Imaginex International Journal of Interdisciplinary Research and Reviews (IIJIRR)
Volume 1, Issue 3, November – December 2026
ISSN: 3140-0094 (Online)
www.iijirr.com

ElderGuard AI: A Privacy-Aware Multimodal Fusion and Edge Intelligence Framework, with a Validation Protocol, for Elderly Home Safety

Senthil Jayapal[1,*], Baraah Amur Mubark Al Rawahi[1], Alaa Amur Salim Al Aghbari[1], Shifa Mohammed Abdullah Salim Al Rawahi[1], Al Adhra Nasser Humaid Mohammed Al Harthi[1], Sabha Said Ali Khamis Al Hudaiwi[1], Ahed Amur Khalfan Khamis Al Difaii[1]

[1]Department of Computing and Information Sciences, College of Computing and Information Sciences, University of Technology and Applied Sciences – Ibra, Sultanate of Oman
[*]Corresponding author. E-mail: senthil.jayapal@utas.edu.om

Abstract

Falls are a leading cause of injury and death among older adults, and many happen at home when nobody is nearby. Small mobile robots could help by checking on a person and calling for help. Yet most published prototypes are judged by how accurately they label short video clips. That figure does not tell a caregiver how often the robot will raise a false alarm, or whether a real alert will reach them when the home Wi-Fi drops. This paper presents ElderGuard AI, a design for a low-cost indoor robot that addresses these practical questions, together with a protocol for testing it. An ESP32 controller handles obstacle avoidance independently of every other task. A separate edge module estimates body pose from camera frames, and a fusion rule weighs each source of evidence by its quality and availability. Events are confirmed over time. When the evidence is too weak, the robot reports UNKNOWN rather than "no fall". Video never leaves the device: only short, signed event records are sent, and each one is tracked until a caregiver acknowledges it. A validation plan with criteria fixed in advance covers public fall datasets, the robot's own camera, network outages and collisions under load. Only the first prototype, which avoids obstacles and sends notifications, has been built so far. Results will be reported once these experiments are complete.

Keywords: Fall detection, Multimodal sensor fusion, Edge intelligence, Privacy by design, Assistive robotics, Ambient assisted living, Uncertainty-aware alerting

1. Introduction

Most older adults would rather stay in their own homes than move into care, and many of them live alone. For this group, a fall is dangerous twice over: first because of the injury itself, and then because hours may pass before anyone notices. The World Health Organization estimates that falls caused about 842,000 deaths worldwide in 2023, roughly 71 % of them among people aged 60 or over [21]. Shortening the time between a fall and a response is therefore a worthwhile engineering goal, provided it does not turn the home into a place of constant surveillance [9], [22].

Each existing option has a familiar weakness. A push-button alarm works only if the person can reach it and press it. Wearable sensors do well in laboratory trials but are often left on the bedside table, and in daily use they raise false alarms. Field studies report anything from about one to several dozen false alarms a day [2], [11]. Fixed cameras see more, but they leave blind spots, and many older adults are reluctant to have cameras inside their homes [10], [17].

A small mobile robot sits between these options. It can take its camera to where the person is and patrol rooms that a fixed camera would miss. Robots were combined with home and body sensor networks for fall detection and rescue as early as 2018 [4]. Several robot fall detectors have since been built [1], [13], [18], and a nursing-home trial is in preparation [14]. Their evaluations, however, tend to stop at the classifier. They report accuracy on short clips, usually with a handful of volunteers, and leave three questions open. How many false alarms does the robot raise in an hour? What does it do when it cannot see the person clearly? And does its alert actually arrive when the network fails?

ElderGuard AI started as a diploma project: an ESP32-based robot that avoids obstacles and sends caregiver notifications. This study extends its design into a system whose whole chain, from camera frame to caregiver acknowledgement, can be measured. Its contributions are:

(i) an edge–IoT architecture that keeps collision avoidance on the ESP32, apart from inference and networking, and never stores or sends raw video, designed in three prototype stages (Section 3);

(ii) a decision model that weighs each source of evidence by its quality and availability, confirms events over time and abstains when it cannot see (Section 4);

(iii) a persistent alert queue that logs every notification until it is acknowledged, together with a validation plan whose acceptance criteria are fixed in advance (Section 5); and

(iv) a set of measures — false alarms per hour, calibration, abstention coverage, end-to-end latency and delivery ratio — that make the trade-offs between sensitivity, alarm burden, privacy and reliability visible (Sections 5–7).

Two terms need a narrow reading. *Autonomy* here means safe, collision-free patrol with automatic alerting; it does not mean whole-home navigation or medical judgement. *Multimodal fusion* means combining camera evidence with any non-visual source whose link to falls has been validated; distance readings are used for navigation only. Figure 1 outlines the research workflow.

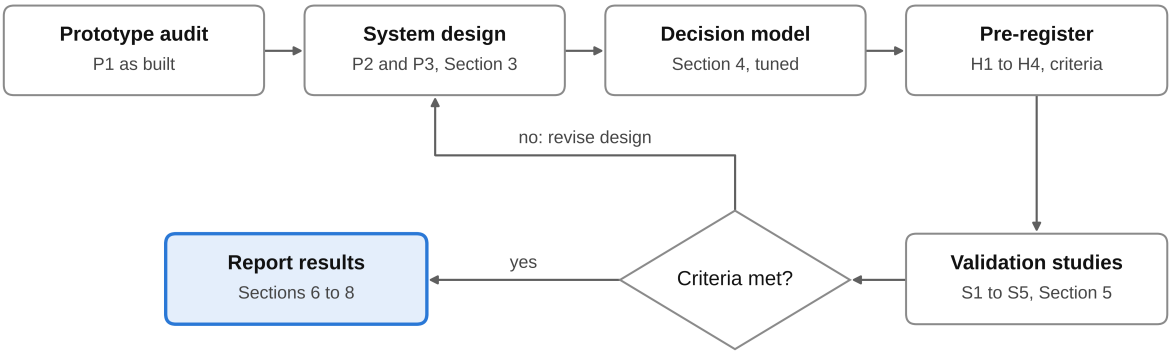

Figure 1. Research workflow. If the acceptance criteria are not met, the work returns to the design stage before any result is reported.

2. Related work

Recent reviews cover fall detection with wearable, ambient, vision and hybrid sensors, and note a gradual shift toward predicting risk before a fall occurs [9], [22]. In a review of 80 studies, systems that combined several sensor types reported the best accuracy and sensitivity, clearly ahead of wearables alone (Figure 2) [8]. The same review is a useful warning, though. Most accuracy figures came from datasets the authors had collected themselves, and fewer than 15 % of studies said how long their models took to run [8]. Combining sensors is not new either: Kwolek and Kępski paired a depth camera with a wireless accelerometer on an embedded board and released the UR Fall dataset [12], and UP-Fall offers synchronised wearable, ambient and video recordings of 17 young adults [15].

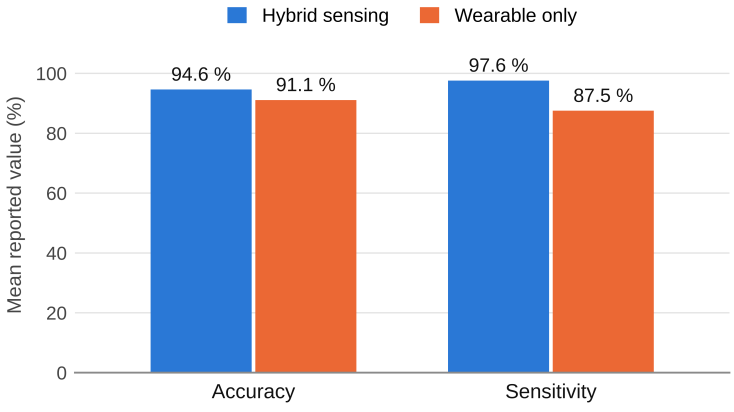

Figure 2. Mean accuracy and sensitivity reported for hybrid and wearable-only fall-detection systems in a systematic review of 80 studies [8]. Hybrid systems report about 10 percentage points higher sensitivity than wearables alone. The values are averages of published results, not a head-to-head comparison.

Results obtained in the laboratory rarely survive contact with daily life. When thirteen published algorithms were run on recordings of real falls, they produced between 3 and 85 false alarms per day [2]. A study of 16 older adults over 15,500 hours detected 12 of 15 falls at 0.049 false alarms per hour [11]. Vision models struggle in a similar way: accuracy drops sharply when models trained on staged recordings meet genuine accident footage [19]. Evaluation choices matter as well, since splitting data by recording rather than by person inflates reported accuracy [3], [20].

Robots bring their own strengths and difficulties. Omobot, a low-cost open-source platform, patrols a home and e-mails responders when it detects a fall. With its camera only 0.15 m above the floor, its authors had to re-project images to a human viewpoint, which improved detection by 6–12 % [1]. EM-Fall places mmWave radar on a humanoid robot and reports 94.7–100 % accuracy with four volunteers, although its false-positive rate reached 10.7 % under occlusion [13]. Other work confirms wearable-triggered alerts with a robot-mounted camera [18], and a nursing-home trial has set a target of no more than 10 % false-positive alerts [14]. Generative AI has also been used to coordinate IoT devices in smart homes [16], the setting in which a safety robot such as ElderGuard would operate. On the hardware side, the ESP32-S3 can now run small person-detection [7] and pose-estimation models [6]. Messaging research has added retransmission after network loss [5], but without any notion of a safety event that someone must acknowledge. As Table 1 shows, none of the robotic systems reviewed reports false alarms per hour, abstains when the evidence is poor, or checks that its alerts arrive.

Table 1. Comparison of related work (✓ addressed; – not reported; for this work, "Designed" and "Planned" mark features that have not yet been measured).

Study	Principal result	Per-hour false alarms	Abstention	Alert delivery tested	On-device privacy
Ahamad et al. [1]	6–12 % detection gain	–	–	–	–
Lu et al. [13]	94.7–100 % accuracy, 4 participants	–	–	–	✓ (radar, no images)
Rahimi Azghadi et al. [18]	96.3 % vision accuracy	–	–	–	✓ (federated)
Malinsky et al. [14]	Protocol only	– (target ≤ 10 % of alerts)	–	Planned	✓ (infrared)
Kangas et al. [11]	80 % sensitivity, 15,500 h	✓	–	–	✓ (no images)
This work	**Framework and test protocol**	**Planned**	**Designed**	**Planned**	**Designed**

3. System design and prototypes

3.1. Architecture

The robot is organised as three paths that share only a small state variable and a persistent event queue (Figure 3). The *safety path* runs on the ESP32 alone: range sensors feed a speed law, which drives the motors at the highest task priority. The *perception path* turns camera frames into pose estimates and updates the decision model described in Section 4. The *communication path* takes confirmed events from the queue and delivers them to the caregiver's phone, which sends back an acknowledgement. Because the paths are separate, a slow model or a dropped Wi-Fi link cannot delay a stop command.

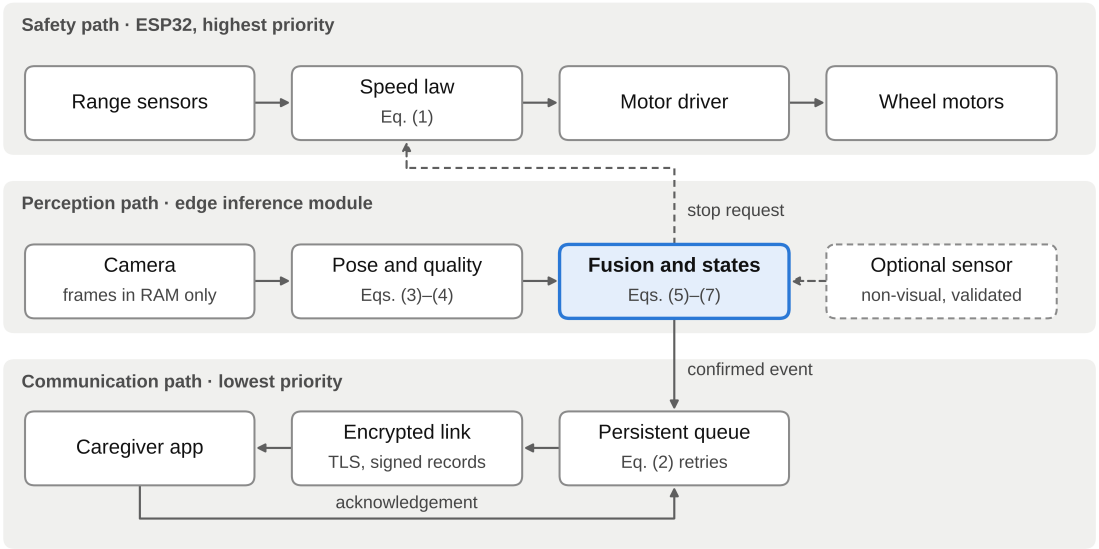

Figure 3. System architecture. Collision avoidance never waits for inference or the network. The perception path can only *request* a stop; it never overrides the collision-avoidance loop. Distance readings are used for navigation only and do not enter the fusion. Dashed lines are optional or advisory.

3.2. Hardware design

The platform is a two-wheel differential-drive base carrying an ESP32 controller, a camera, forward-facing range sensors, a dual H-bridge motor driver and a rechargeable battery. The choice of ESP32 variant matters. The original chip has no practical route to pose estimation, whereas Espressif's ESP-DL runtime for the ESP32-S3 offers a quantised pose model with a 160 × 160-pixel input [6]. The board that performs inference is therefore chosen by measurement, comparing the ESP32-S3 on its own with an ESP32 paired with a small single-board computer.

Figure 4 shows the electrical layout. One battery supplies a regulated logic rail for the ESP32 and a separate rail for the motor driver, so that motor current surges cannot reset the controller. A physical switch cuts power to the camera without affecting obstacle avoidance, and an LED shows when the camera is on.

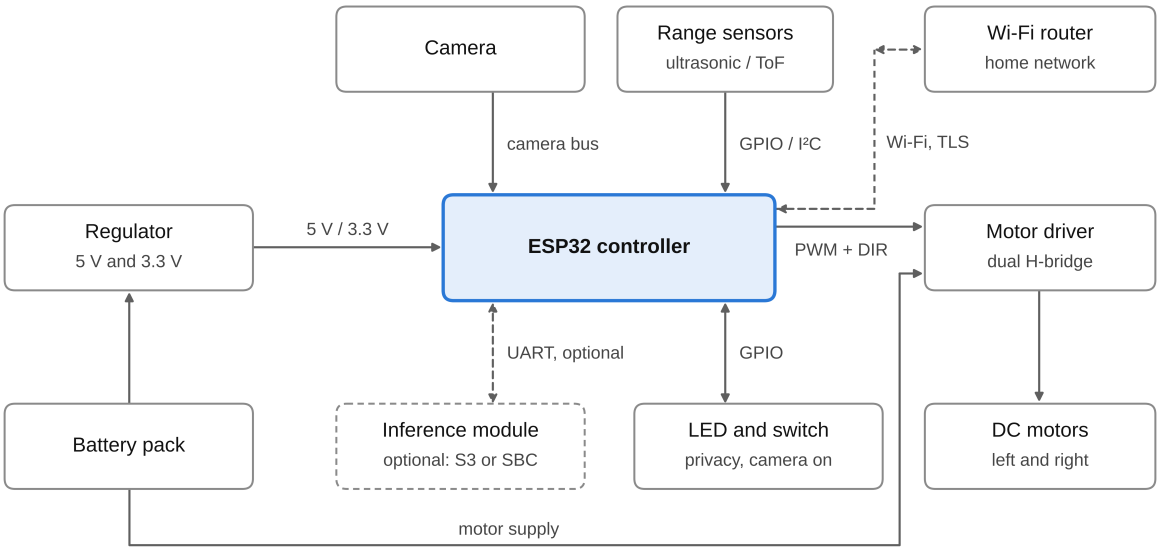

Figure 4. Electrical block diagram. The ESP32 alone drives the motors; inference sits on an optional board. Arrows show power or signal flow; dashed links are wireless or optional.

3.3. Navigation and alert logic

The motor-safety task runs at a fixed period. If d_t is the smallest valid range reading ahead of the robot, the forward speed is

$$v_t = v_{\max}\,\mathrm{clip}\left(\frac{d_t - d_{\mathrm{stop}}}{d_{\mathrm{slow}} - d_{\mathrm{stop}}},\ 0,\ 1\right) \tag{1}$$

where $v_{\max}$ is the top speed, d_{slow} the distance at which slowing begins, and d_{stop} the stopping distance, set larger than the braking distance measured at $v_{\max}$. A missing, implausible or stale reading forces an immediate stop, as does a watchdog reset or a lost heartbeat from the inference module. The perception path may ask the robot to stop, for instance after a confirmed fall, but it can never override this loop. The robot does not approach a fallen person, because that behaviour would need its own hazard analysis.

Every event is written to non-volatile storage before the first attempt to send it, so a reset cannot erase it. It then passes through the states DETECTED, QUEUED, TRANSMITTING, PROVIDER_ACCEPTED, DELIVERED and ACKNOWLEDGED, or ends as EXPIRED or FAILED, and each transition is time-stamped. Failed attempts are retried after a delay that grows exponentially up to a cap:

$$\delta_k = \min\left(\delta_{\max},\ \delta_0\, 2^k\right)(1 + u_k), \qquad u_k \sim \mathcal{U}(0, 0.2) \tag{2}$$

Here k is the retry count, δ_0 the base delay, $\delta_{\max}$ the cap, and u_k a small random jitter that stops several devices from retrying in step. The phone app discards duplicates by event identifier. An alert that is not acknowledged within A seconds (Table 4) is passed to the next contact. A provider's "accepted" status is never taken as proof that a caregiver has seen the alert. Figure 5 shows the perception loop: each frame is scored and then discarded, and only a confirmed event reaches storage.

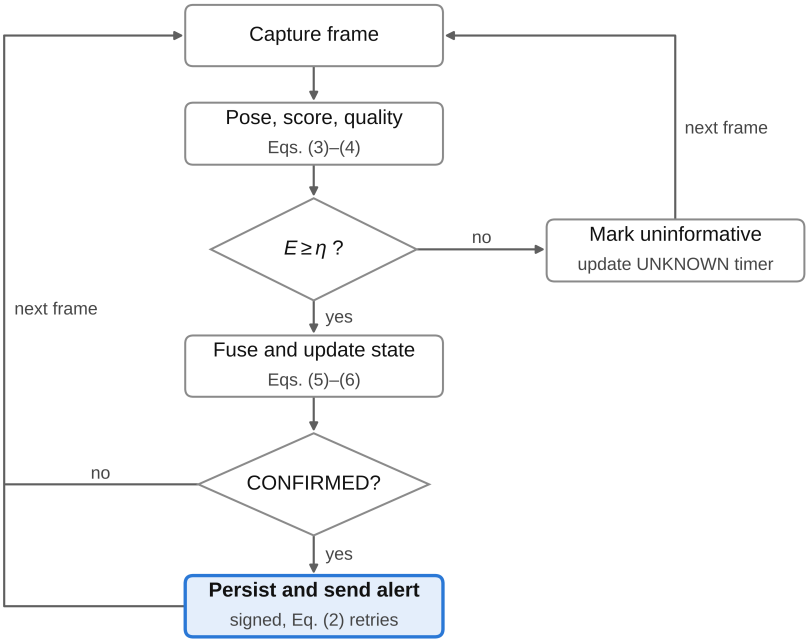

Figure 5. Firmware flowchart of the perception loop. Only a confirmed event leaves the loop as a stored alert. The motor-safety loop runs independently at a higher priority and is not shown.

3.4. Privacy and security design

Privacy is handled as a set of controls, each paired with a check that will be reported with the results (Table 2). The design considers four threats: eavesdropping on the network, an intruder on the local network, a curious cloud provider and theft of the robot. It also guards against two attacks on safety, namely suppressing a real alert and forging false ones. By default, frames exist only in working memory and notifications contain no image. A caregiver who wants visual confirmation must opt in. They then receive a blurred crop of the person, which is deleted once the alert is acknowledged.

Table 2. Privacy and security controls and how each is verified.

Control	Threat addressed	Verification
Frames processed in RAM only; no video stored or sent by default	Eavesdropping, theft, provider	Flash dump after a session contains no image data; audit of transmitted fields
TLS transport (MQTT over TLS or HTTPS) with certificate pinning	Eavesdropping, provider	Packet capture shows no plaintext; rogue certificate rejected
Per-device credentials; local interfaces disabled or authenticated	Local attacker	Port scan; unauthenticated access attempt
Event records signed (HMAC over identifier, time and state)	Forged alerts	Unsigned and replayed events rejected by the app
Heartbeat with caregiver-side “unreachable” warning	Suppressed alerts	Unplugging the robot or blocking its traffic raises a warning within H s
Event logs encrypted; retained for N days	Theft	Inspection of storage after the retention period

Residents can exclude rooms or times from patrols, such as the bathroom, and monitoring always requires their informed consent.

3.5. Prototype configurations

The robot is developed in three stages, each building on the one before (Table 3).

Table 3. Prototype configurations.

Prototype	Adds	Purpose	Status
P1	ESP32, camera, distance sensors, motor driver; obstacle avoidance; caregiver notification	Functional base	Demonstrated; not yet measured
P2	Edge inference module; decision model (Section 4); persistent alert queue	Detection and reliable alerting	To build
P3	Encrypted, signed transport; privacy switch; state and resource logging; optional non-visual sensor	Version used for validation	To build

4. Conceptual model of event detection and fusion

The decision model turns noisy and sometimes missing observations into one of four states: NORMAL, CANDIDATE, CONFIRMED or UNKNOWN (Figure 6). It works in four steps (scoring, quality assessment, fusion and temporal confirmation), and each step can be switched off separately in the ablation study.

4.1. Visual event score and observation quality

For each frame, the inference module estimates 17 body keypoints. From them it derives a few features: the torso angle, the width-to-height ratio of the bounding box, the hip height and its vertical speed over half a second, and the time spent lying down. A small classifier turns these features into a logit z_t, which is calibrated by temperature scaling on validation participants only:

$$p_t^v = \sigma\left(\frac{z_t}{T_v}\right), \qquad \sigma(u) = \frac{1}{1+e^{-u}} \tag{3}$$

The classifier is either gradient-boosted trees or a one-dimensional temporal convolution, whichever validates better at an acceptable cost on the device. Because the camera sits low, frames can optionally be re-projected to a human-height viewpoint before pose estimation, as in [1]. Each frame also receives a quality score. The score falls when the keypoints are uncertain, the image is badly exposed or blurred, or the person fills too little of the picture:

$$q_t^v = \bar{c}_t \, g_{\text{exp}}(t) \, g_{\text{blur}}(t) \, g_{\text{size}}(t) \tag{4}$$

Here $\bar{c}_t$ is the mean confidence of the torso and hip keypoints. The three g terms, each between 0 and 1, penalise poor exposure, motion blur (measured by the variance of the Laplacian) and a small person region. With no person in view, $q_t^v = 0$. To limit blur, the robot pauses briefly whenever a person comes into view.

4.2. Fusion weighted by quality and availability

The default system uses the camera alone. If a validated non-visual sensor c is added, such as a wearable accelerometer or a mmWave unit, the set of modalities becomes $\mathcal{M} = \{v, c\}$. For each modality, $a_t^m \in \{0, 1\}$ records whether a fresh reading has arrived and w_m is a learned reliability weight. The weight actually applied is $\omega_t^m = a_t^m \, q_t^m \, w_m$. The fused probability moves away from a prior π only as far as the available evidence allows:

$$s_t = \sigma\left(\operatorname{logit}\pi + \sum_{m\in\mathcal{M}} \omega_t^m \left[\operatorname{logit} p_t^m - \operatorname{logit}\pi\right]\right) \tag{5}$$

If every source is missing or of no use, s_t falls back to the prior rather than to a confident "no fall". The total weight $E_t = \Sigma\, \omega_t^m$ decides whether an observation counts. Below a threshold η, the observation is ignored. Range readings are deliberately left out of $\mathcal{M}$, because nothing links obstacle distance to falling.

4.3. Temporal confirmation and abstention

Informative observations are smoothed with an exponentially weighted average:

$$\bar{s}_t = \alpha \, s_t + (1-\alpha)\, \bar{s}_{t-1}, \qquad 0 < \alpha \le 1 \tag{6}$$

Two thresholds, $\tau_L < \tau_H$, give the state machine hysteresis. The state becomes CANDIDATE when $\bar{s}_t$ reaches τ_H. It becomes CONFIRMED once $\bar{s}_t$ stays there for L informative observations, or once the person has been lying down for D_{lie} seconds. It returns to NORMAL only after $\bar{s}_t$ has stayed below τ_L for D_{clear} seconds. If no informative observation arrives for U seconds while a person was last in view, the state becomes UNKNOWN. After $U_{\max}$ seconds in that state, the robot asks for a low-priority check-in rather than raising a fall alert. A single noisy frame therefore cannot trigger or cancel an alert, and the robot never claims "no fall" when it simply cannot see.

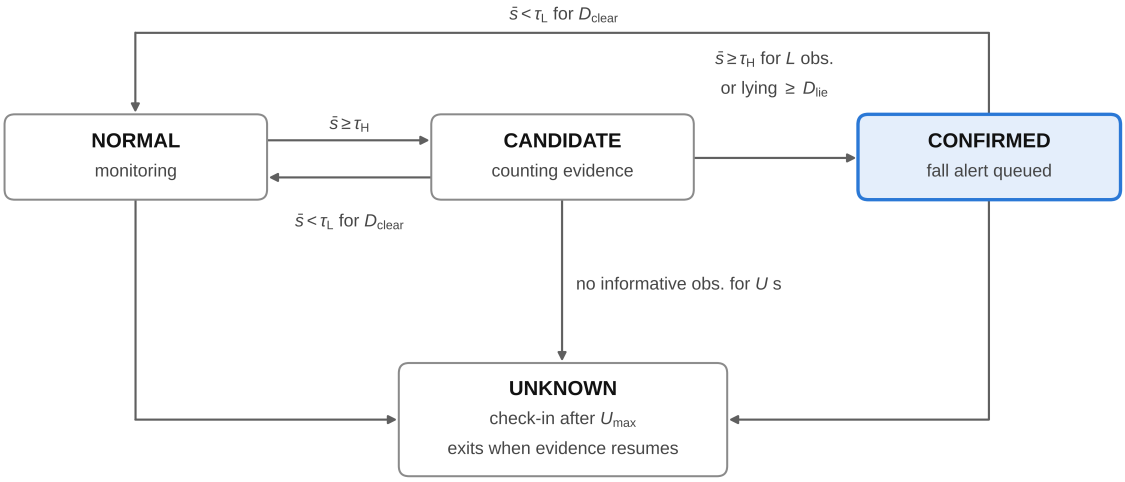

Figure 6. Decision state machine. The robot reports UNKNOWN instead of "no fall" when it cannot see. Any state moves to UNKNOWN when informative observations stop while a person was last in view.

4.4. Parameter selection under an alarm budget

The parameters are not tuned for accuracy. They are chosen to catch as many events as possible while keeping false alarms within a budget B per monitored hour, which caregivers agree in advance:

$$\Theta^* = \arg\max_{\Theta} \text{Recall}(\Theta) \quad \text{subject to} \quad \text{FA/h}(\Theta) \leq B \tag{7}$$

Tuning uses training and validation participants only. Table 4 lists the parameters and where their values come from.

Table 4. Parameters of the decision model.

Parameter	Symbol	Planning value	Source
Calibration temperature	T_v	Fitted	Validation set
Reliability weights	w_m	Fitted	Validation set
Prior event probability	π	0.01	Assumed
Abstention threshold	η	To tune	Eq. (7)
Smoothing factor	α	To tune	Eq. (7)
Hysteresis thresholds	τ_H, τ_L	To tune ($\tau_L < \tau_H$)	Eq. (7)
Confirming observations	L	To tune	Eq. (7)
Lying, clearing, unknown durations	D_{lie}, D_{clear}, U, U_{max}	To tune (s)	Eq. (7)
Alarm budget	B	Agreed with caregivers	Pre-registered
Acknowledgement timeout	A	To be set (s)	Design
Heartbeat warning time	H	To be set (s)	Design

5. Validation methodology

Four hypotheses are tested. Each has an acceptance criterion that will be registered, with a timestamp, in a public registry before any test data are examined.

H1. Temporal confirmation with quality gating (D3) produces fewer false alarms per hour than a fixed-threshold classifier (D1), while losing no more than a pre-set margin ΔR of recall.

H2. Under occlusion and poor light, D3 misses fewer falls than D1, and most of the decisions it withholds are labelled UNKNOWN.

H3. Over at least 30 injected network outages, the persistent queue (C1) delivers more alerts than send-once transmission (C0), with no duplicates.

H4. (a) The loss in F1 between public fixed-camera data and the robot camera is measured with a confidence interval. (b) Collisions are no more frequent under full computational load than at idle.

5.1. Datasets and data handling

UP-Fall [15] provides the training and validation data. Its 17 participants are split by person, for example 11 for training, 3 for validation and 3 for testing, and the split is repeated over several permutations. UR Fall [12] serves only as an external test set. False alarms per hour are estimated from long stretches of ordinary activity: the daily-activity parts of both datasets, plus at least H_{ne} hours of scripted recordings from the robot's camera, with H_{ne} fixed in the registered protocol. Falls seen by the robot camera are performed by consenting adults onto crash mats and by a mannequin; no older adult is ever asked to fall. Nothing from a test participant is used for training, calibration or threshold selection.

5.2. Configurations compared

Table 5. Decision and communication configurations.

ID	Configuration	Purpose
D0	Rule on torso angle and box ratio	Classical baseline
D1	Frame classifier, fixed threshold	Learned baseline
D2	D1 + calibration, smoothing and hysteresis	Effect of temporal confirmation
D3	D2 + quality gating and UNKNOWN	Effect of abstention (full visual model)
D3+H	D3 + viewpoint homography	Effect of viewpoint correction
D4	D3 + non-visual modality (only if installed and validated)	Effect of fusion
C0 / C1	Send-once vs persistent queue with back-off	Alert reliability

For the ablation study, each component is also removed in turn from the full system.

5.3. Performance measures

A fall counts as detected if the system reaches CONFIRMED within a tolerance window W after the labelled start of the fall. Precision, recall, F1 and detection latency are reported per event. False alarms are counted per hour of non-event monitoring:

$$\mathrm{FA/h} = \frac{N_{\mathrm{FP}}}{T_{\text{non-event}}} \tag{8}$$

Calibration is measured by the expected calibration error over M confidence bins B_b, where n is the number of scored observations:

$$\mathrm{ECE} = \sum_{b=1}^{M} \frac{|B_b|}{n} \left|\mathrm{acc}(B_b) - \mathrm{conf}(B_b)\right| \tag{9}$$

Abstention is summarised by coverage, the share of time not spent in UNKNOWN. Alert reliability is measured by the delivery ratio, the number of duplicates, and the median and 95th-percentile time from detection to acknowledgement. Navigation is measured by collisions per 100 runs, minimum clearance and stopping latency. Resource use is measured by frame rate, memory, power draw and battery life.

5.4. Test plan

Table 6. Test plan (planning volumes; final sample sizes follow pilot variance estimates).

Study	Hypothesis	Factors and levels	Volume
S1 Offline detection	H1, H2	D0–D3 on UP-Fall (participant-disjoint) and UR Fall (external)	All sequences, repeated folds
S2 Degradation	H2	Lighting (3 levels), occlusion (0/30/60 %), distance (3 levels), robot moving vs stopped	≥ 10 trials per cell
S3 Robot camera	H4a	Simulated falls and mannequin; D3 vs D3+H	≥ 60 events, ≥ 6 adults
S4 Alert reliability	H3	≥ 100 injected events; ≥ 30 Wi-Fi outages of 10 s–10 min; broker restart; C0 vs C1	2 configurations
S5 Navigation under load	H4b	Corridor, corner, chair legs, glass, moving obstacle; idle vs inference vs inference + alert burst + outage	≥ 100 traversals per load

All studies run on prototype P3.

5.5. Statistical analysis

Confidence intervals (95 %) come from bootstrap resampling with at least 1,000 resamples, clustered by participant or by run, so that correlated frames are not treated as independent. Paired detection results are compared with McNemar's test, latencies with the Wilcoxon signed-rank test, and delivery ratios with Fisher's exact test. Effect sizes are reported throughout, and the Holm–Bonferroni method corrects for multiple comparisons.

5.6. Implementation and ethics

The firmware is written with ESP-IDF or the Arduino-ESP32 core. Models are trained in Python and exported to an 8-bit runtime, and alerts travel over MQTT with TLS. Code, configuration files, trained models and anonymised logs (without video) will be released on acceptance. No study with participants will begin before the institutional ethics committee has approved it.

6. Reporting plan

No measurements have been taken yet, so Tables 7 and 8 show the planned layout only; each cell will hold a value with its 95 % confidence interval. Recall-versus-false-alarm curves, latency breakdowns, the ablation and the privacy checks of Table 2 will be added as figures once data exist.

Table 7. Planned layout for event-level detection results (S1). TBM = to be measured.

Configuration	Recall (UP-Fall)	Precision	F1	FA/h	ECE	Coverage	F1 (UR Fall)
D0	TBM	TBM	TBM	TBM	–	100 %	TBM
D1	TBM	TBM	TBM	TBM	TBM	100 %	TBM
D2	TBM	TBM	TBM	TBM	TBM	100 %	TBM
D3	TBM	TBM	TBM	TBM	TBM	TBM	TBM

Table 8. Planned layout for alert reliability under injected network outages (S4). TBM = to be measured.

Measure	C0 (send-once)	C1 (persistent queue)
Delivery ratio	TBM	TBM
Duplicates	TBM	TBM
Median time to acknowledgement (s)	TBM	TBM
95th-percentile time to acknowledgement (s)	TBM	TBM
Events lost on reset	TBM	TBM

Any hypothesis that is not supported will be reported as clearly as those that are.

7. Discussion

The main tension in the design is between sensitivity and alarm burden. Each extra confirming observation removes some false alarms but delays a genuine alert by roughly one frame interval. Whether that delay is acceptable depends on how serious the event is. For this reason, results will be reported as a curve of recall against false alarms per hour, with latency alongside, rather than as a single best operating point. If D3 can meet the alarm budget only by giving up more recall than ΔR, H1 fails. That outcome would suggest that the camera needs a second, non-visual sensor, making the fusion term in Eq. (5) essential rather than optional.

An UNKNOWN rate that rises in poor light is not a failure; it is an honest report that the camera cannot see. For a caregiver, "the robot cannot see her" is something to act on, whereas a silent miss is not. In the same way, a correct detection lost during a router restart is, for the person on the floor, no different from no detection at all. Because the queue logs every step from detection to acknowledgement, ElderGuard can separate algorithmic misses from communication misses. None of the robots in Table 1 makes this distinction.

The low, moving camera is the largest technical risk. Perspective, motion blur and furniture all differ from the conditions in public datasets, and earlier work found clear accuracy losses at a camera height of 0.15 m [1]. Even fixed cameras lose accuracy when moving from staged to real footage [19]. Study S3 will show whether the stop-and-look behaviour and the optional re-projection recover enough of that loss. Privacy has a price too. Keeping video on the device removes the option of a quick visual check, and encryption and signing add load to a small processor; both costs will be measured.

The work has clear limitations. All results are still pending, and some hypotheses may fail. The participants are young adults and a mannequin in laboratory rooms, so nothing can be generalised to older people or real homes without a supervised field study. Staged falls are not injurious falls. Small samples with correlated frames can also give overconfident estimates, which is why clustered intervals and pre-registered endpoints are used. The results will hold only for the documented hardware, and the privacy checks are self-assessed. The system does not diagnose, predict or prevent falls; it aims only to let a caregiver know sooner.

8. Conclusions

ElderGuard AI treats a low-cost assistive robot as a safety system whose behaviour can be measured from camera frame to caregiver acknowledgement. Collision avoidance runs apart from inference and networking. A fusion model weighs evidence by its quality and availability and abstains when it cannot see. A persistent queue tracks every alert, while video stays on the device. A validation plan with criteria fixed in advance will test whether these choices cut false alarms without losing falls, keep alerts alive through outages and leave collision safety intact. The current prototype only avoids obstacles and sends notifications, so the planned measurements will decide the framework's value. The next steps are to add a validated non-visual sensor, run inference entirely on the ESP32-S3, set the alarm budget with older adults and caregivers, and, after ethics approval, run a supervised pilot in real homes.

Declarations

Conflict of interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding. This research received no external funding.

Data availability. No new data are reported in this paper. UP-Fall [15] and UR Fall [12] are public datasets. Code, configuration files, trained models and anonymised logs (without video) will be released as stated in Section 5.6.

Use of generative AI. During the preparation of this work the authors used an AI-assisted tool for language editing. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Ethics approval and consent. Not applicable at this stage: this paper reports no data from human participants. Studies with participants will begin only after approval by the institutional ethics committee (Section 5.6).

Author contributions. Senthil Jayapal, Baraah Amur Mubark Al Rawahi, Alaa Amur Salim Al Aghbari, Shifa Mohammed Abdullah Salim Al Rawahi, Al Adhra Nasser Humaid Mohammed Al Harthi, Sabha Said Ali Khamis Al Hudaiwi and Ahed Amur Khalfan Khamis Al Difaii contributed to this work. All authors read and approved the final manuscript.

References

[1] S. U. Ahamad, M. Ataei, V. Devabhaktuni, V. Dhiman, "Omobot: A low-cost mobile robot for autonomous search and fall detection," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, 2024, doi: 10.1109/AIM55361.2024.10637198.

[2] F. Bagalà, C. Becker, A. Cappello, L. Chiari, K. Aminian, J. M. Hausdorff, W. Zijlstra, J. Klenk, "Evaluation of accelerometer-based fall detection algorithms on real-world falls," *PLoS ONE*, vol. 7, no. 5, e37062, 2012, doi: 10.1371/journal.pone.0037062.

[3] A. Dehghani, T. Glatard, E. Shihab, "Subject cross validation in human activity recognition," arXiv:1904.02666, 2019.

[4] H. M. Do, M. Pham, W. Sheng, D. Yang, M. Liu, "RiSH: A robot-integrated smart home for elderly care," *Robot. Auton. Syst.*, vol. 101, pp. 74–92, 2018, doi: 10.1016/j.robot.2017.12.008.

[5] M. Domingues, J. N. Faria, D. Portugal, "Dimensioning payload size for fast retransmission of MQTT packets in the wake of network disconnections," *EURASIP J. Wireless Commun. Netw.*, vol. 2024, art. no. 2, 2024, doi: 10.1186/s13638-023-02327-3.

[6] Espressif Systems, "ESP-Vision: On-device vision AI," [Online]. Available: https://vision.espressif.com/en (accessed Nov. 23, 2026).

[7] Espressif Systems, "ESP32-S3 now supports Plumerai's people-detection model," Mar. 31, 2023. [Online]. Available: https://www.espressif.com/en/news/ESP32-S3_Plumerai

[8] P. Gorce, J. Jacquier-Bret, "Fall detection in elderly people: A systematic review of ambient assisted living and smart home-related technology performance," *Sensors*, vol. 25, no. 21, p. 6540, 2025, doi: 10.3390/s25216540.

[9] M. Ishaq, D. C. Guastella, G. Sutera, G. Muscato, "A systematic review of fall detection and prediction technologies for older adults: An analysis of sensor modalities and computational models," *Appl. Sci.*, vol. 16, no. 4, p. 1929, 2026, doi: 10.3390/app16041929.

[10] R. Jaberi, S. Rostam Niakan Kalhori, F. Bahador, S. Heydarian, Z. Javanmard, "Influencing factors and acceptance levels of robotic and smart home health technologies among older adults: A systematic review," *Health Informatics J.*, vol. 32, no. 2, Art. no. 14604582261436632, 2026, doi: 10.1177/14604582261436632.

[11] M. Kangas, R. Korpelainen, I. Vikman, L. Nyberg, T. Jämsä, "Sensitivity and false alarm rate of a fall sensor in long-term fall detection in the elderly," *Gerontology*, vol. 61, no. 1, pp. 61–68, 2015, doi: 10.1159/000362720.

[12] B. Kwolek, M. Kepski, "Human fall detection on embedded platform using depth maps and wireless accelerometer," *Comput. Methods Programs Biomed.*, vol. 117, no. 3, pp. 489–501, 2014, doi: 10.1016/j.cmpb.2014.09.005.

[13] Y. Lu, Y. Hu, S. Yuan, X. Zhou, K. Zuo, B. Lyu, X. Yuan, J. Yang, "EM-Fall: Embodied mmWave sensing for day-and-night fall detection on humanoid robots," arXiv:2606.11109, 2026.

[14] Y. Malinsky, L. McNicoll, S. Gravenstein, "Robots in nursing homes: Helping nurses detect and prevent falls," *Adv. Geriatr. Med. Res.*, vol. 7, no. 1, e250001, 2025, doi: 10.20900/agmr20250001.

[15] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, C. Peñafort-Asturiano, "UP-Fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, 2019, doi: 10.3390/s19091988.

[16] R. Palanisamy, S. Moorthy, S. V. Manikanthan, J. RajaSekhar, A. Surendar, S. Sadulla, "Generative AI-based autonomous orchestration for intelligent IoT-driven smart home systems," *Int. J. Artif. Intell. Mach. Learn.*, vol. 6, no. 1s, pp. 16–24, 2026, doi: 10.51483/IJAIML.6.1s.2026.16-24.

[17] P. Pirzada, A. Wilde, G. H. Doherty, D. Harris-Birtill, "Understanding older adults' perception, acceptance, and adoption of smart home technologies," *PLoS ONE*, vol. 21, no. 4, e0345563, 2026, doi: 10.1371/journal.pone.0345563.

[18] S. A. Rahimi Azghadi, T.-T.-H. Nguyen, H. Fournier, M. Wachowicz, R. Richard, F. Palma, H. Cao, "Privacy-preserving multi-stage fall detection framework with semi-supervised federated learning and robotic vision confirmation," arXiv:2507.10474, 2025.

[19] D. Schneider, Z. Marinov, M. Mistol, Z. Zhong, A. Jaus, R. Düger, R. Baur, M. S. Sarfraz, R. Stiefelhagen, "OmniFall: From staged through synthetic to wild, a unified multi-domain dataset for robust fall detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2026; arXiv:2505.19889.

[20] A. Tello, V. Degeler, A. Lazovik, "Too good to be true: Accuracy overestimation in (re)current practices for human activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2024, pp. 511–517, doi: 10.1109/PerComWorkshops59983.2024.10503465.

[21] World Health Organization, "Falls," fact sheet, Oct. 21, 2026. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[22] X. Yu, Y. Cai, R. Yang, F. Ma, W. Kim, "Revisiting sensor-based intelligent fall risk assessment for older people: A systematic review," *Eng. Appl. Artif. Intell.*, vol. 144, p. 110176, 2025, doi: 10.1016/j.engappai.2025.110176.