Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

NPU-Accelerated Neuro-Fuzzy Safety Assistance with Decision-Level Multimodal Fusion for Teleoperated Assistive Robots

Sarra Ben Halima

sarah.benhlima@enit.utm.tn

National Engineering School of Tunis

Faten Ben Abdallah

National Engineering School of Tunis

Joseph Haggege

National Engineering School of Tunis

Research Article

Keywords: Assisted teleoperation, Shared control, Adaptive neuro-fuzzy inference system (ANFIS), Edge artificial intelligence, Neural processing unit, Decision-level multimodal fusion, Collision-risk reduction, YOLOX object detection

Posted Date: September 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10929168/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

NPU-Accelerated Neuro-Fuzzy Safety Assistance with Decision-Level Multimodal Fusion for Teleoperated Assistive Robots

Sarra Ben Halima, Faten Ben Abdallah and Joseph Haggège

*Automation Research Laboratory (LARA), LR11ES18, National Engineering School of Tunis, University of Tunis El Manar, 1002 Tunis, Tunisia.

*Corresponding author(s). E-mail(s): sarah.benhlima@enit.utm.tn;
Contributing authors: faten.benabdallah@enit.utm.tn;
joseph.haggege@enit.utm.tn;

Abstract

This paper presents an Edge-AI shared-control framework for collision-risk reduction in teleoperated mobile robots. The human operator retains steering authority, while an Adaptive Neuro-Fuzzy Inference System (ANFIS) regulates the admissible translational velocity from servo-scanned infrared range measurements. A pretrained YOLOX-S detector provides complementary visual detection confidence, which modulates the ANFIS output through a lightweight geometric–semantic decision-level fusion law. A deterministic critical-distance override independently takes priority over both learned components and the operator's forward command. ANFIS and YOLOX-S execute concurrently on a single Hailo-8L neural processing unit interfaced with a Raspberry Pi 5, achieving 50 Hz ANFIS updates and 28.7 fps visual inference at 55 % combined NPU utilisation. The complete assistance stack operates locally with no cloud dependency. Experimental results on a custom four-wheeled platform show an ANFIS MSE of 0.027 and an INT8 deployment deviation of only 0.4 % of full scale relative to FP32. Over 10 teleoperation runs per condition, the proposed fusion strategy reduced the collision rate from 0.40 to 0.10 collisions/run, increased minimum obstacle clearance from 5.1 cm to 10.4 cm, and reduced emergency interventions from 1.70 to 0.20 per run.

Keywords: Assisted teleoperation, Shared control, Adaptive neuro-fuzzy inference system (ANFIS), Edge artificial intelligence, Neural processing unit, Decision-level multimodal fusion, Collision-risk reduction, YOLOX object detection

1 Introduction

Teleoperated mobile robots are widely used in assistive, monitoring, and inspection tasks in which a human operator issues motion commands from a remote interface. In such settings the operator's situational awareness is inherently limited: the camera view is narrow, depth cues are weak, and obstacles lying outside the field of view—or transparent and low-contrast surfaces within it—are easily missed. Purely manual teleoperation therefore places the entire burden of collision avoidance on the operator. Fully autonomous navigation removes that burden but also removes the operator's intent from the loop, and requires mapping, localisation, and planning capabilities that exceed the cost, size, and power budgets of low-cost domestic platforms [16]. Shared control offers an intermediate solution: the operator remains responsible for *where* the robot goes, while an onboard safety layer continuously regulates *how fast* it is permitted to go given the sensed obstacle configuration. Classical proportional–integral–derivative (PID) velocity limiters offer mathematical simplicity but are brittle in the presence of unmodelled dynamics and time-varying obstacle configurations.

Fuzzy logic controllers (FLCs), pioneered by Zadeh [2], handle uncertainty through linguistic IF–THEN rules but lack an intrinsic learning mechanism. Artificial neural networks (ANNs) excel at learning complex nonlinear mappings from data but sacrifice interpretability. The Adaptive Neuro-Fuzzy Inference System (ANFIS), proposed by Jang [3, 6], unifies both paradigms: it preserves the linguistically interpretable rule structure of Sugeno-type systems while employing gradient-based and least-squares optimisation to tune all parameters from training data. Applied to teleoperation assistance, ANFIS can learn a smooth, distance- and bearing-aware velocity policy that adapts to varying obstacle configurations without manual retuning.

Geometric range sensors—such as ultrasonic transducers, infrared (IR) proximity detectors, and laser scanners—measure distances to obstacles but provide no semantic interpretation of the environment. Deep-learning-based object detectors, and in particular YOLOX [5] with its anchor-free formulation and decoupled heads, bridge this semantic gap by classifying detected entities in real time. Deploying such models on a mobile platform requires local inference with minimal latency, driving the emergence of Edge AI accelerators such as the Hailo-8L NPU, which is specified for up to 13 TOPS within a 1.5 W to 2.0 W power envelope [7].

Despite these advances, the integrated co-design of neuro-fuzzy control, deep-learning perception, and NPU-level acceleration within a single embedded robotic platform remains largely unexplored. Most existing works treat the controller and the perception pipeline as independent modules, overlooking the synergies that arise when both are co-optimised for the same edge accelerator and fused through a computationally lightweight multiplicative fusion law.

This paper proposes an NPU-accelerated obstacle-aware safety assistance architecture for teleoperated mobile robots. The operator selects the motion direction through a wireless interface, while an ANFIS velocity controller and a YOLOX semantic perception pipeline execute concurrently on a single Hailo-8L NPU and jointly determine the admissible translational velocity. Their outputs are combined through a multiplicative *geometric–semantic decision-level* fusion law—geometric range and bearing on one branch, RGB semantic detection on the other, combined only at the level of

their outputs—in which a visually derived obstacle confidence score attenuates the geometrically derived base velocity, and a critical-distance condition overrides the operator's forward command entirely. The architecture targets assisted teleoperation in elderly-care and domestic monitoring scenarios, where collision-risk reduction and operator safety support must be delivered within strict cost, size, and power constraints, and where remote monitoring cannot assume cloud connectivity. The system performs no path planning and generates no navigation trajectory: steering authority remains entirely with the human operator.

The principal contributions are:

1. A **shared-control architecture** for a teleoperated mobile robot in which the operator determines the motion direction while an Edge-AI safety layer automatically regulates translational velocity.
2. A **dual-inference implementation** executing ANFIS and YOLOX-S concurrently on a single Hailo-8L NPU, with measured latency, throughput, utilisation, queue occupancy, and frame loss under joint load.
3. A **decision-level fusion mechanism**, $\bar{v}_f = \bar{v}_A(1 - s_{\text{obs}})$, combining IR-based geometric information with YOLOX-based semantic detections to attenuate unsafe forward velocity.
4. A **critical-distance safety override** that is independent of both learned components and takes priority over the operator's forward command.
5. A **cost-effective** 360° **servo-scanned dual-IR ranging system** based on two back-to-back infrared distance sensors synchronised with a servo motor, together with a complete deployment pipeline converting a PyTorch-trained ANFIS model into a Hailo-compatible HEF binary through ONNX export, INT8 quantisation, and hardware-aware compilation, with measured INT8 deployment fidelity.
6. **Experimental validation comparing manual, ANFIS-assisted, and ANFIS–YOLOX-assisted teleoperation** in terms of collision rate, obstacle clearance, emergency interventions, stopping distance, command-to-actuation latency, and task completion time.

The remainder of the paper is organised as follows. Section 2 surveys related work and identifies the scientific gap. Section 3 presents the system architecture. Section 4 details the mathematical modelling. Section 5 describes the NPU deployment pipeline. Section 6 reports experimental results. Section 7 provides a critical discussion. Section 8 concludes the paper.

2 Related Work

Fuzzy and neuro-fuzzy approaches have been widely applied to mobile robot navigation because of their ability to handle uncertain sensory information and nonlinear control relationships [2–4]. Recent studies continue to demonstrate their relevance for collision-free navigation and obstacle avoidance. Omrane et al. [11] designed a Sugeno-type fuzzy controller for trajectory tracking and obstacle avoidance on a differential-drive platform, while Mohanty [12] proposed an FCM-clustering ANFIS path planner for uncertain navigation environments. In the context of socially assistive robotics, Soares

et al. [13] developed evolving fuzzy-logic systems for personalised interaction with elderly users in dementia care. Learning-based controllers have also been targeted at embedded platforms: Ben Halima et al. [10] applied deep reinforcement learning to obstacle avoidance for embedded mobile robot navigation, obtaining reactive policies that nevertheless lack the linguistic interpretability of neuro-fuzzy inference. Despite these advances, the deployment of neuro-fuzzy controllers on dedicated Edge-AI accelerators remains comparatively less investigated.

Deep-learning-based perception has meanwhile become central to robotic navigation. Single-stage detectors such as YOLOX [5] provide an effective accuracy–latency compromise for embedded applications. Mavrogiorgos et al. [14] combined YOLOv3 with fuzzy-logic safety zones for human–robot collaborative obstacle avoidance in warehouse environments, and Eirale et al. [15] demonstrated omnidirectional human-centred navigation with person-following for indoor monitoring and assistance. Edge-AI platforms increasingly support such models through local hardware acceleration. Recent surveys [16, 17] highlight NPU/TPU-class acceleration, INT8 quantisation, pruning, and hardware-aware compilation as key technologies for embedded robotic perception, while Tahir et al. [18] surveyed the application of edge computing to robotics across NPU and FPGA targets. Portugal et al. [19] further studied the practical deployment of a service robot in an elderly-care centre, identifying reliable indoor navigation as a critical requirement. In a related edge-deployment study, Ben Halima et al. [9] combined embedded object detection with voice recognition and an LLM-based emotional assistant for elderly-care robots, confirming the feasibility of fully local perception and interaction on low-cost hardware. Nevertheless, hardware acceleration is still predominantly applied to perception, with control algorithms typically executed separately on the host processor.

Multimodal sensing can further improve navigation by combining complementary geometric and semantic information [20, 21]. Vishnu et al. [20] recently proposed multi-modal sensing with group-aware pose estimation for socially compliant indoor navigation, and Qi et al. [21] presented a sensor-fusion survey covering LiDAR–camera–IMU integration for autonomous mobile robots. Most existing systems, however, perform fusion at the measurement, feature, or map level. Comparatively less attention has been given to decision-level fusion in which visual detection confidence directly modulates a neuro-fuzzy motion command. Such a strategy is particularly suitable for embedded assistive platforms because the geometric and visual branches can remain independently deployable while contributing jointly to the final control response.

The literature therefore reveals three closely related limitations, summarised in Table 1: limited NPU-level deployment of neuro-fuzzy controllers; separation between accelerated visual perception and control workloads; and limited investigation of decision-level geometric–semantic fusion as a safety layer for human-in-the-loop robot operation. These gaps are particularly relevant to assistive and elderly-care robotics, where teleoperated platforms must reduce collision risk within strict cost, size, and power constraints, and where full autonomy is often neither necessary nor desirable because the operator's intent carries task context the robot does not have. The proposed architecture addresses these limitations by combining a servo-scanned dual-IR

Table 1: Comparison with representative related work.

Work	Teleop.	Neuro-fuzzy	Visual	Fusion	Target
Omrane et al. [11]	–	Fuzzy	–	–	CPU
Mohanty [12]	–	ANFIS	–	–	CPU
Soares et al. [13]	–	Evolving	–	–	CPU
Mavrogiorgos et al. [14]	–	Fuzzy	✓	✓	GPU/CPU
Eirale et al. [15]	–	–	✓	✓	GPU
Vishnu et al. [20]	–	–	✓	✓	GPU
Ben Halima et al. [10]	–	–	–	–	Embedded GPU
Proposed	✓	ANFIS	✓	Decision level	Hailo-8L NPU

ranging system with ANFIS-based velocity regulation and YOLOX semantic perception, both executing concurrently on a single Hailo-8L NPU, and fusing their outputs through a multiplicative decision-level law.

3 System Architecture

The architecture partitions the task into a low-level embedded layer (sensor acquisition and motor actuation) and a high-level assistance layer (perception, velocity regulation, and fusion), interconnected through serial and wireless channels (Fig. 1).

3.1 Sensor Layer

Two Sharp GP2Y0A41SK0F IR sensors, mounted back-to-back on a servo motor, provide omnidirectional range measurements within the 4 cm to 30 cm envelope. A 6-axis IMU (LSM6DSL accelerometer/gyroscope) integrated on the STM32L4S5 supplies raw motion data for attitude estimation. A Raspberry Pi Camera Module connected to the Raspberry Pi 5 through the CSI-2 interface feeds the visual perception pipeline. Frames are captured at 1280×720 px and letterboxed to the 640×640 RGB input tensor expected by the deployed YOLOX-S network.

3.2 Low-Level Embedded Layer

The STM32L4S5 performs real-time attitude estimation using a quaternion-based 6-axis EKF (Section 4.1) and transmits Euler angles wirelessly. The Arduino MEGA 2560 translates arbitrated motion commands into PWM signals applied to four DC motors through an L298N H-bridge, and forwards IR sensor (distance, angle) pairs to the Raspberry Pi via an FTDI serial bridge.

3.3 High-Level Assistance Layer

The Raspberry Pi 5, equipped with a Hailo-8L AI HAT+, runs two concurrent inference pipelines:

- **ANFIS controller:** infers a base velocity $v_{\text{base}} \in [0, 70]$ from (distance, angle) pairs.

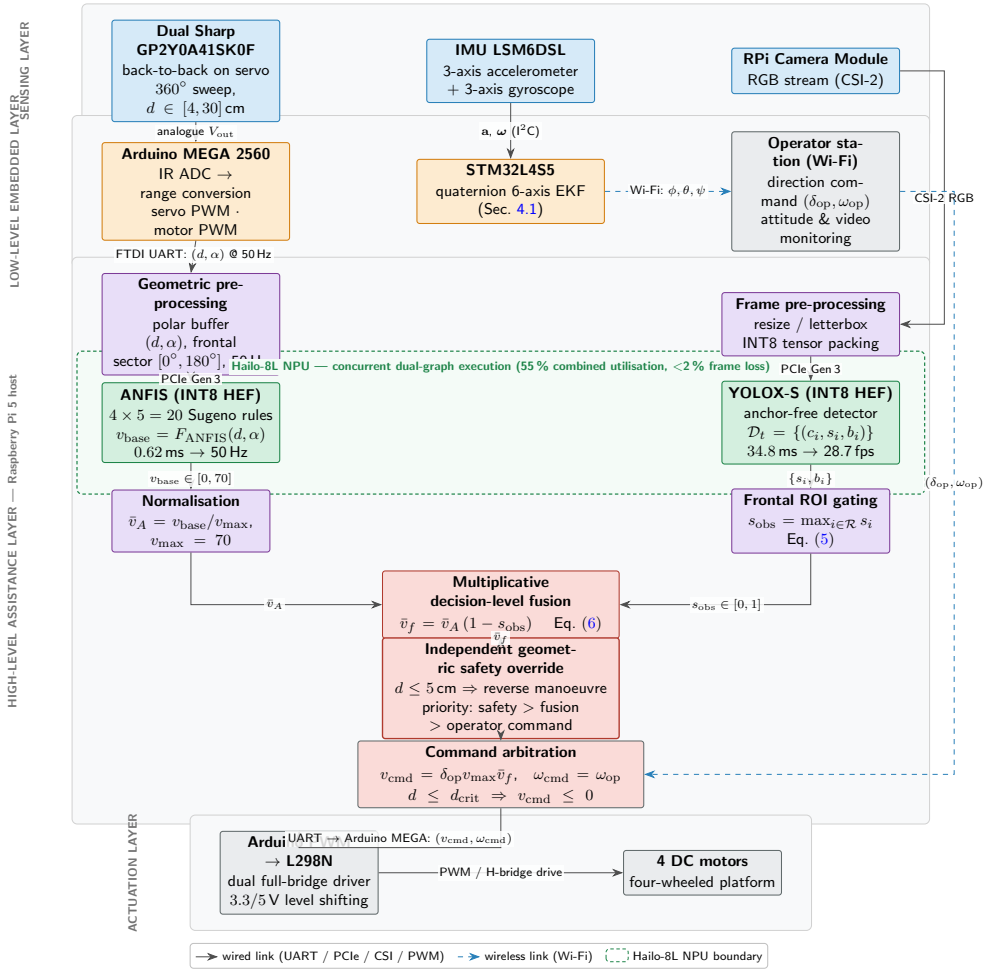


Fig. 1: System architecture of the proposed NPU-accelerated safety assistance framework for teleoperation. The sensing layer supplies servo-scanned dual-IR ranging, inertial, and RGB data; the low-level embedded layer performs range conversion, actuation, and quaternion-based 6-axis EKF attitude estimation; the high-level assistance layer executes the ANFIS velocity controller and the YOLOX-S detector concurrently on a single Hailo-8L NPU and combines their outputs through the multiplicative decision-level fusion law $\bar{v}_f = \bar{v}_A(1 - s_{\text{obs}})$ (Eq. (6)). The resulting speed envelope is arbitrated with the operator's directional command $(\delta_{\text{op}}, \omega_{\text{op}})$ subject to the critical-distance override (Eq. (7)), and the arbitrated command is returned to the Arduino MEGA 2560, which drives the four DC motors through the L298N full-bridge stage.

- **YOLOX detector:** produces an obstacle confidence score $s_{\text{obs}} \in [0, 1]$ from the live camera stream.

The outputs are combined through the multiplicative fusion law (Section 4.4) to produce an admissible translational velocity, which is applied to the operator's directional command and transmitted to the Arduino for motor actuation.

3.4 Teleoperation and Shared-Control Strategy

The proposed platform operates in a shared-control configuration. The human operator selects the desired motion direction through the wireless interface; the autonomous safety layer performs neither path planning nor trajectory generation. Instead, it supervises the operator's command using geometric and semantic obstacle information. ANFIS determines an obstacle-dependent base velocity from the IR distance and bearing, while the YOLOX confidence score further attenuates this velocity. The operator's directional command is preserved unless the critical-distance safety condition is activated, in which case the safety layer overrides forward motion.

The operator interface exposes the following commands:

- discrete directional commands—*forward*, *reverse*, *left*, *right*, *stop*;
- a forward-motion enable $\delta_{\text{op}} \in \{-1, 0, +1\}$ selecting reverse, stop, or forward;
- an angular command ω_{op} setting the turning direction and rate.

The operator does *not* set the translational speed: the magnitude of v_{cmd} is determined entirely by the assistance layer.

3.5 Hardware Integration

A bidirectional logic-level converter bridges 3.3 V (STM32/Raspberry Pi) and 5 V (Arduino) logic. Drive electronics are powered by two 4 V batteries in series; the Raspberry Pi is independently supplied by a 5 V/3 A USB power bank. The two supplies separate the computation and motor power rails and reduce motor-induced disturbances on the logic supply; they do not provide galvanic isolation, since the boards share a common ground through the UART, I^2C, and level-conversion links.

4 Mathematical Modelling and Control Strategy

4.1 Quaternion-Based Extended Kalman Filter

Attitude is represented by a unit quaternion $\boldsymbol{q} = [q_0, q_1, q_2, q_3]^\top$, avoiding the gimbal-lock singularity of Euler angles [1], and propagated from the gyroscope rates $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^\top$ by

$$\dot{\boldsymbol{q}} = \tfrac{1}{2}\, \boldsymbol{\Omega}(\boldsymbol{\omega})\, \boldsymbol{q}, \tag{1}$$

with $\boldsymbol{\Omega}(\boldsymbol{\omega})$ the standard skew-symmetric quaternion-rate matrix. Prediction and covariance propagation follow the usual discrete-time EKF equations. In the correction step the accelerometer supplies, under quasi-static conditions, an absolute gravity reference for roll and pitch; the innovation between the predicted and measured gravity directions updates the quaternion and its covariance. The accelerometer is insensitive

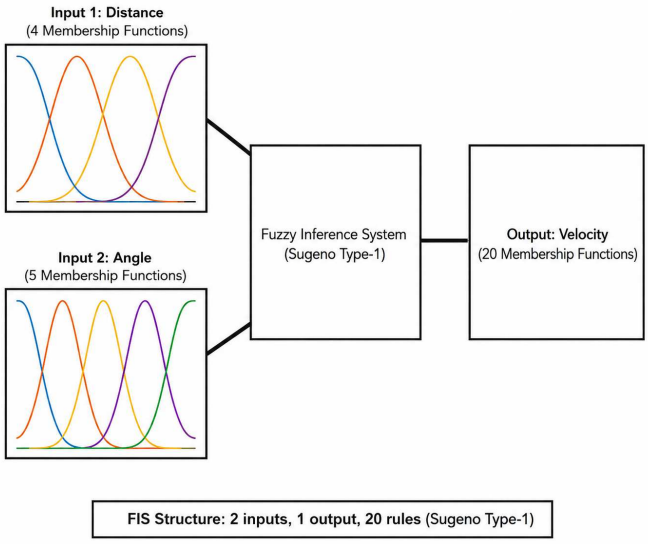


Fig. 2: Two-input, one-output Sugeno-type ANFIS controller.

to rotation about the gravity axis, so yaw is propagated from gyroscope integration alone and retains a residual drift. A magnetometer (LIS3MDL) is present on the hardware but is not used in the correction stage, so the estimator is a 6-axis EKF.

Calibration uses the standard six-position accelerometer method for per-axis offsets and scale factors, and $N = 1000$ static samples for the gyroscope bias; no magnetometer calibration is performed. The resulting attitude estimate serves platform monitoring and operator feedback and does not enter Eq. (6) or Eq. (7).

4.2 Adaptive Neuro-Fuzzy Inference System (ANFIS)

4.2.1 Architecture.

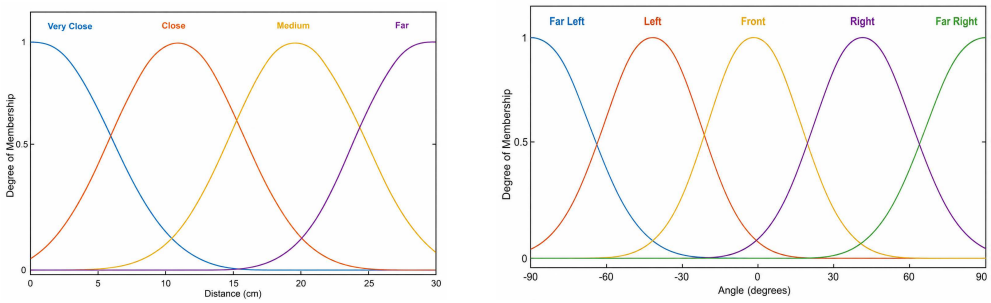
The velocity controller is a first-order Sugeno-type ANFIS with two inputs and one output, governed by $4 \times 5 = 20$ fuzzy rules [3, 4] (Fig. 2).

The two inputs are obstacle distance and bearing angle. The distance $d \in [4, 30]$ cm is partitioned into four Gaussian membership functions: *very close*, *close*, *medium*, and *far*. The bearing $\alpha \in [0°, 180°]$, with $\alpha = 90°$ denoting the frontal direction, is partitioned into five Gaussian membership functions: *far left*, *left*, *front*, *right*, and *far right*. The corresponding membership functions are shown in Fig. 3.

The ANFIS model was trained in MATLAB using the centred internal bearing variable $\theta \in [-90°, 90°]$, with $\theta = 0°$ denoting the frontal direction. The two angular conventions are related by

$$\alpha = \theta + 90°, \tag{2}$$

so that $\theta = -90°$ maps to $\alpha = 0°$ and $\theta = 90°$ maps to $\alpha = 180°$. The bearing membership functions in Fig. 3b and the ANFIS surface in Fig. 4 are expressed in the MATLAB training convention θ; elsewhere, the shifted convention α is used.



(a) Distance input. (b) Bearing input.

Fig. 3: Gaussian membership functions of the ANFIS inputs: (a) distance and (b) bearing. The bearing functions are shown in the MATLAB training variable $\theta \in [-90°, 90°]$; elsewhere, $\alpha = \theta + 90°$ is used.

The ANFIS output is the base velocity $v_{\text{base}} \in [0, 70]$. The bearing input shapes the admissible forward velocity so that, for the same obstacle distance, a laterally located obstacle permits a higher velocity than a frontal one.

4.2.2 Output-surface analysis

The three-dimensional output surface (Fig. 4) shows velocity increasing with distance and decreasing for frontal obstacles ($\alpha \approx 90°$), confirming physically consistent behaviour over most of the input domain. The short-range portion of the surface is examined separately in Section 6.7, where a non-monotonic region near the lower distance bound is characterised and its consequences for the safety architecture are discussed.

4.3 Servo-Scanned Dual-IR Ranging System

Two Sharp GP2Y0A41SK0F IR distance sensors, mounted diametrically on a servo motor, cover the full 360° field of view in a single 180° sweep. At each angular increment, the Arduino converts both sensor voltages to distances and packages the resulting polar point cloud for the Raspberry Pi (Fig. 5). Although the full 360° scan is retained by the sensing layer, only the frontal $[0°, 180°]$ sector is used by the ANFIS velocity controller.

Throughout this paper, the 50 Hz rate denotes the update rate of the individual (d, α) measurements delivered to the assistance layer, and correspondingly of the ANFIS control cycle. It does *not* denote the rate of complete mechanical 360° sweeps: a full sweep spans many such updates and is bounded by the servo traversal time, so the refresh rate of the complete polar point cloud is one to two orders of magnitude lower. The assistance layer operates on a rolling frontal buffer rather than on a completed scan, which is what allows the control cycle to run at sensor update rate.

The selection rule is as follows. Each incoming (d, α) sample in the frontal sector $[0°, 180°]$ is written into a polar buffer indexed by angular bin. At every control cycle

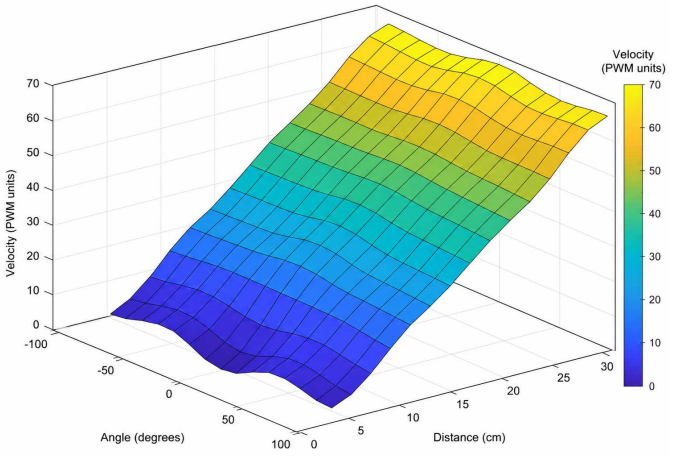


Fig. 4: ANFIS output surface $v_{\text{base}}(d, \theta)$, shown in the MATLAB training variable θ. The bearing convention used in the remainder of this paper is $\alpha = \theta + 90°$ (Eq. 2).

the pair submitted to ANFIS, and the range tested against d_{crit}, is the *nearest valid obstacle currently held in that buffer*, not simply the newest sample. A frontal obstacle detected at one servo position therefore continues to constrain the velocity envelope while the servo sweeps elsewhere, instead of being displaced by a subsequent lateral reading. Entries are discarded once they exceed a fixed retention window, set to the servo traversal time of the frontal sector, so that an obstacle the platform has already passed does not constrain it indefinitely. This window bounds the staleness of the geometric input and is the mechanical counterpart of the freshness rule applied to the visual branch (Section 5.4). Both bounds are collected in Algorithm 1.

4.4 Decision-Level Multimodal Fusion

The geometric branch provides obstacle distance d and bearing α through the IR ranging system. These are processed by ANFIS to generate the base velocity

$$v_{\text{base}} = F_{\text{ANFIS}}(d, \alpha), \tag{3}$$

whose normalised form is

$$\bar{v}_A = v_{\text{base}}/v_{\max}, \qquad v_{\max} = 70. \tag{4}$$

Here v_{base} and $v_{\max}$ are expressed in 8-bit PWM duty-cycle counts on the $[0, 255]$ scale accepted by the Arduino motor driver, so that $v_{\max} = 70$ corresponds to the maximum duty cycle permitted on the platform; the normalised quantities $\bar{v}_A$ and $\bar{v}_f$ are dimensionless and lie in $[0, 1]$.

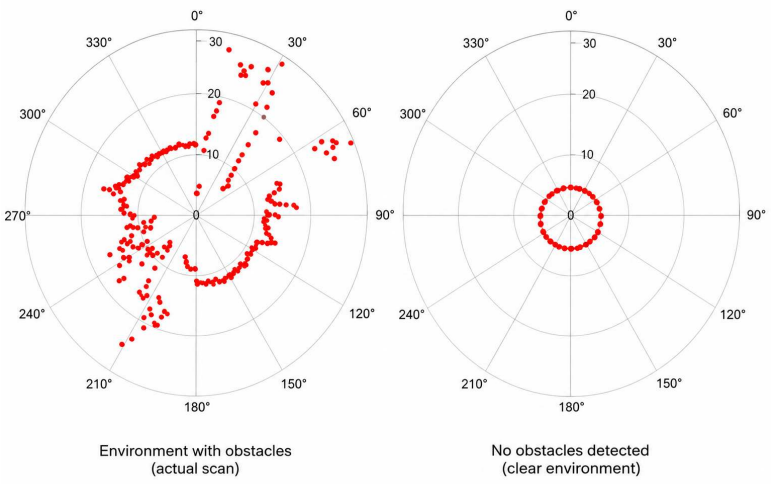


Fig. 5: Servo-scanned IR ranging validation. *Left:* cluttered-lab polar scan. *Right:* controlled circular test confirming sensor–servo synchronisation.

In parallel, YOLOX produces a set of detections $\mathcal{D}_t = \{(c_i, s_i, b_i)\}_{i=1}^{N_t}$. The obstacle confidence entering the fusion law is

$$s_{\text{obs}} = \max_{i \in \mathcal{R}} s_i, \tag{5}$$

where $\mathcal{R}$ contains detections whose bounding boxes intersect the robot's frontal safety region; if $\mathcal{R} = \emptyset$, then $s_{\text{obs}} = 0$.

The quantity s_{obs} is a *visual detection confidence* used as a modulation term, and this is how it should be read wherever it appears below. The fusion law does not distinguish between object classes; only the confidence enters Eq. (6). The visual branch performs no monocular depth estimation, so s_{obs} carries no range information: a distant and a nearby object detected with equal confidence produce equal attenuation. It is therefore not an estimate of obstacle distance or of collision probability. All proximity information reaches the command through the geometric branch alone—through d in Eq. (3) and through the critical-distance condition of Eq. (7).

The two branches are combined through the multiplicative fusion law

$$\boxed{\bar{v}_f = \bar{v}_A \, (1 - s_{\text{obs}})} \tag{6}$$

When $s_{\text{obs}} \approx 0$, the ANFIS command is preserved. As visual confidence increases, velocity is progressively attenuated. For $s_{\text{obs}} \approx 1$, the admissible forward velocity approaches zero.

Equation (6) defines an admissible *speed envelope*, not a motion command. The command actually applied to the drive stage combines this envelope with the

operator's directional input $(\delta_{\text{op}}, \omega_{\text{op}})$:

$$v_{\text{cmd}} = \begin{cases} \delta_{\text{op}}\, v_{\max}\, \bar{v}_A\, (1 - s_{\text{obs}}), & d > d_{\text{crit}}, \\ -v_{\text{rev}}, & d \le d_{\text{crit}}, \end{cases} \qquad \omega_{\text{cmd}} = \omega_{\text{op}}, \tag{7}$$

with $d_{\text{crit}} = 5\,\text{cm}$, and where $v_{\text{rev}} = 60$ is the fixed reverse duty cycle applied by the override, in the same units as $v_{\max}$. It was set empirically as the lowest duty cycle that reliably overcomes static friction on the test surface while keeping the disengagement manoeuvre slow, and it is held for a fixed disengagement interval before control returns to the first branch of Eq. (7). The second branch of Eq. (7) is a purely geometric, deterministic rule: it depends on d alone and is evaluated on the host independently of both inference graphs, so that short-range safety rests on neither the learned ANFIS surface nor the detector confidence.

The override commands an active reverse rather than a simple stop because $d_{\text{crit}} = 5\,\text{cm}$ lies only 1 cm above the lower bound of the IR measurement range: bringing the platform to rest at that separation would leave it inside the least reliable region of the sensor characteristic, with no margin for the operator to manoeuvre out. The disengagement restores a usable sensing margin before authority returns to the first branch of Eq. (7).

The manoeuvre is triggered by the frontal range alone. The rear sector of the 360° scan is acquired by the sensing layer but is not consulted before the reverse command is issued, so the override is not guaranteed to be collision-free with respect to obstacles behind the platform; in the reported trials the course was arranged so that the rear half-space was clear wherever the override engaged. Two remedies follow directly from the existing sensing layer and are identified as future work: gating the reverse on a rear clearance d_r, so that $v_{\text{cmd}} = -v_{\text{rev}}$ only when $d_r > d_{\text{rear}}$ and $v_{\text{cmd}} = 0$ otherwise; or reducing the override to a stop unconditionally and leaving disengagement to the operator. Figure 6 illustrates the information flow.

5 NPU-Accelerated Edge Deployment

5.1 Host–Accelerator Partitioning

The Raspberry Pi 5 acts as the host processor, handling sensor interfacing, data preparation, fusion, and motor-command coordination. The Hailo-8L NPU operates as a companion accelerator, executing both ANFIS and YOLOX inference graphs. The Hailo dataflow architecture exploits computational parallelism through distributed compute clusters, local memory, and a configurable interconnect, while INT8 execution reduces memory traffic and arithmetic cost. The Hailo software stack supports multiple independent network graphs on a single physical accelerator, enabling modular compilation, profiling, and independent model updates.

5.2 ANFIS Deployment Workflow

Deploying the ANFIS controller on the Hailo-8L requires a multi-stage compilation workflow executed on an Ubuntu host before transfer to the Raspberry Pi 5 (Fig. 7).

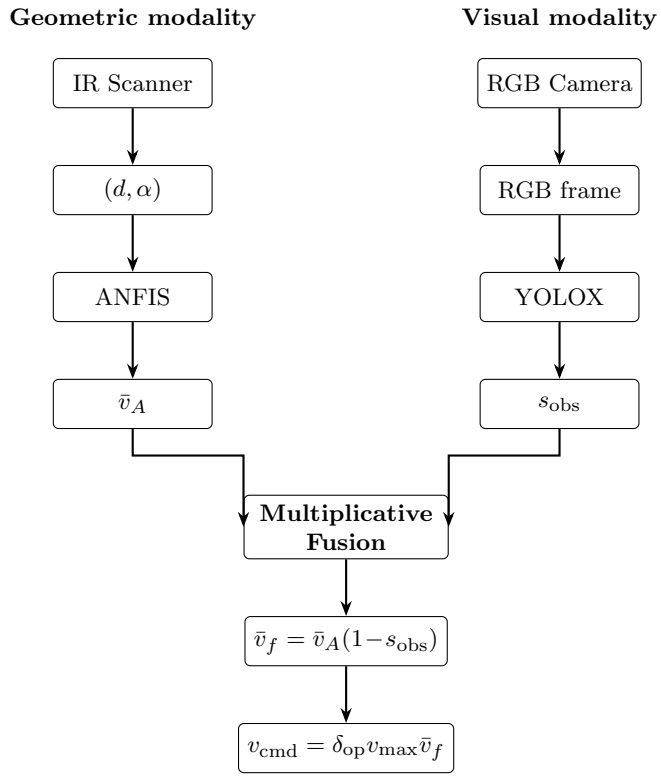


Fig. 6: Decision-level multimodal fusion architecture. The IR ranging system feeds (d, α) to ANFIS; the RGB camera feeds YOLOX. Their outputs $\bar{v}_A$ and s_{obs} are combined at the decision level via Eq. (6), and the resulting speed envelope is applied to the operator's directional command via Eq. (7).

The chain is summarised in Fig. 7; three of its stages carry information not evident from the diagram. The PyTorch–ONNX export was verified to preserve numerical equivalence to within 10^{-4}, so the graph submitted to the compiler is the graph that was trained. Quantisation used 64 representative calibration samples, below the 1024 suggested in the compiler documentation [8]; the consequences of this are measured in Section 6.4 and the shortfall is recorded as a deployment limitation. Hardware-aware mapping converged after four iterations to a 554 kB HEF binary. At run time the binary is loaded through HailoRT, IR measurements arrive at 50 Hz over the serial interface, and the ANFIS output returns to the host for fusion.

5.3 YOLOX Deployment

The visual branch operates on RGB image tensors. Camera frames are acquired by the Raspberry Pi, letterboxed to 640×640 px, and supplied to the Hailo-deployed YOLOX-S network at that input resolution. YOLOX-S is employed as a

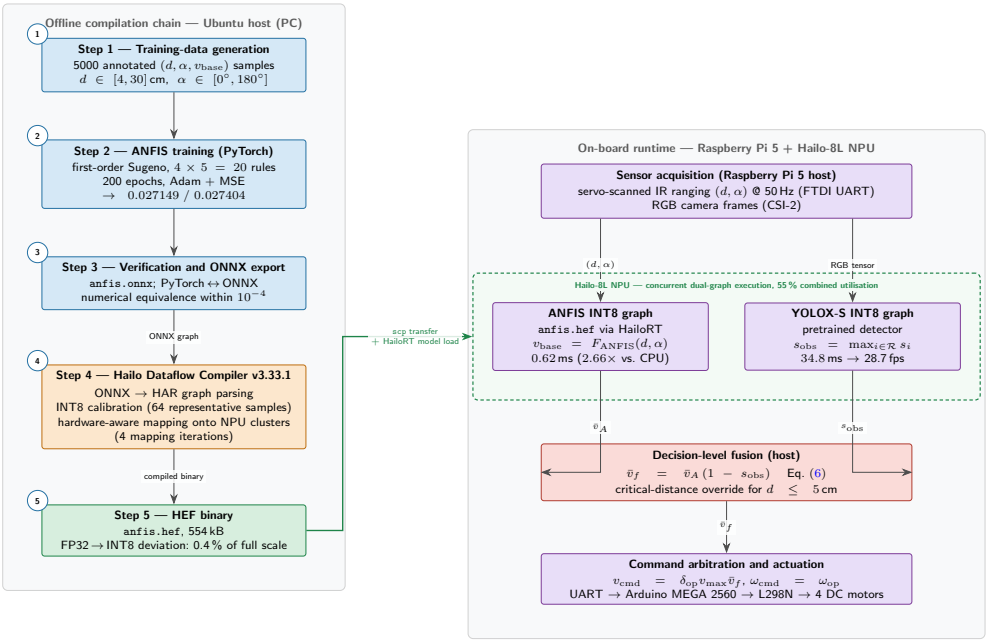


Fig. 7: ANFIS deployment workflow. *Left:* the offline compilation chain executed on the Ubuntu host, from training-data generation and PyTorch training through ONNX export, INT8 calibration, and hardware-aware mapping by the Hailo Dataflow Compiler to the 554 kB HEF binary. *Right:* the on-board runtime, in which the transferred HEF is loaded through HailoRT and executed concurrently with the YOLOX-S graph on the Hailo-8L NPU, their outputs being fused on the Raspberry Pi 5 host and arbitrated with the operator's directional command before transmission to the motor-control layer.

pretrained semantic-perception component; consequently, the reported results characterise embedded deployment performance on the Raspberry Pi 5/Hailo-8L platform rather than detector-training performance. No detector re-training or fine-tuning was performed for the deployment environment, and detection accuracy was therefore not independently benchmarked in the laboratory setting; this is revisited in Section 7.2.

Raw detections are filtered by a confidence threshold and by non-maximum suppression before the frontal gating of Eq. (5) is applied. The region of interest $\mathcal{R}$ is a fixed central band of the letterboxed input, chosen to subtend the width of the platform at the ranges covered by the IR envelope; a detection belongs to $\mathcal{R}$ when its bounding box intersects that band. Because Eq. (5) takes the maximum over $\mathcal{R}$, the attenuation is governed by the single most confident frontal obstacle and is independent of the number of objects present, which keeps $\bar{v}_f$ from collapsing in cluttered scenes merely because many objects are visible.

5.4 Runtime Synchronisation

Because ANFIS and YOLOX operate at different update rates—50 Hz against approximately 28.7 Hz—the Raspberry Pi synchronises their outputs at each control cycle by combining the most recent valid ANFIS velocity with the most recent YOLOX confidence via Eq. (6). The resulting speed envelope is arbitrated with the operator's directional command (Eq. (7)) and transmitted to the Arduino motor-control layer.

Each confidence score carries the acquisition timestamp t_{YOLOX} of the frame from which it was derived, and is admitted into Eq. (6) at control time t only while

$$t - t_{\mathrm{YOLOX}} < T_{\max}, \tag{8}$$

with $T_{\max}$ set to twice the nominal inter-frame interval at 28.7 fps, that is approximately 70 ms. If no detection result satisfies Eq. (8)—because a frame was dropped, delayed, or the visual pipeline stalled—the stale score is discarded and the fusion reverts to $s_{\mathrm{obs}} = 0$, so that the admissible velocity falls back to the purely geometric envelope $\bar{v}_A$ rather than being modulated by an outdated observation. This is the conservative choice with respect to the operator's authority but not with respect to speed: a stalled visual branch removes attenuation rather than imposing it, which is admissible only because the geometric branch and the critical-distance condition of Eq. (7) remain active and depend on no visual input.

Algorithm 1 states the complete control cycle, including the two staleness bounds, the selection rule applied to the polar buffer, and the priority of the critical-distance condition over both learned components and over the operator's command.

6 Experimental Setup and Results

The architecture was evaluated at three levels: embedded sensing and state estimation, Edge-AI execution, and integrated ANFIS–YOLOX safety assistance under operator control.

6.1 Experimental Platform

The robotic platform (Fig. 8) comprises an STM32L4S5 for IMU processing, an Arduino MEGA 2560 for motor control, a Raspberry Pi 5 with Hailo-8L AI HAT+, dual Sharp GP2Y0A41SK0F IR sensors, an onboard camera, and an L298N H-bridge.

6.2 6-Axis EKF Characterisation

Roll and pitch were independently derived from accelerometer measurements as

$$\phi_{\mathrm{acc}} = \operatorname{atan2}(a_y, a_z), \qquad \theta_{\mathrm{acc}} = \operatorname{atan2}\Big(-a_x, \sqrt{a_y^2 + a_z^2}\Big). \tag{9}$$

Gyroscope angular rates were numerically integrated for yaw comparison. Figure 9 compares sensor-derived and EKF-estimated attitudes.

Dispersion was quantified during stationary intervals (Table 2). The noise-reduction ratio is $\eta = (1 - \sigma_{\mathrm{EKF}}/\sigma_{\mathrm{ref}}) \times 100\%$.

Algorithm 1 Assistance-layer control cycle, executed at 50 Hz on the Raspberry Pi 5 host.

Require: frontal polar buffer $\mathcal{B}$; latest detection result $(s_{\text{obs}}, t_{\text{YOLOX}})$; operator input $(\delta_{\text{op}}, \omega_{\text{op}})$; control time t

$\omega_{\text{cmd}} \leftarrow \omega_{\text{op}}$ {steering is never modified}
discard entries of $\mathcal{B}$ older than the retention window
if $\mathcal{B} = \emptyset$ **then**
  $v_{\text{cmd}} \leftarrow 0$ {no valid geometric evidence}
else
  $(d, \alpha) \leftarrow$ nearest valid obstacle in $\mathcal{B}$
  if $d \leq d_{\text{crit}}$ **then**
    $v_{\text{cmd}} \leftarrow -v_{\text{rev}}$, held for the disengagement interval {override, Eq. (7)}
  else
    $\bar{v}_A \leftarrow F_{\text{ANFIS}}(d, \alpha)/v_{\max}$ {Hailo-8L, 0.62 ms}
    if $t - t_{\text{YOLOX}} \geq T_{\max}$ **then**
      $s_{\text{obs}} \leftarrow 0$ {stale detection discarded}
    end if
    $v_{\text{cmd}} \leftarrow \delta_{\text{op}}\, v_{\max}\, \bar{v}_A (1 - s_{\text{obs}})$ {Eqs. (6) and (7)}
  end if
end if
transmit $(v_{\text{cmd}}, \omega_{\text{cmd}})$ to the Arduino MEGA 2560



Fig. 8: Experimental platform. *Left:* front view; *right:* side view.

Roll and pitch dispersion decrease by 70.7% and 69.1%, respectively. The smaller yaw improvement (22.0%) and the measured residual drift of 0.74°/min are consistent with the 6-axis configuration, which lacks an absolute heading observation. Absolute error metrics (RMSE, MAE) are not reported because no independent ground-truth orientation was available. The comparison therefore characterises dispersion reduction

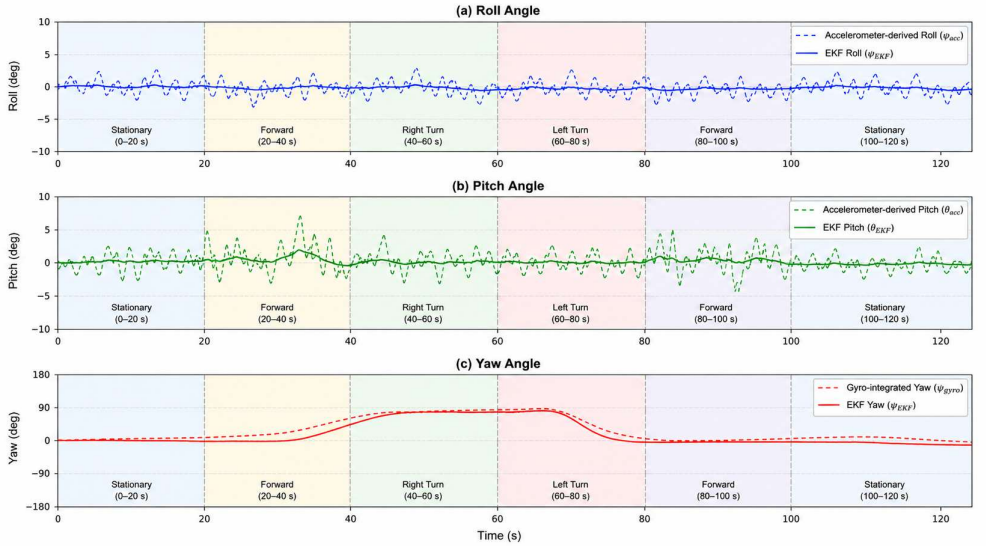


Fig. 9: 6-axis EKF response: (a) roll, (b) pitch, (c) yaw.

Table 2: 6-axis EKF quantitative performance.

Angle	Reference	Ref. σ (°)	EKF σ (°)	η (%)
Roll	Accelerometer	0.82	0.24	70.7
Pitch	Accelerometer	0.94	0.29	69.1
Yaw	Gyroscope	0.41	0.32	22.0

rather than accuracy: the roll/pitch reference is derived from the accelerometer and the yaw reference from the integrated gyroscope—the same sensors that drive the filter.

6.3 ANFIS Training and Validation

6.3.1 Dataset.

The 5000-sample dataset spans $d \in [4, 30]$ cm and $\alpha \in [0°, 180°]$, with mean bearing 89.15° and standard deviation 51.89°, confirming broad frontal coverage (Table 3).

Table 3: Statistics of the two ANFIS input variables over the 5000-sample acquisition dataset.

Variable	Min	Mean	Std	Max
Distance (cm)	4.00	16.82	7.46	29.98
Bearing (°)	0.01	89.15	51.89	179.95

6.3.2 Generation of the target velocity.

The supervision signal deserves explicit description, since it determines what the network actually learns. The target v_{base} associated with each (d, α) pair was not taken directly from free-running operator commands, which would make the model an imitation of one operator's driving style rather than a velocity policy. Instead, targets were assigned from an explicit reference mapping: for each sampled obstacle configuration, the admissible forward duty cycle was defined as a monotonically increasing function of range, scaled by the bearing offset so that a laterally displaced obstacle admits a higher speed than a frontal one at the same range, and capped at $v_{\max} = 70$. Each assigned value was then checked on the platform, the robot being driven at that duty cycle towards the corresponding obstacle configuration and the target retained only when the platform could be brought to rest before contact. The learned surface therefore approximates this reference mapping rather than an operator's behaviour. It remains an approximation, and its departure from monotonicity near the lower distance bound is characterised in Section 6.7, which is why short-range protection is delegated to the independent geometric condition of Eq. (7) rather than to the learned envelope.

6.3.3 Role of the two ANFIS implementations.

Two implementations of the same ANFIS structure were used at different stages of the workflow. The MATLAB implementation was used for initial ANFIS structure design, membership-function shaping, and rule-base analysis, whereas the PyTorch implementation was developed for ONNX export and subsequent Hailo INT8 deployment, since the MATLAB model cannot be exported directly into the Hailo compilation chain. The design sequence is therefore: MATLAB structure and membership-function design $\rightarrow$ PyTorch reimplementation of the same 4×5 rule base $\rightarrow$ ONNX export $\rightarrow$ INT8 quantisation $\rightarrow$ Hailo deployment. The two error figures reported below originate from different internal criteria and are consequently not directly comparable; they are reported separately and only the PyTorch model is deployed. The MATLAB implementation was trained for 100 epochs and reached an internal training error of 0.824 56; because this criterion is not directly comparable with the PyTorch MSE and the MATLAB model is not deployed, no further MATLAB training analysis is reported.

6.3.4 PyTorch training.

The 5000 samples were partitioned into disjoint training and validation subsets, both inputs being min–max normalised over their physical ranges before training. Optimisation used Adam with an MSE loss over 200 epochs, the membership-function centres being initialised by uniform partition of each input range. The loss is evaluated in the units of the network output, so the final MSE of 0.027 corresponds to a root-mean-square error of 0.165 PWM counts, or 0.24 % of the $[0, 70]$ full-scale range. This is the same order as the INT8 quantisation deviation reported in Section 6.4, so neither the fit residual nor the quantisation error dominates the deployed response. Table 4 reports the convergence over 200 epochs.

Table 4: ANFIS training and validation convergence (selected epochs).

Epoch	Train MSE	Val MSE
20	0.143097	0.132629
60	0.057495	0.057796
100	0.045397	0.045685
160	0.031317	0.031563
200	**0.027149**	**0.027404**

The final train–validation gap is $\Delta_{\mathrm{MSE}} = 2.55 \times 10^{-4}$ (0.94% relative), indicating stable generalisation without pronounced overfitting.

6.4 INT8 Deployment Fidelity

The FP32 and INT8 ANFIS outputs were compared on 1000 independent samples (Table 5).

Table 5: INT8 fidelity relative to FP32.

Metric	Value
Test samples	1000
FP32–INT8 MAE	0.27
FP32–INT8 RMSE	0.39
Max absolute error	1.25
MAE normalised to full-scale range	0.4%

The mean absolute error normalised to the full-scale output range, $0.27/70 \approx 0.4\%$, confirms that INT8 conversion preserves the ANFIS response. This quantity is a full-scale-normalised MAE and not a conventional mean relative error, which would be undefined near $v_{\mathrm{base}} = 0$.

6.5 Edge-AI Execution Performance

The inference latency of the ANFIS controller was first evaluated on the host CPU and the Hailo-8L NPU. Table 6 summarises the measured results.

Table 6: ANFIS inference latency.

Metric	CPU	Hailo-8L
Mean latency (ms)	1.65	0.62
P95 latency (ms)	2.15	0.83

NPU deployment yields a 2.66× speedup. The 0.62 ms mean latency is well below the 20 ms period of the 50 Hz ANFIS control cycle, confirming that NPU inference satisfies the real-time requirements of the assistance layer.

YOLOX-S performance was then evaluated independently on the same accelerator, as reported in Table 7.

Table 7: YOLOX-S performance on the Hailo-8L.

Metric	Value
NPU utilisation	42.5 %
Throughput	30.0 fps
Per-frame latency	33.3 ms
Queue occupancy	0.50/8 frames

YOLOX-S achieves 30 fps with a per-frame latency of 33.3 ms while using 42.5 % of the available NPU resources, demonstrating real-time onboard visual inference.

The hardware footprint of the ANFIS controller was also examined across the Hailo-8L clusters. The corresponding resource utilisation is reported in Table 8.

Table 8: Hailo-8L cluster utilisation for ANFIS.

Cluster	Control	Compute	Memory
cluster_0	93.8%	23.4%	37.5%
cluster_1	62.5%	15.6%	25.0%
cluster_4	87.5%	21.9%	23.4%
cluster_5	43.8%	10.9%	10.2%
Total	**71.9%**	**18.0%**	**24.0%**

The ANFIS model requires only 18% of the available compute resources and 24% of memory, leaving sufficient accelerator capacity for the visual perception workload.

Finally, ANFIS and YOLOX were executed simultaneously on the Hailo-8L to evaluate the feasibility of the proposed dual-inference architecture. The concurrent execution results are presented in Table 9.

Under concurrent execution, ANFIS maintains the full 50 Hz update rate with an end-to-end latency of 0.9 ms, while YOLOX sustains 28.7 fps at 34.8 ms per frame. Combined NPU utilisation remains at 55%, queue occupancy is limited to approximately 0.8/8 frames, and frame loss remains below 2%. The utilisation figures reported in Tables 7, 8 and 9 are accelerator-level profiling metrics returned by the Hailo runtime and compiler, and are not additive CPU-style workload percentages; the combined 55% therefore does not correspond to the arithmetic sum of the individually measured values. Relative to independent execution, YOLOX throughput decreases by 4.3% (30.0

Table 9: Concurrent ANFIS–YOLOX execution.

Metric	Value
ANFIS update rate	50 Hz
ANFIS end-to-end latency	0.9 ms
YOLOX throughput	28.7 fps
YOLOX latency	34.8 ms
Combined NPU utilisation	55%
Queue occupancy	∼0.8/8
Frame loss	<2%

to 28.7 fps) and its per-frame latency increases by 1.5 ms (33.3 to 34.8 ms), while the ANFIS update rate is preserved. This limited and quantified degradation demonstrates that both control and perception workloads can coexist on a single Hailo-8L while each retains its required real-time operating rate.

6.6 IR Ranging System Validation

The 360° servo-scanned IR ranging system was validated through two experiments. In a cluttered laboratory, the polar point cloud (Fig. 5, left) confirmed multi-directional obstacle mapping. In the circular test, an obstacle maintained at approximately constant radius produced a continuous circular trace (Fig. 5, right), verifying angular consistency across the full scan. These two experiments establish angular consistency and multi-directional coverage; they do not establish ranging accuracy.

6.7 ANFIS–YOLOX Safety Assistance Experiments

During synchronised hardware trials, the operator held a constant forward command ($\delta_{\text{op}} = +1$) while the platform approached obstacles placed at controlled ranges and bearings. The IR ranging system provides d and α, ANFIS produces $\bar{v}_A$, and YOLOX returns s_{obs}. All reported s_{obs} values are confidence scores recorded from the YOLOX inference pipeline during the obstacle-approach experiments; the associated $\bar{v}_A$ values were obtained from the deployed ANFIS model. These trials characterise the response of the assistance layer, not destination-reaching performance.

6.7.1 Frontal and lateral obstacle response.

Table 10 reports the assisted response across nine configurations.

The velocity reduction equals $s_{\text{obs}} \times 100\%$ by construction of Eq. (6). At 18 cm, $\bar{v}_f = 0.73(1 - 0.35) = 0.475$, a 35% reduction. At 7 cm, $s_{\text{obs}} = 0.90$ suppresses 90% of the ANFIS velocity. The lateral cases show that the assistance layer reduces the forward speed envelope without interfering with the operator's turn: for $\alpha = 45°$, $0.61(1 - 0.60) = 0.244$ while $\omega_{\text{cmd}} = \omega_{\text{op}}$ is passed through unchanged; for $\alpha = 135°$, $0.64(1 - 0.60) = 0.256$ under the same condition.

Table 10: ANFIS–YOLOX safety assistance under representative obstacle scenarios.

Scenario	d (cm)	α (°)	$\bar{v}_A$	s_Y	$\bar{v}_f$	**Operator command**	**Assisted response**
Free path	28	90	0.80	0.03	0.776	Forward	Full envelope
Distant	24	90	0.77	0.15	0.655	Forward	Slight slowdown
Medium	18	90	0.73	0.35	0.475	Forward	Moderate slowdown
Approaching	14	90	0.64	0.55	0.288	Forward	Strong slowdown
Close	10	90	0.51	0.75	0.128	Forward	Very low speed
Very close	7	90	0.58	0.90	0.058	Forward	Near stop
Critical	5	90	0.70	0.95	0.035*	Forward	Reverse override
Left obstacle	12	45	0.61	0.60	0.244	Forward + right turn	Slow, turn preserved
Right obstacle	12	135	0.64	0.60	0.256	Forward + left turn	Slow, turn preserved

$\bar{v}_A = v_{\text{ANFIS}}/v_{\max}$; s_Y: YOLOX confidence (Eq. 5). The *operator command* column reports the input $(\delta_{\text{op}}, \omega_{\text{op}})$ logged during the corresponding trial; the assistance layer acts on v_{cmd} only. *Nominal fused value; the critical-distance safety rule overrides the operator's forward command with a reverse manoeuvre.

6.7.2 Critical-distance safety override.

At $d = 5$ cm, direct fusion yields $\bar{v}_f = 0.035$, a residual positive velocity inappropriate under immediate collision risk. The critical-distance rule therefore overrides the arbitrated command and executes the reverse manoeuvre *even while the operator holds a forward command*, enforcing the priority hierarchy: geometric safety condition > ANFIS–YOLOX velocity fusion > operator forward command.

6.7.3 Component-level comparison: ANFIS alone vs. ANFIS+YOLOX.

To isolate the contribution of the visual branch, the ANFIS-only velocity $\bar{v}_A$ is compared with the fused velocity $\bar{v}_f$ for five representative frontal scenarios (Table 11). Under ANFIS-only assistance, the speed envelope is determined entirely by the geometric distance–angle mapping; no visual evidence modulates the output.

Table 11: Component-level comparison: ANFIS-only vs. ANFIS+YOLOX fused velocity envelope for five frontal configurations. This comparison isolates the arithmetic effect of the attenuation term; the system-level ablation across operating conditions is reported in Section 6.8.

Scenario	d **(cm)**	$\bar{v}_A$	s_Y	$\bar{v}_f$	**YOLOX effect**
Free path	28	0.80	0.03	0.776	−3% (negligible)
Medium	18	0.73	0.35	0.475	−35%
Close	10	0.51	0.75	0.128	−75%
Very close	7	0.58	0.90	0.058	−90%
Critical	5	0.70	0.95	Rev.	Override

Two observations are significant. First, when no obstacle is visually detected ($s_{\text{obs}} = 0.03$), the fused envelope differs from the ANFIS-only envelope by only 3%, confirming that YOLOX introduces no unnecessary braking in obstacle-free conditions and therefore does not degrade the operator's experience when assistance is unnecessary.

Second, at close range the ANFIS-only assistance retains substantial forward velocity: $\bar{v}_A = 0.58$ at 7 cm and $\bar{v}_A = 0.70$ at 5 cm, so that the learned surface is non-monotonic in the immediate vicinity of the lower distance bound. This behaviour originates in the training stage rather than in the fusion law, and four contributing factors can be identified. (i) Sample density decreases near the lower boundary $d \to 4\,\text{cm}$, since obstacles could only be placed reliably down to the minimum measurable range of the IR sensor, so the region is comparatively under-represented in the 5000-sample dataset. (ii) The Gaussian membership functions of the distance input overlap strongly at short range, so the *very close* rule never fires in isolation and the inferred output remains a weighted blend of neighbouring rules. (iii) The first-order Sugeno consequents are unconstrained affine functions of (d, α) and may extrapolate upward at the edge of the training domain. (iv) The symmetric MSE objective imposes no monotonicity constraint on $\partial v_{\text{base}}/\partial d$ and applies no asymmetric penalty on high velocity at small distances, so a residual fitting error in this region is not penalised more heavily than elsewhere. The short-range response of the learned surface can therefore not be relied upon as a safety guarantee.

This is why short-range protection is assigned to the critical-distance condition of Eq. (7), evaluated outside both inference graphs (Algorithm 1) and therefore independent of the defect just described; constraining the learned surface to be monotonically non-increasing as d decreases—through monotonic consequent parameterisation, constrained membership-function placement, or an asymmetric training loss—is left to future work.

In the tested configurations the visual branch contributed two effects the geometric mapping alone did not: progressive confidence-dependent braking at intermediate distances, where the critical-distance rule is inactive, and strong suppression at close range (90% at 7 cm, 95% at 5 cm), which reduced how often the override was reached. Since the detector has not been benchmarked on this environment (Section 7.2), these are observations from the tested configurations rather than a general property of the visual branch.

6.8 Comparative Teleoperation Trials

To evaluate the practical effect of the assistance layer, the same operator completed an identical obstacle course under three conditions: (C1) manual teleoperation, (C2) ANFIS-assisted teleoperation, and (C3) ANFIS–YOLOX fusion. Each condition was repeated over 10 runs with randomised condition order.

The evaluation considered collision rate, minimum obstacle clearance, emergency interventions, assistance interventions, stopping distance, command-to-actuation latency, and course completion time. An assistance intervention was defined as a continuous episode during which the safety layer reduced or reversed the operator-requested translational command. Throughout the paper, *assistance intervention*

denotes such an attenuation episode, whereas *override* is reserved for the deterministic critical-distance action of Eq. (7), which forces $v_{\mathrm{cmd}} \leq 0$ irrespective of the operator's command.

The metrics were defined and measured as follows. A *collision* is any physical contact between the platform and a course obstacle or wall, counted once per contact event by the experimenter observing the run. An *emergency intervention* is an occasion on which the operator issued a stop or reverse command specifically to prevent an imminent contact, identified from the synchronised command log and the run video. *Minimum clearance* is the smallest robot-to-obstacle separation observed over a whole run,

$$d_{\min} = \min_{t} d(t), \tag{10}$$

taken as the minimum of the logged frontal d values. *Stopping distance* is a distinct quantity: the longitudinal distance travelled by the platform between the onset of a deceleration—whether operator-commanded or override-triggered at t_0—and the instant t_1 at which forward motion ceased,

$$d_s = x(t_1) - x(t_0), \tag{11}$$

obtained from wheel-command timestamps and the logged velocity profile. The two metrics therefore measure different things: Eq. (10) is a separation and Eq. (11) is a travelled distance, which is why the reported values (10.4 and 11.3 cm under C3) are of comparable magnitude without being redundant. *Command-to-actuation latency* is the interval between the timestamp at which the operator's command was received at the Raspberry Pi and the timestamp at which the corresponding $(v_{\mathrm{cmd}}, \omega_{\mathrm{cmd}})$ frame was written to the Arduino UART, averaged over the commands of a run. It therefore covers the arbitration path, not the YOLOX inference itself: because the visual branch runs asynchronously and the fusion consumes the most recent admissible confidence (Eq. (8)), the 34.8 ms detector latency does not appear additively in this figure, which is why condition C3 exceeds C2 by only 4.4 ms. *Course completion time* is the interval between the first non-zero command and the moment the platform crossed the end of the course.

Table 12: Experimental teleoperation performance over 10 runs per condition.

Metric	**C1: Manual**	**C2: ANFIS**	**C3: Fusion**
Collisions (total / 10 runs)	4	2	1
Collisions/run	0.40	0.20	0.10
Minimum clearance (cm)	5.1	7.8	10.4
Emergency interventions (total / 10 runs)	17	8	2
Emergency interventions/run	1.70	0.80	0.20
Assistance interventions/run	N/A	6.3	9.2
Stopping distance (cm)	21.6	15.4	11.3
Command-to-actuation latency (ms)	42.3	44.1	48.5
Course completion time (s)	54.8	60.7	65.9

The fused architecture achieved the best observed safety performance. Compared with manual teleoperation, the collision rate decreased by 75%, minimum clearance increased from 5.1 to 10.4 cm, and emergency interventions decreased from 1.70 to 0.20 per run. Stopping distance was also reduced from 21.6 to 11.3 cm. These improvements were obtained at the cost of a moderate increase in command-to-actuation latency (+6.2 ms) and course completion time (+20.3%), reflecting the more conservative velocity profile imposed by the assistance layer. Values are reported as per-condition means over the 10 runs. These experiments are intended as a proof-of-concept comparison: given the limited number of trials and the use of a single operator, the reported differences should be interpreted as observed experimental trends rather than statistically generalisable effects. With the three conditions applied to the same course by the same operator, the runs form matched groups, so a Friedman test followed by pairwise post-hoc comparisons would be the appropriate analysis once the sample is enlarged; a Kruskal–Wallis test is the conservative alternative if the runs are treated as independent. Neither is applied here, since with $n = 10$ per condition the outcome would be dominated by the sample size rather than by the effect.

7 Discussion

7.1 Advantages of the Proposed Architecture

Co-deploying ANFIS and YOLOX on the same Hailo-8L NPU unifies perception and velocity regulation within a single accelerator. The measured concurrent utilisation of 55% (50 Hz ANFIS + 28.7 fps YOLOX) indicates remaining accelerator capacity, although the feasibility of hosting an additional workload in that margin would require separate resource mapping and runtime profiling: deployment is constrained by memory, cluster mapping, scheduling, and bandwidth as well as by aggregate utilisation.

The multiplicative fusion law (6) provides a computationally inexpensive mechanism for combining heterogeneous modalities. Unlike additive fusion, which requires careful weight calibration, the multiplicative formulation progressively attenuates velocity as visual obstacle confidence increases. Importantly, the system enforces a three-level priority hierarchy: (i) the critical-distance geometric safety rule takes absolute precedence, (ii) the ANFIS–YOLOX fusion modulates the speed envelope at non-critical distances, and (iii) the base ANFIS envelope applies when no visual obstacle is detected. This layered structure may also provide complementary information in situations where the geometric sensor response is unreliable, provided that the obstacle concerned remains detectable by the visual branch, since a high visual confidence can reduce the velocity even when the geometric branch does not register an obstacle. This complementary regime was not evaluated experimentally in the present study.

The comparative trials of Section 6.8 bear on the question that component-level inference metrics cannot answer: whether the architecture changes teleoperation behaviour. Three features of Table 12 are worth interpreting rather than restating. First, every safety-related metric improves and none degrades, so the assistance layer is not trading one hazard for another; the whole cost falls on throughput. Second, the ANFIS-only condition lies between manual operation and the fused configuration on

each metric, which indicates that the visual branch contributes assistance beyond geometric velocity regulation alone—though, given the sample size and the multiplicative coupling between $\bar{v}_A$ and s_{obs}, no quantitative apportionment between the branches is attempted. Third, the cost is of a kind that an assistive deployment can absorb: the layer buys clearance and collision-rate reduction with traversal time, which is the appropriate exchange when the operator is remote and the environment is shared with people. What distinguishes this from autonomous obstacle avoidance is that the robot never chooses a different route from the one commanded; it only declines to travel the commanded route faster than the sensed configuration permits.

The servo-scanned IR ranging system represents a pragmatic cost compromise: the two IR sensors and the scanning servo account for a small fraction of the platform cost while providing range and bearing information sufficient for short-range indoor operation, at the expense of the angular resolution and range envelope that a commercial scanning LiDAR would offer.

7.2 Limitations

Several limitations should be acknowledged. First, the ANFIS model was trained offline on 5000 samples under controlled laboratory conditions; out-of-distribution performance has not been evaluated, and the learned surface is non-monotonic near the lower distance bound (Section 6.7), which is why short-range safety is delegated to the independent geometric override rather than to the learned envelope. Second, the IR ranging system's angular resolution is limited by the servo step size, yielding a sparser point cloud than commercial LiDAR. Third, INT8 quantisation used only 64 calibration samples, below the recommended 1024; a larger set could improve fidelity. Fourth, the YOLOX-S detector is used as a pretrained component and its detection accuracy was not independently benchmarked on the deployment environment: no precision, recall, or mAP figures are reported for the laboratory scenes. Because s_{obs} directly modulates the admissible velocity, a false positive causes unnecessary slowing while a missed detection removes the visual attenuation, leaving the geometric branch and the critical-distance override as the only active safety mechanisms. Constructing an annotated onboard test set (of the order of 200–500 frames) and reporting precision, recall, and mAP@0.5 is the most direct way to strengthen the visual branch and is planned as an immediate extension. Fifth, the visual term is a confidence score rather than a risk estimate (Eq. (5)). Sixth, the assistance layer acts on the immediate vicinity of the robot only. The absence of SLAM and global path planning is a design choice rather than a shortcoming: route selection is the operator's responsibility in the proposed shared-control scheme. It does, however, bound the system's applicability to teleoperated operation and precludes autonomous recovery if the communication link is lost. Seventh, all reported trials involved a single operator and 10 runs per condition; inter-operator variability in command style, its interaction with the assistance layer, and the statistical significance of the reported differences have not been characterised. Eighth, the quaternion-based EKF supports platform monitoring and operator feedback but does not participate in the fusion law; its inclusion documents the embedded integration of the platform rather than contributing to the safety mechanism itself. Ninth, both sensing branches are stale by a bounded amount (Algorithm 1), and the

effect of that staleness under faster motion or moving obstacles has not been characterised; all trials used static obstacles. Tenth, the override reverses without consulting the rear sector (Section 4.4), so the architecture protects against frontal contact only. Eleventh, the ranging accuracy of the IR system was not quantified against reference distances, even though d enters the control law directly and $d_{\text{crit}} = 5\,\text{cm}$ lies only 1 cm above the sensor's lower measurement bound—so the effective trigger distance of the override is known only to within the unmeasured ranging error at short range. This is the measurement gap with the most direct bearing on the safety claim.

8 Conclusion and Future Work

This paper has presented an NPU-accelerated safety assistance architecture for teleoperated mobile robots, integrating an ANFIS velocity controller, a YOLOX visual detector, and a multiplicative decision-level multimodal fusion law on a single Hailo-8L NPU. Both inference workloads execute concurrently at 50 Hz (ANFIS) and 28.7 fps (YOLOX) with 55% combined NPU utilisation, demonstrating that control and perception can share a single edge accelerator with only minor degradation under concurrent execution: YOLOX throughput falls from 30.0 to 28.7 fps and its per-frame latency rises from 33.3 to 34.8 ms, while the ANFIS update rate is fully preserved.

In comparative teleoperation trials the fused configuration improved every safety-related metric relative to manual operation—most notably a four-fold reduction in collision rate and a doubling of minimum obstacle clearance (Table 12)—at the cost of an approximately 20% longer course completion time.

The complete system—from attitude estimation through neuro-fuzzy velocity regulation and visual perception to motor actuation— executes entirely at the edge on a platform assembled from commodity single-board computers, a consumer NPU module, and low-cost proximity sensors. The compact form factor and fully edge-based execution are compatible with the constraints of low-cost assistive platforms. All reported trials were laboratory obstacle-course runs with a single operator and static obstacles, so elderly-care operation remains a target application rather than a demonstrated capability.

Future work follows directly from the limitations above. The first priority is to quantify the IR ranging error at reference distances bracketing d_{crit} and to reposition the threshold if the measurement warrants it, since the deterministic override rests on that figure. The remaining priorities are to make the override rear-aware, either by gating the reverse on a measured rear clearance or by reducing it to an unconditional stop; to benchmark the pretrained visual branch on an annotated onboard test set (precision, recall, mAP@0.5); to impose monotonicity constraints on the learned ANFIS response at short range; and to evaluate the architecture across multiple operators with an enlarged trial count and repeated-measures statistical analysis.

Declarations

Funding. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of interest. The authors declare no competing interests.

ORCID iDs. S. Ben Halima: https://orcid.org/0009-0002-2195-8800.
F. Ben Abdallah: https://orcid.org/0000-0002-2449-2199.
J. Haggège: https://orcid.org/0000-0002-6068-0568.

Author contributions. S. Ben Halima: Conceptualisation, Methodology, Software, Investigation, Formal analysis, Validation, Visualisation, Writing . **F. Ben Abdallah**: Methodology, Validation, Supervision, Writing review & editing. **J. Haggège**: Supervision, Project administration, Writing review & editing.

References

[1] Grewal, M.S., Andrews, A.P.: Kalman Filtering: Theory and Practice Using MATLAB, 3rd edn. Wiley-IEEE Press, Hoboken (2008)

[2] Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965). https://doi.org/10.1016/S0019-9958(65)90241-X

[3] Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Upper Saddle River (1997)

[4] Borne, P., Benrejeb, M., Haggège, J.: Les Réseaux de Neurones: Présentation et Applications. Editions Technip, Paris (2007)

[5] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding YOLO Series in 2021. arXiv preprint arXiv:2107.08430 (2021)

[6] Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. **23**(3), 665–685 (1993). https://doi.org/10.1109/21.256541

[7] Hailo Technologies Ltd.: Hailo-8L™ AI Accelerator Datasheet. Hailo, Tel Aviv (2024). https://hailo.ai/products/ai-accelerators/hailo-8l-ai-accelerator/

[8] Hailo Technologies Ltd.: Hailo Dataflow Compiler User Guide, v3.33.1. Hailo, Tel Aviv (2025)

[9] Ben Halima, S., Ben Abdallah, F., Haggège, J.: Edge AI-based object detection via voice recognition with an LLM-based emotional assistant for elderly care robots. Int. J. Inf. Technol. Comput. Sci. (IJITCS) **18**(4), 50–67 (2026). https://doi.org/10.5815/ijitcs.2026.04.04

[10] Ben Halima, S., Ben Abdallah, F., Haggège, J.: Deep reinforcement learning-based obstacle avoidance for embedded mobile robot navigation. In: Proceedings of the 2026 IEEE International Conference on Advanced Systems and Emergent

Technologies (IC_ASET), Hammamet, Tunisia, 26–28 March 2026. https://doi.org/10.1109/IC_ASET69920.2026.11502382

[11] Omrane, H., Masmoudi, M.S., Masmoudi, M.: Fuzzy logic based control for autonomous mobile robot navigation. Comput. Intell. Neurosci. **2016**, 9548482 (2016). https://doi.org/10.1155/2016/9548482

[12] Mohanty, P.K.: Path planning of mobile robots under uncertain navigation environments using FCM clustering ANFIS. Wirel. Pers. Commun. **137**(2), 1251–1276 (2024). https://doi.org/10.1007/s11277-024-11437-0

[13] Soares, E., Angelov, P., Gu, X.: Evolving fuzzy logic systems for creative personalized socially assistive robots. Eng. Appl. Artif. Intell. **117**, 105432 (2022). https://doi.org/10.1016/j.engappai.2022.105432

[14] Mavrogiorgos, V., Kouliaridis, V., Kambourakis, G.: Obstacle avoidance technique for mobile robots at autonomous human-robot collaborative warehouse environments. Sensors **25**(8), 2387 (2025). https://doi.org/10.3390/s25082387

[15] Eirale, A., Martini, M., Chiaberge, M.: Human-centered navigation and person-following with omnidirectional robot for indoor assistance and monitoring. Robotics **11**(5), 108 (2022). https://doi.org/10.3390/robotics11050108

[16] Loganathan, A., Ahmad, N.S.: A systematic review on recent advances in autonomous mobile robot navigation. Eng. Sci. Technol. Int. J. **40**, 101343 (2023). https://doi.org/10.1016/j.jestch.2023.101343

[17] Dafinca, M., et al.: Edge AI in practice: a survey and deployment framework for neural networks on embedded systems. Electronics **14**(24), 4877 (2025). https://doi.org/10.3390/electronics14244877

[18] Tahir, N., et al.: Edge computing and its application in robotics: a survey. J. Sens. Actuator Netw. **14**(4), 65 (2025). https://doi.org/10.3390/jsan14040065

[19] Portugal, D., Alvito, P., Christodoulou, E., Samaras, G., Dias, J.: A study on the deployment of a service robot in an elderly care center. Int. J. Soc. Robot. **11**(2), 317–341 (2019). https://doi.org/10.1007/s12369-018-0492-5

[20] Vishnu, T.P., Ray, D., Thiyagarajan, K., Chowdhury, A.R.: Multi-modal sensing for socially compliant safe robot navigation in human-aware indoor environments. IEEE Sens. J. **25**(21), 40428–40439 (2025). https://doi.org/10.1109/JSEN.2025.3522186

[21] Qi, H., et al.: A review of sensing technologies for indoor autonomous mobile robots. Sensors **24**(4), 1222 (2024). https://doi.org/10.3390/s24041222