dreamerobot-d10s-plus>

<sup>4</sup> <https://valetudo.cloud/>

systematically navigates the user's living space in a meandering pattern, capturing images from various angles and positions. During a search run, it moves

between custom points and rotates on the spot to take images. These runs are faster but still provide multiple views for each room section.

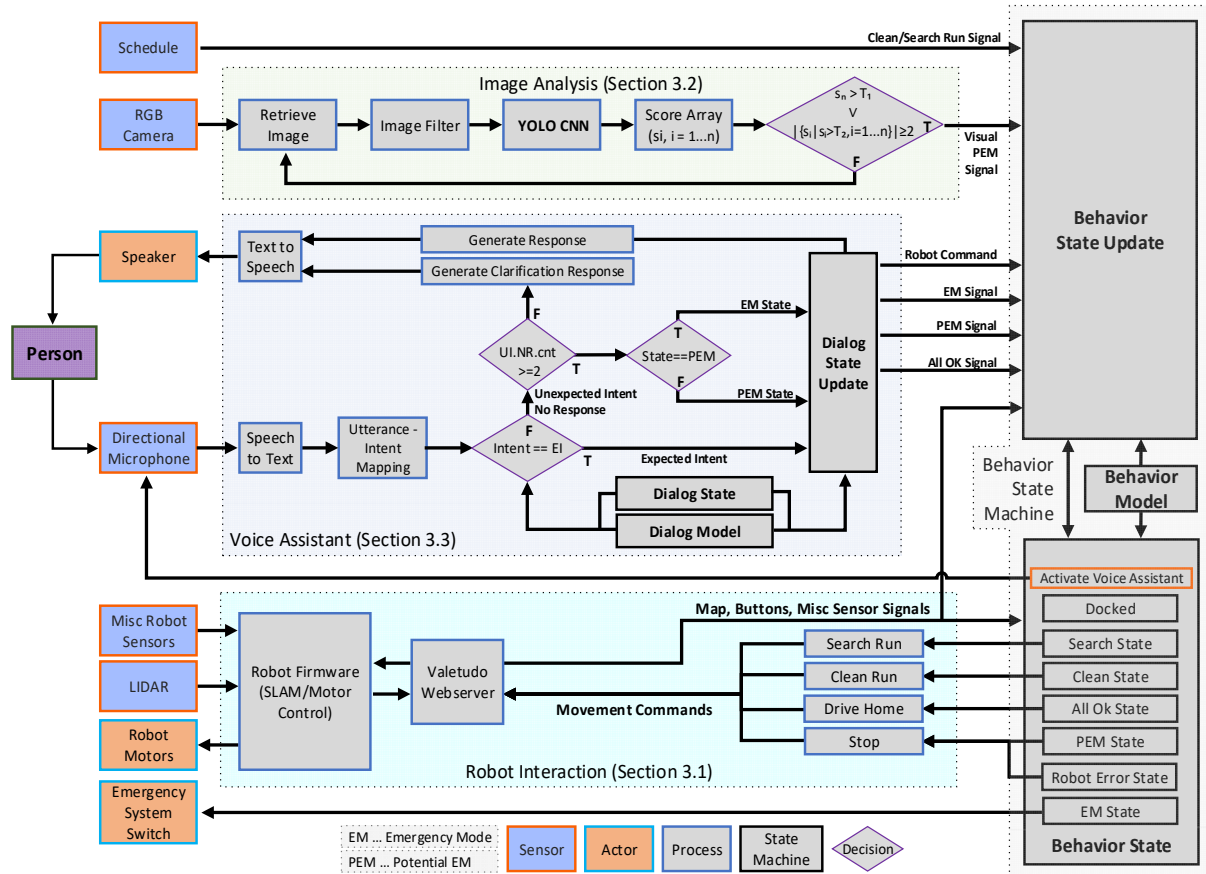


Fig. 4. System components – behavior overview.

If a lying person is detected but declines help, the robot returns to the base station or continues cleaning (Fig. 4, bottom-right). If a user asks for help but does not respond further, a search run is automatically started.

### 3.3. Image Analysis

During a run, the robot vacuum cleaner's RGB camera continuously records images, which are retrieved by the MCU every two seconds (Fig. 4, top left).

The image filter then discards images too similar to previous ones to prevent duplicate- and near-duplicate detections due to similar views, which is important for the later multi-view analysis. To achieve this, new images are first down-sampled by a factor of four, transformed from RGB to greyscale values and brightness-equalized (histogram equalization). The transformed images are stored in a FIFO array with a capacity of fourteen, this number explained further down. They are then compared with the transformation of the other previous images: If the mean pixel difference is less than 10 %, the new image

is discarded; otherwise, the image is passed to the YOLO CNN.

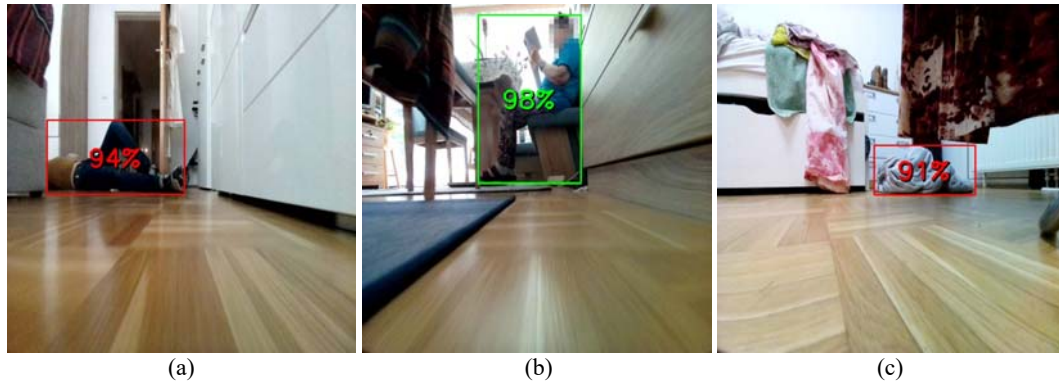
The YOLOv8 [24] CNN was adapted and fine-tuned for detecting fallen people by our project partners at Bosch. The original classification head layer was removed, and three new classes were introduced: "person", "lying person" and "no person". The model was fine-tuned on a combination of public and custom datasets (21k images), labeled for these classes, and evaluated on separate test sets (5k images). To meet the project's requirements, the custom dataset includes images from the robot's onboard camera, as this specific viewpoint is underrepresented in public datasets, particularly for lying individuals. The images were recorded in various test settings and later expanded during the field study to further improve the model.

To maintain the model's ability to extract generalizable features during fine-tuning, we tested freezing between the first 4 and the first 20 layers, with the best performance achieved when freezing the first 13–17 layers. To prevent overfitting, different training epoch counts were tested, validated by monitoring a YOLO-specific loss function [25] for up to 1200 epochs. The best performance was observed

around epoch 50. The final model configuration was the YOLOv8 x-large model with the first 15 layers frozen during fine-tuning.

As finding lying people was the priority, the model was first optimized for a high recall, achieving 0.91. In combination with analyzing multiple views, this proved to be very effective in field evaluations.

Optimizing precision was the secondary objective (preventing false alarms). This slightly reduced recall but showed no noticeable negative effects during field tests for overall lying person detection. The final performance achieved was a recall of 0.86 and a precision of 0.93. Three example classifications from the field study are shown in Fig. 5.



**Fig. 5.** Detection score results with bounding boxes from images taken by the robot during the field study: (a) “lying person” (0.94); (b) “person” (0.98); (c) misclassification of a pile of clothing as a “lying person” (0.91).

During a cleaning run, the robot’s meandering path captures images from multiple angles of most positions in a room. This automatically makes it likely to obtain views suitable for person detection, which depends on the person’s pose and position relative to the camera. In contrast, the supplementary search runs are designed to be quick and unobtrusive, with the robot navigating to a limited set of pre-defined points (search points) distributed across the living area. When at a search point, it rotates in seven steps while taking images at each step. When the search points are appropriately chosen, a person lying between two search points is thus captured from multiple overlapping view angles within the last fourteen images.

Continuing in Fig. 4, after the “YOLO” block, the classification scores of the images are stored in the “score array”. Two heuristically determined rules are then applied: a detection is considered valid if, within the last fourteen images, one of them applies:

- One score exceeds 0.91;
- At least two scores exceed 0.85.

When this condition is met, the “Visual Potential Emergency” (Visual PEM) signal is set, and the BSM switches to the PEM state, stopping the robot and initiating verbal communication via the voice assistant (“Activate Voice Assistant” signal). This will then lead to the activation of the verbal interaction.

### 3.4. Verbal Interaction

Verbal communication is implemented with the Alexa Skills Kit (ASK) [26], using an Amazon Echo Dot smart speaker mounted on the robot, see Fig. 3. Voice processing is done in the AVS Cloud. To

interpret user’s utterances, they are transcribed to text and compared with predefined utterances to determine the best match. Predefined utterances are grouped into intents to interpret their meaning (i.e., “help”, “I need assistance”, “call someone”, etc. are grouped into the “help” intent). The intent resulting from the user’s utterance determines how the conversation flow progresses. In the ASK, each intent has its own independent customizable handler function leading to a distributed control system. However, to create a centralized behavior, we use all intent functions as triggers for the DSM, passing their intent along as input. The DSM controls the dialog, guided by a dialog model. It can process all possible user inputs by dividing them into expected intents (EI), unexpected intents (UEI) and non-responses (NR). Whether an intent is expected depends on the current state of the conversation and the dialog model configuration. For example, the “help” intent (allowing the call for help), and the “go-home” intent (used to end a conversation) are both always expected. However, if a call for help is already ongoing, the “start cleaning” intent is not expected and thus has no dialog flow linked to it.

Each conversation state can only be left with specific exit intents and entered with specific entrance intents. In case of a UEI or NR, the previous voice assistant statement can be repeated and expected answers are provided (clarification response). Definition of entrance and exit intents, combined with the responses generated when a state is entered, form the basis of the dialog state (see Fig. 6).

States also execute additional functionalities for communication with other system components and emergency handling. Linking of states’ exit intents to other states’ entrance intents forms the dialog flow, which is configured via a JSON file. Simply adapting

this JSON file allows for configuration and expansion of the dialog model. The final model was developed in a user-centered design process, including potential users, to adapt the responses and expected user utterances in accordance with their behavior and remarks. It contains 398 utterances grouped into 21 intents and over 40 dialog states. For more details on the DSM and its development please refer to [27]. A similar principle is used for configuring the BSM, with generalized input signals originating from the system components replacing intents.

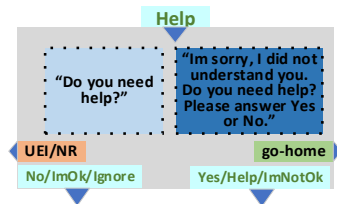


Fig. 6. Dialog state with clarification response, entrance and exit intents.

Fig. 4 shows the DSM's operation in emergency scenarios. Starting from the microphone: the Alexa device records statements and sends them to AVS where they are transcribed (speech to text). The text is compared with a list of utterances to interpret its intent (utterance-intent mapping). If the intent is expected, the conversation is continued according to the dialog model, followed by the system switching to the appropriate next state (dialog state update). The dialog model's text response is synthesized into a sound file, transferred to the Alexa smart speaker, and played.

If a lying person is detected, the voice assistant queries the user's status, and one of the following will occur:

- If the user at any point declines help with EIs, the All-OK signal is sent to the BSM;
- If the user asks for help with EIs, the emergency mode (EM) signal is sent to the BSM;
- If a UEI/NR is detected twice, the system switches to the Potential Emergency Mode (PEM). The user will then be asked again if they need help, in a stern manner. If again a UEI/NR is detected twice, or the user asks for help with EIs, the EM signal is sent to the BSM.

If the user proactively requests help, and the request is confirmed after a follow-up question, the EM signal is sent.

If the All-OK signal is sent, the robot continues its previous action (e.g., cleaning) or drives home. The EM signal leads to the activation of the emergency system switch and the notification of the ASB emergency service.

### 3.5. Fall Hazard Detection & Notification

The base of fall hazard detection is identifying and locating objects that pose a fall hazard. While the

unaltered YOLOv8 CNN can classify various objects, determining their position requires projecting the image-based detections into the robot's map. This depends on pose information at the time of image capture or the ability to correlate detections with LiDAR data. Pose data is not exposed frequently enough by the firmware and can't be correlated with timestamps with recorded images. Furthermore, we don't have access to internal characteristics such as the camera matrix. So instead, we use results of the already implemented obstacle detection performed by the robot's firmware, which the robot uses to avoid obstacles during vacuuming.

Although the details of the proprietary object detection are not publicly documented, empirical observations indicate that it performs best for nearby obstacles, while multi-view detections can further refine their estimated positions, likely by means of state estimation techniques. While this object detection theoretically provides an easy-to-use functionality with over a hundred object classes, there are issues regarding object persistence.

Object locations can fluctuate between map updates (while the robot is moving), sometimes object detections disappear completely. Object classifications show similar inconsistencies, and all detected obstacles are cleared when a new cleaning run starts. To achieve a persistent but up-to-date obstacle map based on the firmware's detections, several methods were implemented. These include persistent storage, consistency checks, area-based obstacle definitions, presence-map removal logic, and post-run filtering:

During obstacle recording, a temporary obstacle file is maintained and updated if the firmware reports a change in its obstacle findings. Obstacles are only deemed permanent if they appeared in each consequent firmware-triggered map update for 20 times (corresponding to about 1 minute). When returning to the dock, the temporary file is merged with a persistent file containing attributes such as object classification, coordinates, and user notification status. Since positional information for the same obstacle fluctuates, an obstacle is assigned an area rather than a point (10 cm radius). New detections within this area update existing obstacles rather than create new ones.

Only cleaning runs are used for data collection, as the robot follows a meandering path that covers the entire floor area, and objects must be close to the robot to be reliably detected. Comprehensive coverage is also essential for removing objects from the persistent file, since the only reliable way to confirm that an object is no longer present is for the robot to traverse its previous location and fail to detect it again.

To support this process, a "presence" map (Fig. 7) is introduced, marking all areas visited by the robot. The map is generated by dividing the robot's trajectory into 5 cm segments (map resolution) and adding a circular area corresponding to the size of the robot for each segment (17 cm radius), contributing values to a 2D map image. An object is deemed removed when

the accumulated and normalized presence values over its object area exceed a predefined threshold of 0.75.

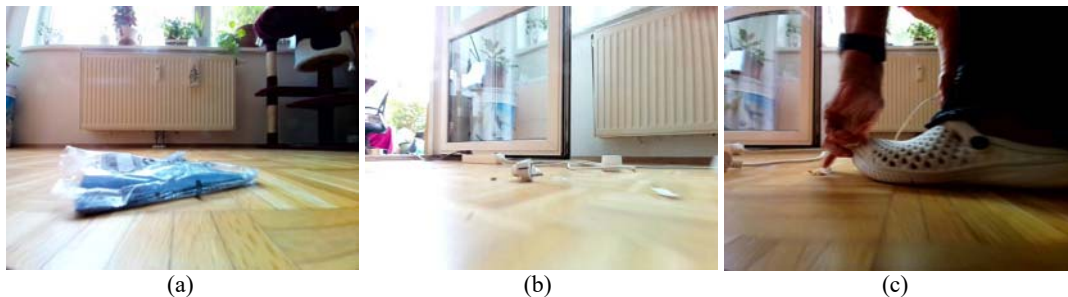


**Fig. 7.** Presence map, showing visited areas in white (except walls). Pink 'x's mark positions of obstacles that could not be removed as the robot did not visit the area.

Compared to a binary overlap check, this cumulative approach captures both direct and peripheral passes, reducing the chance that previous misclassifications of fixed objects persist in the map.

The normalizing factor is derived by simulating an ideal case in which the robot moves in a straight line directly over the object's center. For each 5 cm step, the number of overlapping pixels between the robot's footprint and the object area is added to a sum. The sum of all these discrete overlaps represents the maximum theoretical coverage used for normalization. This value is about 2000, so the chosen absolute threshold value is 1500, indicating the robot has passed mostly over the area. This threshold proved to be suitable during the study.

Since obstacle detections can fluctuate significantly during a cleaning run, the persistent list is updated only after a cleaning run has been completed. This approach ensures that temporary or incorrect detections are filtered out before the final update. User notifications are likewise generated exclusively from the persistent list to avoid unnecessary alerts caused by transient or moving objects. For example, shoes worn by a standing or walking person could otherwise be repeatedly reported as obstacles during a single run. Images taken by the robot of detected obstacles of type "fabric" and "cable" can be seen in Fig. 8(a) and Fig. 8(b) respectively. These images were taken approximately when the obstacles were detected for the first time and thus it was concluded that the obstacle detection referred to them.



**Fig. 8.** Obstacles detected by the robot that pose a potential fall hazard, about which the user was notified and led to: (a) Discovered type "textile"; (b) "Cable"; (c) User seen removing the cable after being notified.

### 3.6. User Obstacle Notification

The obstacle notification dialog is fully integrated in the Alexa skill implementation DSM, simply adding obstacle-related intents and dialog states. While its dialog flow forms its own side branch, parallel to the emergency dialog, they are also linked via entrance and exit intents. For example, the dialog can flow to the emergency branch if, during the obstacle dialog, the user does not respond anymore, calls for help, or issues other robot commands (e.g. drive home). And in the other direction when the user answers again and declares that no help is needed, the obstacle dialog will continue.

Fig. 9 (left) shows the obstacle detection and notification flow. As mentioned, obstacles are recorded during cleaning runs and integrated into the persistent file at their completion. However,

notifications for obstacles already stored in the persistent file can occur during either a cleaning or a search run. Since timely user notification is important, a search run is also automatically triggered after a cleaning run if new obstacles have been added to the persistent file.

The obstacle Verbal Interaction (VIA) starts (Fig. 9 right), if a not-lying person is detected and the persistent obstacle file contains obstacles pending user notification. The Alexa skill is then triggered and supplied with obstacle information. The user is prompted that an obstacle has been found, including its type, and asked whether they want to be guided to it. If so, the robot will drive to the obstacle. Upon arrival, the user is asked to confirm or deny the obstacle as a fall hazard, as the detected object could also be placed there on purpose or be misclassified. In case of a confirmed fall hazard, it is suggested that it

be removed, otherwise the obstacle will be ignored in the future.

Fig. 8 (b) shows the robot's image of the obstacle type "cable" after guiding the user to it, and Fig. 8(c) depicts the user removing it. These images where

taken during the field study. If the user instead chose not to be guided to the obstacle in the first place, the user can command the robot to ignore the obstacle in general, or to be reminded of it in the future.

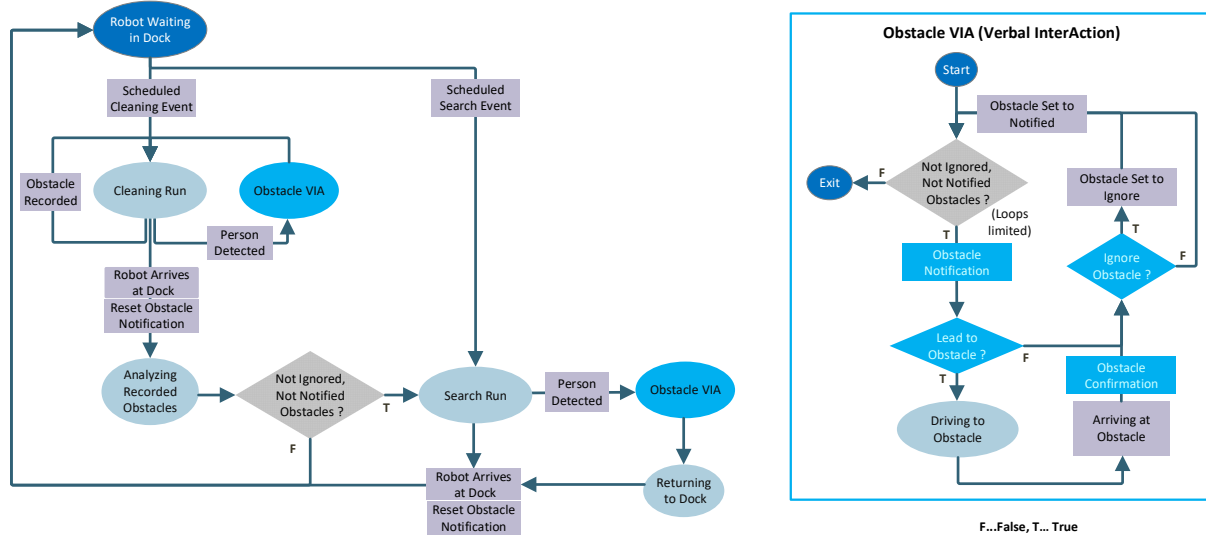


Fig. 9. Obstacle detection and notification flow charts.

This process repeats for a maximum of three times to avoid annoying the user, and only takes place once per run, meaning once per detection of not-lying person per run. The user can also request guidance to all currently known, non-ignored obstacles. Ignoring an obstacle suppresses detections within a 1 m radius, since the robot cannot indicate the object's exact position and the user will typically inspect the surrounding area when alerted to a potential fall hazard.

#### 4. Evaluation

After the initial user-centered development, the system was evaluated by six individuals (5 female, 1 male) in their homes for one month each. They were living alone and independently and were in the age range of 60 to 80.

After explaining the system, a schedule and a set of search points were defined. Participants kept a daily log, grading their experience. Entrance and exit interviews were conducted, as well as lying person simulations to evaluate the system's readiness. In these simulations, participants were guided through the process of the fall detection to learn the robot's behavior and share feedback. With over 100k recorded images, user feedback, and automated system and voice assistant logs, improvements were implemented after each test round. Dialog and robot settings were adjusted based on user feedback, and the CNN was further fine-tuned.

The combination of image analysis and audio confirmation yielded only a single false alarm during the six-month test period. The respective misclassification can be seen in Fig. 5(c). The participant was not present during the interaction attempt. During lying simulations and many additional lab tests, the multi-view lying detection worked reliably. Many detections had to be confirmed with a second image during the multi-view analysis, which showed that the CNN model could be improved further.

Although initial concerns suggested the robot itself might pose a fall hazard, its bright color, noise during vacuuming, and verbal notifications of its actions prevented this. While this could be improved with warning sounds or blinking lights, this might reduce user acceptance.

A concluding user survey showed high scores for trust in lying detection ability (4.8/5), trust in privacy protection (4.9/5) and usefulness in their lives (5/5). Although the number of study participants is low, a total of 184 days of continuous on-site testing, including 208 search and cleaning runs, supports the system's robustness under real-world conditions.

This extensive testing duration in combination with the user-centered development also substantiates the qualitative user acceptance results.

#### 5. Outlook

Future advances in robot vacuum hardware may allow data processing to be performed directly on the

robot, enabling the integration of the entire system via software updates only.

To reduce response time, the system could be triggered by a privacy-preserving fall detection sensor, combining event detection speed with the reliability of lying-state recognition and the user acceptance of a robot vacuum.

Large Language Models (LLM) could enhance natural communication. Since their tendency to hallucinate still prohibits them from making critical decisions [28], a hybrid approach with a structured behavioral model appears promising at present.

The robot's integrated object detection identifies a wide range of objects, but only some of these are relevant for detecting fall hazards. Even though we already filter object classes accordingly (e.g., detected pets, bathroom scales, etc.), further contextual analysis of the obstacle would improve overall accuracy. For instance, a detected shoe may represent a hazard when located in the middle of a hallway, but not when positioned on a shoe mat. A carpet by itself is not necessarily a fall hazard, but if its edges are raised or curled it might be. Such contextual distinctions are irrelevant for a robot vacuum's cleaning capability but have to be considered for improving fall hazard detection. Vision Language Models (VLM) can take such context into account and could be used in the future [29].

## Acknowledgments

Smart Companion is a multidisciplinary project conducted as a collaboration between the University of Applied Sciences St. Pölten, Robert Bosch AG, the Arbeiter-Samariter-Bund Österreich Gruppe Linz (ASB), and the Academy for Ageing Research at the "Haus der Barmherzigkeit".

It was funded by the Austrian Research Promotion Agency (FFG), project number: 887562. The project was approved by the Ethics Commission of the Medical Faculty of Johannes Kepler University Linz (EK Nr: 1022/2024).

This article is an extended version of our previously published paper [30], presented at the 11<sup>th</sup> International Conference on Sensors and Electronic Instrumentation Advances (SEIA' 2025).

## References

- [1]. T. Mujirishvili, C. Maidhof, F. Florez-Revue, M. Ziefle, et al., Acceptance and privacy perceptions toward video-based active and assisted living technologies: scoping review, *Journal of Medical Internet Research*, Vol. 25, 2023, e45297.
- [2]. M. A. Heard, S. Suresh, M. Leonard, J. B. Burns, Poor outcomes of patients from delayed care after ground level falls, *The American Surgeon*, Vol. 89, Issue 7, 2023, pp. 3153-3156.
- [3]. N. T. Newaz, E. Hanada, The methods of fall detection: a literature review, *Sensors*, Vol. 23, Issue 11, 2023, 5212.
- [4]. B. Heinbüchner, M. Hautzinger, C. Becker, K. Pfeiffer, Satisfaction and use of personal emergency response systems, *Zeitschrift für Gerontologie und Geriatrie*, Vol. 43, Issue 4, 2010, pp. 219-223.
- [5]. C. Naschenweng, Analyzing fall patterns for 'smart companion' behavior adaption: insights through an exploratory study based on extensive Austrian elderly care data, FH St. Pölten University of Applied Sciences, Phaidra Repository, 2024, <https://phaidra.fhstp.ac.at/detail/o:5980>
- [6]. N. Thakur, C. Y. Han, A study of fall detection in assisted living: identifying and improving the optimal machine learning method, *Journal of Sensor and Actuator Networks*, Vol. 10, Issue 3, 2021, 39.
- [7]. J. Kubitzka, I. T. Schneider, B. Reuschenbach, Concept of the term long lie: a scoping review, *European Review of Aging and Physical Activity*, Vol. 20, Issue 1, 2023, 16.
- [8]. L. Abou, et al., Sensitivity of Apple Watch fall detection feature among wheelchair users, *Assistive Technology*, Vol. 34, Issue 5, 2022, pp. 619-625.
- [9]. A. Jakl, et al., Smart companion: active AI assistant to support autonomous living, FH St. Pölten University of Applied Sciences, Phaidra Repository. <https://phaidra.fhstp.ac.at/detail/o:7507>
- [10]. U.S. Consumer Product Safety Commission, Senior injuries and deaths associated with consumer products: 2024 report, 2024, <https://www.cpsc.gov/content/Senior-Injuries-and-Deaths-Associated-with-Consumer-Products-2024-Report>
- [11]. N. Shishov, V. Komisar, D. S. Marigold, J.-S. Blouin, S. N. Robinovitch, Interactions during falls with environmental objects: evidence from real-life falls in long-term care captured on video, *BMC Geriatrics*, Vol. 24, Issue 1, 2024, 726.
- [12]. Z. Song, et al., Hazards in daily life? Enabling robots to proactively detect and resolve anomalies, *arXiv preprint*, 2024, arXiv:2411.00781.
- [13]. S. P. Konara Mudiyanse, C. T. Yao, S. D. Maitheepala, B. O. Lee, Emerging digital technologies used for fall detection in older adults in aged care: a scoping review, *Journal of the American Medical Directors Association*, Vol. 26, Issue 1, 2025, 105330.
- [14]. F. X. Gaya-Morey, C. Manresa-Yee, J. M. Buades-Rubio, Deep learning for computer vision based activity recognition and fall detection of the elderly: a systematic review, *Applied Intelligence*, Vol. 54, Issue 19, 2024, pp. 8982-9007.
- [15]. Y. Cho, et al., WiFi based human fall and activity recognition using transformer based encoder decoder and graph neural networks, *arXiv preprint*, 2025, arXiv:2504.16655.
- [16]. J. Kantorovitch, J. Väre, V. Pehkonen, A. Laikari, H. Seppälä, An assistive household robot – doing more than just cleaning, *Journal of Assistive Technologies*, Vol. 8, Issue 2, 2014, pp. 64-76.
- [17]. S. Maldonado-Bascón, C. Iglesias-Iglesias, P. Martín-Martín, S. Lafuente-Arroyo, Fallen people detection capabilities using assistive robot, *Electronics*, Vol. 8, Issue 9, 2019, 915.
- [18]. M. Bouazizi, A. Lorite Mora, T. Ohtsuki, A 2D-LIDAR-equipped unmanned robot-based approach for indoor human activity detection, *Sensors*, Vol. 23, Issue 5, 2023, 2534.
- [19]. E. Tîrziu, A. -M. Vasilevschi, A. Alexandru, E. Tudora, Real-time fall monitoring for seniors via

- YOLO and voice interaction, *Future Internet*, Vol. 17, Issue 8, 2025, 324.
- [20]. L. Killian, et al., Fall prevention and detection in smart homes using monocular cameras and an interactive social robot, in *Proceedings of the Conference on Information Technology for Social Good (GoodIT'21)*, 2021, pp. 7-12.
- [21]. R. S. Sadasivam, et al., Robot-assisted home hazard assessment for fall prevention: a feasibility study, *Journal of Telemedicine and Telecare*, Vol. 20, Issue 1, 2014, pp. 3-10.
- [22]. J. Moore, et al., Enhancing fall risk assessment: instrumenting vision with deep learning during walks, *Journal of NeuroEngineering and Rehabilitation*, Vol. 21, Issue 1, 2024, 106.
- [23]. S. Ishii, A. Chikhalikar, A. A. Ravankar, J. V. S. Lucas, Y. Hirata, Context-aware risk estimation in home environments: a probabilistic framework for service robots, *arXiv preprint*, 2025, arXiv:2508.19788.
- [24]. M. Yaseen, What is YOLOv8: an in-depth exploration of the internal features of the next-generation object detector, *arXiv preprint*, 2024, arXiv:2408.15857.
- [25]. Ultralytics, Loss Functions, <https://docs.ultralytics.com/reference/utils/loss/>
- [26]. A. Kumar, et al., Just ASK: building an architecture for extensible self-service spoken language understanding, *arXiv preprint*, 2018, arXiv:1711.00549.
- [27]. D. Feilacher, S. Lehner, M. Heller, B. Girsule, A. Jaki, Robust dialog design for emergency situations with AI-based voice assistants, in *Proceedings of the International Conference on Creative Media Technologies (ICConCMT) 2023 and 2024*, 2024, pp. 65-75.
- [28]. Z. Bai, et al., Hallucination of multimodal large language models: a survey, *arXiv preprint*, 2024, arXiv:2404.18930.
- [29]. Z. Li, X. Wu, H. Du, F. Liu, et al., A survey of state of the art large vision language models: alignment, benchmark, evaluations and challenges, *arXiv preprint*, 2025, arXiv:2501.02189.
- [30]. D. Feilacher, B. Girsule, S. Lehner, M. Heller, et al., Smart Companion: an audio-visual lying person detection system integrated in a robot vacuum cleaner, in *Proceedings of the 11<sup>th</sup> International Conference on Sensors and Electronic Instrumentation Advances (SEIA '25)*, Ponta Delgada, Portugal, 24-26 September 2025, pp. 57-63.



Published by International Frequency Sensor Association (IFSA) Publishing, S. L., 2025 (<http://www.sensorsportal.com>).



## Online Experimentation: Emerging Technologies and IoT

Maria Teresa Restivo, Alberto Cardoso, António Mendes Lopes (Editors)

*Online Experimentation: Emerging Technologies and IoT* describes online experimentation, using fundamentally emergent technologies to build the resources and considering the context of IoT.

In this context, each online experimentation (OE) resource can be viewed as a "thing" in IoT, uniquely identifiable through its embedded computing system, and considered as an object to be sensed and controlled or remotely operated across the existing network infrastructure, allowing a more effective integration between the experiments and computer-based systems.

The various examples of OE can involve experiments of different type (remote, virtual or hybrid) but all are IoT devices connected to the Internet, sending information about the experiments (e.g. information sensed by connected sensors or cameras) over a network, to other devices or servers, or allowing remote actuation upon physical instruments or their virtual representations.

The contributions of this book show the effectiveness of the use of emergent technologies to develop and build a wide range of experiments and to make them available online, integrating the universe of the IoT, spreading its application in different academic and training contexts, offering an opportunity to break barriers and overcome differences in development all over the world.

*Online Experimentation: Emerging Technologies and IoT* is suitable for all who is involved in the development design and building of the domain of remote experiments.

Hardcover: ISBN 978-84-608-5977-2  
e-Book: ISBN 978-84-608-6128-7

Order: [http://www.sensorsportal.com/HTML/BOOKSTORE/Online\\_Experimentation.htm](http://www.sensorsportal.com/HTML/BOOKSTORE/Online_Experimentation.htm)