CS-067-v1.0
2025/03/31
PU - Public

HORIZON
EUROPE

D3.5
Risk awareness module

Jan Zahálka (CVUT)

CORESENSE
Horizon Europe Project #101070254
HORIZON-CL4-2021-DIGITAL-EMERGING-01-11

CORESENSE

CORESENSE

The **CORESENSE Project** is a four year research project focused in the development of a *technology of machine understanding* for the construction of dependable autonomous systems. It is a project funded by the European Union.

Programme:	Horizon Europe
Project number:	101070254
Project name:	CoreSense: A Hybrid Cognitive Architecture for Deep Understanding
Project acronym:	CORESENSE
Topic:	HORIZON-CL4-2021-DIGITAL-EMERGING-01-11
Type of action:	HORIZON Research and Innovation Actions
Granting authority:	European Commission-EU
Project start date:	1 October 2022
Project end date:	30 September 2026
Project duration:	48 months

Most of CORESENSE results are open science but IPR regulations may still hold. Requests for permission to reproduce this document or to prepare derivative works of this document should be addressed to the CORESENSE Consortium IPR management.

UNIVERSIDAD POLITECNICA DE MADRID (UPM)
TECHNISCHE UNIVERSITEIT DELFT (TUD)
FRAUNHOFER GESELLSCHAFT ZUR FORDERUNG DER ANGEWANDTEN FORSCHUNG (FHG)
UNIVERSIDAD REY JUAN CARLOS (URJC)
PAL ROBOTICS SL (PAL)
IRISH MANUFACTURING RESEARCH COMPANY LIMITED BY GUARANTEE (IMR)
CZECH TECHNICAL UNIVERSITY IN PRAGUE (CVUT)

For information about other CORESENSE products, reports or articles, please visit our Web site:

http://www.coresense.eu/

NO WARRANTY — Please read the fine print:

THIS CORESENSE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. THE CORESENSE CONSORTIUM NOR ANY OF THE INDIVIDUAL CORESENSE PARTNERS OR RESEARCHERS MAKE NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, ACCURACY, COMPLETENESS, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CORESENSE DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Funded by the European Union

The CORESENSE project is funded by the EC Horizon Europe programme though grant HE #101070254 inside the HORIZON-CL4-2021-DIGITAL-EMERGING-01-11 topic. Views and opinions expressed in this document are however those of the author(s) only and do not necessarily reflect those of the European Union or the Horizon Europe Programme. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

Title: D3.5 Risk awareness module

Subtitle:

Author: Jan Zahálka (CVUT)

Reference: CS-067 v 1.0

Date: 2025/03/31

Dissemination level: PU - Public

URL: http://www.coresense.eu/doc/CS-067.pdf

Executive Summary

This deliverable presents the concept, design, and initial implementation of the *CoreSense Risk Awareness Module (RiskAM)*, a general framework for risk awareness in robotics. RiskAM is motivated by the growing demand for safe robotics behavior, especially in dynamic, open-world and/or human-centric environments. The document introduces a comprehensive architecture structured into two complementary stages: *offline risk mitigation*, responsible for preparatory safety measures and system validation; and *online risk awareness*, which operates in real time to assess the situational risk during robot deployment.

To make the RiskAM framework concrete, the deliverable presents a prototype implementation for a flagship risk awareness task: visual navigation in human-robot environments, with a focus on assessing risk to nearby humans. The prototype uses RGB camera input—the most universally available modality in robotic platforms—to compute a scalar risk score for each video frame. This risk is derived from three interpretable components: human *proximity*, *gaze* direction (as a proxy for awareness), and x*-position* (alignment with the robot's motion path). The output risk score is mapped to one of four discrete brackets to facilitate integration with downstream control systems.

The RiskAM prototype is evaluated on a real-world dataset, CS-RoboCup23, collected by the CoreSense Social Testbed. The evaluation demonstrates strong performance across the five critical characteristics of risk awareness: *accuracy* (95.3% of predictions support safe behavior), *timeliness* (real-time inference with 9–10 FPS on modest hardware), *semantic richness* (interpretable outputs via visual component breakdown), *robustness* (errors tend to overestimate risk, rather than underestimate it), and *adaptivity* (general operation without task- or input-specific reconfiguration).

The deliverable further identifies key areas for future development, including evaluation across diverse datasets, active parameter tuning, optional integration of multimodal inputs, expansion of risk components, modeling of temporal dynamics, and refinement of risk-level thresholds. Long-term directions include generalizing RiskAM to a broader range of robotic tasks and moving toward a unified pipeline that integrates both online risk estimation and offline mitigation.

In summary, RiskAM represents a modular, interpretable approach to real-time risk awareness

in robotics. The prototype establishes a viable foundation for future research and system development, and offers a concrete step toward equipping autonomous systems with the capacity to reason about and respond to situational risk in complex scenarios.

We acknowledge the financial support through the contract #101070254, *CoreSense: A Hybrid Cognitive Architecture for Deep Understanding* granted by the European Commission inside the Horizon Europe programme.

Document Versions

Version	**0.1** | **2024/03/31**
Changes	Concept document
Authors	J. Zahálka (CVUT)
Version	**0.5** | **2024/12/10**
Changes	First draft with early implementation
Authors	J. Zahálka (CVUT)
Version	**1.0** | **2025/03/31**
Changes	Complete RiskAM prototype, deliverable D3.5
Authors	J. Zahálka (CVUT)

Contents

1 Name

CoreSense Risk Awareness Module (RiskAM)

2 Context

In its current form, RiskAM is tailored to support robotics tasks that involve **visual navigation** in **human-robot environments**. Within the CoreSense project, this makes RiskAM particularly well suited for deployment in the Social Robots and Manufacturing testbeds, where human presence and dynamic interaction are central concerns. However, the architecture of RiskAM is intentionally **extensible**: its modular design facilitates the addition of new tasks, risk components, and sensory modalities, enabling broader applicability across diverse robotic platforms and operational domains.

3 Problem

3.1 Hazards and risks

In virtually all use cases, robots encounter **hazards**—dangerous situations, conditions, or circumstances that may disrupt their operations or cause damage. The consequences of such disruptions vary widely, ranging from minor delays in task completion and financial costs to severe injuries or even fatalities if a human is struck. To mitigate these risks, roboticists and robotics stakeholders must strive to minimize **risk**: the likelihood of hazardous operational disruptions.

Risk management and mitigation is a well-established discipline in robotics. A time-tested approach to risk management relies on physical barriers combined with rigorous safety protocols such as ISO 10218 [9]. *Physical barriers* such as walls, cages, or light curtains ensure that the robot remains confined within its designated operating environment. *Safety protocols* define safe operational scenarios and specify the necessary precautions to minimize hazards. Together, physical barriers and safety protocols can effectively *eliminate* risks in structured, controlled environments such as manufacturing. However, as robotics expands into increasingly open environments and complex human-robot interactions, controlling the robot's workspace solely through physical barriers and predefined safety protocols becomes infeasible.

3.2 Risk awareness

The solution is to make the robots themselves **risk-aware**: capable of perceiving and assessing the risks associated with their current state. This assessment then informs decision-making, guiding the robot to anticipate and mitigate potential hazards. In an ideal scenario, a perfectly risk-aware robot would possess a complete model of all possible hazards and respond accordingly, allowing it to safely navigate open environments without external constraints [3, 14]. However, in practice, risk awareness must work in conjunction with other cognitive understanding capabilities, motion planning, and traditional safety measures [9]. The interplay of these approaches ensures both adaptivity and reliability in complex human-robot interactions.

Since perfection is an elusive goal, we decompose the concept of risk awareness quality into five key characteristics:

- *Accuracy* — The risk assessment should faithfully reflect the actual hazards the robot is encountering and provide actionable insights to guide the robot's behavior effectively.

- *Timeliness* — The risk assessment must be delivered early enough for the robot to take successful hazard avoidance or mitigation actions.

- *Adaptivity* — The risk awareness model should generalize to different robots performing similar tasks, minimizing the need for costly reconfiguration or retraining.

- *Semantic richness* — The risk awareness process should incorporate meaningful, interpretable information so that stakeholders can understand and trust the system's decision-making (explainability, transparency).

- *Robustness* — The risk awareness system must maintain reliable performance across diverse environments, sensing conditions, and unforeseen scenarios, ensuring safety even under uncertainty.

Accuracy and *timeliness* are not difficult to motivate; they are the basic two requirements for any risk awareness model to be useful in the first place. Obviously, we need an accurate assessment, and we need it in time, otherwise it is useless in practice. It is important to note that accuracy and timeliness are in a direct trade-off. Improving accuracy generally entails making the model more complex, which in turn increases its run time, negatively impacting timeliness. Risk awareness models must bear this trade-off in mind.

Adaptivity is motivated chiefly by the fact that an open world is inherently dynamic. Rather than relying on highly specialized models tailored to a single workspace or configuration, a risk awareness model should be quickly adaptable to new circumstances and setups with only moderate calibration. This flexibility also has economic benefits, which in turn enhance overall safety—if a risk awareness model is easy to implement, stakeholders are less likely to forgo its deployment due to cost or complexity. Moreover, adaptivity contributes to a deeper understanding of risk. By necessity, adaptive models must possess a strong "core" that captures setup-agnostic knowledge relevant to risk awareness for a given task or even an entire class of tasks. The adaptive, calibrated component then refines this general knowledge to account for setup-specific variations, ensuring transferability.

Deeper understanding also motivates the need for *semantic richness*. To improve risk awareness, we must not only enhance the robot's ability to assess hazards but also ensure that we understand what factors contribute to its decision-making process. This transparency is essential for debugging, optimizing performance, and building trust in robotic systems. Moreover, there is a strong legal motivation: human society ultimately defines safety standards, and individual stakeholders are responsible for ensuring compliance. For them to fulfill this role, they must be able to interpret how the robot assesses and mitigates risks.

Finally, the *robustness* requirement stipulates that a risk awareness module—crucial for the overall robustness of the robot—must itself be resilient to variability and uncertainty. In open-world environments, the robot may encounter both foreseen and unforeseen hazards, including sensor noise, dynamic obstacles, and adversarial conditions. It is imperative that the risk awareness model can gracefully handle these situations, ensuring reliable risk assessments even under imperfect conditions. This requirement introduces significant challenges. Enhancing robustness typically increases computational complexity, which in turn impacts *timeliness* by delaying hazard response times. Additionally, as seen in AI security, robustness often comes at the cost of *accuracy*, as models designed to withstand perturbations may sacrifice precision in favor of generalization [21]. Achieving the right balance requires careful architectural design, including uncertainty-aware models, robust training methodologies, and computationally efficient risk assessment strategies.

3.3 Related work

To the best of our knowledge, as of early 2025, no comprehensive risk awareness framework has been established. While risk mitigation is widely recognized as a critical challenge for robotics adoption in open-world human-robot collaboration scenarios, risk awareness itself remains a relatively niche topic. To illustrate this point: only 0.65% of accepted papers at the robotics conferences ICRA and IROS between 2020 and 2024 contain the word “risk” in their title. Despite its limited prominence, several works have contributed to advancing risk awareness in robotics. In this section, we review these approaches, analyze the robotics tasks they address, and assess how they align with the five risk awareness quality properties defined in Section 3.2.

Risk awareness revolves around *risk modeling*. Since risk cannot be measured exactly—doing so would require a perfect model of the environment, including all possible hazards—the best we can achieve is *estimation*. The dominant approach in recent literature was introduced by Majumdar and Pavone [13], where risk is quantified in terms of the actual monetary costs of robot failure due to a given hazard. As a risk measure supported by mathematical axioms, Majumdar and Pavone propose *conditional value at risk (CVaR)*. CVaR has been successfully applied in various domains, particularly in navigation.

Fan et al. incorporate CVaR into stochastic planning [3]. Cai et al. combine CVaR with premapping involving self-supervised data collection and training into a framework modeling aleatoric (inherent) and epistemic (out-of-distribution at test-time) uncertainty [2]. An earlier work by Hakobyan and Yang applies CVaR to prevent collisions with randomly moving obstacles [7]. Beyond CVaR-based approaches, some works tie risk modeling directly to specific tasks. For instance, Randriamiarintsoa et al. define mobile navigation risk as the maximum potential energy absorbed by the robot’s wheels in a collision [14].

Another approach is to *learn* risks rather than defining an explicit risk function. For instance, the aforementioned work by Cai et al. [2], which uses CVaR as well, leverages learning to assess the traversability of different terrains. Similarly, Schneider et al. employ distributional reinforcement learning to achieve risk-aware quadrupedal locomotion [18]. In human-robot interaction settings, Sun et al. use deep reinforcement learning for risk-aware navigation in human crowds, allowing the robot to adapt to pedestrian movement patterns [19]. Beyond reinforcement learning, Liu et al. integrate neural radiance fields (NeRF) with 3D Gaussian splatting for risk-aware environment masking, effectively “disabling” hazardous or problematic regions in the robot’s operational space [11]. This approach allows the robot to leverage implicit 3D scene understanding to improve safety and navigation decisions.

From the standpoint of the five risk awareness quality characteristics, all related work methods experimentally evaluate their *accuracy* and *timeliness*. However, they exhibit limitations or remain uncertain with respect to the other characteristics. Firstly, they are not *adaptive*. Approaches based on explicit risk functions often require detailed mapping of the robot’s operational environment, with predefined risk values assigned to specific regions. Meanwhile, learning-based methods require training complex models for a single task configuration, making generalization to new environments or tasks non-trivial. Secondly, they lack *semantic* interpretability. While some methods offer visualization possibilities, most approaches do not provide explicit explanations of their internal risk assessment mechanisms, making them difficult to interpret for stakeholders. Finally, *robustness* remains an open question. While these approaches clearly improve the overall robustness of robotic systems by enhancing risk awareness, their behavior in truly unexpected scenarios is often unclear. The extent to which they can generalize to out-of-distribution hazards or function reliably under sensor degradation and adversarial conditions remains largely untested in the reviewed works.

4 Idea

In this chapter, we introduce the **CoreSense Risk Awareness Module (RiskAM)**, designed to address the risk awareness challenges outlined in Chapter 3. Risk awareness in robotics is a complex, open research area; therefore the development of robust, generalizable solutions that solves the risk awareness problem demands substantial collaborative effort from the wider robotics community. In this chapter, we present the conceptual framework of RiskAM in its entirety. While the full realization of this framework remains an ambitious objective, we argue that establishing such a framework is essential for guiding future research and development in this domain. The initial prototype, presented in Chapter 5 of this deliverable, can be regarded as a tangible instantiation of this vision.

4.1 RiskAM structure

We decompose the RiskAM framework into two *stages* separating preparatory measures from runtime operation:

- *Offline stage* — This stage encompasses all risk mitigation procedures performed during the system setup and deployment phase, prior to actual robot operation.
- *Online stage* — This stage refers to the runtime component of RiskAM, which continuously monitors and evaluates the current risk state based on live sensor data and contextual information.

While the core notion of *awareness*—the ability to perceive and interpret risk in context—is realized during the online phase, the offline stage plays an equally critical role in the overall RiskAM framework. It focuses on *mitigation*: proactively reducing expected risk. It is difficult to understand or implement risk awareness in isolation from these preparatory offline processes. For this reason, we consider both stages as integral and interdependent elements of the RiskAM architecture.

4.2 Offline stage: Risk mitigation

The offline stage of the RiskAM framework encompasses all preparatory activities aimed at reducing the likelihood and severity of risk during robot operation. This includes the definition and enforcement of safety regulations, adherence to relevant operational standards, and implementation of domain-specific safety protocols. Where applicable, additional steps should be taken to physically secure the environment, constrain operational boundaries, and reduce exposure to known hazards.

A cornerstone of the offline stage is the explicit formulation of the risk tolerance acceptable for the robotic system in its given context. In real-world deployments, especially in dynamic or unstructured environments, it is infeasible to demand perfect reliability or absolute safety guarantees. Instead, roboticists and robotics shareholders should have a clear idea about risk thresholds, taking into account the potential impact of failure events. This enables meaningful risk mitigation strategies and facilitates informed decision-making.

When AI/ML components such as perception or decision-making models are integrated into the system, additional safeguards must be introduced during the offline phase. AI-based modules are inherently vulnerable to adversarial manipulation, distributional shift, and model degradation over time. These vulnerabilities can manipulate [5, 12] or damage [6, 20] the robot's cognitive reliability and lead to unsafe behavior if left unchecked. To address this, risk-aware development practices should include robustness testing, adversarial training, and validation under diverse operational conditions [8, 17].

The offline stage also plays a critical role in optimizing the performance of the online risk awareness module. By frontloading computationally intensive tasks, the system reduces the computational burden during runtime. This is particularly important for RiskAM, which must operate under tight latency constraints to provide timely and context-aware assessments of emerging risk. Delayed risk estimates undermine the robot's ability to react effectively, compromising both safety and task performance. Reducing online computational load further enables the risk awareness module to align with other safety-critical system components that operate under strict timing constraints, such as control loops, actuation deadlines, and real-time monitoring frameworks. Ensuring that the risk estimation remains accurate and temporally synchronized with the rest of the robot control stack is essential for cohesive and safe system behavior.

In summary, the offline stage establishes the foundation for a successful online phase. It defines acceptable risk thresholds, secures the physical and algorithmic components of the system, and enables the runtime component to remain responsive, efficient, and compatible with the overall safety architecture of the robot.

4.3 Online stage: Risk awareness

In Chapter 3, we introduced five key characteristics that define effective risk awareness in robotic systems: *accuracy*, *timeliness*, *adaptivity*, *semantic richness*, and *robustness*. This section discusses how each of these properties should be addressed in the design and implementation of the online component of RiskAM to achieve reliable risk awareness.

In principle, an ideal risk awareness module would excel in all five characteristics. However, in practice, trade-offs must be made due to resource constraints and the complexity of real-world environments. For realistic system development, we categorize these characteristics into two types: *optimizing* criteria, where performance should be maximized, but there is not necessarily a minimum threshold; and *satisficing* criteria, where a minimum threshold must be reliably met for the system to be considered functional and safe. In the remainder of this section, we examine each characteristic and discuss its categorization, practical implications, and implementation strategies within the online stage of RiskAM.

4.3.1 Accuracy

Accuracy is intentionally listed first among the core characteristics of risk awareness, as it serves as a necessary requirement for any practical utility. An inaccurate risk estimate—regardless of how timely, adaptive, semantically rich, or robust it may be—can lead to inappropriate or even dangerous decisions. Uniquely among the five characteristics, *accuracy* is simultaneously both an *optimizing* and a *satisficing* criterion. On one hand, we seek to maximize the precision of risk estimates as much as possible. On the other hand, there exists a critical accuracy threshold below which the risk awareness module becomes functionally unusable.

The accuracy of risk awareness depends on how effectively the module translates its *inputs* into *outputs* that are both meaningful and reliable for hazard-avoidance decisions. Inputs are inherently task-dependent, ranging from sensory data and internal state to images or videos. While outputs may also vary slightly by application, the overarching objective remains the same: the output must support swift and unambiguous decision-making under uncertainty. To support this, we propose a default risk function $r : I \rightarrow R$ that maps an input $i \in I$ from the task-dependent input space I to a real value in the interval $[0, 1]$. Here, 0 represents no perceived risk, and 1 represents maximal risk requiring immediate action to avoid potential harm. This scalar formulation avoids the complications of multicriteria optimization, offering a compact and interpretable signal for downstream decision modules. The $[0, 1]$ range also has intuitive semantic meaning, aligning well with human understanding, e.g., percentages or probability-like interpretations. While we do not claim this is the only viable representation, it offers a practical and extensible baseline that does not preclude storing or computing additional structured outputs for diagnostic or post-hoc analysis.

4.3.2 Timeliness

Timeliness is a strict *satisficing* criterion: if the risk signal arrives too late, it is as if it did not arrive at all. This imposes a hard computational constraint on the entire perception–inference–decision pipeline. There must be a sufficient time window bounded by the existing planning and control loops for the RiskAM to operate in the first place. This depends not only on the module's own efficiency but also on the overall responsiveness of the robot's control architecture and the pre-optimization done in the offline stage.

The module itself must be designed to operate within strict latency bounds, even when risk estimation involves complex, high-dimensional inputs. This demands careful trade-offs in system design and thorough timing evaluations. Evaluation should go beyond average-case latency, incorporating worst-case response time analysis and stress testing under peak load. For critical applications, formal timing guarantees or integration with real-time operating systems (RTOS) may be necessary to ensure predictable behavior under all conditions. In AI/ML-based implementations, efficiency becomes particularly critical: many state-of-the-art models prioritize accuracy or robustness, often at the expense of inference speed. For risk awareness, lightweight architectures or specialized runtime optimizations should be preferred to ensure that decision-critical signals are available exactly when needed.

4.3.3 Adaptivity

Adaptivity is an *optimizing* metric that reflects the ability of RiskAM to generalize across different robotic platforms, tasks, and environments. The long-term objective is to minimize the number of highly contextual submodules, which are often tightly coupled to specific hardware, workspace layouts, or narrowly defined operational scenarios. Instead, we aim for general submodules that can be reused across tasks with minimal tuning or retraining.

A realistic long-term target is generalization at the task level where a single submodule supports instances of a given robotics task, such as navigation or manipulation, across diverse deployments. Achieving this likely requires aggregating insights from many task-specific deployments under varying conditions and integrating them into more general, transferable components.

Among the five characteristics, adaptivity occupies a special position: it is the only one that describes RiskAM's quality *across* different robotics setups. The other four are evaluated *within* a single setup. Adaptivity therefore reflects RiskAM's long-term scalability and practical utility in the broader robotics landscape.

4.3.4 Semantic richness

Semantic richness is an *optimizing* criterion that enhances the interpretability and transparency of RiskAM's outputs. While accuracy focuses on producing a simple, reliable estimate to guide decisions, semantic richness provides contextual information that helps understand *why* a particular risk assessment was made. This added depth is instrumental in building operator trust, supporting debugging, and enabling informed human oversight.

Semantic richness can be achieved by designing RiskAM from components that produce semantically meaningful intermediate outputs. Modern foundation models, such as large language models (LLMs), offer the possibility of generating high-level semantic interpretations from low-level robotics data. Although their inference time is often too high for real-time use, they are well suited for post hoc analysis and operator-facing reporting. This allows the system to retain its real-time responsiveness while benefiting from the interpretive power of semantically aware models.

4.3.5 Robustness

Robustness is a *satisficing* metric that ensures RiskAM remains reliable under degraded or unforeseen conditions. These may include sensor noise, partial observations, distributional shifts, or transient failures in upstream modules. If the risk awareness module becomes unstable or erratic under such perturbations, it can no longer be trusted to support safe decision-making.

Robustness can be improved through several complementary strategies. These include sensor redundancy (e.g., combining vision with depth or proprioception), conservative design principles such as fail-safe defaults and early warnings, and the integration of confidence estimation or uncertainty-aware models. RiskAM should also be tested under diverse and adversarial conditions, including edge cases and failure modes, to identify points of fragility. Incorporating data from these scenarios during retraining can further enhance robustness. Systems should include fallback mechanisms, such as escalation to human oversight or temporary shutdown, when confidence in risk estimation drops below an acceptable threshold. These safeguards help ensure that even in uncertain or unstable situations, the robot can default to safe behavior.

5 Solution

In this chapter, we describe the first RiskAM prototype. In Section 5.1, we motivate the flagship task we chose for the prototype. Section 5.2 describes the method. Section 5.3 presents a showcase of the RiskAM prototype on a real dataset obtained in the CoreSense Social robots testbed.

5.1 Prototype task

The diversity of robotic platforms, their assigned tasks, sensory inputs, operational capabilities, and environmental contexts precludes the development of a universal, one-size-fits-all RiskAM at the outset. Instead, we propose a bottom-up development strategy. This involves progressively covering the risk awareness space by targeting specific, high-priority subproblems that are representative and practically relevant. This naturally raises the following question: *Which task should be addressed first?*

We focused our prototype development on **visual navigation in human-robot environments**, namely on **risks posed to humans**. This scenario represents a high-priority, flagship task, as it involves key conceptual and technical challenges with valuable lessons for subsequent expansions of RiskAM. *Visual navigation* is a core task for open-world robotics, the next frontier in the deployment of autonomous systems. It is inherently complex, as it depends on a semantically rich and high-dimensional sensory modality: visual input from onboard cameras. This modality introduces computational and semantic interpretation challenges, but also enables fine-grained perception of dynamic environments. *Human-robot interaction* represents another critical area of robotics. Risks to humans carry the most severe consequences among all risk categories: from discomfort and perceived intrusiveness to physical injury or loss of life. Moreover, humans are highly unpredictable and difficult to model using static rules or assumptions, making them particularly challenging agents to account for in predictive risk estimation systems. As such, the chosen task lies at the intersection of two major research priorities and constitutes a challenging proving ground for the development of risk awareness.

5.2 Method

In this section, we present our prototype solution for assessing risks to humans in visual navigation in human-robot environments. In Section 5.2.1, we define the inputs and outputs of RiskAM. Section 5.2.2 presents the risk components, or individual aspects of human risk we consider in our solution. Section 5.2.3 defines how the individual components come together into a single risk score.

5.2.1 Inputs and outputs

Inputs. The RiskAM prototype is designed to operate with a single input modality: *RGB images* (further denoted $\mathbf{X}_i$, where i denotes the image ordering) captured from the robot's onboard camera. This choice is based on the central role of visual input in visual navigation. RGB imagery represents the minimal, yet semantically rich, sensory input necessary for scene understanding in navigational contexts. Incorporating additional sensory modalities, e.g., depth information, likely substantially improves the *accuracy* of risk estimation. Our reduction to RGB-only input is a deliberate design decision intended to maximize *adaptivity*: RGB cameras are ubiquitously available across visual navigation setups. We neither claim that RGB input is inherently sufficient for all use cases of visual navigation, nor do we advocate restricting sensory inputs in broader risk awareness applications—our prototype merely establishes a strong, minimal baseline for risk awareness.

Outputs. For each input RGB image (i.e., each video frame), the RiskAM prototype produces a single scalar output: a *risk score* $r \in [0, 1]$. This score matches the default risk function proposal in Section 4.3.1 and represents an abstracted estimate of the level of risk to humans associated with the current scene. To further support actionable decision-making, the score is discretized into four intuitive brackets:

- *No risk* ($r = 0$): The current scene presents no identifiable risk, and the robot may continue its operation without modification.
- *Low risk* ($r \in (0, 0.3]$): The robot should remain aware of its surroundings but may proceed with its nominal behavior.
- *Medium risk* ($r \in (0.3, 0.6]$): The robot must explicitly factor human presence into its decision-making. Navigation and planning routines should account for potential human motion and proximity.
- *High risk* ($r \in (0.6, 1]$): The robot is likely on a direct collision course or in a situation of imminent interference with humans. It must prioritize risk mitigation through immediate behavioral adjustment, such as slowing down, rerouting, or stopping altogether.

This bracketed structure is designed to enable direct coupling of RiskAM outputs with downstream planning or control systems, facilitating adaptive and risk-aware robot behavior.

5.2.2 Modeling risk

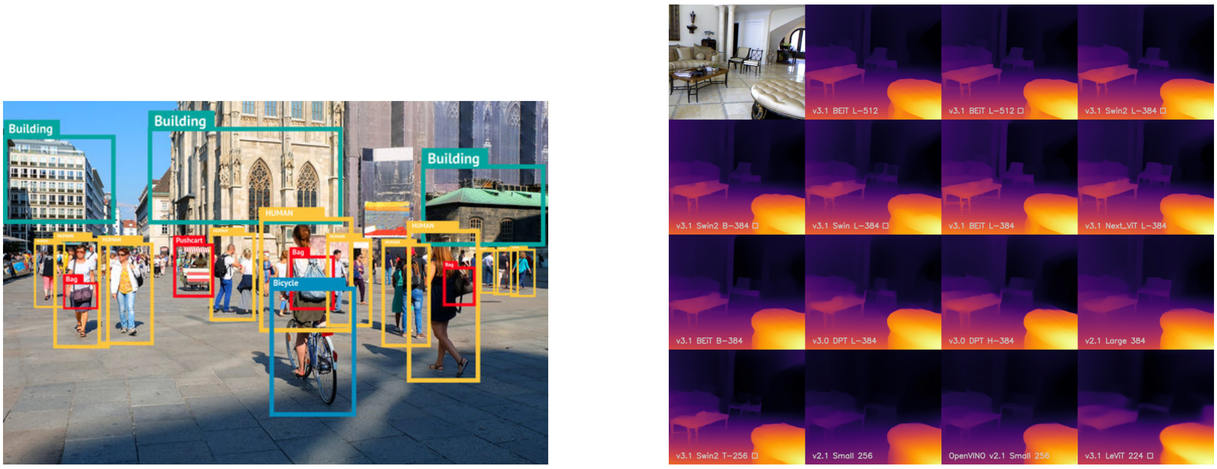
The core design principle of the RiskAM prototype is to decompose overall scene risk into a set of semantically meaningful components, each of which is evaluated independently. These component-wise assessments are subsequently synthesized into a unified scalar risk score, as described in Section 5.2.3. This modular strategy directly contributes to maximizing *semantic richness*, as it allows the model to explicitly account for multiple interpretable aspects of the scene, such as human presence, visibility, proximity, or mutual awareness. However, the space of potentially relevant risk factors is vast, and incorporating all of them would incur prohibitive computational cost. To remain suitable for real-time deployment, the model must strictly adhere to the constraint of *timeliness*. This motivates careful selection of risk components based not only on their theoretical relevance but also on their computational feasibility under real-world conditions.

Simultaneously, the framework is designed to maximize *accuracy* within these operational constraints. Component selection and model architecture are both guided by the goal of producing risk estimates that are as faithful as possible to the actual danger posed to humans in the robot's vicinity, without compromising the ability to respond in time.

Each individual risk component is estimated using state-of-the-art machine learning models. A learning-based approach was deliberately chosen to enhance *adaptivity*, enabling the system to generalize across diverse and previously unseen environments. This is critical for open-world robotics, where the vast majority of operational contexts cannot be explicitly modeled in advance.

In the RiskAM prototype, we define and evaluate *three risk components*, each capturing a distinct and complementary aspect of human-robot interaction risk in visual navigation:

- *Proximity* — This component assesses how close any detected humans are to the robot. Smaller distances correspond to higher potential for interference or collision.
- *Gaze* — This component evaluates whether the detected humans are looking in the direction of the robot. Gaze awareness is treated as a proxy for mutual awareness, which has significant implications for risk mitigation in shared spaces.
- x*-position* — This component measures the horizontal displacement of humans from the center vertical axis of the image. Humans located closer to the image center are more likely to be directly in the robot's intended path, increasing the associated risk.



(a) Object detection (YOLOv11)

(b) Depth estimation (MiDaS v3.1)

Figure 5.1: Preliminaries: object detection and depth estimation.

Preliminaries. The computation of the three risk components requires two preliminary steps, conceptualized in Figure 5.1. Firstly, all three components rely on detecting humans in the scene. For this purpose, we employ the state-of-the-art YOLOv11 pose estimation model [10], which jointly outputs human bounding boxes and 2D keypoint-based body poses. Each detected human in the scene is associated with a bounding box $\mathbf{B}_j$, where $j \in \{1, \dots, n_h\}$ and n_h denotes the number of detected humans. Each bounding box $\mathbf{B}_j$ corresponds to a subset of pixel indices within the proximity matrix $\mathbf{X}$ and any of its derivatives. Second, the proximity component additionally requires depth information $\mathbf{X}_d$ which maps each pixel in $\mathbf{X}$ to a corresponding depth value to estimate the physical distance between detected humans and the robot. Reliable depth information can be obtained from dedicated sensors. We, however, opt for an estimation-based approach to maximize *adaptivity* and assess the feasibility of learning-based monocular depth inference. Specifically, we use the MiDaS v3.1 DPT model [1, 15, 16] to estimate relative depth from single RGB frames. This approach avoids reliance on specialized hardware while maintaining sufficient depth fidelity for risk assessment purposes.

Proximity. To enhance *robustness*, missing values in $\mathbf{X}_d$ are first replaced with zeros, and outlier values below the 1st percentile and above the 99th percentile are clipped. The resulting depth matrix is then min-max normalized and inverted to form a proximity matrix $\mathbf{X}_p$, where higher values correspond to closer objects:

$$\mathbf{X}_p = 1 - \frac{\mathbf{X}_d - \min(\mathbf{X}_d)}{\max(\mathbf{X}_d) - \min(\mathbf{X}_d)}. \tag{5.1}$$

In degenerate cases where the dynamic range of $\mathbf{X}_d$ is negligible, $\mathbf{X}_p$ is set to a zero matrix. Then, the proximity matrix is adjusted using gamma correction to better align with human perceptual sensitivity. This yields the final gamma-normalized proximity map $\mathbf{X}_p^\gamma$:

$$\mathbf{X}_p^\gamma = (\text{clip}(\mathbf{X}_p, 0, 1))^\gamma , \tag{5.2}$$

where $\gamma \in (0, 1]$ is a tunable parameter and clip$(\cdot)$ denotes element-wise clipping to the $[0, 1]$ interval. The powering to the γ is done element-wise.

To represent the proximity of a human in a conservative and robust manner, we define the proximity value for the j-th human as the 10th percentile of $\mathbf{X}_p^\gamma$ over the pixels enclosed by the bounding box:

$$p^j = \text{percentile}_{10}\left(\mathbf{X}_p^\gamma[\mathbf{B}_j]\right), \tag{5.3}$$

where the $[\cdot]$ operator denotes pixel selection, e.g., $\mathbf{X}[\mathbf{B}_j]$ denotes "pixels of $\mathbf{X}$ within bounding box $\mathbf{B}_j$". The proximity risk component score for the j-th human is then defined as the global percentile of v_p^j within the full $\mathbf{X}_p^\gamma$ map:

$$r_p^j = \text{percentile\_rank}\left(\mathbf{X}_p^\gamma, p^j\right), \tag{5.4}$$

where percentile_rank$(\cdot, \cdot)$ returns the percentile rank of a given value within the entire matrix. This score captures how close the human is to the camera as a normalized proximity risk estimate.

Gaze. The gaze risk component estimates whether a detected human is likely visually aware of the robot, under the assumption that mutual awareness reduces the likelihood of unsafe interactions. This component leverages 2D pose keypoints provided by the YOLOv11 pose estimator, particularly focusing on the facial region (nose and eyes) and, if available, the shoulders.

Let K_j denote the set of keypoints detected for the j-th human. If no keypoints are detected (i.e., $K_j = \emptyset$), the gaze component score is set to zero—it is safer and more robust to assume the human is *not* aware of the robot:

$$r_g^j = 0 \quad \text{if } K_j = \emptyset. \tag{5.5}$$

For the remaining cases, we compute whether the head appears to be oriented toward the robot using facial symmetry. Let $\mathbf{n}, \mathbf{e}_L, \mathbf{e}_R \in K_j$ represent the nose, left eye, and right eye keypoints, respectively. If both eyes are present, the method proceeds by computing the midpoint between the eyes as a proxy for the facial center:

$$\mathbf{m}_f = \frac{1}{2}(\mathbf{e}_L + \mathbf{e}_R), \tag{5.6}$$

and the estimated face width as:

$$w_f = \|\mathbf{e}_R - \mathbf{e}_L\|_2. \tag{5.7}$$

The horizontal face offset is defined as the absolute horizontal distance between the nose and the eye midpoint:

$$\Delta x = |\mathbf{n}_x - \mathbf{m}_{f,x}|, \tag{5.8}$$

where the x in subscript denotes the horizontal (image column) coordinate. This offset is normalized by the face width to obtain a scale-invariant measure:

$$o_f = \frac{\Delta x}{w_f}. \tag{5.9}$$

The gaze component score r_g^j is then computed as a soft thresholded function:

$$r_g^j = \min\left(1,\ \max\left(0,\ \frac{\theta_\mathsf{u} - o_f}{\theta_\mathsf{l}}\right)\right), \tag{5.10}$$

where θ_l and θ_u are tunable parameters: wider range indicates higher error tolerance. If the nose is centered between the eyes (i.e., o_f is small), the score approaches 1, indicating a high likelihood that the person is looking at the robot. As the offset increases, the score decays toward 0.

x-position. The x-*position* component represents the likelihood that a detected human lies within the robot's forward motion path. In visual navigation, forward motion is typically aligned with the central vertical axis of the image. Thus, humans near the horizontal center of the image pose greater risk than those located peripherally.

Let B_j be decomposed into top-left (x_1, y_1) and bottom-right (x_2, y_2) corner coordinates: $\mathbf{B}_j = (x_1, y_1, x_2, y_2)$, and let W be the image width. The horizontal center of the bounding box is computed as:

$$c_x^j = \frac{x_1 + x_2}{2}, \tag{5.11}$$

and the image center is:

$$c_x^{\mathsf{img}} = \frac{W}{2}. \tag{5.12}$$

The normalized horizontal offset from the image center is then:

$$o_x^{\mathbf{B}^j} = \frac{|c_x^j - c_x^{\mathsf{img}}|}{c_x^{\mathsf{img}}}. \tag{5.13}$$

This offset is converted into a risk score via a quadratic weighting function, which penalizes larger displacements from the center more strongly, yielding the x-position component score:

$$r_x^j = 1 - \left(o_x^j\right)^2. \tag{5.14}$$

This formulation provides a computationally efficient and geometrically meaningful approximation of motion-path risk.

Bounding box tracking. To ensure temporal consistency and robustness against noisy detections, the RiskAM prototype incorporates a lightweight bounding box tracker that introduces *spatio-temporal continuity* across video frames. This mechanism links human detections over time, enabling the system to reason about persistent entities rather than processing each frame in isolation.

The tracker operates by maintaining a dynamic set of active bounding boxes, each associated with a persistence counter. For each new frame, incoming human detections are matched against tracked boxes using the standard *intersection over union (IoU)* metric. If the IoU between a new detection and an existing tracked box exceeds 0.5, the detection is considered a match and the associated track is updated. Otherwise, a new track is initialized.

To filter out unstable or transient detections, only bounding boxes that have been consistently observed over a minimum number of consecutive frames (typically three) are considered *confirmed* and passed to the risk component computation. This approach suppresses false positives and provides smoother, more reliable risk estimation in dynamic environments. The technique effectively balances computational simplicity with temporal coherence and contributes to the overall *accuracy* and *robustness* of the system.

5.2.3 Risk score

Once the individual risk components are computed for each detected human, they are combined into a single, scalar risk awareness score $r \in [0, 1]$ for the entire frame. For increased *robustness* and reduced frame-to-frame volatility, r is temporally smoothed using a mean filter on the rolling history of the most recent n_f frame-level risk scores $r_t, t \in 1..n_f$:

$$r = \frac{1}{n_f} \sum_{t=T-n_f+1}^{T} r_t, \tag{5.15}$$

where T is the current frame index. In our work, we empirically set $n_f = 5$. This score reflects the overall level of risk present in the current scene from the perspective of human-robot interaction. The final scalar score r is mapped to one of the predefined risk brackets described earlier, enabling threshold-based behavioral adaptation in the robot's control system.

If and only if there are no humans detected in the scene, the frame-level risk score r_t is equal to zero (no risk):

$$r_t = 0 \text{ iff } n_h = 0 \tag{5.16}$$

When there are humans in the scene, the frame-level risk score depends on r_p^j, r_g^j, and r_x^j, the *proximity*, *gaze*, and *x-position* risk scores for the j-th detected human, respectively. Each component is weighted by a weight parameter: w_p, w_g, and w_x, where

$$w_p + w_g + w_x = 1. \tag{5.17}$$

For the j-th human in the frame, an intermediate risk score is computed as:

$$r_t^j = w_p \cdot (1 - r_p^j) + w_g \cdot (1 - r_g^j) + w_x \cdot r_x^j. \tag{5.18}$$

Here, the proximity and gaze components are *inversely* proportional to safety: smaller distances and lack of mutual gaze awareness increase risk. Conversely, the x-position component contributes directly to risk, with higher values indicating more central—and thus more dangerous—positions in the image.

The frame-level risk score is then computed as the maximum across all detected humans:

$$r_t = \max\left(\varepsilon, \min\left(1, \max_j r_t^j\right)\right), \tag{5.19}$$

where ε is a small positive constant used to distinguish very low risk with humans present in the scene from scenes with no risk due to lack of humans.

5.3 Showcase

This section presents an experimental showcase of the RiskAM prototype. Section 5.3.1 describes the dataset used for testing. Section 5.3.2 details the evaluation protocol and experimental setup. Section 5.3.3 reports the results, with particular emphasis on RiskAM prototype's performance across the five key awareness characteristics: *accuracy*, *timeliness*, *adaptivity*, *semantic richness*, and *robustness*.

5.3.1 Dataset

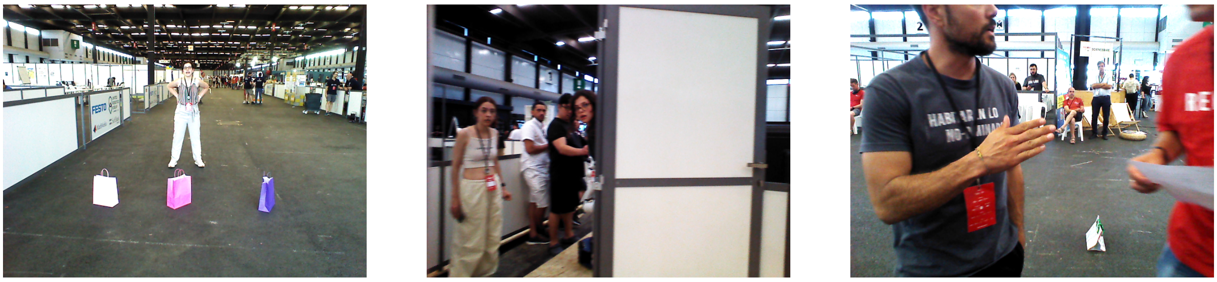


Figure 5.2: Example frames from the CS-RoboCup23 dataset

We demonstrate the performance of the RiskAM prototype on a real-world dataset collected by the CoreSense Social Testbed during the RoboCup@Home 2023 competition [4]. This dataset, henceforth referred to as *CS-RoboCup23* for short, consists of a total of 31,507 RGB video frames captured during autonomous robot operation. The recording spans several task scenarios, including environment mapping, the Receptionist task, and the Carry My Luggage tasks. Example frames from the dataset are shown in Figure 5.2.

While CS-RoboCup23 includes additional sensory data modalities—most notably depth maps—we restrict our experiments to RGB input only, in line with our objective of maximizing *adaptivity*, as discussed in Section 5.2.1. For quantitative evaluation, each video frame in the dataset has been manually annotated with a risk level label (no risk, low risk, medium risk, high risk). These human annotations serve as ground truth for assessing accuracy.

5.3.2 Evaluation setup

Evaluating the five awareness characteristics. To evaluate the *accuracy* of the RiskAM prototype, we run the system on the full CS-RoboCup23 dataset and compare its predicted risk scores against the manually annotated ground truth risk levels. Each predicted scalar risk score $r \in [0, 1]$ is mapped to one of the four discrete risk level brackets (no risk, low risk, medium risk, high risk).

The resulting predictions are categorized into three classes:

- *Correct*: The predicted risk bracket matches the ground truth label.
- *Overestimate*: The predicted risk bracket is higher than the ground truth label, indicating that the system assessed the scene as riskier than it actually was.
- *Underestimate*: The predicted risk bracket is lower than the ground truth label, meaning the system failed to detect the full extent of the present risk.

The primary objective is to maximize the proportion of *correct* predictions. From a safety standpoint, *underestimates* are significantly more concerning than *overestimates*. An underestimate indicates that the system failed to detect the true level of risk, which can lead to unsafe behavior, such as failing to yield, slow down, or stop when necessary. In contrast, overestimates result in overly cautious behavior—this may reduce efficiency, but it does not compromise safety. As such, overestimates are considered less problematic than underestimates, and their impact is comparatively minor.

Timeliness is assessed by measuring the average inference time of the RiskAM prototype per frame, thereby verifying its suitability for real-time or near-real-time deployment in robotic systems.

Adaptivity is not evaluated as a quantitative metric, but is instead supported by the experimental setup and underlying design principles. In particular, the use of generic inputs (RGB imagery) and the reliance on learning-based models that capture semantically relevant features of the scene both contribute to the system's ability to generalize across environments. In addition, to demonstrate adaptivity, we showcase results aggregated over the three tasks present in the dataset (mapping, Carry My Luggage, Receptionist) without changing the RiskAM configuration between them. Finally, as long as the results across the other characteristics remain meaningful, the adaptivity criterion is considered satisfied.

Semantic richness is demonstrated qualitatively through visualizations of the component-wise outputs of the RiskAM.

Finally, *robustness* is evaluated by analyzing the distribution of prediction errors, with particular focus on the relative frequency of *underestimates* versus *overestimates*. A robust system should minimize both types of errors while avoiding high-risk underestimates.

Evaluating parameters. To establish empirical intuition and identify best practices for deployment, we additionally evaluate the influence of the free parameters defined in Section 5.2.

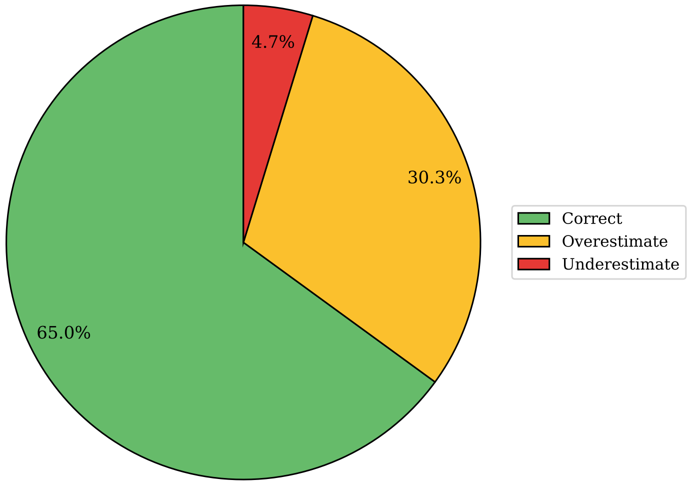


Figure 5.3: Risk prediction categorization for all outputs for our best parameter configuration ($w_p = 0.65$, $w_g = 0.25$, $w_x = 0.1$, $\gamma = 1$, $\theta_l = 0.1$, $\theta_u = 0.2$).

For each parameter, we systematically test a range of plausible configurations and observe their effect on prediction accuracy and error distribution.

We vary the weights assigned to the three risk score components—*Proximity* (w_p), *Gaze* (w_g), and x-*position* (w_x)—while maintaining the constraint $w_p + w_g + w_x = 1$ (Equation 5.17). The following six configurations are tested:

$$\begin{aligned}(w_p,\ w_g,\ w_x) \in \{\,&(0.7,\ 0.25,\ 0.05),\ (0.475,\ 0.475,\ 0.05),\ (0.25,\ 0.7,\ 0.05),\\ &(0.65,\ 0.25,\ 0.1),\ (0.45,\ 0.45,\ 0.1),\ (0.25,\ 0.65,\ 0.1)\,\}\end{aligned}$$

We also assess the impact of the gamma correction parameter γ used in the computation of the proximity matrix, testing three values:

$$\gamma \in \{0.5,\ 0.75,\ 1.0\}$$

Finally, we evaluate the sensitivity of the gaze component to its thresholding behavior by testing two configurations of the lower and upper bounds (θ_l, θ_u):

$$(\theta_l, \theta_u) \in \{(0.1,\ 0.2), (0.15,\ 0.3)\}$$

These experiments provide insight into how the model's behavior responds to key hyperparameters, aiding in the development of principled defaults and adaptive tuning strategies.

Hardware setup. All experiments were conducted on a desktop machine from 2019 running Ubuntu 24.04. The system was equipped with an Intel® Core™ i5-9400F CPU (6 cores), 32 GB of RAM, and a single NVIDIA GeForce RTX 2060 GPU. The RiskAM prototype ran successfully on this configuration without requiring specialized hardware, demonstrating its computational feasibility on modest consumer-grade platforms.

5.3.3 Results

Accuracy. The accuracy results for the best-performing parameter configuration ($w_p = 0.65$, $w_g = 0.25$, $w_x = 0.1$, $\gamma = 1$, $\theta_l = 0.1$, $\theta_u = 0.2$), selected to maximize the proportion of correct predictions while minimizing underestimates, are shown in Figure 5.3.

The key finding is that **95.3%** of all predictions support safe, risk-aware robot behavior. Specifically, **65.0%** of predictions match the ground truth risk bracket exactly, and **30.3%** overestimate the risk level. Only **4.7%** of predictions fall into the underestimate category. Given the safety-critical nature of risk awareness and consequently, the emphasis on conservative risk estimation, we consider this a strong result for a first prototype implementation.

Timeliness. The average inference time of the RiskAM prototype on our modest hardware setup (described in Section 5.3.2) is **0.102 seconds per frame**, corresponding to a processing rate of approximately 9–10 frames per second. This performance supports real-time use in typical service robotics scenarios and confirms that the prototype meets the *timeliness* criterion.

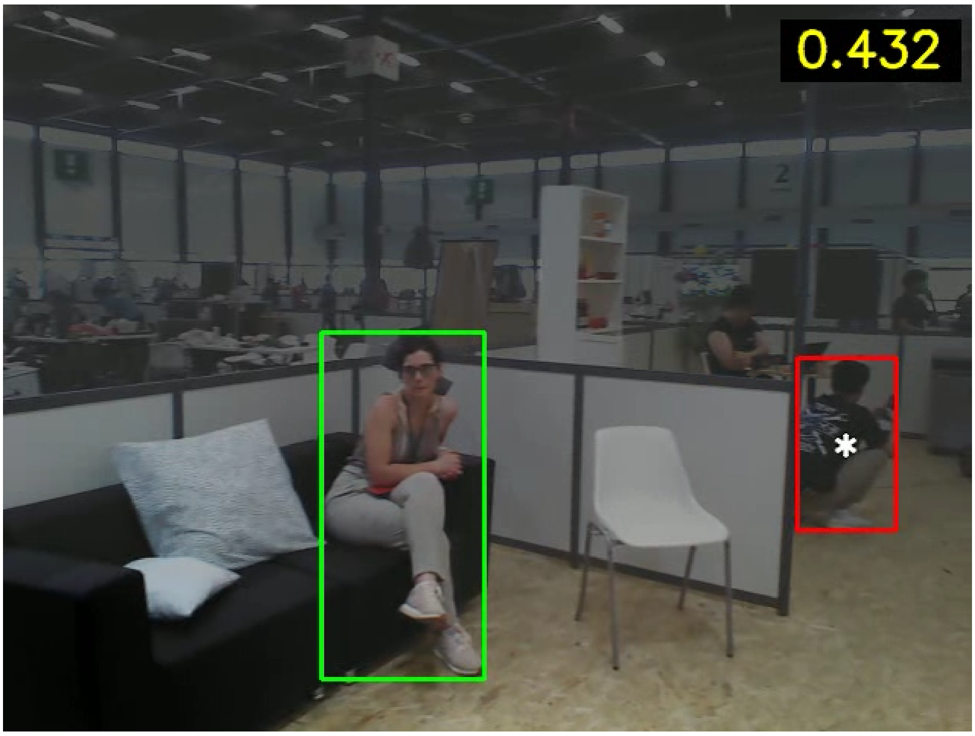
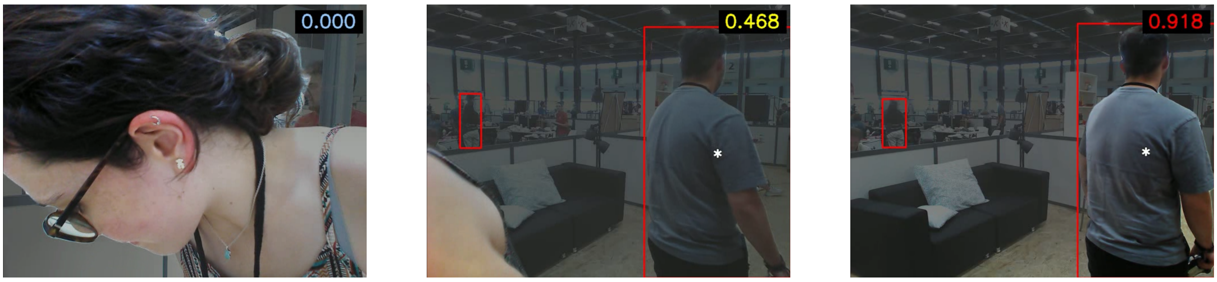


Figure 5.4: A showcase of RiskAM's semantic output.

Semantic richness. The semantic structure of RiskAM's output is illustrated in Figure 5.4, which visualizes risk components for a representative set of frames. The scalar risk score r is displayed in the top-right corner of each frame and color-coded according to its risk level bracket as defined in Section 5.2.1.

Human bounding boxes are rendered directly on the image and color-coded according to the *gaze* component r_g^j: red indicates that the human is likely not aware of the robot or the human pose could not be determined ($r_g^j = 0$); green denotes high confidence in mutual awareness ($r_g^j = 1$); and yellow reflects uncertain awareness with the given detected human pose ($0 < r_g^j < 1$). The bounding box associated with the highest per-human risk score r_t^j in the frame is highlighted with an asterisk to draw attention to the most critical actor in the scene. Additionally, the proximity map $\mathbf{X}_p$ is visualized using a smoke-like effect: image regions fade to gray as their proximity value decreases, indicating lower spatial risk. Together, these visual elements convincingly demonstrate the *semantic richness* of the RiskAM output and make the interpretable components that contribute to the final risk score explicit.



(a) Failed detection: human too close

(b) Failed detection and skewed proximity map

(c) Correct risk value (compare with (b))

Figure 5.5: Examples of RiskAM's semantic outputs for accuracy diagnostics.

Beyond interpretability, the visualization technique described above serves as a powerful tool for diagnosing issues related to *accuracy*. Figure 5.5 presents selected examples where the semantic decomposition of risk enables clear identification of failure modes.

Subfigure (a) illustrates the importance of reliable human detection, particularly in cases where individuals are either very close to the camera or only partially visible. In such scenarios, incomplete bounding box coverage can distort the computed risk components. Subfigures (b) and (c) reveal subtleties in the interplay between proximity and human detection. In subfigure (b), the proximity map $\mathbf{X}_p$ is strongly influenced by a close object in the bottom-left corner—actually a part of a human. However, because the visible region is too limited to reliably trigger a bounding box, this individual is not explicitly detected. The only person detected in the frame, a man with his back turned, is mistakenly considered relatively far from the robot. As a result, the frame receives an erroneously low risk score r, underestimating the true risk level posed by the close-range humans. Subfigure (c) shows a similar scene, but without the occluding close-range presence, leading to a correct risk assessment.

These examples highlight how the semantically grounded visualization facilitates targeted debugging of the model's predictions. By isolating the influence of each risk component, practitioners can obtain actionable insights to improve detection accuracy, refine model behavior, and mitigate specific failure cases in future iterations of RiskAM.

Robustness. The RiskAM prototype demonstrates a strong tendency to favor conservative risk estimation. In particular, the best-performing parameter configuration $w_p = 0.65$, $w_g = 0.25$, $w_x = 0.1$, $\gamma = 1$, $\theta_l = 0.1$, $\theta_u = 0.2$ results in **86.5%** of all prediction errors being *overestimates*, and only **13.5%** being *underestimates*. This skew toward overestimation is conducive to *robustness*, as it ensures that risk is rarely understated—a critical property in safety-sensitive human-robot interaction scenarios. Overestimates do not compromise safety and can be systematically refined in later iterations without endangering users or bystanders.

Adaptivity. Overall, the experiments confirm that the RiskAM prototype performs well across all other risk awareness chracteristics. Given that the method and experimental setup were explicitly designed to support and ensure generalization, most notably through the use of generic RGB inputs and component-wise semantic modeling, this outcome provides strong empirical support for the prototype's *adaptivity* in visual navigation scenarios.

Parameters. The results of the parameter sweep experiments are summarized in Figure 5.6. For the component weights, the findings suggest that best performance is achieved when *proximity* is given the highest influence, *gaze* receives a moderate weight, and the contribution of the x*-position* component is kept minimal. This reflects the relative reliability and informativeness of the components.

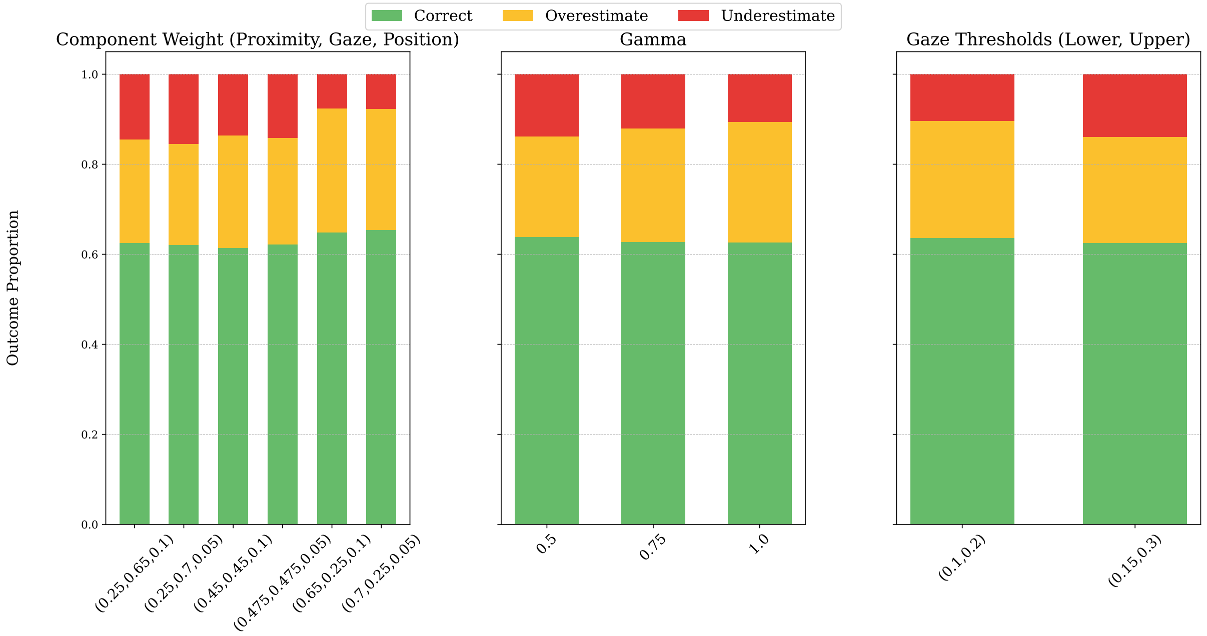


Figure 5.6: The output categorization broken down across parameters (w_p, w_g, w_x), γ, (θ_l, θ_u).

The evaluation of the gamma correction parameter suggests an accuracy-robustness trade-off. Lower values of γ (e.g., $\gamma = 0.5$) improve the proportion of *correct* predictions but simultaneously increase the proportion of *underestimates*, which are less desirable from a safety perspective. Higher values of γ yield slightly lower accuracy but produce a more conservative error distribution.

The gaze threshold experiments indicate that tighter tolerances for face alignment, i.e., lower values of θ_l and θ_u, improve predictive quality. These configurations enforce stricter conditions for interpreting gaze as directed toward the robot, reducing false confidence in mutual awareness and improving overall safety alignment.

6 Consequences

In this final chapter, we discuss implications and consequences for risk awareness derived from the RiskAM prototype and its results. Section 6.1 summarizes the main positive results. Section 6.2 provides directions for future work on RiskAM. Finally, Section 6.3 concludes this document.

6.1 Discussion

The experimental results demonstrate that the RiskAM prototype successfully achieves its intended objectives across a range of critical evaluation criteria. The system delivers a strong level of *accuracy*, with 95.3% of all predictions supporting safe robot behavior and only 4.7% underestimating the risk.

RiskAM also meets the *timeliness* requirement, achieving real-time performance (approximately 9–10 FPS). This confirms the computational efficiency of the approach and its practicality for deployment on standard robotic platforms.

The *semantic richness* of the prototype is clearly evidenced in its component-wise decomposition of risk, which yields interpretable visual outputs and facilitates both human understanding and targeted debugging. The visualization framework not only improves transparency but also serves as a diagnostic tool for identifying and correcting error sources in both perception and risk assessment.

Adaptivity is strongly supported by the design principles of RiskAM. The successful evaluation on a real-world dataset collected under realistic conditions and featuring multiple tasks further supports the generalizability of the method.

Finally, RiskAM demonstrates a desirable level of *robustness*, consistently erring on the side of caution. The system strongly favors overestimation over underestimation, a conservative behavior pattern that aligns well with safety requirements in human-robot interaction.

Overall, the RiskAM prototype succeeds as a proof of concept. It demonstrates effective risk awareness with strong performance across all five desirable risk awareness characteristics. As such, it provides a solid foundation for future refinement and extension, and serves as a strong core upon which more advanced risk awareness capabilities can be built.

6.2 Future work

Naturally, the current RiskAM prototype has several limitations. Risk awareness in robotics is a broad and complex challenge, requiring concerted effort from the research community. Even within the limited scope of visual navigation in human-robot environments, many important questions remain unresolved. The present work addresses only a subset of the broader problem space and should be viewed as a first step toward comprehensive and context-aware risk awareness systems.

We identify several immediate directions for future development of RiskAM within the current task setting. First, to rigorously evaluate *adaptivity* and *robustness*, it is necessary to test the prototype on a diverse set of datasets. While CS-RoboCup23 serves as a challenging and insightful showcase, it remains a single-domain benchmark. Expanding evaluation to datasets with varying environments, human densities, and robot embodiments will provide a complete picture of performance generalization.

Second, the development of a user-friendly parameter tuning tool would significantly improve *adaptivity*. For example, an active learning-based assistant could guide users through the process of calibrating RiskAM by asking for annotations on a small number of representative frames. This would allow vital domain-specific task context to be incorporated seamlessly, without requiring manual fine-tuning of parameters.

Third, the integration of additional sensory modalities, such as depth data, audio cues, or LIDAR. These modalities can provide complementary information to RGB input and are likely to enhance both *accuracy* and *robustness*, particularly in complex or ambiguous scenes. However, any such extension must preserve the system's *adaptivity*. Relying on rigid combinations of modality-specific inputs risks reducing generalizability and limiting deployment across diverse platforms. Instead, future iterations of RiskAM should aim for an optional multimodal architecture, in which additional sensory inputs can be incorporated when available, but are not required for basic operation. This would allow the system to flexibly adapt to hardware constraints while still benefiting from richer perceptual input when present.

Fourth, the set of modeled risk components could be expanded, provided that *timeliness* is preserved. Promising candidates include more expressive representations of gaze direction (e.g., explicit gaze angles) or emotion detection, which may provide additional cues for human intent or awareness.

Fifth, the current system operates primarily on individual frames. Extending the model to incorporate spatio-temporal dynamics that analyze short video segments or sequences instead of single images could enhance risk assessment by capturing trends, motion, and continuity of human behavior.

Finally, the current formulation of the risk score defined as a scalar value in the range $[0, 1]$ and interpreted via four discrete risk brackets represents a clear and practical starting point. However, the mapping between numeric values and categorical risk levels, as well as the thresholds used to define the brackets, should be further validated. Future work should investigate whether this structure truly supports intuitive and effective decision-making in downstream robot control, or whether alternative formulations such as continuous risk feedback, probabilistic interpretations, or task-adaptive thresholds result in a better alignment with human expectations and real-world safety demands.

Beyond the specific context of visual navigation in human-robot environments, the potential for advancing risk awareness in robotics is vast. Future work may extend RiskAM to a broader range of tasks and settings, progressively moving toward a fully-fledged pipeline that integrates both RiskAM stages: online risk estimation and offline risk mitigation. Expanding the dictionary of modeled risks and refining the associated risk functions will further enable robots to operate safely and intelligently in increasingly complex, dynamic, open-world, and human-centric environments.

6.3 Conclusion

This work presents the first prototype of RiskAM, a risk awareness module designed to support safe and adaptive robot behavior in visual navigation tasks in human-robot environments. The prototype decomposes overall risk into interpretable components and delivers strong results across key performance criteria, including accuracy, timeliness, semantic richness, robustness, and adaptivity.

Through its modular architecture, lightweight input requirements, and interpretable outputs, RiskAM lays a solid foundation for the integration of risk-aware perception into real-world robotic systems. While still early in its development, the prototype marks a meaningful step toward equipping robots with a general capacity for situational risk assessment—one that is both scalable across tasks and aligned with the safety demands of human-robot interaction.

Bibliography

[1] Reiner Birkl, Diana Wofk, and Matthias Müller. Midas v3.1 – a model zoo for robust monocular relative depth estimation. *arXiv preprint arXiv:2307.14460*, 2023.

[2] Xiaoyi Cai, Siddharth Ancha, Lakshay Sharma, Philip R Osteen, Bernadette Bucher, Stephen Phillips, Jiuguang Wang, Michael Everett, Nicholas Roy, and Jonathan P How. EVORA: Deep evidential traversability learning for risk-aware off-road autonomy. *IEEE Transactions on Robotics*, 2024.

[3] David D Fan, Kyohei Otsu, Yuki Kubo, Anushri Dixit, Joel Burdick, and Ali-Akbar Agha-Mohammadi. STEP: Stochastic traversability evaluation and planning for risk-aware off-road navigation. *arXiv preprint arXiv:2103.02828*, 2021.

[4] Irene González Fernández, Juan Diego Peña Narváez, José Miguel Guerrero Hernández, Rodrigo Pérez Rodríguez, Alejandro González Cantón, and Francisco Javier Rodríguez Lera. Robocup 2023-2024 rosbag dataset. *Data in Brief*, 57:111086, 2024.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[7] Astghik Hakobyan and Insoon Yang. Wasserstein distributionally robust motion planning and control with safety constraints using conditional value-at-risk. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 490–496. IEEE, 2020.

[8] Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ml safety. *arXiv preprint arXiv:2109.13916*, 2021.

[9] International Organization for Standardization. *ISO 10218-1:2025: Robotics — Safety requirements — Part 1: Industrial robots*. International Organization for Standardization, Geneva, Switzerland, 2025.

[10] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024.

[11] Guangyi Liu, Wen Jiang, Boshu Lei, Vivek Pandey, Kostas Daniilidis, and Nader Motee. Beyond uncertainty: Risk-aware active view acquisition for safe robot navigation and 3d scene understanding with fisherrf. *arXiv preprint arXiv:2403.11396*, 2024.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[13] Anirudha Majumdar and Marco Pavone. How should a robot assess risk? towards an axiomatic theory of risk in robotics. In *Robotics Research: The 18th International Symposium ISRR*, pages 75–84. Springer, 2017.

[14] Elie Randriamiarintsoa, Johann Laconte, Benoit Thuilot, and Romuald Aufrère. Risk-aware navigation for mobile robots in unknown 3d environments. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1949–1954. IEEE, 2023.

[15] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *ICCV*, 2021.

[16] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3), 2022.

[17] Kan Ren, Yulong Zheng, Bo Zhang, Shu-Tao Yang, Ying Chen, Pengfei Wang, Zheng Zhao, and Xiao Wang. Adversarial attacks and defenses in deep learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1342–1360, 2020.

[18] Lukas Schneider, Jonas Frey, Takahiro Miki, and Marco Hutter. Learning risk-aware quadrupedal locomotion using distributional reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11451–11458. IEEE, 2024.

[19] Xueying Sun, Qiang Zhang, Yifei Wei, and Mingmin Liu. Risk-aware deep reinforcement learning for robot crowd navigation. *Electronics*, 12(23):4744, 2023.

[20] Jan Zahálka. Trainwreck: A damaging adversarial attack on image classifiers. *arXiv preprint arXiv:2311.14772*, 2023.

[21] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

HORIZON
EUROPE

CORESENSE

Title	**D3.5 Risk awareness module**
Subtitle	
Author	Jan Zahálka (CVUT)
Date	2025/03/31
Reference	CS-067
Version	1.0
URL	http://www.coresense.eu/doc/CS-067.pdf

coresense.eu