electronics

MDPI

Article

Robustness Assessment of AI-Based 2D Object Detection Systems: A Method and Lessons Learned from Two Industrial Cases

Anne-Laure Wozniak [1,2,*], Sergio Segura [3] and Raúl Mazo [2]

[1] Kereval, 35235 Thorigné-Fouillard, France
[2] Lab-STICC, ENSTA-Bretagne, 29806 Brest, France; raul.mazo@ensta-bretagne.fr
[3] SCORE Lab, I3US Institute, Universidad de Sevilla, 41012 Seville, Spain; sergiosegura@us.es
* Correspondence: anne-laure.wozniak@kereval.com

Abstract: The reliability of AI-based object detection models has gained interest with their increasing use in safety-critical systems and the development of new regulations on artificial intelligence. To meet the need for robustness evaluation, several authors have proposed methods for testing these models. However, applying these methods in industrial settings can be difficult, and several challenges have been identified in practice in the design and execution of tests. There is, therefore, a need for clear guidelines for practitioners. In this paper, we propose a method and guidelines for assessing the robustness of AI-based 2D object detection systems, based on the Goal Question Metric approach. The method defines the overall robustness testing process and a set of recommended metrics to be used at each stage of the process. We developed and evaluated the method through action research cycles, based on two industrial cases and feedback from practitioners. Thus, the resulting method addresses issues encountered in practice. A qualitative evaluation of the method by practitioners was also conducted to provide insights that can guide future research on the subject.

Keywords: robustness; software testing; object detection; artificial intelligence

Citation: Wozniak, A.-L.; Segura, S.; Mazo, R. Robustness Assessment of AI-Based 2D Object Detection Systems: A Method and Lessons Learned from Two Industrial Cases. *Electronics* **2024**, *13*, 1368. https://doi.org/10.3390/electronics13071368

Academic Editor: Yue Wu, Hong Zhu, Junhua Ding and Aktouf Oum-El-Kheir

Received: 31 January 2024
Revised: 19 March 2024
Accepted: 2 April 2024
Published: 4 April 2024

1. Introduction

Object detection systems (ODS) based on artificial intelligence (AI) can perform various tasks, including locating, tracking, and counting objects, as well as detecting anomalies. They are thus finding numerous applications for computer vision (CV) in the real world [1], ranging from facial recognition [2] to animal monitoring [3] to defect detection [4]. However, despite recent progress in AI-based object detection systems, their integration into real-world safety-critical systems, such as autonomous cars or medical diagnosis, raises concerns about their reliability in practice. Indeed, deep learning models used in these systems have been shown to be sensitive not only to perturbations imperceptible to humans [5], but also to natural or common perturbations that may occur during their real-world operation [6,7]. Robustness is, therefore, described as a key requirement for high-risk AI systems in the Artificial Intelligence Act (AI Act), the proposed European Union regulation on artificial intelligence.

To meet the need for robustness assessment, several methods have been proposed to test the robustness of image-based object detection models [8,9], usually by measuring their performance against perturbed input images, an instance of the so-called metamorphic testing technique [10,11]. However, while these methods are suitable for assessing the robustness of deep learning models, they can be difficult to apply in practice due to the lack of guidelines to adapt them to the specificities of the system under test. In previous work [12], we tested the robustness of an industrial AI-based road object detection system through metamorphic testing and identified several challenges in the design and execution of the tests in practice, such as in the selection of relevant perturbations or appropriate

metrics. Among other lessons, we learned that it may be necessary to adapt the test method through domain knowledge while remaining sufficiently generic to be able to compare the results from one system to another. Therefore, a method with clear guidelines is needed to ensure that the robustness assessment is relevant and rigorous and that the results are comparable.

Motivated by our previous industrial case study, we propose a method for the robustness assessment of AI-based 2D object detection systems from a practitioner's perspective. It should allow image-based ODSs to be evaluated in a black-box setting, which is usually a strong constraint in an industrial context or in real-world scenarios. The method was developed and evaluated through two cycles of action research [13] during which we assessed the robustness of two industrial AI-based object detection systems—a road monitoring system and a medical diagnosis system. In particular, these two industrial cases helped us to identify issues encountered when testing the robustness of AI-based systems, and to find ways of addressing them. The proposed method defines the overall robustness testing process based on metamorphic testing, which is currently one of the most popular techniques for testing machine learning-based systems [14,15]. Furthermore, it identifies a set of complementary steps and metrics to be used in order to support the application of metamorphic testing and overcome issues encountered in practice. Finally, a metamorphic relation for assessing robustness is defined, which reflects the acceptable trade-off between the distance between sets of circumstances and the difference in performance.

In summary, after giving an overview of AI-based object detection systems and metamorphic testing (Section 2), and presenting the two industrial cases (Section 3) as well as the research method followed (Section 4), this paper makes the following original research and engineering contributions:

- A metrics-driven method for the robustness testing of AI-based 2D object detection systems (Section 5), including a combination of relevant metrics and the definition of a metamorphic relation for assessing robustness.
- A prototype tool to assist practitioners in applying the proposed method and analyzing the results obtained (Section 6).
- An extensive evaluation of the proposed method on an industrial case (Section 7).

Then, we address threats to validity in Section 8. Section 9 presents related work, from reference systems to other methods and frameworks for testing the robustness of AI-based systems. Finally, we present the conclusions and future work in Section 10.

2. Background

2.1. AI-Based 2D Object Detection Systems

Object detection is a computer vision task whose objective is to detect and locate objects in an image or video. In practice, an object detection model identifies the position and boundaries of each object by drawing a bounding box around it and assigns it a class from a predefined set (see Figure 1 for an example). Object detection has a wide range of applications, including healthcare systems, autonomous vehicles, surveillance systems, and anomaly detection in industry.

Over the years, many model architectures have been proposed to perform this task, based mainly on deep learning, as it has been shown to perform better [16]. Among those architectures, a distinction is made between one-stage detectors and two-stage detectors (see Figure 2). In two-stage detectors, a first model is used to extract potential regions of interest (Region Proposal) and a second model is used to refine the location of each object within those regions and classify it. For example, models from the R-CNN family [17] fall into this category. In contrast, one-stage detectors locate and classify objects in a single operation. This is the case, for example, of models from the YOLO family [18]. One-stage detectors generally have lower accuracy but are often faster than two-stage detectors.

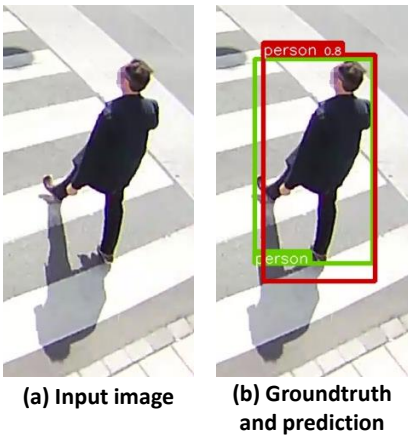


Figure 1. Detection of an object (person) on a pedestrian crossing. In green, the groundtruth (true location and true class); in red, the predicted bounding box and class (with confidence score).

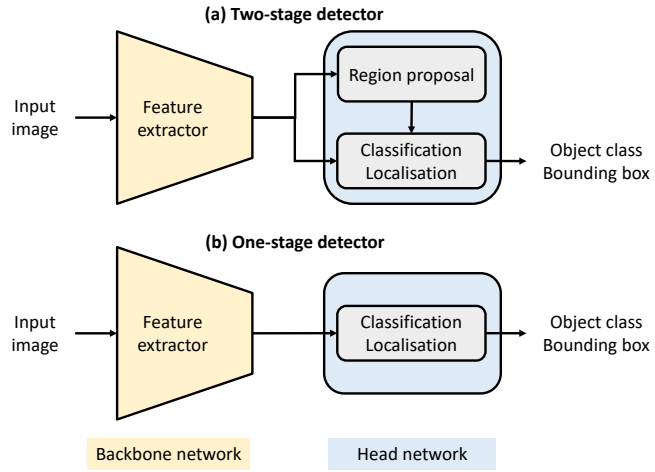


Figure 2. Overview of (**a**) a two-stage detector and (**b**) a one-stage detector.

Regardless of their architecture, object detection models based on deep learning are mostly evaluated by measuring the accuracy of their detections, both in terms of localization and classification. The overall performance of an object detection model is usually given by the average precision (AP) for each object class or the mean average precision (mAP) averaged over all classes [19]. These metrics are based on the notions of true positive (TP), false positive (FP) and false negative (FN). In the case of object detection, a true positive is defined with respect to a threshold on the Intersection over Union (IoU) and the top-1 class prediction. For instance, in some cases, TPs are detections with an IoU above 0.5 and the correct predicted class. If at least one of these two conditions is not met, it is a false positive. In the previous example (Figure 1), the predicted class is correct and the IoU is 0.83. It is, therefore, a true positive. However, the definition of these metrics may vary slightly depending on the benchmarks considered. The most recent research papers tend to use the evaluation metrics of the COCO benchmark, which include 12 metrics for measuring performance (https://cocodataset.org/#detection-eval (accessed on 1 April 2024)).

2.2. Metamorphic Testing

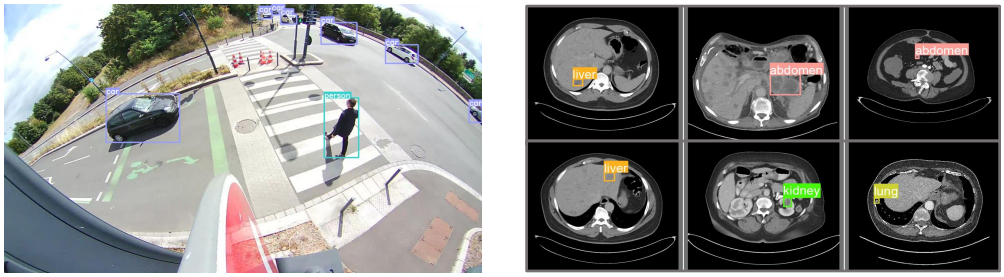
Metamorphic testing [10,11] is a property-based testing technique that alleviates the oracle problem and facilitates the generation of test cases. It is based on the definition of metamorphic relations (MRs) which are necessary properties of the system under test and which relate to multiple inputs and their expected outputs. In the process of metamorphic testing, follow-up test cases are generated from metamorphic relations and existing test

cases (so-called source test cases). The source and follow-up test cases are then executed and if their outputs violate the MRs it indicates a failure of the system under test.

Metamorphic testing has been applied to a variety of domains [20,21], such as web services and applications or embedded systems. In the field of machine learning, it has been widely adopted in response to the oracle problem [14,15]. Several metamorphic relations have been proposed, generally based on transformations in the training or test dataset that should not affect the expected outputs or in a controlled and certain way [15]. In the case of object detection systems, Wang et al. [22] introduced MetaOD, a metamorphic testing tool for AI-based object detectors. The tool ensures that the following metamorphic property is met: inserting a realistic object into the background of an image should not change the object detection results, except for the inserted object.

3. Industrial Applications

The proposed method has been developed and evaluated using two industrial AI-based 2D object detection systems (Figure 3). In the following, we introduce these two cases, which have also been used as running examples throughout the paper.



(**a**) (**b**)

Figure 3. Object detection systems under test. (**a**) Road object monitoring system; (**b**) Medical diagnosis system.

3.1. Road Monitoring System

The first object detection model under test is an AI-based Road Monitoring System (RMS) for traffic regulation developed by the company Lacroix [12]. It aims to detect road objects, such as vehicles and pedestrians, in images from cameras placed at road junctions (see Figure 3a).

The object detection model is based on an architecture derived from EfficientNet [23] for the backbone network and CenterNet [24] for the head network. It is a fast one-stage detector (Figure 2), but its performance is still comparable to that of a two-stage detector. The model was trained on an in-house dataset consisting of 30,000 high-resolution images taken in real-life conditions, extracted from videos captured in France and Vietnam where cameras were placed at road junctions, day and night. It can detect six classes of objects: person, car, bicycle, motorcycle, truck, and bus. The test dataset consists of 2645 daytime images taken in France.

The main objective of the tests is to assess the robustness of the system against changes in its hardware or software environment (e.g., change in the parameters of the image signal processor, or use of different camera technologies). This is a priority for the company developing the system since it is intended to be deployed in several cities that are likely not to use the same equipment. In addition, robustness testing must be performed in a black-box setting, to avoid having to share sensitive information.

3.2. Medical Diagnosis System

The second object detection model under test is an AI-based Medical Diagnosis System (MDS) based on the DeepLesion dataset [25]. It aims to detect and localize lesions in radiological images from CT scans (see Figure 3b). It was developed by an independent team of the company Kereval, for experimental purposes and to demonstrate its expertise.

It is considered to be similar to an industrial system in that it was developed following a typical development process, although having no real-world application.

The DeepLesion dataset was released in 2018 by the National Institutes of Health's Clinical Center. It is one of the largest CT scan datasets, including over 32,000 annotated lesions and representing more than 4400 unique patients. It, therefore, has a great diversity and is well suited to deep neural network training. The model used to do so is YOLOv5, a one-stage detector from the YOLO family of detectors [18]. It was trained on about 6500 images from the DeepLesion dataset and can detect eight classes of lesions: bone, abdomen, mediastinum, liver, lung, kidney, soft tissue, and pelvis. The test dataset consists of approximately 1600 CT images captured under the same conditions as the training dataset.

Despite its great diversity (variety of patients and studies), the training dataset contains only images that have already been processed and annotated by radiologists, with relatively few noisy images. The aim of the tests is, therefore, to assess the robustness of the system in the absence of human supervision, in the event of equipment failures that could occur during operation or misuse of the equipment.

4. Research Method

The method presented in this paper was developed iteratively and evaluated through action research [13]. Action research aims "to study a system and concurrently to collaborate with members of the system in changing it in what is together regarded as a desirable direction" [26].

Susman [27] developed a detailed model of the action research method. It is thus, an iterative process in which each research cycle has five stages: diagnosing, action planning, taking action, evaluating, and specifying learning. In the diagnosing stage, the problem is defined and the data required to carry out a detailed diagnosis are collected. During the action planning stage, possible solutions are identified to address the problem. Then, in the taking action stage, one of the solutions is selected and implemented. Data resulting from the application of the chosen solution are collected and analyzed during the evaluating stage. Finally, the specifying learning stage allows stakeholders to interpret the findings, with respect to the success or failure of the solution. At this point, a new cycle of the action research process begins and the problem is re-evaluated in the diagnosing stage. This process continues until the problem is solved and the stakeholders are satisfied with the result.

To build this method, we carried out two cycles of the action research method, as presented in Figure 4. Each cycle corresponds to the evaluation of the robustness of a different industrial object detection system.

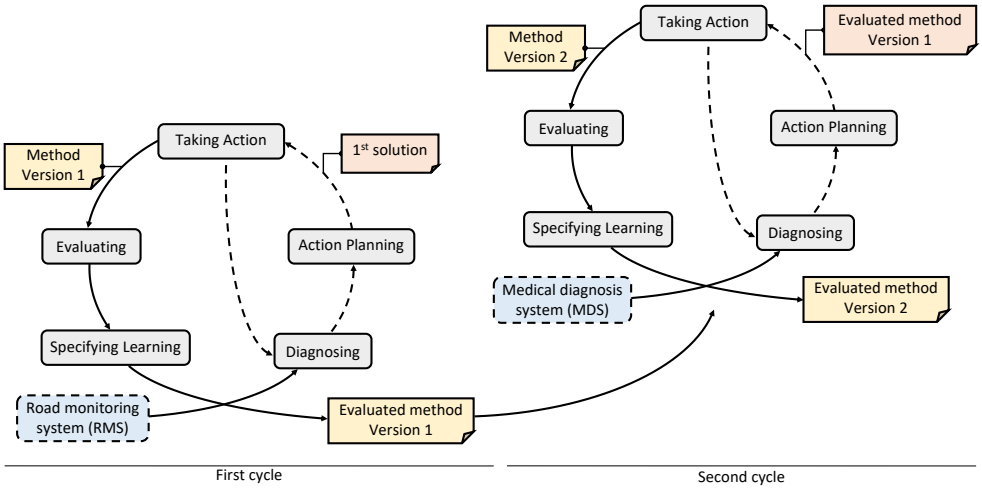


Figure 4. Action research process.

In the first cycle, we identified the metrics to be included in a method for the robustness assessment of AI-based object detection systems. As a starting point, we used the results of a previous study in which we performed the robustness assessment of the road monitoring system (RMS) using a state-of-the-art approach [12]. This study allowed us to confirm the effectiveness of the metamorphic testing approach in that context, and to identify the shortcomings and gaps in the methodology regarding the metrics and domain knowledge. To address those issues and elaborate a first solution for the method, we used the Goal Question Metric (GQM) approach [28,29] during the taking action stage. The GQM method is a top-down approach to developing goal-oriented measurements. It defines a goal, refines this goal into questions, and defines metrics to answer these questions. An advantage of this approach is that it limits the measurements collected to what is strictly necessary since each metric is justified by an objective. Based on the questions defined in the GQM model, we also adapted and completed the test method derived from the case study. The resulting method was then evaluated and improved in the next cycle.

In the second cycle, we improved the overall robustness assessment process presented in this paper. To do so, we inspected the evaluated first version of the metrics-driven method by assessing the robustness of the medical diagnosis system (MDS). During the taking action stage, we further detailed the different steps in the testing process, incorporating new guidelines based on feedback from practitioners. This new use case also allowed us to validate that the method and the recommended metrics can be successfully applied to another domain.

5. The Method

In this section, we introduce a method for robustness assessment of ODS by presenting, in detail, the definition of the goal and questions of the GQM approach, the overall robustness testing process, and the choice of metrics at each step.

5.1. Overview

The aim of our method is to provide clear guidelines for the assessment of the robustness of AI-based 2D object detection systems during their life-cycle. In GQM terms, this translates to the following goal:

Purpose: Evaluate
Issue: The robustness of
Object: An AI-based 2D object detection model
Viewpoint: From a practitioner's viewpoint

Following Solingen et al. [29] guidelines, we define several questions to characterize this goal in a quantifiable way, summarized in Table 1.

The first group of questions aims to characterize the object of the GQM model, i.e., an object detection model, with respect to the overall goal and the issue. To do so, we use the definition of robustness, which is, according to ISO/IEC TR 24029-1:2021 [30], the "ability of an AI system to maintain its level of performance under any circumstances". Thus, the object of our GQM model can be decomposed along two axes: the performance of the system on the one hand, and the circumstances of its use on the other. The first two questions, therefore, relate to the measurement of the system's performance and the description of the circumstances (see Q1 and Q2 in Table 1).

A second group of questions seeks to further characterize these attributes, i.e., performance and circumstances. To this end, Q3 aims to assess the significance or impact of a given circumstance on the functioning of the system. In addition, Q4 focuses on the differences between sets of circumstances, as comparing them can provide a good idea of their characteristics. Similarly, one way to characterize the performance of the system is to analyze its evolution by comparing performance measures under different sets of circumstances, and so this is the subject of Q5.

Finally, a question characterizes the relationship between performance and circumstances to establish the robustness of the system and achieve the overall goal of the GQM model (see Q6 in Table 1).

Table 1. GQM model: questions.

	Question
Q1	What are the circumstances of use of the system?
Q2	What is the level of performance of the object detection model?
Q3	How significant is a circumstance in the functioning of the system?
Q4	What is the difference between two sets of circumstances?
Q5	What is the difference between two measures of performance?
Q6	What is the impact of a change in circumstances on performance?

Each of these questions is answered at a different stage of the robustness assessment process, for which several metrics are recommended.

Figure 5 presents the overall process, based on metamorphic testing, which is currently one of the most effective techniques for testing machine learning-based systems [14,15]. The process has three main phases: the *specification* phase establishes references for the tests (target circumstances and their significance for the system), the *test case generation* phase allows the generation of test cases in target circumstances, and the *evaluation* phase assesses the robustness by comparing the source circumstances (resp. performance) with the target circumstances (resp. performance).

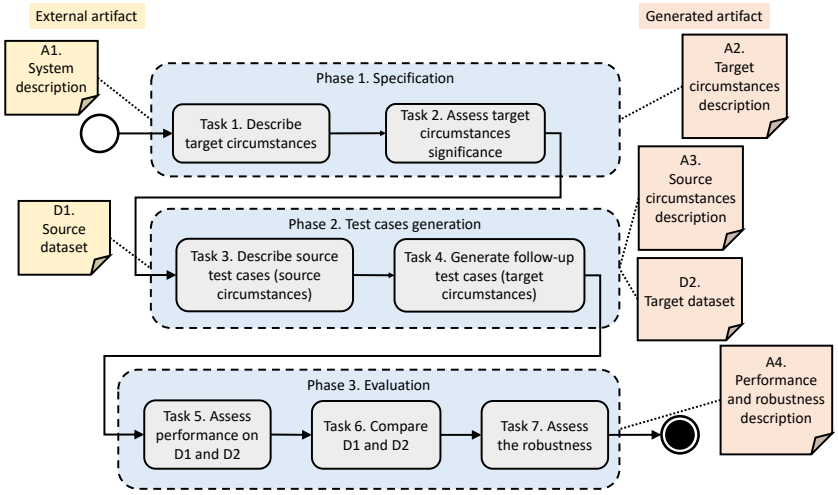


Figure 5. Business Process Model and Notation (BPMN) diagram of the robustness testing method. Yellow artifacts (A1 and D1) are external data, provided by the developer of the system, while orange artifacts are generated during the tasks.

In the following, we review each task of the process and associate each question of the GQM model with a set of appropriate metrics.

5.2. Task 1. Describe Target Circumstances

The aim of this task is to provide a statistical or probabilistic description of the target circumstances of use of the system, based on a description of the system and its environment. Target circumstances are the set of circumstances against which a practitioner wishes to assess the robustness of an AI-based system. For example, considering the MDS case, target circumstances could be misuses of the scanner, such as a patient moving during the scan.

In fact, according to ISO/IEC TR 24029-1:2021 [30], the aim of robustness testing should be to evaluate the system's performance against atypical data, different from that expected under operational conditions. Target circumstances should, therefore, be distinguished from

the operational design domain (ODD) which is the set of operational conditions under which the system is designed to function [31,32]. In particular, the target circumstances must include potential hazards that the system may face during operation (e.g., change of input data domain or hardware failure). Several works address the issue of finding hazards in the field of computer vision (CV). For example, Zendel et al. [33] built a checklist of more than a thousand potential hazards in CV systems, not only in the vision algorithm itself but also in the equipment with which it interacts and the environment observed by the system.

In practical terms, the description of the target circumstances serves as a reference during the robustness assessment for selecting and generating follow-up test cases. In this step, not only are the possible circumstances listed, but also their distribution is described, either by knowing their probability (M1) or by measuring their relative frequency (M2) within a real-world representative data sample (see Table 2). It is important to note that the sum of the relative frequencies or probabilities of the circumstances does not necessarily have to be 1, especially when circumstances are not mutually exclusive and may occur simultaneously. For example, in the MDS case, a different scanner may be used, resulting in a different density contrast, while the patient may be incorrectly positioned, leading also to a rotated cross-sectional image. On the other hand, if the circumstances are mutually exclusive (i.e., cannot occur simultaneously), the sums are expected to equal 1. For example, this is the case when circumstances are defined in relation to the time of day.

Table 2. GQM model: Q1 related metrics.

Metric	Name	Range	Definition
M1	Probability	[0, 1]	How likely the circumstance is to occur. Relative frequency over an infinite number of trials.
M2	Relative frequency	[0, 1]	How often an event occurs within the total number of observations.

Note, that the target circumstances can be defined at several levels, depending on the point of view adopted (e.g., the end user or the developer). We recommend that this step be carried out from different perspectives, as they are often complementary. As an example, Table 3 shows target circumstances for the MDS case study from the point of view of a radiographer and from the point of view of the developer of the AI model that performs the object detection. In the first case, the circumstances relate to the way the image acquisition system is used, i.e., the scanner. In the second case, the circumstances relate to the directly observed impact on the images. In that case, each high-level circumstance defined by the radiographer can be associated with one or more circumstances defined by the developer, as in the above example (Box 1).

Table 3. Target circumstances from two different perspectives (order does not matter).

Point of View	Target Circumstances
Radiographer	Two different scanners are used An old scanner is used The patient is poorly positioned The patient is moving The patient has a large body habitus The radiation dose administered was reduced (with image reconstruction) The radiation dose administered was reduced (without image reconstruction)
AI developer	Density contrast may vary significantly Some pixel lines may be shifted The cross-section may be truncated Images may be blurred due to motion blur Images may be blurred (Gaussian blur) There may be salt and pepper noise in the images The cross-sectional image may be rotated between −20° and 20° Image resolution may be altered during preprocessing

Box 1. Example: MDS

Consider, for the MDS case, the following (hypothetical) context of use, based on discussions with a practitioner in the field.

MDS may be used on images from different scanners. This can have a visible impact on the images, which can be estimated experimentally. In our case, we estimate that in 15% of cases, the radiographic contrast varies significantly from the others, in 10% of cases the image contains salt and pepper noise, in 10% of cases the image is slightly blurred, and in 10% of cases the image is pixelated due to lower resolution. Another anomaly observed in scanner images (1 in 5 cases, for our system) is the presence of lines of horizontally shifted pixels, which can distort the analysis or make the image unusable by the AI model.
In addition to equipment configuration, variability can also arise from use and patient factors. For example, a large body habitus may lead to truncated cross-sections or images with truncation artifacts. It is estimated that this occurs in 3 out of 20 images. Similarly, it is estimated that 1 in 10 images are blurred due to patient movement during the scan and that 1 in 10 images include a slightly rotated view of the section because of the patient positioning.

Finally, the target circumstances of the MDS case study are as follows:

Target Circumstances	**Relative Freq.**
Density contrast may vary significantly	0.15
Some pixel lines may be shifted	0.2
The cross-section may be truncated	0.15
Images may be blurred due to motion blur	0.1
Images may be blurred (Gaussian blur)	0.1
There may be salt and pepper noise in the images	0.1
The cross-sectional image can be rotated between -20° and 20°	0.1
Image resolution may be altered during preprocessing	0.1

5.3. Task 2. Assess the Significance of the Target Circumstances

Test case selection and prioritization is a challenge in the field of AI-based object detection, as the input space can be very large [34]. In software testing, the notion of risk is an integral part of the testing process, and the selection and prioritization of test cases based on risk, also known as risk-based testing, is a recommended practice [35].

In this method, as target circumstances may be potential hazards, it is relevant to prioritize test cases based on the risk associated with a target circumstance. In addition, this information can also be reused and included in the definition of the metamorphic relation used to assess the robustness (see Section 5.8).

In practical terms, the significance of a target circumstance is determined by the risk associated with a system failure in that circumstance, and the assessment of the significance of the target circumstances has two main objectives: to enable test cases to be prioritized and to enable the variables involved in the metamorphic relation to be weighted. Thus, the higher the significance of the circumstance, the higher the priority given to that circumstance. Similarly, the higher the significance of the circumstance, the smaller the variation in performance should be.

Based on the risk assessment model proposed by Kinney et al. [36], we define significance as the product of three factors (see Table 4): the exposure of the system to a circumstance, the likelihood of that circumstance occurring in practice, and the severity of a system failure in that circumstance (potential loss or consequences).

Table 4. GQM model: Q3 related metrics.

Metric	Name	Range	Definition
M3	Exposure	1–5	Frequency to which the system is exposed to a possible circumstance.
M4	Likelihood	1–5	Probability of that circumstance occurring in practice.
M5	Severity	1–5	Severity of a system failure in that circumstance (potential loss or consequences).
M6	Significance	1–125	$= Exposure \times Likelihood \times Severity$

In order to facilitate the definition of significance scores, we recommend assigning a value to each target circumstance independently and using fixed scales, ranging from 1 to 5 to quantify each of the three factors (see Table 5). An example in the case of RMS is given below (Box 2).

Table 5. Scales.

Value	Exposure	Likelihood	Severity
1	Rare	Rare	Insignificant
2	Unusual	Unlikely	Minor
3	Occasional	Possible	Marginal
4	Frequent	Likely	Critical
5	Continuous	Certain	Catastrophic

Box 2. Example: RMS

Consider the following circumstance for the RMS case study: *The camera has a very low shutter speed*. This circumstance translates as blurred input images for the object detection model. Its significance can be defined as follows:

- Exposure: the system is exposed to this circumstance each time the camera is changed, which should be rare. The exposure value is 1.
- Likelihood: in practice, it is very unlikely that the camera will have a too-low shutter speed, as the camera will be tested and calibrated prior to its use. The likelihood value is 2.
- Severity: a failure of the AI-based object detection system in this circumstance could be critical because it will have a continuous impact on the road actors (pedestrians and vehicles), potentially leading to accidents, until the camera is removed or calibrated. The severity value is 5.

Finally, the significance of the circumstance *The camera has a very low shutter speed* is 10.

At the end of this task and the specification phase, a document describing the target circumstances, including their distribution and significance is generated and saved for later use.

5.4. Task 3. Describe Source Test Cases (Source Circumstances)

Once the test objectives have been defined (i.e., target circumstances and their significance), the test case generation phase can begin. This phase takes as input the target circumstances description generated during the previous tasks and requires the use of a dataset, which may be the test dataset (as opposed to the training dataset), and which we will simply refer to as the *source dataset*.

As ML models are usually developed under the closed-world assumption, where training and test data are drawn from the same distribution, all the images in the source dataset should lie in the operational design domain (ODD), and therefore, contain no target circumstances. However, when data are acquired in a real environment, some images or parts of images may fall outside the ODD [37]. For example, in the case of road monitoring, this could be the unexpected presence of wild animals on the road, whereas the training data only contain objects common in urban areas (e.g., cars, pedestrians, bicycles). These samples are commonly referred to as out-of-distribution (OOD) samples.

It is, therefore, necessary to check whether the source dataset includes the target circumstances. Thus, the objective of this step is to systematically identify and quantify the target circumstances that are present in the source dataset.

In practice, we recommend following the list of previously established target circumstances and measuring the relative frequency of each of them within the source dataset (see Table 2 and Box 3 for an example). However, in the absence of metadata or precise knowledge of the dataset, it can be difficult to perform this step, especially on large datasets. To overcome this difficulty, it is possible to perform this step on a representative sample of the source dataset and extrapolate to the entire dataset. In addition, determining whether an image is drawn from the same distribution as the training data is a complex problem, known as out-of-distribution detection. Several methods have been proposed to perform this task, but it is beyond the scope of this study [38,39].

Box 3. Example: MDS

In the following, this step is carried out from the point of view of the developer of the AI model in the MDS case study. The following table lists the relative frequencies of each target circumstance listed in Section 5.2 within the test dataset.

Finally, the source circumstances of the MDS case study are as follows:

Target Circumstances	**Relative Freq.**
Density contrast may vary significantly	0.05
Some pixel lines may be shifted	0
The cross-section may be truncated	0.1
Images may be blurred due to motion blur	0.05
Images may be blurred (Gaussian blur)	0
There may be salt and pepper noise in the images	0
The cross-sectional image can be rotated between -20° and 20°	0
Image resolution may be altered during preprocessing	0

5.5. Task 4. Generate Follow-Up Test Cases (in Target Circumstances)

The objective of this task is to generate a dataset consisting solely of images in target circumstances (i.e., outside the operational design domain), in order to assess the robustness of the model in these circumstances. In the context of metamorphic testing, these images are called follow-up test cases, generated on the basis of transformations representative of the target circumstances. This task takes as input all the documents generated during the previous tasks (description of the source and target circumstances) and the source dataset.

The first step is to determine the transformations to be applied to the images in the source dataset to generate the follow-up test cases. These transformations are then randomly applied to images from the source dataset in order to generate the follow-up test cases in proportions similar to the frequency of the target circumstances. We call this newly generated dataset of follow-up test cases, the *target dataset*. See Figure 6 for examples of source and follow-up test cases, for the MDS case.

Note, that if necessary, the significance value of each circumstance can be used to prioritize the transformations to be applied (e.g., if we do not wish to assess robustness to all the target circumstances at the same time, which may be the case for assessing situations independently of each other).

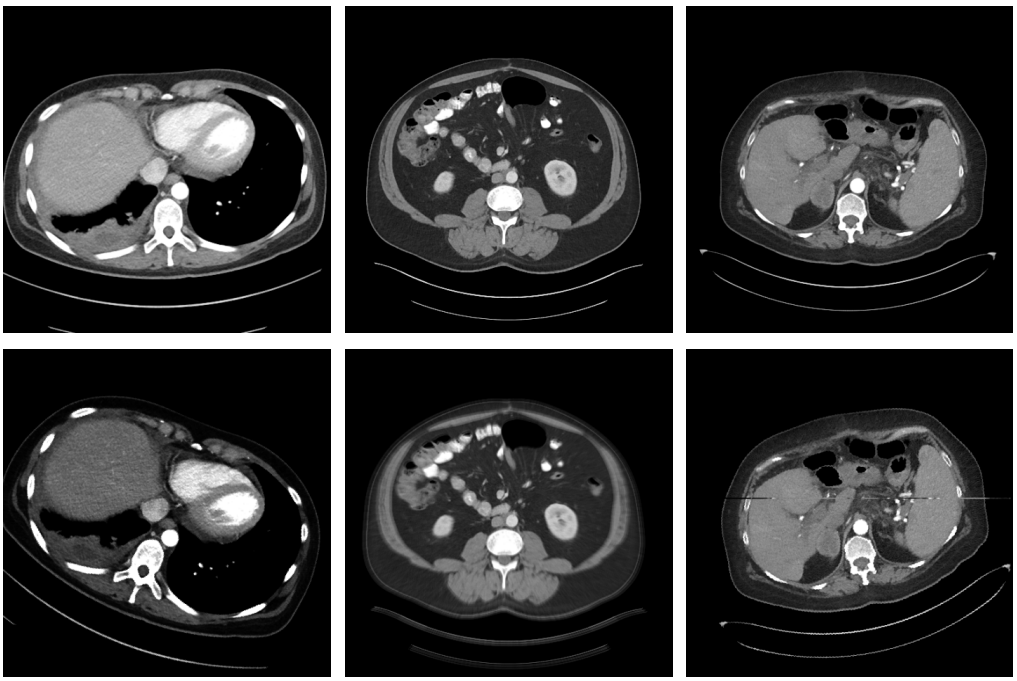


Figure 6. Examples of source test cases (source circumstances), on the top, and follow-up test cases (target circumstances), on the bottom, for the MDS case.

An example of the application of this task in the case of the RMS is given below (Box 4).

Box 4. Example: RMS

In the RMS case study, we have defined a certain number of target circumstances related to changes in camera or image signal processor (ISP) parameters. These target circumstances can be translated into transformations that can be applied directly to the input images. For example, the circumstance *The camera has a very low shutter speed* results in blurred input images. We can, therefore, define a Motion Blur transformation to apply to the images in the source dataset to create follow-up test cases in this target circumstance.
See Wozniak et al. [12] for more details and examples on the mapping between transformations and the real phenomena, as well as examples of source and follow-up test cases for this industrial case.

Task 3 allowed us to determine that none of the images in the source dataset contain the target circumstances. We thus, directly apply the Motion Blur transformation to the images in the dataset so that the relative frequency of the follow-up test cases matches the relative frequency of the corresponding target circumstance.

5.6. Task 5. Assess Performance on Source and Target Datasets

In accordance with the definition of robustness (see Section 5.1), the performance assessment is a fundamental step in quantifying the robustness of an AI-based object detection system with respect to changes in the circumstances of use of the system. The objective of this task is to evaluate the performance of the AI-based system on both the source dataset and the newly generated target dataset(s) corresponding to the target circumstances. These two performance evaluations are similar in terms of process and metrics. Table 6 lists the metrics used to evaluate the performance of object detection models.

Table 6. GQM model: Q2 related metrics.

Metric	Name	Range	Definition
M7	Number of true positive (TP)	$\mathbb{N}$	Number of objects correctly detected.
M8	Number of false positive (FP)	$\mathbb{N}$	Number of nonexistent detected objects or existing objects wrongly localized.
M9	Number of false negative (FN)	$\mathbb{N}$	Number of objects not detected.
M10	Precision	[0, 1]	Efficiency of the model. $P = \frac{\text{TP}}{\text{TP + FP}}$
M11	Recall	[0, 1]	Effectiveness of the model. $R = \frac{\text{TP}}{\text{TP + FN}}$
M12	Mean average precision (mAP)	[0, 1]	Area under the Precision-Recall curve averaged over all classes. It measures the overall accuracy and the trade-off between Precision and Recall. We distinguish between: mAP across IoUs: AP, $AP_{0.5}$, $AP_{0.75}$ mAP across object scales: AP_S, AP_M, AP_L
M13	Mean average recall (mAR)	[0, 1]	Maximum recall given a fixed number of detections per image, averaged over categories and IoUs. We distinguish between: AR across detections: AR_1, AR_{10}, AR_{100} AR across object scales: AR_S, AR_M, AR_L

According to Padilla et al. [19], the most widely used metrics, whether in object detection challenges or by the scientific community, are those measuring the accuracy of detections. The overall performance of an object detection model is thus, usually given by the mean average precision (mAP) and the mean average recall (mAR).

These metrics are based on the notions of true positive (TP), false positive (FP) and false negative (FN). In the case of object detection, a true positive is defined with respect to a threshold on the Intersection over Union (IoU) and the top-1 class prediction. For instance, in some cases, TPs are detections with IoU above 0.5 and correct predicted class. If at least one of these two conditions is not met, it is then a false positive.

For each detected object, IoU is defined as follows:

$$IoU = \frac{\text{area of intersection}}{\text{area of union}}$$

Although these are the most commonly used metrics, not all of them are useful for all object detection systems. The choice of one or the other metric varies according to the application case and the objectives of the system. For example, in the case of the road monitoring system, one of the most important metrics is the number of false negatives, as it must be as close as possible to 0 to avoid potentially fatal incidents. On the other hand, in the case of the medical diagnosis system, the number of false negatives is not as important as in the case of the road monitoring system, because each CT scan image can be reviewed by a human expert at any time. Box 5 summarises the results obtained for the MDS case.

Box 5. Example: MDS

The following table shows the performance of MDS on the source and target datasets.

	Precision	**Recall**	**$AP_{0.5}$**	**$AP_{0.75}$**	**AP**
Source dataset	0.564	0.51	0.526	0.396	0.34
Target dataset	0.498	0.384	0.414	0.264	0.246

5.7. Task 6. Compare Source to Target Datasets

As previously mentioned, assessing robustness requires comparing the performance of the system in different circumstances (i.e., against the source and target datasets). However, given the stochastic nature of the target dataset generation process, the difficulty that the target dataset represents for the system is not always the same from one run to another. For example, from one image to another, a change in contrast (with the same parameters) will not always have the same impact. Similarly, combinations of circumstances on the same image will not always be the same.

To overcome this issue it is, therefore, necessary to have a common scale that quantifies the difficulty of the target dataset, with respect to the source dataset. In practice, this can be conducted by comparing the two datasets and measuring a distance, either between the samples or between their features. In the first case, the metric is directly calculated sample by sample, by measuring the distance or similarity between the original and the follow-up images. It is averaged over all the samples from the datasets. In the second case, the characteristics of the datasets are compared, for example, by using feature extraction or clustering, and the metric measures the distance between feature vectors or clusters. Table 7 lists several types of metrics that can be used in these two cases.

Note, that the choice of one metric or another highly depends on the use case and the types of circumstances being tested (see example below in Box 6).

Table 7. GQM model: Q4 related metrics.

Metric	Name	Definition
M14	Distance between images	E.g., Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) and its variants [40]. These metrics are used in the field of image quality. They are full-reference metrics, which means that they compare an original image with a distorted image (in our case, the generated follow-up image).
M15	Distance between distributions	E.g., Fréchet Inception Distance (FID) [41]. Rather than comparing images pixel by pixel, it is possible to compare other characteristics. This is the objective of metrics such as Fréchet Inception Distance (FID), which compares the distribution (mean and standard deviation) of images processed by an Inception-type neural network. It is currently a benchmark metric for assessing the quality of generative models.
M16	Distance between feature vectors	E.g., Euclidean distance, cosine similarity. After using a feature extraction algorithm for each of the images (source and follow-up), the distance separating the feature vectors is measured using a metric such as Euclidean distance or cosine similarity.
M17	Inter-cluster distance	E.g., linkage distance [42]. There are several ways of defining the distance between two clusters, depending on whether we are considering a single point (Single linkage), all the points (Complete linkage), the average of all points (Average linkage), or the centroid of the clusters (Centroid linkage). Note: inter-cluster or linkage distances are usually calculated based on well-known distance measures, such as Euclidean, Manhattan, or Mahalanbois distances.

Box 6. Example: RMS

For this use case, we used the structural similarity index (SSIM) [43] to measure the distance between circumstances. SSIM is a full-reference metric that measures the similarity between a source image and its follow-up, in terms of luminance, contrast, and structure. Its value ranges from 0 to 1, where an SSIM of 1 indicates that the two images being compared are identical. The distance between circumstances is thus, given by $1 - \text{SSIM}$.

This choice was motivated by the fact that in this use case, the camera is fixed in time and the circumstances tested give rise to transformations applied uniformly to the image. For example, there are no changes to specific objects in the image. The only structural modifications are, therefore, those introduced by blur or noise transformations. A full-reference metric is thus relevant in this configuration.
In addition, we have chosen to use a metric that is close to the human perception of image quality (by comparison with MSE or PSNR, for example), because the transformations must be realistic and the system's performance comparable to that of human vision.

In practice, SSIM is computed for each pair of images (source, follow-up) in the datasets, and then averaged over all the values. The distance between the datasets is given by $1 - \overline{\text{SSIM}}$. As an example, the following table shows the evolution of the averaged SSIM as a function of the parameter (amount of expansion and blurring) of the Chromatic aberration transformation, defined in the context of the circumstance: *the camera is equipped with a poor quality lens*.

Parameter	**0**	**0.1**	**0.25**	**0.5**	**0.75**	**1**
SSIM	1	0.904	0.777	0.733	0.711	0.697
Distance	0	0.096	0.223	0.267	0.289	0.303

5.8. Task 7. Assess the Robustness

The objective of the last task is to aggregate the information from the previous stages to assess the robustness of the system in the target circumstances. In the following, the metamorphic relation used in this testing method is introduced.

As previously mentioned, we base our approach on the definition of robustness in [30], which considers an AI system to be robust if it is able to maintain its level of performance under any circumstances. In practice, we relax this definition by introducing the notion of distance between the circumstances: the smaller the distance between the source and target circumstances, the closer the performance of the model in these circumstances should be. In fact, as the tested circumstances lead to atypical data different from those expected in operational conditions, it is unlikely in practice, to have strictly equal performance. Furthermore, this relaxed definition includes the notion of global robustness as defined by other authors (i.e., whatever two points are contained in the same region, the model returns a similar prediction) [15].

We can, therefore, express the difference in performance ΔP (see Table 8) as a function of the distance d. Then, a metamorphic relation for a robust model is:

$$\forall (x, x'), \Delta P = |P_x - P_{x'}| \leq \epsilon(d)$$

where x and x' are two sets of circumstances, P_x (resp. $P_{x'}$) is the performance of the model under given circumstances x (resp. x'), and ϵ is a piecewise linear function. It means that we can define several d_i for which the maximum threshold on ΔP will be different. In practice, this function ϵ depends on the use case and the choices made at the previous stages. It represents the acceptable trade-off between the distance between sets of circumstances and the change in performance. It can be determined experimentally. See the example below for more details (Box 7).

Table 8. GQM model: Q5 related metrics.

Metric	Name	Range	Definition
M18	Performance difference (ΔP)	[0, 1]	$\Delta P = \lvert P_x - P_{x'} \rvert$ Absolute value of the difference between performance on the source and target datasets.

Once the ϵ function is defined, all that remains is to use the previous results to check that the model is robust, i.e., that it verifies the metamorphic relation.

Box 7. Example: MDS

For the MDS use case, several performance metrics have been calculated (see Section 5.6). The distance between the source and target datasets was measured with CW-SSIM [44], a variant of SSIM which provides better results for images that have been rotated or truncated. The results are summarized in the table below.

In order to determine the $\epsilon(d)$ function in the metamorphic relation, we first seek to identify one or more significant distance values d. In the previous task, the CW-SSIM metric was computed for each pair of images (source, follow-up) in the datasets, and it is, therefore, possible to determine the threshold at which it becomes significantly more difficult for the human eye to correctly process the image. In practice, we rely on a domain expert to define this threshold. In this use case, it is estimated at $d = 0.25$ on average. This means that the performance of the model should remain more or less the same (e.g., $\Delta P \leq 0.01$) for $d \leq 0.25$. Beyond this value, the difference in performance may increase linearly with d.

We thus, have the following metamorphic relation:

$$\forall(x, x'), \Delta P = |P_x - P_{x'}| \leq \epsilon(d)$$

where

$$\epsilon(d) = \begin{cases} 0.01 & \text{if} d \leq 0.25 \\ d & \text{else} \end{cases}$$

In this example, as shown in the table below, $d \leq 0.25$. Whatever the performance metric, the model does not verify the above relation and is, therefore, not globally robust.

Performance ΔP			Distance d
AP	**Precision**	**Recall**	**CW-SSIM**
0.094	0.066	0.126	0.154

6. Tooling

A prototype tool, available online (https://drive.google.com/file/d/1i-kqAeABWZPYyJii3JDUHSbDHsPncmFq/view?usp=drive_link) (accessed on 1 April 2024) , has been developed to assist practitioners in utilizing the method. This tool has been implemented as a three-tab spreadsheet and allows (i) the different phases of the testing process to be monitored, (ii) information about the system, such as measurements, to be kept throughout the process, and (iii) the robustness of the system to be checked (w.r.t. a metamorphic relation). The content of each of the three tabs is described below, and screenshots of the tool used for the MDS case are available in Figure A1.

The first tab, *Specification*, is used to define the target circumstances (Task 1) and their significance (Task 2). In practice, the user fills in a table summarizing all the target circumstances of the system and their characteristics, including probability, exposure, likelihood, and severity. The tool then calculates significance scores and displays the distribution of circumstances according to the three factors to help identify the most critical circumstances for the system.

The second tab, *Test cases generation*, focuses on describing the source dataset (Task 3) and is used to define missing follow-up test cases in target circumstances (Task 4) for the

robustness assessment. First, a pre-populated table based on the previous tasks is made available for the user to enter circumstances that are already in the source dataset used for testing the system. Users indicate their relative frequency in the dataset. Then, the tool automatically updates a list of circumstances to be tested as a priority. To do so, it uses the difference between the probability and the relative frequency to determine whether follow-up test cases should be generated. The list is prioritized according to the circumstance's significance score.

Finally, the last tab, *Evaluation*, is used to perform all the remaining tasks for the robustness assessment: performance evaluation (Task 5), measurement of the distance between datasets (Task 6), and definition of the robustness property to be verified (Task 7).

7. Evaluation and Lessons Learned

Following the two cycles of action research, the proposed testing method was empirically evaluated through a case study [45] whose goal was to assess the relevance and applicability of the method in an industrial context. The main research question was defined as follows: *How useful is the proposed method for practitioners to test the robustness of AI-based 2D object detection models?* As mentioned in the introduction of this research (Section 1), from our previous work we have learned that current state-of-the-art methods can be difficult to apply in practice as they often lack clear guidelines for adapting them to the systems under test while still allowing to compare the results from one system to another. We thus, focus on the usefulness of the proposed method for practitioners, in terms of its applicability and its ability to provide clear guidelines.

In the following, we report on the design, results and lessons learned from the study.

7.1. Design

We conducted the case study on the Medical Diagnosis System (MDS) developed by Kereval (see Section 3.2), and the primary mean of data collection was a focus group involving practitioners from the company. Focus groups are an effective method for conducting qualitative assessments of new approaches and collecting potential problems and lessons learned based on feedback from practitioners [46]. We followed the recommendations of Kontio et al. [46] in designing and conducting the focus group session.

The point of view adopted in this evaluation is that of external testers. We, therefore, selected five participants from Kereval, all of whom are experienced engineers or testers with experience working on AI-based systems, but who were not involved in the development of the MDS.

The focus group session took the form of a structured discussion in which the following four main questions were addressed.

Q1: What problems did the company's actors encounter in applying the state-of-art methods? In this study, we aim to assess the usefulness of the proposed method in relation to current practices and the issues that may be encountered in practice when testing robustness. Therefore, the goal of this question is to identify practical issues and shortcomings related to the methods commonly used by practitioners.

Q2: What difficulties do the company's actors encounter in applying the proposed method? The goal of this question is to identify potential problems in using the method, as they may reveal unclear guidelines or unresolved practical issues.

Q3: What is the company's actors perception on the benefits of the proposed method for the practitioners? We seek to identify the strengths of the proposed method, such as the practical issues it addresses.

Q4: What could prevent the use of the proposed method in other contexts? We seek to identify the limitations of the proposed method in terms of applicability, and potential areas of improvement.

During the session, one researcher acted as moderator to facilitate the discussion and probe deeper when necessary. An additional observer took part in the session, which was also video-recorded for data collection purposes. In total, the session lasted 2.5 h, divided

into three parts. First, the moderator briefly recalled the purpose of the focus group and the principle of the proposed method. Then, the discussion focused on the application of the method to the MDS case in order to identify the potential difficulties for an external tester in this context. Finally, the participants were invited to discuss more broadly the usefulness of the method in relation to current practices and methods, the benefits it could bring and the potential limitations of its application in different contexts.

Data were analyzed from the video recording and the notes taken during the session, using the pattern matching technique [47] in which the empirically observed patterns (findings from the focus group session) were compared with the expected patterns that we had formulated based on the outcomes of the action research.

7.2. Results

In this section, we synthesize the results for each main question addressed during the focus group session.

7.2.1. Current Practical Issues (Q1)

The most frequently cited problem during the discussion, and the one with which the majority of participants (three out of five) agreed was the difficulty of *determining a threshold on the system's performance* that would enable testers to assess its robustness. This difficulty led some participants, during similar projects, not to explicitly define a metamorphic relation as presented in our proposed method (Task 7) but to use a method similar to load testing in which the transformations applied to the input images are increasingly severe.

The difficulty of *defining relevant transformations* on the data to test the system was also mentioned, together with the difficulty of *applying transformations* to the images, i.e., generating test cases from a source dataset in the case of systems where domain knowledge is required and the transformations to be applied are complex. The lack of a framework to guide the definition of the transformations has already led one participant to carry out the tests using an exploratory approach during one of his projects.

7.2.2. Difficulties in Using the Method (Q2)

The participants agreed that Task 1 (*Describe target circumstances*) and Task 2 (*Assess the significance of the target circumstances*) were the most difficult to carry out in practice as a tester, particularly the assignment of a value to certain factors involved in the significance score (exposure, likelihood, and severity). They also identified that this latter difficulty could arise from the sometimes subjective nature of these factors. On the other hand, two of the participants drew parallels with traditional software testing practices, such as risk assessment and the determination of safety integrity levels (SIL) as described in the IEC 61508 standard [48], validating that these difficulties are common in the context of software testing and that the tester indeed usually discuss these issues with a domain expert.

To a lesser extent, the question of *the choice of some metrics* was raised as a difficulty in practice, particularly for measuring the distance between datasets or circumstances as it requires a good understanding of the system and its environment. However, the majority of participants (four out of five) felt that these choices were indeed a matter of the AI tester's expertise. Thus, the tester can discuss with the domain expert to better understand the system but retains responsibility for the final decision.

7.2.3. Benefits (Q3)

In terms of the benefits perceived by the company's practitioners, the most frequently cited response was *clear, structured guidelines* for the testing process. One participant highlighted the *time saved* by defining a clear process with ordered steps. Another participant felt that such a method can *help in formulating the hypotheses* under which the robustness of an AI-based system is validated. In addition, most of them (four out of five) agreed that the proposed method can *help determine the threshold on system performance and the definition*

of relevant transformations, and two of the current practical issues previously mentioned (cf. Q1).

7.2.4. Limitations (Q4)

Overall, the participants felt that the proposed method would be difficult to apply to *systems whose circumstances of use are complex to describe and strongly linked to the application domain*. In practice, they consider that such "domain" circumstances cannot be easily expressed in terms of transformations to be applied to images and, if they are, may not be feasible. One of the participants gave the example of a transformation that would involve transforming a daytime image into a night-time image: such a transformation is very complex, difficult to automate, and therefore, very costly. Similar examples have been given in industrial cases where the tester alone cannot determine the visual effect of a given circumstance, and therefore, the associated transformation.

7.3. Conclusion and Lessons Learned

This study allowed us to determine how the proposed method could be useful to software testers in relation to the problems they currently encounter in practice. The focus group participants identified that the proposed method provides concrete answers to their practical issues from a software tester's point of view, although certain questions remain unanswered. The lessons learned from this study are summarized below.

Lesson 1: Need for circumstances analysis when testing the robustness of AI-based systems. The current practical issues mentioned by software testers are related to the application domain of the system under test and require the tester to rely on documentation or discussions with the development team or domain experts to determine the relevant cases to test. This confirms that this activity must be an integral part of the testing process in the same way as risk analysis in traditional software testing for high-risk systems.

Lesson 2: The choice of metrics is a matter for the AI tester's area of expertise. One of the lessons learned from this study is that AI testers need to be made aware of or trained in the choice of performance metrics for AI models and the measurement of distance between datasets. Indeed, the software testers interviewed believe that this should be part of an AI tester's area of expertise.

Lesson 3: More needs to be conducted to address the issue of complex, domain-related circumstances. According to the practitioners interviewed, there is currently no practical and cost-effective solution for AI-based systems whose circumstances cannot be expressed in the form of simple or automatable transformations. A current alternative is to acquire a dataset specifically for these circumstances of use, but this solution is also very costly and often unfeasible.

8. Threats to Validity

In this section, we report on the possible threats to validity that we have identified in this research and discuss how we mitigated them, following the recommendations of Wohlin et al. [49].

8.1. Construct Validity

In action research, the learning effect may be a threat to construct validity. We avoided the learning effect by using different cases and involving new practitioners in each cycle of the action research. To ensure the construct validity of the empirical evaluation, we have used an application case that we believe to be representative of a real situation of testing the robustness of an AI-based 2D object detection system. Measurement bias was also mitigated by using video recording and having an additional observer during the focus group session.

8.2. Internal Validity

To ensure that the results of the focus group session were representative of reality, we selected practitioners with different profiles in terms of experience and role within the company. In addition, to minimize the threat posed by the fact that one of the researchers and the team of practitioners interviewed worked in the same company, the latter was encouraged to criticize the proposed method, highlighting the difficulties and limitations encountered in its application and to detail their responses. Furthermore, none of the participants had previously contributed to the development of the method.

8.3. External Validity

In the action research, we used two different AI-based 2D object detection use cases that have two distinct application domains, and two different groups of people to apply the method, which allowed us to mitigate to some extent, the threat to external validity. Regarding the case study, the focus group adopted the viewpoint of an external tester to apply the proposed method and we do not intend to generalize the results beyond this scope.

8.4. Conclusion Validity

To improve the reliability of our research, we have used well-established research methods (e.g., action research and focus group method) and described in detail the procedures to conduct the research, following good practices.

9. Related Work

9.1. Standards, Guidelines and Methodologies for Testing AI Robustness

From best practices to emerging standards, several reference systems for the development and testing of AI-based systems have surged in recent years. At the international and European level, ISO/IEC, CEN-CENELEC and ETSI are among the standards organizations involved. In the following, we focus on the reference systems related to the robustness of AI-based systems.

The ISO/IEC JTC 1/SC 42 committee, which deals with standardization in the field of artificial intelligence, is developing the ISO/IEC 24029 family of standards on the robustness of systems using neural networks. This standard currently has three parts: ISO/IEC TR 24029-1:2021 [30] on an overview of the subject, published in 2021 as a 31-page document, ISO/IEC 24029-2:2023 [50] on the use of formal methods in this context, published in 2023 as a 23-page document, and ISO/IEC AWI 24029-3 [51] on the use of statistical methods, which is still under development. At the same time, this committee has been assigned to review the ISO/IEC TR 29119-11:2020 [52] standard in the field of software testing, which describes the test methods applicable to AI-based systems, the metrics that can be used, and which maps the various test stages to those of the life cycle of an AI-based system.

CEN and CENELEC have established the CEN-CENELEC JTC 21 in response to the European Commission White Paper [53] on AI. This committee is responsible for the development and adoption of standards for AI and related data, as well as providing guidance to other Technical Committees concerned with AI. In particular, it identifies and adopts international standards already available or under development from other organizations such as the ISO/IEC JTC 1/SC 42 committee.

For its part, the International Software Testing Qualifications Board (ISTQB) published a Syllabus [54] that serves as a basis for certification in AI testing. Among the topics covered in the syllabus are the various quality characteristics to be verified in AI-based systems, including robustness, and the methods, techniques, and environments for testing these systems.

AMLAS [55] is a methodology for the safety assurance of machine learning components in autonomous systems. It supports the development of safety cases in a systematic way. However, this high-level framework does not provide practical guidance on the choice

of test methods during the model verification stage or on the appropriate metrics to be used for the robustness assessment of a given system.

Although such reference systems help to structure progress in the field of AI testing and guide the development of new methods and tools, they often lack practical applications. Built to be applicable to different types of AI-based systems, they often result in high-level and generic guidelines. Furthermore, their complexity and inertia in the face of the emergence of new, more relevant methods make them difficult to use when selecting a suitable testing approach in practice.

9.2. Robustness Testing of AI-Based Object Detection Systems

Several methods have been proposed in the literature to test the robustness of AI-based models [14,15]. In computer vision, these methods usually involve measuring the performance of the model against perturbed input images that reflect uncertainties in its operational environment. Input images can be classified into two categories [15]: adversarial inputs and natural inputs. Adversarial inputs are crafted from perturbations that are imperceptible to humans but can cause AI models to make incorrect predictions [5]. Alternatively, natural inputs can be generated by applying common perturbations that simulate real-world scenarios, such as sensor failures or changes in the environment [6,7].

Given their nature and implications for the security of AI-based systems, the topic of adversarial input generation has received considerable attention in recent years, as highlighted by Akhtar et al. in two successive surveys [56,57]. This trend has also been observed in the field of AI-based 2D object detection systems, where numerous techniques have been proposed in the literature for generating adversarial inputs [9].

Regarding natural inputs, several methods have been proposed for defining perturbations and generating perturbed input images, guided by knowledge of the application domain [58], coverage metrics [59,60], or properties that the system must satisfy, such as metamorphic relations [22,61]. In addition, several perturbation benchmarks have been proposed in the literature to assess robustness [7,62–64]. However, few of these methods have been developed and evaluated in the context of 2D object detection systems. As mentioned in Section 2.2, Wang et al. [22] developed MetaOD, a tool based on metamorphic testing. It aims to verify that inserting objects into the background of an image does not change the results for the other objects. For their part, Zhao et al. [63] proposed a natural perturbation benchmark for testing models in the field of computer vision, including object detection models.

In contrast to related work, we do not aim for a specific application domain (e.g., autonomous vehicles) or a specific set of perturbations. Instead, we aim to fill the gap between academia and the industry, providing practitioners and researchers with practical guidance on the application of effective state-of-the-art methods and a set of relevant metrics to overcome issues encountered in practice, such as the elicitation of relevant perturbations.

10. Conclusions and Future Work

In this paper, we have presented the development of a method for assessing the robustness of AI-based 2D object detection systems. The goal of this method is to meet the need for clear guidelines identified in practice in previous work [12]. It was developed using action research, based on two industrial cases and feedback from practitioners. As a result of these experiments, we identified key steps in a testing process based on the metamorphic testing technique and a set of metrics to be used at each step.

An initial qualitative evaluation of the proposed method was conducted to assess its usefulness for practitioners and its ability to provide clear guidelines. The lessons learned from this study will also help to guide future work. In particular, we identified that more efforts should be concentrated on AI-based systems whose circumstances of use are difficult to translate into a set of simple, automatable transformations. Thus, further industrial case studies should be conducted to further improve the proposed method and complement the guidelines in this context.

In addition to the above, future work will involve the further implementation of a software tool to assist practitioners in using the method. An initial prototype has been developed to help with case studies, and initial feedback will enable us to continue our work on the subject. Other future work will also concern the specification phase in line with current risk analysis practices for critical software systems in order to deepen the analysis of target circumstances. In particular, we will study how they can be formulated at several levels of detail and address the question of the testability of circumstances.

Author Contributions: Conceptualization, A.-L.W., S.S. and R.M.; investigation, A.-L.W.; writing—original draft, A.-L.W.; writing—review and editing, S.S. and R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the French National Association of Research in Technology (ANRT) [CIFRE N°2020/0754]; grants PID2021-126227NB-C22 and PID2021-126227NB-C21, funded by MCIN/AEI/10.13039/501100011033/FEDER, UE and grant TED2021-131023B-C21, funded by MCIN/AEI/10.13039/501100011033 and by European Union "NextGenerationEU"/PRTR.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author. A publicly available dataset was used in one of the industrial case evaluated in this study. A description of the dataset is provided in [25] and the data can be found here: https://nihcc.app.box.com/v/DeepLesion/folder/51877983116 (accessed on 1 April 2024). Restrictions apply to the availability of the data used in the second industrial case presented in this article. Data were obtained from the company Lacroix and are not publicly available. A description of the dataset is provided in [12].

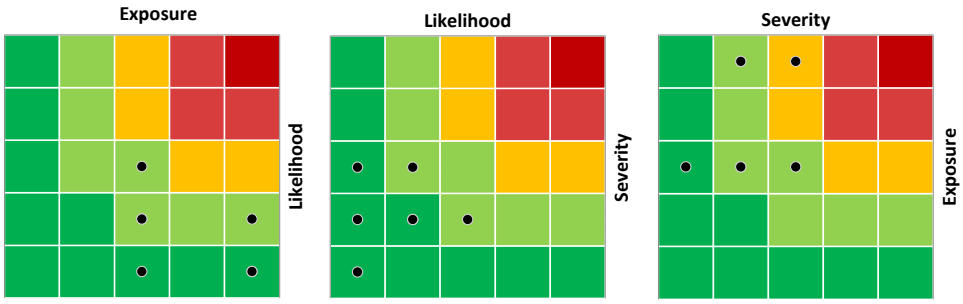
Conflicts of Interest: Author Anne-Laure Wozniak was employed by the company Kereval. The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A

1. Specification

ID	Circumstance	Probability	Exposure	Likelihood	Severity	Significance
1	Density contrast may vary significantly	0.15	3	2	3	18
2	Same pixel lines may be shifted	0.2	3	3	2	18
3	The cross-section may be truncated	0.15	5	2	2	20
4	Images may be blurred due to motion blur	0.1	5	1	3	15
5	Images may be blurred (Gaussian blur)	0.1	3	1	3	9
6	There may be salt and pepper noise in the images	0.1	3	1	1	3
7	The cross-sectional image may be rotated between -20° and 20°	0.1	5	1	2	10
8	Image resolution may be altered during preprocessing	0.1	3	2	2	12

Insert new lines before this one.



2. Test Cases Generation (results)

Missing circumstances (%): **63%**
Circumstances over/under - represented (%): **100%**
Total coverage of circumstances (%): **0%**

ID	Circumstance	Probability	Significance	Relative Frequency
3	The cross-section may be truncated	0.15	20	0.1
1	Density contrast may vary significantly	0.15	18	0.05
2	Same pixel lines may be shifted	0.2	18	0
4	Images may be blurred due to motion blur	0.1	15	0.05
8	Image resolution may be altered during preprocessing	0.1	12	0
7	The cross-sectional image may be rotated between -20° and 20°	0.1	10	0
5	Images may be blurred (Gaussian blur)	0.1	9	0
6	There may be salt and pepper noise in the images	0.1	3	0

3. Evaluation

MEASURED PERFORMANCE

Metric	Perf. on source dataset	Perf. on target dataset	ΔP
Precision	0.564	0.498	**0.066**
Recall	0.510	0.384	**0.126**
mAP50	0.526	0.414	**0.112**
mAP75	0.396	0.264	**0.132**
mAP	0.340	0.246	**0.094**

Insert new lines before this one.

MEASURED DISTANCE

Metric	Distance d
CW-SSIM	**0.154**

PIECEWISE LINEAR FUNCTION PARAMETERS

IF d <	Param a	Param b
0.25	0	0.01
1	1	0

Insert new lines before this one.

ROBUSTNESS

Metamorphic relation: d <0.25 ⇒ ΔP <0.01

This property holds for **0** performance metrics.
The system is not robust with respect to the metamorphic relation.

Figure A1. Elements from the tooling, for the MDS case.

References

1. Wu, X.; Sahoo, D.; Hoi, S.C. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
2. Wang, M.; Deng, W. Deep face recognition: A survey. *Neurocomputing* **2021**, *429*, 215–244. [CrossRef]
3. Christin, S.; Hervet, E.; Lecomte, N. Applications of deep learning in ecology. *Methods Ecol. Evol.* **2019**, *10*, 1632–1644. [CrossRef]
4. Prunella, M.; Scardigno, R.M.; Buongiorno, D.; Brunetti, A.; Longo, N.; Carli, R.; Dotoli, M.; Bevilacqua, V. Deep Learning for Automatic Vision-Based Recognition of Industrial Surface Defects: A Survey. *IEEE Access* **2023**, *11*, 43370–43423. [CrossRef]
5. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
6. Dodge, S.; Karam, L. Understanding how image quality affects deep neural networks. In Proceedings of the 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, 6–8 June 2016; pp. 1–6.
7. Hendrycks, D.; Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In Proceedings of the 7th International Conference on Learning Representations, ICLR'19, New Orleans, LA, USA, 6–9 May 2019.
8. Drenkow, N.; Sani, N.; Shpitser, I.; Unberath, M. A systematic review of robustness in deep learning for computer vision: Mind the gap? *arXiv* **2021**, arXiv:2112.00639.
9. Serban, A.; Poll, E.; Visser, J. Adversarial Examples on Object Recognition: A Comprehensive Survey. *ACM Comput. Surv.* **2020**, *53*, 1–38. [CrossRef]
10. Chen, T.Y.; Cheung, S.C.; Yiu, S.M. *Metamorphic Testing: A New Approach for Generating Next Test Cases*; Technical Report HKUST-CS98-01; Department of Computer Science, The Hong Kong University of Science and Technology: Hong Kong, China, 1998.
11. Segura, S.; Towey, D.; Zhou, Z.Q.; Chen, T.Y. Metamorphic Testing: Testing the Untestable. *IEEE Softw.* **2020**, *37*, 46–53. [CrossRef]
12. Wozniak, A.; Duong, R.; Benderitter, I.; Leroy, S.; Segura, S.; Mazo, R. Robustness Testing of an Industrial Road Object Detection System. In Proceedings of the 2023 IEEE International Conference On Artificial Intelligence Testing (AITest), Athens, Greece, 17–20 July 2023.
13. O'Brien, R.P. An overview of the methodological approach of action research. In *Theory and Practice of Action Research*; Richardson, R., Ed.; Universidade Federal da Paraiba: Joao Pessoa, Brazil, 2001.
14. Riccio, V.; Jahangirova, G.; Stocco, A.; Humbatova, N.; Weiss, M.; Tonella, P. Testing machine learning based systems: A systematic mapping. *Empir. Softw. Eng.* **2020**, *25*, 5193–5254. [CrossRef]
15. Zhang, J.M.; Harman, M.; Ma, L.; Liu, Y. Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE Trans. Softw. Eng.* **2022**, *48*, 1–36. [CrossRef]
16. Zaidi, S.S.A.; Ansari, M.S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A survey of modern deep learning based object detection models. *Digit. Signal Process.* **2022**, *126*, 103514. [CrossRef]
17. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2016**, arXiv:1506.02640.
19. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niterói, Brazil, 1–3 July 2020; pp. 237–242.
20. Segura, S.; Fraser, G.; Sanchez, A.B.; Ruiz-Cortés, A. A Survey on Metamorphic Testing. *IEEE Trans. Softw. Eng.* **2016**, *42*, 805–824. [CrossRef]
21. Chen, T.Y.; Kuo, F.C.; Liu, H.; Poon, P.L.; Towey, D.; Tse, T.H.; Zhou, Z.Q. Metamorphic Testing: A Review of Challenges and Opportunities. *ACM Comput. Surv.* **2018**, *51*, 1–27. [CrossRef]
22. Wang, S.; Su, Z. Metamorphic Object Insertion for Testing Object Detection Systems. In Proceedings of the 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, VIC, Australia, 21–25 September 2020; IEEE Computer Society: Washington, DC, USA, 2020; pp. 1053–1065.
23. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
24. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October 2019–2 November 2019; pp. 6568–6577.
25. Yan, K.; Wang, X.; Lu, L.; Summers, R.M. DeepLesion: Automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *J. Med Imaging* **2018**, *5*, 036501. [CrossRef]
26. Gilmore, T.; Krantz, J.; Ramirez, R. Action based modes of inquiry and the host-researcher relationship. *Consult. Int. J.* **1986**, *5*, 160–176.
27. Susman, G.I. Action research: A sociotechnical systems perspective. *Method Strateg. Soc. Res.* **1983**, *95*, 95–113.
28. Solingen, R.V.; Berghout, E.W. *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*; McGraw-Hill: New York, NY, USA, 1999.
29. Solingen, R.V.; Caldiera, V.R.; Basili, G.; Rombach, H.D. The goal question metric approach. In *Encyclopedia of Software Engineering*; Marciniak, J., Ed.; John Wiley and Sons: West Sussex, UK, 2002.

30. *ISO/IEC TR 24029-1:2021*; Artificial Intelligence (AI)—Assessment of the Robustness of Neural Networks—Part 1: Overview. International Organization for Standardization: Geneva, Switzerland, 2021.
31. *ISO/TS 14812:2022*; Intelligent Transport Systems—Vocabulary. International Organization for Standardization: Geneva, Switzerland, 2022.
32. *ISO 21448:2022*; Road Vehicles—Safety of the Intended Functionality. International Organization for Standardization: Geneva, Switzerland, 2022.
33. Zendel, O.; Murschitz, M.; Humenberger, M.; Herzner, W. CV-HAZOP: Introducing Test Data Validation for Computer Vision. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2066–2074.
34. Marijan, D.; Gotlieb, A.; Kumar Ahuja, M. Challenges of Testing Machine Learning Based Systems. In Proceedings of the 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 4–9 April 2019; pp. 101–102.
35. *ISO/IEC TR 29119-1:2022*; Software and Systems Engineering—Software Testing—Part 1: General Concepts. International Organization for Standardization: Geneva, Switzerland, 2022.
36. Kinney, G.F.; Wiruth, A. *Practical Risk Analysis for Safety Management*; Technical Report; Naval Weapons Center China Lake CA: Ridgecrest, CA, USA, 1976.
37. Du, X.; Wang, X.; Gozum, G.; Li, Y. Unknown-Aware Object Detection: Learning What You Don't Know From Videos in the Wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 13678–13688.
38. Yang, J.; Zhou, K.; Li, Y.; Liu, Z. Generalized Out-of-Distribution Detection: A Survey. *arXiv* **2022**, arXiv:2110.11334.
39. Salehi, M.; Mirzaei, H.; Hendrycks, D.; Li, Y.; Rohban, M.H.; Sabokrou, M. A Unified Survey on Anomaly, Novelty, Open-Set, and Out of-Distribution Detection: Solutions and Future Challenges. *arXiv* **2022**, arXiv:2110.14051.
40. Zhai, G.; Min, X. Perceptual image quality assessment: A survey. *Sci. China Inf. Sci.* **2020**, *63*, 1–52. [CrossRef]
41. Heusel, L.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017.
42. Yim, O.; Ramdeen, K.T. Hierarchical cluster analysis: Comparison of three linkage measures and applications to psychological data. *Quant. Methods Psychol.* **2015**, *11*, 8–21. [CrossRef]
43. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef]
44. Sampat, M.P.; Wang, Z.; Gupta, S.; Bovik, A.C.; Markey, M.K. Complex wavelet structural similarity: A new image similarity index. *IEEE Trans. Image Process.* **2009**, *18*, 2385–2401. [CrossRef]
45. Runeson, P.; Höst, M.; Rainer, A.; Regnell, B. *Case Study Research in Software Engineering: Guidelines and Examples*; John Wiley & Sons, Ltd.: Hoboken, NJ, USA, 2012.
46. Kontio, J.; Bragge, J.; Lehtola, L. The Focus Group Method as an Empirical Tool in Software Engineering. In *Guide to Advanced Empirical Software Engineering*; Springer: London, UK, 2008; pp. 93–116.
47. Yin, R. *Case Study Research: Design and Methods*, 3rd ed.; SAGE Publications: Southend Oaks, CA, USA, 2003.
48. *IEC 61508*; Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. International Electrotechnical Commission: Geneva, Switzerland, 2010.
49. Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A. *Experimentation in Software Engineering*; Springer: Berlin/Heidelberg, Germany, 2012.
50. *ISO/IEC TR 24029-2:2023*; Artificial Intelligence (AI)—Assessment of the Robustness of Neural Networks—Part 2: Methodology for the use of formal methods. International Organization for Standardization: Geneva, Switzerland, 2023.
51. *ISO/IEC AWI 24029-3:2023*; Artificial Intelligence (AI)—Assessment of the Robustness of Neural Networks—Part 3: Methodology for the use of statistical methods. International Organization for Standardization: Geneva, Switzerland, 2023.
52. *ISO/IEC TR 29119-11:2020*; Software and Systems Engineering—Software Testing—Part 11: Guidelines on the Testing of AI-Based Systems. International Organization for Standardization: Geneva, Switzerland, 2020.
53. European Commission White Paper. *On Artificial Intelligence—A European Approach to Excellence and Trust*; White Paper; European Commission: Brussels, Belgium, 2020.
54. Syllabus. *Certified Tester AI Testing (CT-AI)*; Syllabus; International Software Testing Qualifications Board (ISTQB): Edinburgh, UK, 2021.
55. Hawkins, R.; Paterson, C.; Picardi, C.; Jia, Y.; Calinescu, R.; Habli, I. Guidance on the assurance of machine learning in autonomous systems (AMLAS). *arXiv* **2021**, arXiv:2102.01564.
56. Akhtar, N.; Mian, A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey. *IEEE Access* **2018**, *6*, 14410–14430. [CrossRef]
57. Akhtar, N.; Mian, A.; Kardan, N.; Shah, M. Advances in Adversarial Attacks and Defenses in Computer Vision: A Survey. *IEEE Access* **2021**, *9*, 155161–155196. [CrossRef]
58. Rubaiyat, A.H.M.; Qin, Y.; Alemzadeh, H. Experimental resilience assessment of an open-source driving agent. In Proceedings of the 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), Taipei, Taiwan, 4–7 December 2018; pp. 54–63.

59. Pei, K.; Cao, Y.; Yang, J.; Jana, S. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. In Proceedings of the 26th Symposium on Operating Systems Principles, SOSP'17, Shanghai, China, 28–31 October 2017; pp. 1–18.
60. Tian, Y.; Pei, K.; Jana, S.; Ray, B. DeepTest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th International Conference on Software Engineering, ICSE'18, Gothenburg, Sweden, 27 May–3 June 2018; pp. 303–314.
61. Zhang, Z.; Wang, P.; Guo, H.; Wang, Z.; Zhou, Y.; Huang, Z. DeepBackground: Metamorphic testing for Deep-Learning-driven image recognition systems accompanied by Background-Relevance. *Inf. Softw. Technol.* **2021**, *140*, 106701. [CrossRef]
62. Michaelis, C.; Mitzkus, B.; Geirhos, R.; Rusak, E.; Bringmann, O.; Ecker, A.S.; Bethge, M.; Brendel, W. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming. *arXiv* **2020**, arXiv:1907.07484.
63. Zhao, B.; Yu, S.; Ma, W.; Yu, M.; Mei, S.; Wang, A.; He, J.; Yuille, A.; Kortylewski, A. OOD-CV: A Benchmark for Robustness to Out-of-Distribution Shifts of Individual Nuisances in Natural Images. In Proceedings of the Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022; Proceedings, Part VIII; Springer: Berlin/Heidelberg, Germany, 2022; pp. 163–180.
64. Dong, Y.; Kang, C.; Zhang, J.; Zhu, Z.; Wang, Y.; Yang, X.; Su, H.; Wei, X.; Zhu, J. Benchmarking Robustness of 3D Object Detection to Common Corruptions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 1022–1032.