Fraunhofer IKS

Dependable Person Detection using AI in Industrial Environments

Iwo Kurzidem, Andrea Matic-Flierl, Poulami Sinhamahapatra, Tom Haider and Karsten Roscher

iks.fraunhofer.de

Abstract:

Contemporary manufacturing facilities aim to enhance flexibility and efficiency while ensuring the safety of workers. The inclusion of autonomous machines can boost productivity even further, but it is vital to demonstrate their safe operation alongside human personnel.

Conventional safety measures often restrict the operational capabilities of autonomous machines too much. To address this, perception systems that integrate Artificial Intelligence (AI) have emerged as a promising solution, but AI currently faces its own technical and legislative challenges regarding safety. Recently the automotive industry has pioneered efforts to develop AI specific standards, e.g. ISO/PAS 8800, which includes a detailed ML Safety Lifecycle emphasizing an iterative AI development process. This paper outlines the activities involved in developing a dependable person detection system within an industrial context. It highlights the individual steps of the ML Safety Lifecycle's iterative approach to safety assurance. The approach encompasses defining safety requirements, data collection, algorithm selection, performance evaluation and mitigation strategies. Throughout the paper we provide details about each phase of the ML Safety Lifecycle followed by a lessons learnt part. For safety requirement elicitation and data collection, relevant norms and standards are listed. Regarding the algorithm selection, performance evaluation and potential mitigation strategies, we describe general considerations and procedures. Followed by illustrative examples from our own experience, this paper offers suggestions for ensuring safe person detection using AI in industrial environments.

Keywords:

Machine learning · Person detection · Safety · Regulatory landscape · Robots · Industrial applications.

Contents

1 Introduction

Modern industrial environments strive to achieve high levels of flexibility and efficiency in production, while maintaining a safe workspace for employees. The ability to deploy autonomous machines, such as (collaborative) robots, without highly restrictive safety limitations, additionally accelerates productivity. However, while there is a demand for even greater flexibility and efficiency, it is crucial to demonstrate and ensure the safety of human personnel working alongside autonomous machines. Conventional safety measures, such as light barriers or safety cages, are currently used to protect workers from potential harm, while deployed robots often operate at much slower speeds than technically possible. These measures do provide safety, but they also impose significant limitations on production capabilities, thereby excessively restricting the potential of shared workspace for humans and robots. To overcome these limitations, perception systems that incorporate Artificial Intelligence (AI) have recently emerged as a promising solution. AI promises to deliver even greater gains in flexibility and efficiency. Unfortunately, AI still faces both technical and legislative challenges that prevents its deployment in safety critical environments.

This white paper provides an outline for a dependable person detection using AI in industrial settings. It outlines crucial tasks and activities for achieving safe AI, using the ML *Safety Lifecycle* as proposed in ISO/PAS 8800 [6]. Additionally, the paper highlights specific examples and experiences from our own development.

Example

Such grey boxes contain specific examples and details from our own development of a dependable person detection using AI in industrial settings.

AI and Safety. It is imperative that Machine Learning (ML)-based algorithms adhere to the same established safety standards as conventional software, thereby demonstrating their capability to safely perform the intended tasks under all specified conditions. Industrial safety, including autonomous machines, is ensured through development according to the relevant safety norms and standards.

However, for ML there are still no explicit approaches or methods within available guidelines, regulations, or standards for how they can be integrated into safety critical applications. Figure 1 gives an overview of the current, relevant ISO standards regarding safety principles of machinery within industrial environments. Boxes with a pale green background indicate documents primarily concerned with classical software systems, i.e., traditional programming and control systems without a specific focus on ML. Whereas boxes with a pale yellow background represent documents addressing pure machine learning aspects, including data quality models and AI-specific safety considerations.

While there are updates and ongoing activities related to the use of machine learning in ISO 12100 and its referenced standards (cf. Fig. 1), these documents currently do not offer a comprehensive development methodology on how to safely implement AI within these frameworks. They primarily emphasize that if AI is employed, it must adhere to the same requirements established for conventional software, but they offer limited guidance about recommended ML models, useful methods to mitigate errors or functional insufficiencies, and suitable metrics to ascertain safety. While the overarching goal is clear, safe AI, the methodologies and tools to attain this goal are currently unclear. Besides these established documents, the most promising upcoming standard is ISO/IEC AWI TS 22440; however, it is still under development and cannot yet be applied. Fraunhofer IKS is a member of the joint working group and contributes to its development.

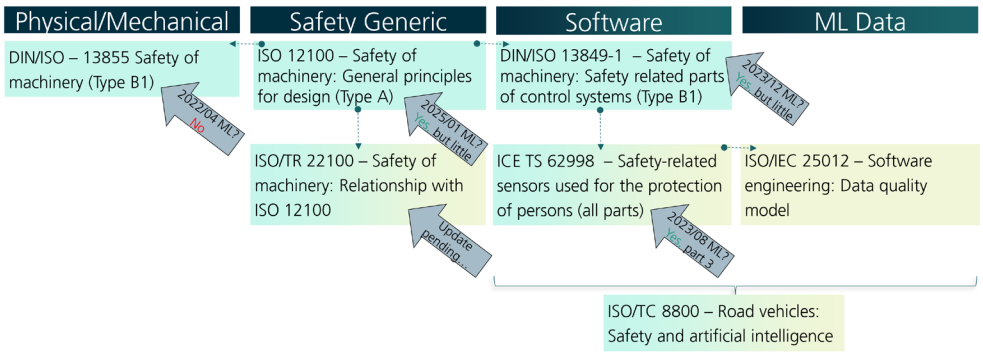


Fig. 1: Overview of relevant ISO standards and references related to industrial machinery (non-exhaustive). The big arrows indicate the latest version and if they include AI specific updates.

So far, only the automotive industry has begun to address the challenges associated with the deployment of ML in safety-critical applications through distinct, specific standards. Namely, Safety of the Intended Functionality (SOTIF) [4] and ISO/PAS 8800. Especially ISO/PAS 8800 includes a detailed ML Safety Lifecycle that emphasizes the need for an iterative and thorough validation and verification of AI systems to ensure their safe operation. In addition, it provides an overview of potential methods and metrics for design, development and run-time tasks.

The ML Safety Lifecycle. One of the rather paradoxical aspects of ML is that it is particularly useful when an exact formal specification is not possible,e.g. what defines a human? This, in turn, makes it very difficult to fully specify the problem from the outset. This inherent complexity means development will begin with an incomplete understanding of the problem at hand.

As development progresses, it's essential to recognize that knowledge will accumulate and our understanding of the problem and its solutions will gradually become clearer. This is why adopting an iterative approach for developing ML safety functions is crucial. By continuously analyzing and evaluating functional insufficiencies, we incrementally improve the system until it is safe. Additional details about the ML Safety Lifecycle and its motivation are given by Simon Burton et al. in [1].

The proposed ML Safety Lifecycle involves an iterative process that progresses with overall system development. Its core activities are:

- Definition of safety requirements based on the ML system
- Data specification and collection for training and testing
- Selection of ML algorithms and their design
- Evaluation of performance
- Mitigation strategies

These steps are repeated until there is enough evidence for a safety assurance argument, which is then evaluated for confidence. Additional, subsequent steps include the incorporation of knowledge gained during system operation, as well as the continuous assessment of remaining uncertainties about the assurance argument.

Evidence in this context refers to measurement results as well as safety-related properties of the AI system. Complex ML algorithms are essentially black-box models with incomprehensible inner workings, so the amount of quantitative data that can be obtained through testing is limited and subject to interpretation. This implies that qualitative arguments are necessary to enhance the safety of the system. Consequently, safety-related properties are described conceptually rather than with formal mathematical definitions. Safety-related properties are a subset of general AI properties; they are independent of specific applications and may include aspects such as robustness or

as robustness or domain shift (see ISO/IEC 22989 [5] for details).

In summary, this iterative approach provides a foundation for ML development despite incomplete specifications, while continuously enhancing safety assurance throughout development and deployment. The following sections will outline what these activities include for the development of a dependable person detection.

Use Case In order to provide a tangible base for the individual activities of the the ML Safety Lifecycle, we first introduce a representative use case. The model-agnostic functional specification includes the following:

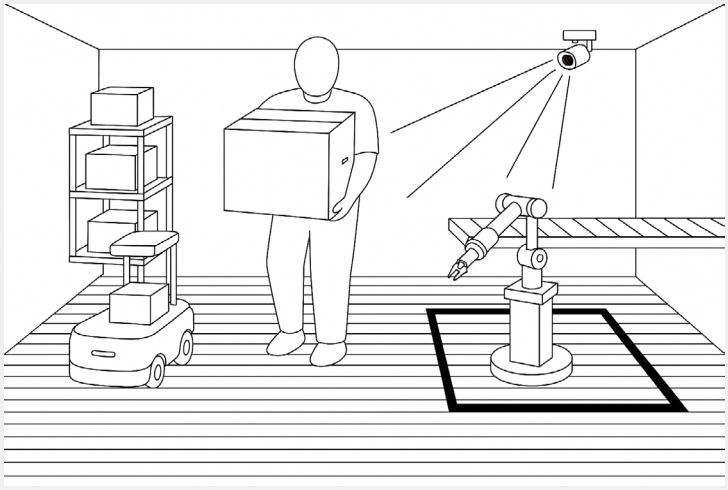
- **Input:** Sensor data obtained from the continuous monitoring of a designated area
- **Output:** An individual is either detected or not present within the designated area

Descriptively, the use case involves an indoor industrial environment. Within this environment, trained workers are allowed to move freely and may also carry objects with them, e.g. tool boxes. All mandatory occupational safety regulations apply. Apart from staff, automated guided vehicles (AGVs) also operate within this area, they are typically transporting goods. Besides AGVs, there are also permanently installed machines, e.g. robot-arms, that perform a certain tasks within a defined area and with a defined reach.

Our Use Case in Detail

Amid this general setting we realize our safety function using AI. The safety function shall reliably detect humans (and not confuse them with AGVs) in order to trigger a safety stop for the fixed machines (i.e., robotarm). When an AGV is detected in the vicinity of the robot arm, the AGV *and* robot-arm continue to operate as before, as a collision between both is not deemed a safety hazard.

However, when human personnel is detected, the robot arm shall initiate a safety stop to prevent any possible collision or harm. We assume that AGVs are equipped with their own non-AI safety functions to stop when encountering persons (this aspect is outside the scope of this study). A schematic illustration of this use case is shown below.



The figure shows our idea of a collaborative, shared space between robots and humans. This area is continuously monitored. The safety function uses a ML-based person detection algorithm. Illustration generated using AI (ChatGPT).

2
Development according to the ML Safety Lifecycle

The main idea of the ML Safety Lifecylce is to start with an incomplete specification and using the lifecycle's activities to iteratively move towards the final solution, while continuously improving safety along the way. In essence, the ML Safety Lifecycle shows how to realize and operationalize this in process.

The ML Safety Lifecycle is applied within the context of the defined use case (cf. Section 1) and details about the single procedures and tasks are provided in this chapter. Figure 2 visualizes and summarizes the main steps. The inner loop depicts the steps involving iterative cycles of data collection, training, evaluation, and optimization. Additionally, it includes a dedicated mitigation step to assess the effects and root causes of functional insufficiencies and their impact on safety. An outer loop encompasses safety activities after deployment, such as continuous monitoring. In Fig. 2 this outer loop is simplified and illustrated by the dashed path. The exact details of the continuous assurance activities are not part of this study.

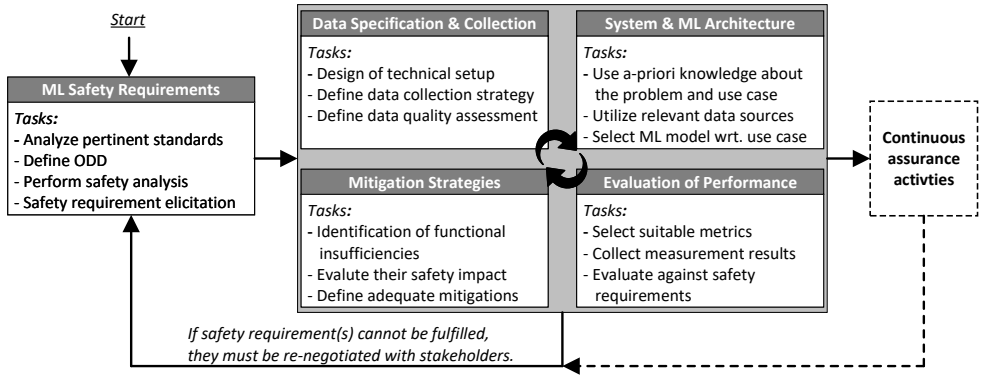


Fig. 2: Main development phases and corresponding tasks of the ML Safety Lifecycle. Adapted from [1].

2.1
ML Safety Requirements

To realize safe and reliable software systems, the software development process must follow relevant safety standards and guidelines to ensure compliance and to facilitate the requirement elicitation process. In every domain or sector, there exists a governing or established norm that serves as a framework for best practices and common conventions. These norms guide the development process by providing structured methodologies, tools, and criteria that ensure consistency, quality, safety and security. This also applies to ML development, but certain development processes must be adapted to align with the particularities of machine learning.

Figure 1 shows an overview of the landscape of ISO standards and other relevant documents related to safety in industrial environments. The governing norm and therefore the entry point is ISO 12100. This norm (type A) contains safety regulations in accordance with the equipment and product safety act Beside this, it includes development process descriptions and lists relevant and related norms and standards, as well as use case dependent norms (type B). These type B norms often contain specific details regarding certain steps within the high-level process descriptions of type A norms, which arise from the use of a particular technique or implementation approach.

Besides the norms and standards given in Fig. 1 and the *Operational Design Domain* related ones (discussed below), the following norms are relevant given the model-agnostic use case (cf. Section 1):

- ***ISO/TS 15066 (2016):*** Robots and robotic devices - Collaborative robots
- ***ISO IEC 25012 (2008):*** Software engineering - Software product Quality Requirements and Evaluation - Data quality model
- ***ISO 10218-1 (2025):*** Robotics - Safety requirements - Part 1: Industrial robots
- ***ISO/IEC TR 5469 (2014):*** Artificial intelligence - Functional safety and AI systems
- ***DIN EN IEC 62046 (2019):*** Safety of machinery - Application of protective equipment to detect the presence of persons
- ***ISO/PAS 8800 (2024):*** Road vehicles - Safety and artificial intelligence

Please note that this list is not exhaustive.

In order to derive safety requirements, a safety analysis is required. This safety analysis is performed on the intended functionality, taking into account the use case and requirement guidelines from the aforementioned documents. The safety analysis reveals potential hazards and corresponding harms, that are mitigated or even prevented entirely by suitable safety requirements.

Top Level Safety Requirement

Performing a risk assessment according to ISO 12100 and ISO 13849-1, we derived the following systems level safety requirement (SR):

SR01: *Improbable forceful collision of robot-arm with human personnel.*

This SR01 is partitioned into several sub-goals. Some of these are allocated to the ML component and require a corresponding *Performance Level* (according to ICE TS 62998-1).

During the process of requirement elicitation, derived requirements are allocated to different system components, e.g. to the ML model, and need to be evaluated. For instance, SR01 can be decomposed into subgoals involving certain physical and mechanical requirements, such as structural safeguarding. Additionally, suitable sub-goals regarding software logic must also be derived, which must be fulfilled by the ML component. Depending on where particular safety criteria are allocated, the evaluation and mitigation processes of the ML Safety Lifecycle might be simplified or complicated.

This particular step, the decomposition of high-level application safety requirements into specific safety requirements for ML models is currently non-trivial. We strongly advocate a pragmatic approach, as a complete specification right from the start is infeasible. It is also impossible to derive definite safety requirements wrt. ML model performance. Instead, we fully utilize the iterative ML Safety Lifecycle to incrementally solve this problem (additional details can be found in Section 2.4).

As pointed out before, appropriate safety-related properties should be defined to enhance system safety beyond limited measurement results. These safetyrelated properties should be hypothesized in the early stages of development, based on previous product experiences, expert knowledge, or identified through iterative safety analysis.
To achieve this, the safety analysis may consider the effects of altering safety-related properties, specific noise robustness related to the hardware characteristics and other factors on the overall safety of the system. Identifying suitable safety-related properties of AI systems will facilitate the derivation of additional, refined ML safety requirements.

As mentioned in the introduction, one of the somewhat puzzling paradoxes of ML is that it becomes particularly effective when exact formal specifications cannot be fully defined upfront. Regrettably, this makes it very difficult to test and verify an implementation using ML against such a (fuzzy) specification. However, this does not mean that there is

no specification at all; instead, the training data represent the specification. One way to obtain and structure the training (and test) data is to specify a so-called Operational Design Domain (ODD) [8].

Operational Design Domain. Clearly specifying the ODD is paramount for defining the conditions and limitations of system operation. The ODD definition also serves as proxy for function specification and can be used to derive quantitative metrics about coverage aspects [9]. In order to create an ODD description in industrial environments, tailored to the presented use case (cf. Section 1), the following standards and norms are relevant (not exhaustive):

- *ICE 60721-1 (1990):* Classification of environmental conditions - Part 1: Environmental parameters and their severities
- *DIN EN ISO 7250-1 (2017):* Basic human body measurements for technologicaldesign
- *DIN EN 60654-1 (1994):* Industrial-process measurement and control equipment; operating conditions; part 1: climatic conditions

The listed norms cover the two main aspects of the use case, namely the expected environmental influences and the human personnel. None of these norms include any ML-specific aspects, nor is there a need for them to do so. Yet, it is advantageous for the content and structure of the ODD to reflect the specifics of ML development. Again, the automotive sector offers help through PAS 1883 (2020) [3].

ODD Defintion and an Example Element

To facilitate the data specification for ML, using the ODD, we structured it similar to PAS 1883. Considerations for the ODD, specific to the defined use case, include the following:

- *Environment:* The environment is characterized by several factors including (air) particulates, illumination levels, ambient temperature etc. Mechanical loads and noise/sound are not considered. We assume that suitable safety measures are in place to address vibrations and electromagnetic influences. Compliance with standards DIN EN 60721-1 and 60654-1 ensures all relevant environmental conditions are considered in the ODD definition.
- *Dynamic Elements:* In this context dynamic refers to objects that physically change their position via movement. Dynamic elements in Dependable Person Detection using AI in Industrial Environments [9] the environment encompass humans, vehicles, and movable/carry-on items. Regarding human anatomy and speed of movement, the documents of DIN EN ISO 7250-1 and ISO/TS 15066 are used as guideline.
- *Scenery:* The scenery in the environment is defined by specific characteristics and layout, primarily within a production hall setting. Relevant factors include background and geometrical structures, as well as potential occlusion and/or shadowing effects.

One concrete ODD element from our whole ODD definition is shown below. The ODD is a good place to explicitly state and collect assumptions about the operating conditions and (technical) limitations.

Attribute	Dynamic element
Aspect	Human person
Category	Speed of movement
Limits	Human walking speed: 1600 mm/s, Human extremities speed: 2000 mm/s, Human acceleration: 2000 mm/s²
Assumption(s)	These limits apply to any human personnel subject to the safety function as defined in the use case. Assumption(s)

All elements of our ODD are structured in such a way.

As with all phases of the ML Safety Lifecycle, the ODD itself is subject to refinement and iterative updates as function development proceeds. Identified insufficiencies may require that the ODD is extended, especially when it becomes evident that certain environmental factors, initially deemed not relevant or simply unknown, may actually influence safe system behavior. For instance, if the models exhibit sensitivity to specific conditions, it becomes essential for these factors to be incorporated into the ODD. This adaptability enhances the robustness (i.e., safety-related properties) and thus safety of the system, while ensuring alignment with real-world conditions and effects.

2.2
Data Specification and Collection

In ML, the data specification provides the closest approximation to the function specification. Therefore, the data collection process should systematically investigate all dimensions of the defined ODD (see previous Section 2.1). By thoroughly examining each entry of the ODD, it is possible to gather suitable and comprehensive datasets that not only meet the requirements of the model but also improve and accelerate downstream tasks of the ML Safety Lifecycle. A systematic investigation can also involve the assertion of various factors, including human behavior (i.e., decision-making) and potential edge cases given technical realizations, to ensure that the collected data is representative of real-world situations. While this systematic investigation is crucial, it is also important to recognize potential challenges in data collection, such as input and labelling quality, as well as representatives, which must be addressed according to ISO/IEC 25012.

The technical setup used for the data collection significantly impacts the development of the ML model. The setup encompasses various aspects, including the choice of sensors, their specifications and operating conditions. The effort and effectiveness of the data collection process relies heavily on how these elements are selected wrt. the intended functionality. Properly designed setups can enhance data quality and ensure that the gathered information accurately represents the scenarios the system will encounter in real-world applications.

For instance, the use of RGB cameras under varying lighting conditions can significantly and unknowingly impact the quality and interpretation of the gathered data. This can lead to unintended biases, such as having all good examples captured in bright, daytime conditions while all bad examples are recorded in dark conditions during nighttime. Such biases can distort the model's understanding and performance, as it may not generalize well. To mitigate such spurious correlations in the ML model, ISO/IEC 25012 and ISO/PAS 8800 should be consulted.

Data Collection Setup and Process

Our internal lab setup simulates the real-world use case described in Section 1, with a robotic arm positioned centrally in an industrial production hall with persons working alongside it.

For data collection, we use three video cameras to capture both RGB and depth images. While the primary goal is person detection, this setup is selected with future applications in mind, allowing us to gather 3D information for determining spatial positions and distances to the robotarm. Furthermore, the depth data enhances the stability of our motion detection algorithms, as will be described in Section 2.3. The cameras are mounted on the ceiling and monitor the robot arm from different angles to minimize occluded areas. This provides a good coverage of the room and increases detection reliability: If a person is not present in the images of one camera, the other two can compensate for that.

We adopt an incremental and iterative approach for the data collection and analysis, focusing on finding scenarios where typical AI-based detection algorithms

struggle to provide reliable predictions. This process is coordinated and aligned with our ODD definition. In an iterative approach, we expand our (test) dataset and use different detection algorithms, thereby evaluating their suitability for the different test cases. If we find out that a specific situation is not represented enough in the dataset, or if we identify a new challenge, we repeat the data collection process to gather additional relevant data. Labeling of the collected data is done internally, with labels and bounding boxes created around persons and AGVs. This labeled data serves as a foundation for evaluating the performance of the algorithms employed in our study.

2.3
System and ML Architecture

Generally, the selected ML model(s) should match the intended functionality of the application to ensure optimal performance and accuracy, for instance, using vision models for image detection.

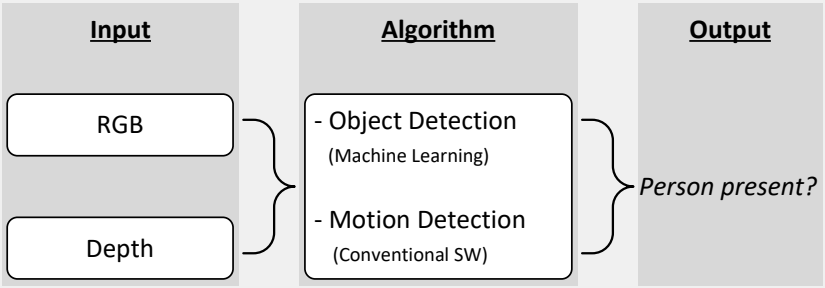
To achieve this alignment, careful consideration must be given to the selection of possible inputs from the environment. When selecting input sources, it is essential to consider the diversity and relevance of the sources from which the data is collected. Utilizing multiple data sources can provide a more comprehensive view of the problem space, allowing the model to learn from a wider range of scenarios and conditions. For instance, combining data from various sensors, such as RGB cameras, LiDAR, and/or ultrasonic sensors, one can enhance the richness of the input data. Each sensor type captures different aspects of the environment, which can lead to improved system robustness. Furthermore, employing different sensor types can help to mitigate biases that may arise from relying on only a single data source. By integrating data from multiple sensors, the model can become more resilient to variations in environmental conditions, such as lighting changes or obstructions. This multi-sensory approach not only enhances data quality but also enables the model to make more informed predictions and ultimately increase system safety.

Apart from robustness and redundancy, utilizing a-priori knowledge about potential errors and challenging scenarios can guide the technical setup phase. Leveraging expert insights allows for the implementation of corrective measures early on, thereby mitigating the impact of known, difficult scenarios. SOTIF coined the expression triggering condition, which encapsulates this concept aptly. Before and during function development a "library of potential triggering conditions" serves as a valuable resource to facilitate development. Additionally, such a library offers a comprehensive collection of scenarios that can inform and enhance projects based on other use cases.

In addition to careful input selection, exploring combinations of different algorithms for the same task can yield significant benefits. Different algorithms may have distinct strengths and weaknesses, and their performance can vary based on the specific characteristics of the data. By combining different algorithms it is possible to leverage the strengths of each individual algorithm, leading to a safer and more robust system overall.

Ultimately, the choice of input data, sensor types, and algorithm combinations should be guided by a thorough understanding of the intended functionality, use case and the specific (safety) requirements of the application.

Our ML Pipeline



For the detection of persons and AGVs, we could use common ML-based object detection algorithms, which are publicly available. These predict whether there is a person and/or AGV in the room, and also provide their 2D location within the image via bounding boxes. In addition, we also utilize a non-AI-based motion detection algorithm from classical computer vision. This is intended as a safety fallback mechanism to complement the AI-based algorithms: If the AI-based system fails to detect an object due to occlusion, unusual poses, or unexpected appearances, the motion detection algorithm can still identify movement within the scene. This dual approach ensures greater reliability and safety in scenarios where AIbased detection may be compromised, thus enhancing the dependability of our person detection system.

2.4
Evaluation of Performance

The evaluation of a ML model typically involves a variety of metrics that assess how well the model performs. Among the most fundamental metrics are the number of true positive (TP), false positive (FP), and false negative (FN) instance.In the context of person detection, a TP prediction represents a correctly detected person, a FP prediction is an object which is mistakenly identified as a person, and FN instances are cases where the model fails to detect a person. These metrics help to assess how well a model recognizes patterns and where it may be making errors.

However, a strong performance does not inherently guarantee a model's reliability or safety. To ensure that ML models function safely in real-world scenarios, both random and systematic errors must be evaluated [2]. In our case, systematic errors correspond to functional insufficiencies, which may arise from wrong assumptions or inadequate data representation. Meanwhile, random errors, usually stemming from noise within the data, can also compromise a model's decisionmaking capabilities. Thus, while performance metrics provide valuable insights, a comprehensive analysis of potential errors is necessary to understand the model predictions and their reliability.

One effective strategy for enhancing safety is the use of multiple ML models. By employing a combination of several models the overall system can achieve a higher number of TPs. This, in turn, reduces the number of FNs, as the diverse models can collectively capture a wider range of relevant patterns and scenarios that a single model might miss.

SR01: ML Model Performance Evaluation

During our study, the safety requirement SR01 was decomposed into several sub-goals. One of the derived requirements is allocated to the ML model component; as Machine Learning Safety Requirement (ML SR).

MLSR01: *Sufficiently low False Negative (FN) rate within the defined ODD.*

We focus on FNs, as they can lead to a violation of SR01. Using ISO 13849-1 and ICE TS 62998, we define "sufficiently low" *by Performance Level B* (PL B). This means the measured FN-rate shall satisfy:

$$\text{DANGEROUS FAILURE PER HOUR} < 10^{-5},$$

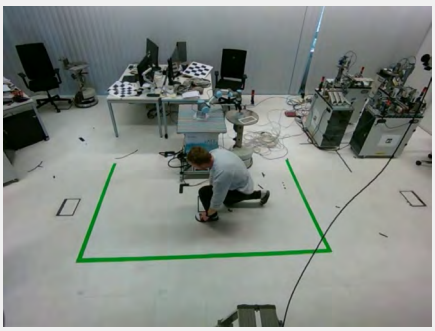
Our measurement results are obtained by using an in-house test-dataset. This set contains a total of 2,520 frames with 7,532 person annotations.On this test set we measure 1,897 FNs. Please be aware that this testset contains challenging examples, such a occluded persons, diverse appearances, including fully covered persons and others. Our measurement therefore results into a FN-rate of about 0.2518. This metric on its own does not satisfy MLSR01.

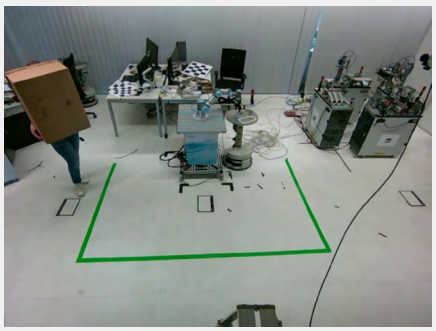
However, from an application and use case perspective we do not have to base our final decision on the raw ML algorithm output. Instead, we can use suitable post processing techniques to improve the ML model output, e.g. by aggregating the output over time. Using the inequality

$$(\text{FN-RATE})^n < \text{PL B}.$$

with our measurement $(0.2518)^n < 10^{-5}$, results in $n \geq 9$. So aggregating *nine consecutive frames* satisfies MLSR01 with our current system performance. This, however, is only possible if the samples are statistically independent. Another way to express this is to say that there are no systematic errors or functional insufficiences.

Some of our identified functional insufficiencies are illustrated below. On the left we see an unusual person pose. Here, we simulate a person tying their shoe, while on the right we can see a person carrying a box. From this angle, most of their human features are occluded.





In the next development phase we investigate possible mitigation strategies. Their implementation and evaluation will require another iteration of the ML Safety Lifecycle.

2.5
Mitigation Strategies

In the previous section, the performance of the deployed algorithms is provided, using metrics tailored to assess their performance wrt. the derived safety requirements. This measurement process forms the basis for evaluating the model's behavior in relation to safety implications and requirements. Specifically, it is essential to determine whether the observed behavior of the model meets the criteria necessary for a safe deployment. These criteria should include identified functional insufficiencies, that could lead to system errors.

Evaluating the safety impact involves analyzing not only the accuracy of the model's predictions but also its robustness, availability and residual risk associated with its deployment. For example, if the model is intended for use cases involving person detection, it must demonstrate: a sufficient accuracy in object detection and a safe behavior for known edge cases, such as unpredictable pedestrian movements or changing structural features. If the measured performance indicates that the model may pose safety risks, such as a high rate of FNs in relevant situations, then further action is required to ensure safe operation.

When the evaluation reveals deficiencies in the model's performance, it necessitates the implementation of adequate mitigation steps. These mitigation strategies can be applied to any of the various components of the system, but the solution is best allocated where it allows for an easy, efficient and correct implementation.

The identification of hazardous performance insufficiencies triggers another iteration of the ML Safety Lifecycle. As a consequence, possible solutions shall be realized within any of the previous steps:

ML Safety Requirements. If formulated safety requirements cannot be fulfilled, as no suitable mitigation strategy exists, they need to be re-negotiated wrt. the use case. According to the lifecycle it may also be required to extend, adapt or modify the safety requirements for single components or the system as a whole, either because it is evident that this requirement is technically not feasible within the component or because the state safety goal cannot be reached.

Data Specification and Collection. One of the primary areas to address functional insufficiencies is related to the data used for model training. If the training dataset lacks sufficient examples of rare but critical scenarios, it may lead to gaps in the model's understanding. In this case, additional data collection efforts may be needed to include a broader range of scenarios, thereby improving the model's ability to generalize and respond correctly to unexpected events.

System and ML Architecture. Adjustments to the ML model or the system as a whole may be necessary to address certain insufficiencies. Solutions could involve experimenting with different architectures that are better suited for handling specific errors or enhancing the model's safety performance (e.g. by learning from diverse data inputs). Adding independent input sources or improving signal quality can also increase system performance and therefore safety.

Evaluation of Performance. Selecting appropriate metrics for evaluating safety is important and often non-trivial. Traditional performance metrics, especially aggregated ones like mean Average Precision (mAP), may not fully reflect the ML model's true error behavior. Metrics suitable for the intended functionality, use case and, in particular, the safety requirements should be prioritized (e.g., absolute numbers for false positive and false negative detections). Finally, proper interpretation of computed metrics is of paramount importance; for instance, a ML model may show high mAP values, however, systematic errors may still be present.

As discussed previously, it is simply infeasible to only measure quantitative performance due to a lack of test samples, statistical significance etc. Therefore, safety-related

properties need to be evaluated to achieve a more comprehensive analysis. For instance, for all clear images in the input space, if noise perturbations characterized by an L1 norm of less than 0.001 are added to the image, the ML model shall not introduce more than 0.01% new errors [5]. These safetyrelated properties are naturally part of certain phases of the lifecycle.

Identified Functional Insufficiencies

For our use case, we have identified multiple safety-critical situations, which can be grouped into the following three categories:

- ***Occlusion:*** These are situations in which persons are partially or even fully hidden from one camera view, e.g. due to a box they are carrying.Depending on the level of occlusion, common object detection algorithms may fail to detect the person. In research, a variety of methods have been proposed to solve this problem. These contain e.g. pose estimation algorithms, which are used to identify and track the positions of key points or joints on a person's body. Another approach are part based models which focus on detecting individual body parts, such as the monitoring framework presented in [7]. In our first iteration we opted for multi-camera views as mitigation strategy (cf. Section 2.2).
- ***Unusual poses:*** In most computer vision datasets for person detection people are typically walking or standing upright. When training a ML model, this gives a certain bias for these kinds of poses. As a consequence, a person lying on the floor, crouching or in another unusual pose may not be detected by a common object detector. An example for such a situation is shown in Section 2.4, where a person is bending down to tie their shoe. Common mitigation strategies are e.g. pose estimation algorithms or to extend the training dataset by images with varying human poses. We selected non-AI motion detection as solution.
- ***Unexpected appearance:*** Not only the poses, but also the general appearance of a person may influence the performance of an object detection algorithm. For instance, if a person is wearing a fire protection suit, a uniform, or anything else that significantly alters their visual characteristics, the object detection system might fail to recognize them as a person. This issue arises because many training datasets do not encompass a wide variety of clothing styles or protective gear, which can lead to a lack of robustness in the detection algorithms. To mitigate this problem, one can enhance the training datasets with diverse examples that include individuals in unusual clothing or gear. Additionally, employing advanced techniques such as domain adaptation and transfer learning can help the model to generalize better to such unexpected appearances.

3
Summary

The work described within this paper has outlined the main development stages of an integrated and iterative development and assurance cycle for the application of AI in industrial environments. The proposed approach made use of the ISO/PAS 8800 ML Safety Lifecycle to structure the safety assurance via definition of safety requirements, data collection, algorithm selection, performance evaluation and mitigation strategies. Additionally, this paper showcased selected lessons learnt for each development step.

The paper emphasized the necessity of a specification derived from the defined use case and documented through an ODD definition to guide through the development phases. Specific documents that support this ODD construction for industrial settings have been listed. The process of safety requirement elicitation needs to be grounded on several different norms, starting from ISO 12100 and branching over to adjacent standards, connecting safety of machinery to AI. Both the ODD and safety requirements highlighted the need for better industryspecific standards regarding the use of ML for safety-relevant functions, as no consolidated standards yet exist outside of the domain of automated driving. Regarding data specification and collection for training and testing, the report strongly recommends to make use of the ODD, as it represents an approximation of the function specification within the domain of AI.

Apart from the ODD, the technical setup must match the intended functionality, insofar as it ensured proper data quality. All of this becomes part of the resulting system architecture. An additional key consideration centers around the combination of different, independent algorithms, some of which are conventional non data-driven algorithms. The performance evaluation has demonstrated that safety requires adequate metrics. In particular, common AI performance measurements, such as mAP, are unsuitable as they obscure safety-critical errors. This requires to switch from a purely model-driven development to an application as a whole, and the inclusion of safety-related properties for qualitative assurances.

For the derived top level safety requirement we decided to analyze the relevant error category FN in order to calculate the ML model's error-rate. Our measurement yielded a FN-rate of approximately 0.25. Although this FN-rate does not meet our ML safety requirement on its own, suitable post-processing, such as aggregating nine consecutive frames, can improve the ML model output, provided the samples are statistically independent and free from systematic errors or functional insufficiencies. Finally, possible strategies to address identified functional insufficiencies have been listed and explained. This work has shown how an iterative ML development approach can be leveraged to minimize work effort while simultaneously maximizing results from previous activities.

4
Outlook

As this work has shown, there is currently no clear, straightforward approach to achieve safe systems using AI. Instead, many small, incremental steps allow to improve the system bit by bit, to eventually satisfy all safety requirements. Unfortunately, quite many of these steps currently require manual work. For instance, the ODD creation, the selection of suitable metrics, the visual inspection of frames for FN and FP results to detect systematic errors, etc.

Intriguingly, AI itself promises to elevate some of these tedious tasks. Coined AI for safety, there is a growing field that leverages AI to support the development process of software systems. In particular generative AI, such as Large Language Models (LLMs), could be utilized for this purpose.

As discussed earlier, standardized ODD descriptions for domains like autonomous trains or automated driving already exist, however, this is currently not the case for industrial automation. Scenarios like human-robot collaboration in production facilities, warehouses etc. require their own ODD description. This is partly due to the vast array of use case specifications which vary widely based on application. A possible solution to bridge this gap would be to use LLMs to understand and extract requirements based on existing safety standards and use case documents for safe operation, while integrating the know-how of safety experts in establishing an ODD tailored for the respective use case. The generated ODD can be used to analyse favourable or deficient conditions required for standardized safe operation.

Additionally, using LLMs during development will accelerate the process and ideally decrease the time each new iteration of the ML Safety Lifecycle takes.

Acknowledgments. This work was funded by the Bavarian Ministry for Economic Affairs, Regional Development, and Energy as part of a project to support the thematic development of the Institute for Cognitive Systems.

5
References

1. Burton, S., Herd, B.: Addressing uncertainty in the safety assurance of machine learning.Frontiers in Computer Science 5 (Apr 2023). https://doi.org/10.3389/fcomp.2023.1132580
2. Burton, S., Kurzidem, I., Schwaiger, A., Schleiss, P., Unterreiner, M., Graeber, T., Becker, P.: Safety Assurance of Machine Learning for Chassis Control Functions. In: International Conference on Computer Safety, Reliability, and Security. pp. 149-16. Springer, Cham (2021)
3. Institution, T.B.S.: PAS 1883:2020 Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS) – Specification. BSI (2020)
4. International Organization for Standardization: Safety Of The Intended Functionality – SOTIF (ISO/PAS 21448). ISO (2019)
5. International Organization for Standardization: ISO/IEC 22989 Information Technology – Artificial Intelligence - Artificial Intelligence Concepts and Terminology. ISO (20222)
6. International Organization for Standardization: ISO/PAS 8800. Tech. rep., ISO (in work)
7. Schwaiger, F., Matic, A., Roscher, K., Guennemann, S.: Preventing Errors in Person Detection: A Part-Based Self-Monitoring Framework. In: 2023 IEEE Intelligent Vehicles Symposium (IV). vol. 1–8. IEEE (2023). https://doi.org/10.1109 IV55152.2023.10186644
8. Standard, SAE.: J3016. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems 4, 593–598 (2014)
9. Weiss, G., Zeller, M., Schoenhaar, H., Drabek, C., Kreutz, A.: Approach for Argumenting Safety on Basis of an Operational Design Domain. In: Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI. pp. 184–193 (2024). https://doi.org/10.1145/3644815.3644944