applied system innovation

MDPI

Review

Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers

Nils Hütten *, Miguel Alves Gomes, Florian Hölken, Karlo Andricevic, Richard Meyes and Tobias Meisen

Chair of Technologies and Management of Digital Transformatio, School of Electrical, Information and Media Engineering, University of Wuppertal, Rainer-Gruenter-Straße 21, 42119 Wuppertal, Germany; alvesgomes@uni-wuppertal.de (M.A.G.); hoelken@uni-wuppertal.de (F.H.); karlo.andricevic@uni-wuppertal.de (K.A.); meyes@uni-wuppertal.de (R.M.); meisen@uni-wuppertal.de (T.M.)
* Correspondence: nhuetten@uni-wuppertal.de

Abstract: Quality assessment in industrial applications is often carried out through visual inspection, usually performed or supported by human domain experts. However, the manual visual inspection of processes and products is error-prone and expensive. It is therefore not surprising that the automation of visual inspection in manufacturing and maintenance is heavily researched and discussed. The use of artificial intelligence as an approach to visual inspection in industrial applications has been considered for decades. Recent successes, driven by advances in deep learning, present a possible paradigm shift and have the potential to facilitate automated visual inspection, even under complex environmental conditions. For this reason, we explore the question of to what extent deep learning is already being used in the field of automated visual inspection and which potential improvements to the state of the art could be realized utilizing concepts from academic research. By conducting an extensive review of the openly accessible literature, we provide an overview of proposed and in-use deep-learning models presented in recent years. Our survey consists of 196 open-access publications, of which 31.7% are manufacturing use cases and 68.3% are maintenance use cases. Furthermore, the survey also shows that the majority of the models currently in use are based on convolutional neural networks, the current de facto standard for image classification, object recognition, or object segmentation tasks. Nevertheless, we see the emergence of vision transformer models that seem to outperform convolutional neural networks but require more resources, which also opens up new research opportunities for the future. Another finding is that in 97% of the publications, the authors use supervised learning techniques to train their models. However, with the median dataset size consisting of 2500 samples, deep-learning models cannot be trained from scratch, so it would be beneficial to use other training paradigms, such as self-supervised learning. In addition, we identified a gap of approximately three years between approaches from deep-learning-based computer vision being published and their introduction in industrial visual inspection applications. Based on our findings, we additionally discuss potential future developments in the area of automated visual inspection.

Keywords: automated visual inspection; industrial applications; deep learning; computer vision; convolutional neural network; vision transformer

check for updates

Citation: Hütten, N.; Alves Gomes, M.; Hölken, F.; Andricevic, K.; Meyes, R.; Meisen, T. Deep Learning for Automated Visual Inspection in Manufacturing and Maintenance: A Survey of Open-Access Papers. *Appl. Syst. Innov.* **2024**, *7*, 11. https://doi.org/10.3390/asi7010011

Academic Editor: Mario Di Nardo

Received: 15 November 2023
Revised: 27 December 2023
Accepted: 16 January 2024
Published: 22 January 2024

1. Introduction

Industrial production and maintenance are under constant pressure from increasing quality requirements due to rising product demands, changing resources, and cost specifications. In addition, there are constantly changing framework conditions due to new and changing legal requirements, standards, and norms. Ultimately, the increasing general flow of information via social media and other platforms leads to an increased

risk of reputational damage from substandard products. These influences, combined with the fact that quality assurance is still predominantly performed or supported by human inspectors, have led to the need for advances in continuous quality control. Since vision is the predominant conscious sense of humans, most inspection techniques in the past and even today are of a visual nature [1]. However, manual visual inspection (VI) has several drawbacks, which have been studied, for example, by Steger et al., Sheehan et al., and Chiang et al. [2–4], specifically including high labor costs, low efficiency, and low real-time performance in the case of fast-moving inspection objects or large surface areas. According to Swain and Guttmann [5], minimal error rates of 10^{-3} can be reached for very simple accept/reject tasks. Though highly dependent on the inspection task, Drury and Fox [6] observed error rates of 20% to 30% in more complex VI tasks in their studies. In addition, decreasing efficiency and accuracy occur during human inspection due to fatigue and resulting attention deficits.

As a way to counteract these effects, automation solutions were pushed in the 1980s. The goal was to increase efficiency and performance and reduce costs while minimizing human error. Accordingly, computer vision (CV) methods were introduced to VI, which was initially only relevant for the automation of simple, monotonous tasks. In the beginning, they served more as a support for inspectors [7], but as development progressed, whole tasks were solved without human involvement. This was the beginning of automated visual inspection (AVI).

With deep learning (DL) becoming the technology of choice in CV since 2010, showing better generalization and less sensitivity to application conditions than traditional CV methods, new models have reached performance levels that even surpass those of humans in complex tasks like image classification, object detection, or segmentation [8–10]. Accordingly, the development and use of such DL techniques for use in industrial AVI are reasonable and understandable. In order to provide an overview of the current state of research and development and to understand what the current focus is, we provide answers to the following guiding questions in our study.

1. What are the requirements that have to be considered when applying DL-based models to AVI?
2. Which AVI use cases are currently being addressed by deep-learning models?
3. Are there certain recurring AVI tasks that these use cases can be categorized into?
4. What is the data basis for industrial AVI, and are there common benchmark datasets?
5. How do DL models perform in these tasks, and which of them can be recommended for certain AVI use cases?
6. Are recent state-of-the-art (SOTA) CV DL models used in AVI applications, and if not, is there untapped potential?

From these questions, we derive key insights and challenges for AVI and give an outlook on future research directions that we believe will have a positive impact on the field in industry as well as research. From these questions, we further derive the structure for our study as follows. In Section 2, we describe our literature research process and the constraints we defined. This is followed by Section 3, where we derive requirements for DL-based AVI and categorize the surveyed literature based on the use case as well as on how these use cases are solved to answer questions one to three. Section 4 deals with the evaluation of the industrial approaches and data characteristics, summarizes developments in academic research, and identifies promising models and methods from it that are not yet utilized in application use cases (questions 3, 4, and 5). Section 5 summarizes our survey and gives an outlook on possible future research directions.

2. Methodology of Literature Research

In our work, we analyzed the SOTA by surveying recent papers solving AVI use cases with DL-based approaches. Figure 1 succinctly illustrates the employed literature research methodology for our comprehensive review. Therefore, we followed the systematic literature research procedure proposed by Vom Brocke et al. [11] and adhered to their

recommended process, starting with the investigation of survey papers on VI with DL to justify the necessity of our research. We used scholarly search engines like Google Scholar, Web of Science (WOS), and Semantic Scholar and found many related survey publications on VI with DL. However, all publications put their focus on DL-based VI in specific industry sectors or methodologies but not on industrial VI as a whole.

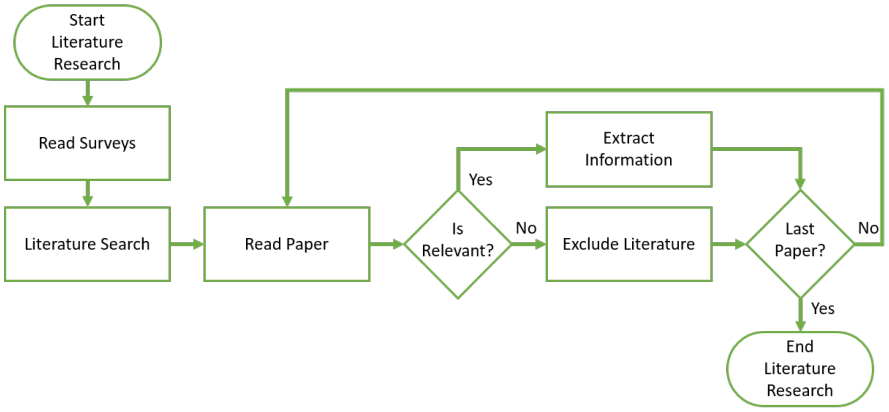

Figure 1. Flow chart of the literature research process.

Zheng et al. [12], e.g., focused their survey on surface defect detection with DL techniques. The same applies to the surveys by Jenssen et al. [13], Sun et al. [14], Yang et al. [15], Nash et al. [16], Liu et al. [17], and Donato et al. [18], who concentrated on different industry sectors, like steel production, railway applications, power lines, or manufacturing in general. There are also many publications in the areas of civil engineering and structural health monitoring. Ali et al. [19], Ali et al. [20], Hamishebahar et al. [21], and Intisar et al. [22] studied DL-based approaches to crack detection. Chu et al. [23], Qureshi et al. [24], and Ranyal et al. [25] reviewed the general SOTA in the AVI of pavements and roads, while Kim et al. [26] focused more specifically on the detection of potholes. A wider scope is covered by Zhou et al. [27] and Hassani et al. [28], as they consider structural health monitoring as a whole, not just cracks or pavements and roads. Chew et al. [29] highlighted the consideration of falling facade objects in their review, while Luleci et al. [30] emphasized the application of generative adversarial networks (GANs). Mera et al. [31] surveyed the literature on class imbalance for VI. Tao et al. [32] covered a wide range of industrial applications but restricted their survey to unsupervised anomaly detection.

Similarly, Rippel et al. [33] reviewed the literature with a focus on anomaly detection with DL in AVI. In addition to an overview, they also discuss the advantages and disadvantages of the reviewed anomaly detection approaches.

The only survey we found that deals with VI as a whole, i.e., not only one VI use case or industry sector, was written in 1995 by Newmann et al. [34]. However, this work was published over 20 years ago and did not deal with DL and, therefore, does not cover new challenges that emerged with the era of digitization.

Given the lack of a domain-overarching perspective on VI, we gathered relevant publications utilizing the WOS online research tool, as it is recognized as the leading scientific citation and analytical platform and lists publications across a wide area of knowledge domains [35]. The research field we are interested in is an intersection of two topics: DL-based CV and AVI in industrial use cases. Thus, we gathered search terms with the intention of covering the most important aspects of each of the two areas. Table 1 lists all defined search terms. We formulated search queries to find publications that contain at least one term from each of the two categories and conducted our literature search on 4 March 2023.

Table 1. Search terms used for literature research.

Category	Search Terms
DL-based computer vision	Deep Learning, Neural Network, Convolutional Neural Network, CNN, Transformer, Semantic segmentation, Object Detection
Automated visual inspection in industrial use cases	Industrial Vision Inspection, Industrial Visual Inspection, Vision Inspection, Visual Inspection, Damage Detection, Damage Segmentation, Error Detection

Furthermore, we defined additional constraints to refine the results of the query. The considered time range started from 1 January 2010 in order to also include less high-profile publications before the first largely successful CV model "AlexNet" was proposed in 2012. Furthermore, we only considered open-access publications written in English to be transparent and comprehensible in our work by only using references accessible to the scientific community. The query with the aforementioned constraints resulted in 6583 publications. As a next refinement step, we excluded all science categories that are not associated with an industrial context (detailed listing in Table A1). After filtering, we obtained exactly 808 publications for further investigation. For the next research step, we read the publications, and in that process, we defined them as relevant or not relevant based on set constraints. The constraints for a publication being relevant to our survey are manifold and include that the authors described their approach appropriately with information about the task, the method, the used data, and performance. Moreover, we also restricted ourselves to publications that use 2D image data and in which an industrial context is clearly identifiable. The decision to exclusively consider publications that employ AVI on 2D imagery is motivated by the widespread availability and affordability of cameras compared to alternative devices, such as hyperspectral or 3D cameras as well as light detection and ranging (LiDAR) devices.

For example, this excludes publications that deal with a medical context, remote sensing, or autonomous driving. All these constraints resulted in a publication corpus of 196 publications that we investigated further.

3. Categorization of Visual Inspection (Tasks)

In this section, we first discuss the term VI in more detail, as it is used in our publication corpus to lay the foundation for our further analysis. Next, we derive a hierarchy, which structures all VI use cases from the gathered literature (Section 3.2). In addition to the VI use case, we also group the publications by the AVI task, with the aim of categorizing them by the methodology that is used to solve them. These are closely related to the CV tasks classification, object detection, and segmentation. These AVI tasks provide the structure for the following Sections 3.3.1 to 3.3.4. In each subsection, we review and analyze publications that aim to automate the corresponding AVI task using DL-based methods.

In general, VI describes the visual assessment of the state of a manufacturing product or assembly, a vehicle, a building, or infrastructure and its comparison to specified requirements. The publication corpus shows two application contexts in which VI is employed: maintenance and manufacturing. In manufacturing, the inspection is executed, e.g., after critical points during or at the end of manufacturing processes, like the final machining step of a component responsible for key functions of the overall system or the point at which two partial assemblies are combined into one. By doing so, it is confirmed that the finished product meets the predefined quality standards [34,36]. In a maintenance context, See et al. [37] define VI as the periodical monitoring of features that indicate a certain state of the inspected object that impairs its functionality or operational safety and can lead to additional negative impacts like injury, fatality, or the loss of expensive equipment.

3.1. Requirements for Deep-Learning Models in AVI

Several requirements have to be considered when introducing DL-based AVI to a previously manual inspection process or even to an already automated process that uses

classical CV methods. To answer our second question, *"What are the requirements that have to be considered when applying DL-based models to AVI?"*, we analyzed our publication corpus with regard to requirements, with either a direct textual mention or indirect mention through evaluation procedures, as well as reported metrics. These requirements can be grouped in two dimensions: on the one hand, between general and application- or domain-specific requirements or, on the other hand, between hard technical and soft human factors. The most general technical challenge is performance, as visualized in Figure 2.

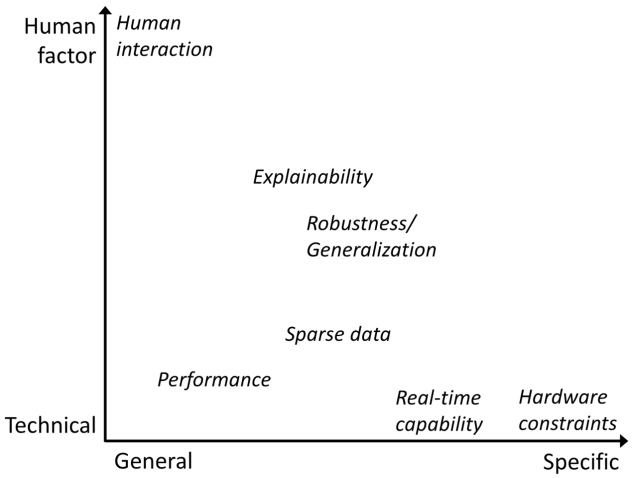

Figure 2. AVI requirements grouped by their combined properties with regard to specificity and whether they are technical-factor- or human-factor-driven.

In the case of the automation of previously manual processes, human-level performance is usually used as a reference value, which is intended to guarantee the same level of safety, as mentioned by Brandoli et al. for aircraft maintenance [38]. If the target process is already automated, the DL-based solution needs to prevail against the established solution. Performance can be measured by different metrics, as some processes are more focused on false positives (FPs), like the one investigated by Wang et al. [39], or false negatives (FNs). Therefore, it cannot be considered a purely general requirement, as it is affected by the choice of metric. Real-time capability is a strictly technical challenge, as it can be defined by the number of frames per second (FPS) a model can process but is mostly specific, as it is mainly required when inspecting manufactured goods on a conveyor belt or rails/streets from a fast-moving vehicle for maintenance [39–42]. Hardware constraints are the most specific and rare technical challenge found in our publication corpus. This usually means that the models have to run on a particular edge device, which is limited in memory, the number of floating-point operations per second (FLOPS), or even the possible computational operations it can perform [43]. Sparse (labeled) data are primarily a technical challenge, where the emphasis is put on the fact that models with more parameters generally perform better but require more data samples to optimize those parameters, as well. The labeling process introduces the human factor into this requirement because a consistent understanding of the boundary between different classes is necessary in order to produce a coherent distribution of labels with as few non-application-related inconsistencies or outliers as possible. This is especially true if there are few samples and if multiple different persons create the labels. Models need to perform well with these small labeled industrial datasets [44–48] or, even better, work without labeled data [49,50]. One of the key advantages of DL-based models compared to classic CV methods is their generalization capability, which makes them robust against partly hidden objects, changing lighting conditions, or new damage types. This characteristic is required for many use cases where it is not possible to enforce controlled conditions or have full visibility, such as rail track inspection [42,51,52], or it is an added benefit when a model is able to extrapolate to previously unseen damages [53]. As this requirement is not easily quantifiable and

application-specific to a certain degree, we place it centrally in both dimensions. Part of any industrial transformation process is the people involved, whether they are directly affected as part of the process or indirectly affected through interfaces with the process. To improve the acceptance of change processes, it is necessary to convince domain experts that they can trust the new DL solution. In addition, explainability can also be helpful from a model development perspective to determine the reason for certain model confusions that lead to errors [54].

3.2. Overview of Visual Inspection Use Cases

In order to answer our second guiding question, *"Which AVI use cases are currently being addressed by DL models?"*, we examined the reviewed publications to determine whether it is possible to summarize the solved VI tasks into specific use cases. We identified a hierarchy of VI use cases based on the surveyed literature, that visualized in Figure 3.

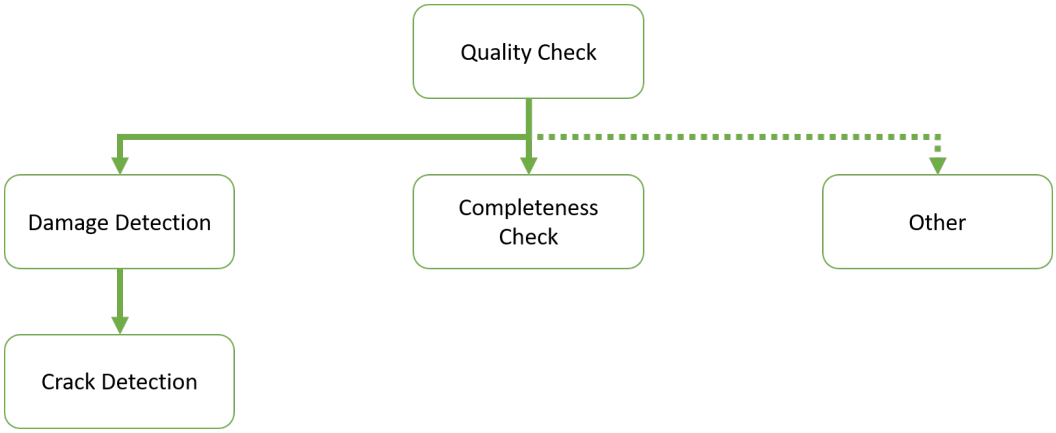

Figure 3. Hierarchical structure of top-level VI use cases based on the surveyed literature.

As previously mentioned, VI is getting more challenging due to ever-increasing quality requirements, and all use cases can be considered to be at least **quality inspection**. In our literature review, quality inspection use cases are those that do not detect defects or missing parts but the state of an object. For example, determining the state of woven fabrics or leather quality is a use case we considered to be only quality inspection [55,56]. **Damage detection**, also referred to as defect detection in the literature, summarizes all VI use cases that classify or detect at least one type of damage. An example of damage detection use cases is the surface defect detection of internal combustion engine parts [57] or the segmentation of different steel surface defects [58]. **Crack detection** can be considered a specialization of damage detection use cases and has its own category because of its occurrence frequency in the surveyed literature. The crack detection use case deals solely with crack classification, localization, or segmentation. The application context is usually the maintenance of public buildings, for example, pavement cracks [59,60] or concrete cracks [61,62]. In addition to detecting defects, another VI use case is to check whether a part is missing or not. **Completeness check** summarizes these use cases. A completeness check can be the determination of whether something is missing, or to the contrary, the determination of whether something is present. O'Byrne et al. [63] proposed a method to detect barnacles on ship hulls. Another example is provided by Chandran et al. [51], who propose a DL approach to detect rail track fasteners for railway maintenance. The last VI use case class we defined as **other**, which includes VI use cases that cannot directly be seen through only quality inspection and are not of the damage detection or completeness check type. Example use cases are plant disease detection [64,65] or type classification [66]. Figure 4 shows the distribution of the VI use cases over the investigated literature. Most publications (53.57%) deal with damage detection use cases. The second most (27.55%) researched VI use case is crack detection, followed by quality inspection (6.63%) as well

as other use cases (6.63%), and the least occurring type is completeness check use cases (5.61%).

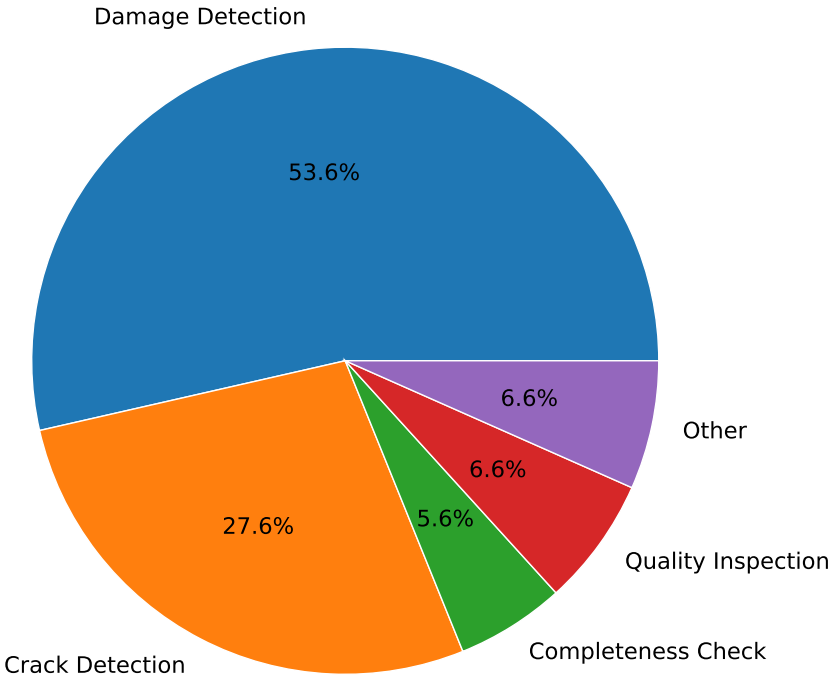

Figure 4. Distribution of reviewed publications by VI use cases.

3.3. Overview on How to Solve Automated Visual Inspection with Deep-Learning Models

In the following, we aim to answer our third guiding question, *"Are there certain recurring AVI tasks that these use cases can be categorized into?"*, by investigating with which DL approach the VI use cases can be solved. For this, we determined four different AVI tasks to categorize the approaches. Each of these tasks aims to answer one or more questions about the inspected object. Binary classification tries to answer the question, Is the inspected object in the desired state? This applies mainly to accept/reject tasks, like separating correctly produced parts from scrap parts, regardless of the type of deficiency. Multi-class classification goes one step further, trying to answer the question, In which state is the inspected object? By additionally identifying the type of deficiency, it is possible to, e.g., distinguish between parts that are irreparably damaged and parts that can still be reworked to pass the requirements or determine the rework steps that are necessary. Localization further answers the question, Where do we find this state on the inspected object? This adds information about the locality of a state of interest, as well as enabling the finding of more than one target. It can be utilized, e.g., to check assemblies for their completeness. The fourth AVI task, multi-class localization, answers the question, Where do we find which state on the inspected object? For example, the state of a bolt can be present, missing, rusty, or cracked. Thus, the set of states is not fixed and depends, among other things, on application-specific conditions, as well as on the object under inspection.

These four AVI tasks are closely related to the three most common CV tasks, image classification, object detection, and segmentation, which are visualized in Figure 5.

In image classification, the goal is to assign a corresponding label to an image. Object detection is performed by a method or model that searches for objects of interest. Usually, the object is indicated by a rectangular bounding box, and simultaneously, object classification is performed for each object. Unlike pure classification, multiple objects can be detected and classified. Image segmentation is the process of separating every recognizable object into corresponding pixel segments. This means that both classification AVI tasks are performed by image classification models, while both localization tasks are performed by either an object detection model or a segmentation model.

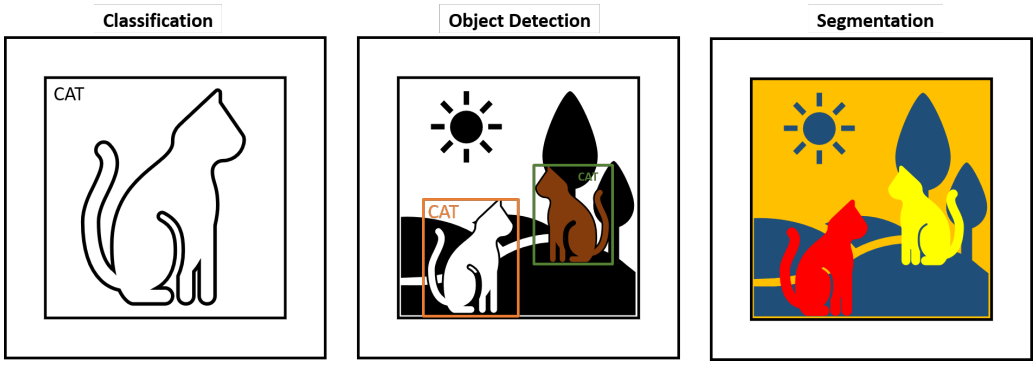

Figure 5. Visualization of the three different CV tasks—classification, object detection with two bounding boxes, and segmentation.

Figure 6 shows the composition of our publication corpus with regard to the application context, industry sector, and AVI task.

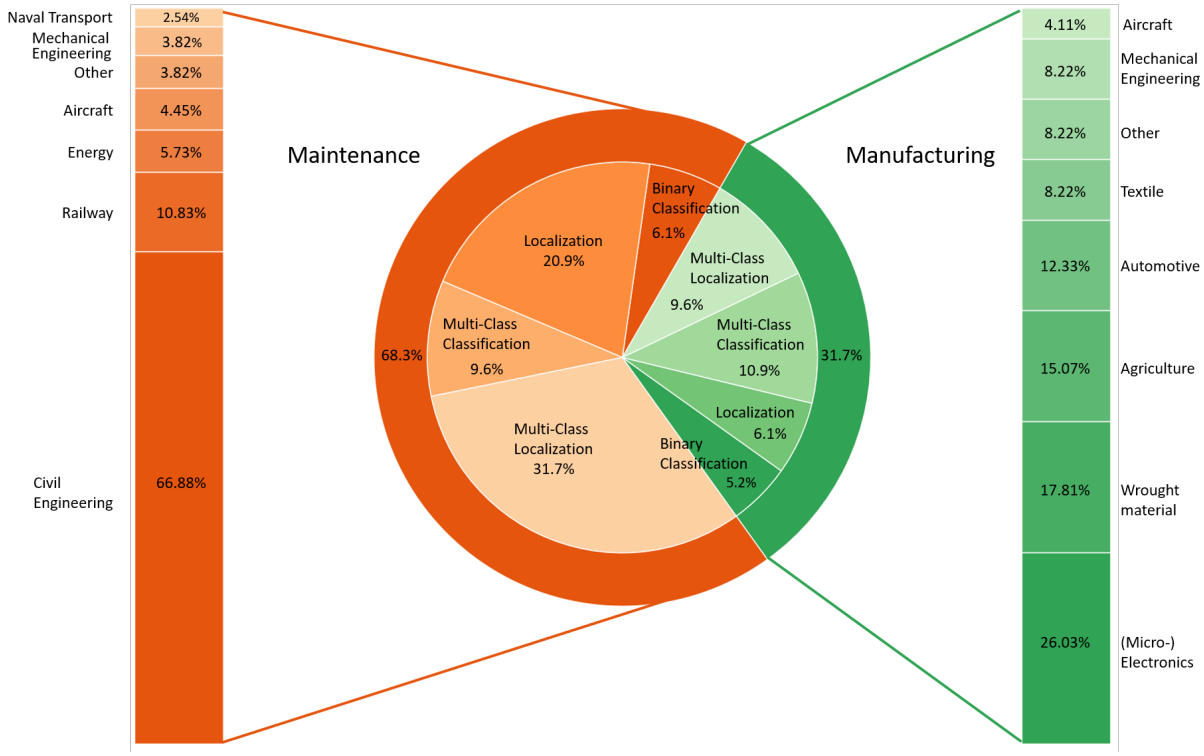

Figure 6. Distribution of reviewed publications by inspection context, VI task, and associated industrial sector.

The number of papers in the maintenance context outweigh those addressing manufacturing by two to one, as depicted by the outer pie chart in the center. Each of those contexts is associated with several different industrial sectors in which AVI is applied. The share of the industry sectors in each context is plotted on the left- and right-hand sides. The biggest shares in maintenance are held by the fields of civil engineering, railway, energy, and aircraft. These sum up to a total of 87.89% of all maintenance publications. The manufacturing sectors, (micro-) electronics, wrought material, agriculture, automotive, and textiles, add up to a total of 79.46% of all manufacturing papers. In addition to the industry sectors, we also group the applications per context by the AVI task. The distribution of VI tasks for each industry context is visualized by the inner pie chart. For maintenance applications, 77.01% of their total 68.3% is covered by basic and multi-class localization tasks. Only 15.7% of the tasks can be attributed to classification tasks. In manufacturing, the VI tasks are spread

across 16.1% classification and 15.7% localization publications. The multi-class variants are clearly more frequent for both, with 9.6% for localization and 10.9% for classification.

In the following subsections, one for each AVI task, we investigate the collected literature and utilized models. Only the best-performing architecture is mentioned if multiple are utilized. Models that are derived from established base architectures like Residual Networks (ResNet) [67] are still considered to belong to that architecture family unless they are combined or stacked with another architecture. We also subsumed all versions of the "you only look once" (YOLO) architecture [68] under YOLO. Models that are custom designs of the authors and not based on any established architectures are subsumed under the categories multi-layer perceptron (MLP), convolutional neural network (CNN), or Transformer based on their main underlying mechanisms.

3.3.1. Visual Inspection via Binary Classification

In the surveyed literature, 21 publications describe a way to approach AVI with binary image classification; these are summarized in Table 2. Following the general trend of VI use cases, damage detection is addressed ten times with binary classification.

Adibhatla et al. [50] used a ResNet, Selmaier et al. [46] used an Xception architecture, and Jian et al. [69] used a DenseNet to classify whether damage is visible or not. Crack detection is addressed seven times with binary classification. In four publications, the authors propose a CNN architecture for AVI crack detection. In the other two publications, crack detection was performed with an AlexNet or a Visual geometry group model (VGG). Ali et al. [70] proposed a sliding window vision transformer (ViT) as a binary classifier for crack detection in pavement structures. Binary classification is also utilized for completeness checks and plant disease detection (other). For plant disease detection, Ahmad et al. [64] used an MLP, while O'Byrne et al. [63] used a custom CNN for a completeness check use case.

Table 2. Overview of VI use cases and models that solve these problems via binary classification.

VI Use Case	Model	Count	References
Crack Detection	AlexNet	1	[71]
	CNN	4	[72–75]
	VGG	1	[76]
	ViT	1	[70]
Damage Detection	AlexNet	1	[77]
	CNN	1	[78]
	DenseNet	3	[38,69,79]
	Ensemble	1	[80]
	MLP	1	[81]
	ResNet	1	[50]
	SVM	1	[82]
	Xception	1	[46]
Quality Inspection	AlexNet	1	[83]
	MLP	1	[84]
Other	MLP	1	[64]
Completeness Check	CNN	1	[63]

3.3.2. Visual Inspection via Multi-Class Classification

Table 3 presents an overview of 42 publications that solve various use cases of AVI through multi-class classification and the models that are used to solve them. The models used to solve these use cases include popular DL architectures such as AlexNet, CNN, DenseNet, EfficientNet, GAN, MLP, MobileNet, ResNet, single-shot detector (SSD), and VGG. Twenty-one publications describe approaches for damage detection, of which six approaches are based on custom CNNs. The other four authors used ResNet-based architec-

tures. Kumar et al. [85] proposed an MLP architecture to perform damage detection. Also, an EfficientNet and a single-shot detector (SSD) were employed for multi-class damage detection. Five publications cover crack detection use cases. For example, Alqahtani [86] used a CNN, and Elhariri et al. [87] as well as Kim et al. [88] used a VGG. Also, DL models like ResNet, DenseNet, and an ensemble architecture are proposed by some authors. Completeness checks were performed with the help of a ResNet by Chandran et al. [51] or an SSD, as shown by Yang et al. [89]. In seven publications, the authors used custom CNNs, DenseNet, ResNet, or VGG in quality inspection use cases. Also, other use cases can be addressed by different DL-based CV models or MLPs.

Table 3. Overview of VI use cases and models that solve these use cases via multi-class classification.

VI Use Case	Model	Count	References
Crack Detection	AlexNet	1	[90]
	CNN	1	[86]
	EfficientNet	1	[91]
	ResNet	1	[92]
	VGG	2	[87,88]
Damage Detection	AlexNet	1	[93]
	CNN	6	[66,94–98]
	CNN LSTM	1	[99]
	EfficientNet	2	[100,101]
	Ensemble	1	[56]
	GAN	2	[102,103]
	MLP	1	[85]
	MobileNet	1	[104]
	ResNet	4	[105–108]
	VGG	2	[60,109]
Completeness Check	ResNet	1	[51]
	SSD	1	[89]
Quality Inspection	CNN	3	[110–112]
	DenseNet	1	[113]
	ResNet	2	[55,114]
	VGG	1	[115]
Other	CNN	1	[65]
	EfficientNet	1	[116]
	MLP	1	[117]
	MobileNet	1	[118]
	ResNet	1	[54]
	VGG	1	[119]

3.3.3. Visual Inspection via Localization

As previously mentioned, localization is used to detect where an object of interest is located. Table 4 summarizes which VI use cases are addressed with localization and the appropriate models. In a total of 50 publications, localization was employed for AVI. Contrary to classification approaches, crack detection is the most addressed VI use case, with a total of 26 publications investigating it. The most utilized approach for crack detection is the CNN, which was applied in eight publications. Furthermore, in three other publications, extended CNN architectures were used. Kang et al. [120] introduced a CNN with an attention mechanism, and Yuan et al. [121] used a CNN with an encoder–decoder architecture. Andrushia et al. [122] combined a CNN with a long short-term memory cell (LSTM) to process the images recurrently for crack detection. Among custom CNN approaches, six authors used UNet to detect cracks, mostly in public constructions. Damage detection via localization occurred 16 times and was addressed with at least twelve different DL-based models. Three authors decided to approach it with DL-based models of the Transformer family. For example, Wan et al. [123] utilized a Swin-Transformer to

localize damages on rail surfaces. Completeness checks can be executed with YOLO and/or regional convolutional neural networks (RCNNs). Furthermore, YOLO can be used for vibration estimation, as shown by Su et al. [124]. Oishi et al. [125] proposed a Faster RCNN to localize abnormalities on potato plants.

Table 4. Overview of VI use cases and models that solve these use cases via localization.

VI Use Case	Model	Count	References
Crack Detection	CNN	8	[47,59,62,126–130]
	CNN LSTM	1	[122]
	Attention CNN	1	[120]
	Custom encoder–decoder CNN	1	[121]
	DeepLab	3	[131–133]
	Ensemble	3	[134–136]
	Fully convolutional network (FCN)	2	[137,138]
	Faster RCNN	1	[139]
	UNet	6	[140–145]
Damage Detection	DenseNet	1	[146]
	Faster RCNN	1	[147]
	GAN	1	[148]
	Mask RCNN	2	[149,150]
	ResNet	1	[48]
	SSD	1	[151]
	Swin	1	[123]
	Transformer	1	[152]
	UNet	3	[153–155]
	VAE	1	[49]
	ViT	1	[156]
	YOLO	2	[157,158]
Completeness Check	Mask RCNN	1	[159]
	YOLO	1	[160]
Quality Inspection	YOLO	1	[161]
Other	Faster RCNN	1	[125]
	UNet	2	[44,162]
	YOLO	2	[124,163]

3.3.4. Visual Inspection via Multi-Class Localization

The majority of the literature reviewed used multi-class localization for VI. In 83 publications, it is shown how to approach different use cases, like crack or damage detection, with multi-class localization. Table 5 provides a detailed overview. As for the two classification approaches, damage detection is the most investigated VI use case, with 58 publications. Therein, YOLO and Faster RCNNs are the two most used models, with over ten publications. They are followed by CNNs and Mask RCNN models, which are utilized more than five times. FCN, SSD, and UNet can also be used as approaches to multi-class damage detection. Huetten et al. [164] conducted a comparative study of several CNN models highly utilized in AVI and three vision transformer models, namely, detection transformer (DETR), deformable detection transformer (DDETR), and Retina-Swin, on three different damage detection use cases on freight cars. Multi-class localization was used in 15 publications for crack detection. In five publications, the authors performed crack detection with a YOLO model. Crack detection can also be performed with AlexNet, DeepLab, FCN, Mask RCNN, and UNet, which was shown in different publications. In three different publications, the authors show how to conduct quality inspection with YOLO. YOLO can be used in a tobacco use case, as well (other), as shown by Wang et al. [165].

Table 5. Overview of VI use cases and models that solve these use cases via multi-class localization.

VI Use Case	Model	Count	References
Crack Detection	AlexNet	1	[166]
	DeepLab	2	[61,167]
	FCN	1	[168]
	Mask RCNN	5	[169–173]
	UNet	1	[174]
	YOLO	5	[175–179]
Damage Detection	CNN	7	[180–186]
	DETR	1	[187]
	EfficientNet	1	[188]
	FCN	3	[189–191]
	FCOS	1	[192]
	Faster RCNN	10	[193–202]
	Mask RCNN	6	[53,203–207]
	MobileNet	1	[39]
	RCNN	1	[45]
	SSD	5	[208–212]
	Swin	1	[164]
	UNet	4	[58,213–215]
	VGG	1	[216]
	YOLO	16	[41,57,217–230]
Completeness Check	CNN	1	[231]
	Ensemble	1	[232]
	Faster RCNN	2	[233,234]
	YOLO	2	[42,52]
Quality Inspection	YOLO	3	[40,235,236]
Other	YOLO	1	[165]

4. Analysis and Discussion

In this section, we focus on answering guiding questions three, *"What is the data basis for industrial AVI and are there common benchmark datasets?"*; four, *"How do deep learning models perform on these tasks and which of them can be recommended for certain AVI use cases?"*; and five, *"Are recent SOTA CV deep learning models used in AVI application, and if not, is there untapped potential?"*. We sought answers to these questions by analyzing the performance of the models as well as comparing the SOTA in DL-based AVI applications and academic CV research. First, we look into the datasets and learning paradigms that are utilized. This is followed by an analysis of the performance of the applied DL models in the AVI tasks they have been categorized into. This section is concluded by summarizing developments in academic research and identifying promising models and methods from it.

4.1. Inspection Context and Industrial Sectors

There are many different industry sectors that apply AVI in maintenance and manufacturing. In maintenance, the clear majority (66.88%) of publications are from the civil engineering sector, while, in manufacturing, the distribution of a similar total share is more even, with 26.03% from (micro-) electronics, 17.81% from wrought material production, 15.07% from agriculture, and 12.33% from automotive. This may be due to the fact that many manufacturing applications were already targeted for automation with classic CV, as the environmental conditions that affect image quality, as well as variance, are easier to control, so the expected process improvements were generally lower. Most maintenance use cases are performed outdoors in varying light, weather, and seasonal conditions. Therefore, DL-based CV was needed to address use cases in the civil engineering or railway sector, such as bridge, road, or building facade inspection, as well as rail surface, fastener, or catenary inspection. Looking at the preferred AVI tasks, we see a clear majority of 77.01% for localization in maintenance, while, in manufacturing, the shares of classification (50.47%)

and localization (49.53%) are evenly split. The reason for this is that, in maintenance, it is usually necessary to assess a whole system with many different parts, so the need for localization arises, while, in manufacturing, it is easier to limit the inspection to individual parts, which is also possible with classification.

4.2. Datasets and Learning Paradigms

The dataset dimensions vary from 45 images [48] to more than 100,000 images [81,94,183,231], with the average number of images at 14,614 and a median of 1952. Figure 7 visualizes the number of datasets grouped by the decimal power of their dimension, ranging from less than 100 to less than 1,000,000. The two largest groups cover the intervals [101, 1000] and [1001, 10,000], with most of the datasets tending toward the lower boundary of their respective group. These two groups also feature the most open-access datasets: in total, 77/243 (31.69%) of the papers used at least one open-access dataset.

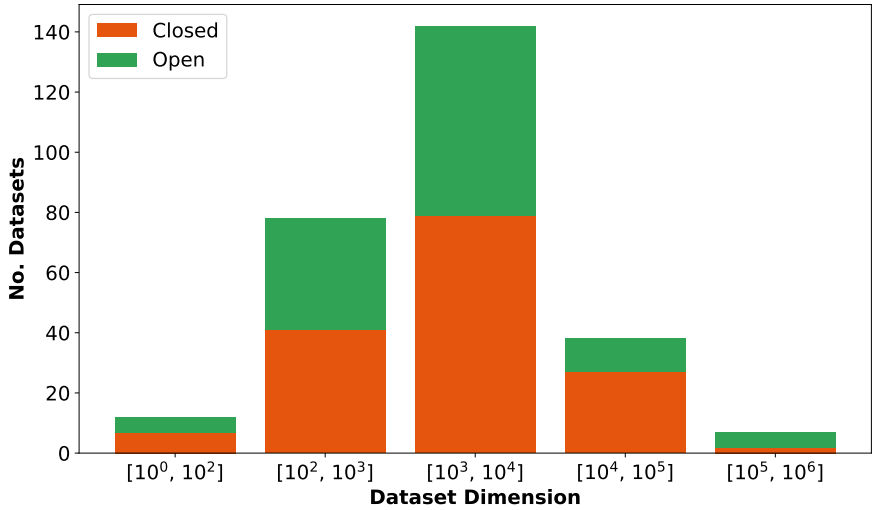

Figure 7. Distribution of dataset dimensions (number of samples) utilized by publications in our publication corpus.

Another important factor in relation to datasets is the uniformity of the distribution of their samples across the classes, often formulated more negatively as class imbalance. Class imbalance should be considered in learning processes since it can result in incorrect classification, detection, or segmentation. Therefore, imbalance in datasets has to be quantified differently across different learning tasks. For classification, each sample is directly associated with only one class. Object detection may feature more than one class per image, several instances of the same class, or no annotated class at all. So, to quantify the imbalance of an object detection dataset, the number of annotated objects, as well as the number of images featuring no objects, is required. Dataset imbalance for segmentation tasks can be quantified in the same way as for object detection, but the most exact measure would be to evaluate the pixel areas covered by each class in relation to the overall pixel area of the dataset, which is only reported by Neven et al. and Li et al. [58,121]. Given the number of classes n, the total number of samples, and the number of samples per class, we propose to quantify the balance in a dataset by Equation (1). The distance between the hypothetical balanced class ratio and the actual ratio between samples of a particular class and the total number of samples is used to quantify the imbalance for this class. The Euclidean distance is chosen as the distance metric for numerical stability reasons. Summing up these distances for each class yields zero for a perfectly balanced dataset and one for a completely imbalanced dataset, so subtracting it from one achieves a metric measuring the balance.

$$\text{Balance} = 1 - \sum_{c=1}^{n} \sqrt{\left(\frac{samples_c}{samples_{total}} - \frac{1}{n}\right)^2} \quad (1)$$

In total, our publication corpus encompasses 244 results produced on 204 unique datasets. For 47 publications, there is no detailed information on the training data, while 54 do not specify their test data, and in 44 cases, neither training nor testing data are sufficiently described. This means that, despite many authors providing open access to their data bases, very few researchers in the same field make use of them. This may be either due to available datasets being deemed insufficient in size, label precision, image quality, or other reasons or because they are not advertised enough and therefore not recognized by peer researchers. Overall, this makes the results less comparable and harder to reproduce and hinders further development of existing research results by others, which is detrimental to the progress of AVI.

Figure 8 visualizes the relationship between model performance measured by the F1-score and the balance score defined in the last section. We chose to only include publications in this plot that report the F1-score because it is the only reported metric that deals well with balanced as well as imbalanced datasets. The chart on the left shows a mostly linear correlation between balance and performance in classification tasks, with a few upward and downward outliers. These are predominantly from the multi-class versions. Binary classification has a lot more publications based on entirely balanced datasets.

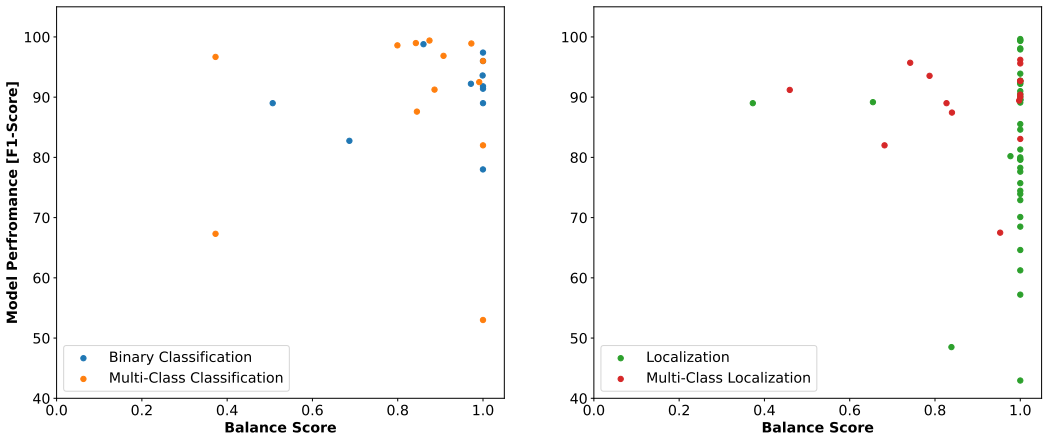

Figure 8. Performance of models in classification (**left**) and localization tasks (**right**) plotted against the balance score.

The first thing that stands out for the localization tasks depicted in the right graph is the accumulation of (binary- or single-class) localization datasets with a balance score of one. This is primarily attributable to the fact that it is seldom specified whether and how many images without inspection targets are contained in the datasets, specifically for segmentation. The general correlation between class balance and performance is visible as well, but the gradient of an imaginary regression line would be slightly lower compared to classification. This can be explained by the higher resilience of localization models to dataset imbalance based on the property that there are generally fewer areas of interest or foreground in an image than background, and in this task, they have to be explicitly marked.

In our literature review, we found that 14 datasets were utilized more than once. Table 6 lists them in descending order by the number of publications in which they have been used. Most of them are concerned with enforcing a binary differentiation between damaged and intact or background, respectively, via the computer vision tasks classification or segmentation (SDNET 2018, CFD, RSSDD, Deep Crack, Crack Tree, Özgenel crack dataset, Crack 500, Crack LS 315, Aigle RN). This places them at the bottom end regarding class-induced complexity. In many cases, framing the learning like this may be the only way of making it manageable at 500 or fewer samples with a certain performance goal in

mind. The cost of data acquisition and annotation certainly plays a role, as well. Lastly, the applications all come from a maintenance context, where the basic recognition of damages in the field can be a satisfactory first step, and a more detailed inspection will be performed by the personnel repairing the damage.

Table 6. Characteristics of benchmark datasets in automated visual inspection in industrial applications. Pixel area percentages are based on [121].

Dataset Name	# Samples	Resolution	Learning Task	Class Distribution	B-Score	# Publications	References
NEU Surface Defect Database	1800	200 × 200	Classification	Rolled-in Scale 300 Patches 300 Crazing 300 Pitted Surface 300 Inclusion 300 Scratches 300	1.0	9	[237]
SDNET 2018	56,000	256 × 256	Classification	Crack 8484 Intact 47,608	0.51	6	[238]
Crack Forest Dataset (CFD)	118	480 × 320	Segmentation	Crack 118	0.32	5	[239]
Road Damage Dataset 2018	9054	600 × 600	Object Detection	Longitudinal Crack, Wheel Mark 2768 Longitudinal Crack, Construction Joint 3789 Lateral Crack 742 Lateral Crack, Construction Joint 636 Alligator Crack 2541 Rutting, Bump, Pothole 409 Cross-Walk Blur 817 White-Line Blur 3733	0.75	5	[211]
GRDDC 2020	21,041	600 × 600, 720 × 720	Object Detection	Longitudinal Crack 8242, Laterial Crack 5480, Alligator Crack 10613, Pothole 7008	0.85	4	[240]
Rail Surface Defect Dataset (RSDD)	195	1024 ×*	Segmentation	Defect 195	-	4	[241]
Severstal Dataset	87,995	256 × 256	Segmentation	Holes 1820 Scratches 14576 Rolling 2327 Intact 69,272	0.37	4	[242]
Deep Crack	537	544 × 384	Segmentation	Crack 3.54% Background 96.46%	0.34	3	[243]
Crack Tree	206/260	800 × 600	Segmentation	Crack 206/1.91% Background -/98.09%	0.32	3	[243,244]
Özgenel Crack Dataset	40,000	227 × 227	Classification	Crack 20,000 Intact 20,000	1.0	3	[245]
Crack 500	500	2560 × 1440	Segmentation	Crack 4.33% Background 95.67%	0.35	2	[246]
Crack LS 315	315	512 × 512	Segmentation	Crack 1.69% Background 98.31%	0.32	2	[243]
Aigle RN	38	311 × 462, 991 × 462	Segmentation	Crack 38	-	2	[247]
Magnetic Tile Surface Dataset	1344	196 × 245	Segmentation	Blowhole 115 Break 85 Crack 57 Fray 32 Uneven 103 Intact 952	0.40	2	[248]

* variable image height

Statements about the imbalance of the segmentation datasets are possible in different levels of detail. Li et al. [121] report the pixel area percentages of cracks and background for DeepCrack, CrackTree 260, Crack LS 315, and Crack 500, which are very pronounced, with the highest balance being 0.35 for Crack 500. The balance reaches as low as 0.32 for Crack LS 315 and CrackTree, which poses a significant challenge compared to anything deemed feasible in classification problems. CFD, RSDD, and Aigle RN do not allow a statement to be made, as there is no information about images without damage, the number of class instances, or the area ratios between classes. The two classification datasets SDNET 2018 and Özgenel crack dataset are far more balanced, with scores of 0.51 and 1.0, as well as offering significantly more data samples.

The Severstal and Magnetic Tile Surface segmentation datasets have higher class-induced complexity, with four and six classes compared to only two. The extracted information about them only allows for an instance-based evaluation of balance, which leads to balance scores that are slightly higher compared to the crack segmentation datasets, with 0.37 and 0.40, respectively. They are also both from the production context in quality assurance applications, where the focus may be not only on recognizing any damages but also on categorizing them in more detail to be able to draw conclusions about process parameters and ultimately improve the quality in the future. The NEU surface dataset is also from the steel industry like the Severstal dataset but with a completely balanced class distribution at six total classes, albeit with only 1800 samples at a similar resolution.

GRDDC 2020 and Road Damage Dataset 2018 offer a more nuanced view of road damage with four and eight classes, respectively. Despite their higher number of classes, they are a lot less imbalanced compared to the binary crack segmentation datasets, but this is only partly due to the learning task of object detection being pursued.

A clear recommendation can be made for Crack 500 as a crack segmentation benchmark, as it offers a comparatively high balance score, the most data samples for this specific CV task, and the highest-resolution images, as well. This means it is possible to increase the number of samples even further by subdividing them, which may be necessary to reduce the computational cost or use an efficient training batch size, depending on the used model. For crack classification, the Özgenel crack dataset seems to be recommendable, as it is well balanced and has a tolerable size difference from SDNET. In all other areas, a targeted combination of datasets, such as the NEU surface defect database and Severstal dataset for steel surface detection, to compensate for each other's weaknesses appears to be the best solution.

The majority of industrial AVI use cases are based on datasets that are, at most, 10% of the size of the Microsoft common objects in context benchmark dataset (MS COCO) or 1% of the size of the ImageNet benchmark dataset. One would therefore expect that most authors resort to methods that do not require labeled training data or strategies that reduce the number of training samples required.

Still, the vast majority of the surveyed literature covers supervised learning applications (97.37%), with only three publications employing unsupervised anomaly detection [48,50,125] and one using semi-supervised learning [130]. Regarding the training processes employed, we can identify three groups. The first group, which accounts for 49.32%, utilizes transfer learning; i.e., the models are initialized with weights from one of the major CV challenges, and in most cases, an implementation from one of the many available open-source repositories is also used. This reduces the training time as well as lowers the number of required samples for a sufficiently large training dataset and thus the effort to create such a dataset. The second group, which makes up 30.59% of the publication corpus, trains their models from scratch. This results in a much larger need for training data and training time but is also the only way if the author creates their own architecture, which 21.0% of the authors did. In the third group, which is made up of the remaining 20.09%, there is no indication in the publication of whether the model/s were pretrained or not. In none of the publications using datasets with more than 50,000 samples did the authors utilize transfer learning. This seems to be a viable approach since the datasets are sufficiently large. At the same time, a comparison with transfer learning approaches would be insightful since the underlying models have been pretrained on much larger datasets. However, it is counterintuitive and incomprehensible that transfer learning from benchmarks, or the addition of samples from an open dataset from the same domain, was not used in 20 of the 68 cases with fewer than 1000 samples. Five of the eight classification models in this group performed below the median compared to other models with their respective architectures, while only five of the twelve localization models are in this performance range. So, there is no clear indication of whether transfer learning would have improved performance in these cases. In addition to transfer learning, self-supervised or unsupervised learning could prove useful since they alleviate the effort of labeling.

4.3. Performance Evaluation by AVI Task

The performance of the DL models utilized is not as easily comparable because of the heterogeneity of reported metrics as well as datasets. In total, ten different metrics were reported, namely, accuracy (Acc), precision (Pre), recall (Rec), F1-score (F1), true-negative rate (TNR), true-positive rate (TNR), mean average precision (mAP), mean average precision at 50% as well as 75% intersection over union (mAP50, mAP75), and mean intersection over union (mIoU). Figure 9 shows how often a certain metric was utilized for applications that use classification, object detection, and segmentation (from top to bottom).

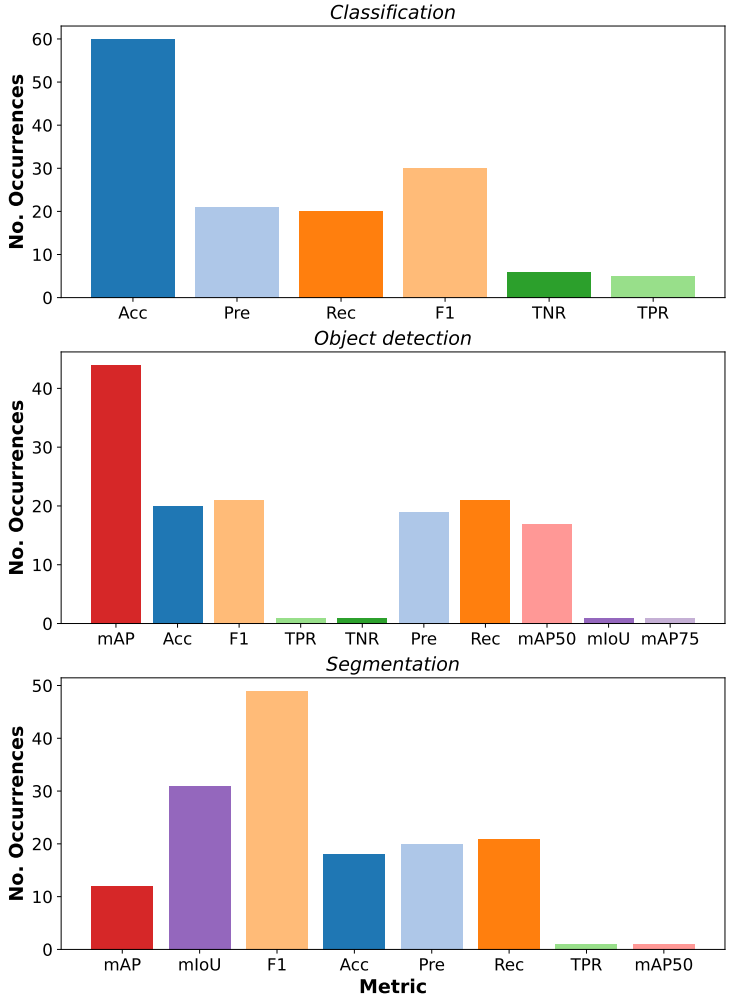

Figure 9. Distribution of metrics used in papers in our publication corpus grouped by CV task.

The first thing that stands out is the fact that accuracy is still the most common metric for classification tasks, despite having low descriptive quality, especially with imbalanced datasets, which are quite common, as stated in Section 4.2. The F1-score, which is the harmonic mean of precision and recall, is the second most common metric. Precision and recall are tied for third place. Among object detection use cases, the mean average precision (mAP) is the most reported metric, probably because it is also the official metric of the MS COCO object detection challenge. The F1-score and recall are also very relevant in object detection tasks, with the second-most occurrences in the surveyed literature, while accuracy is in third place. Segmentation tasks are predominantly evaluated by their F1-score and mean intersection over union (mIoU) between predictions and the ground truth. Recall is in the top three as well for the last CV task.

In the following, we will first analyze the performance of the models employed in the application use cases to derive recommendations for each AVI task based on the

most reported metrics. These are accuracy as well as F1-score for (binary and multi-class) classification and mAP as well as F1-score for (binary and multi-class) localization. If precision and recall were reported but F1-score was not, we calculated it based on them and take it into account as well. To be able to provide recommendations for segmentation models as well, we will also look at the mIoU, as this seems to be a metric reported almost exclusively for this CV task. After this, we will also look at the performance on the benchmark datasets from Table 6 and determine whether there are any differences.

Custom CNN LSTM networks show the best median accuracy, as depicted in Figure 10, but this does not have the highest expressive value, with a sample size of four and all results coming from the same paper [74].

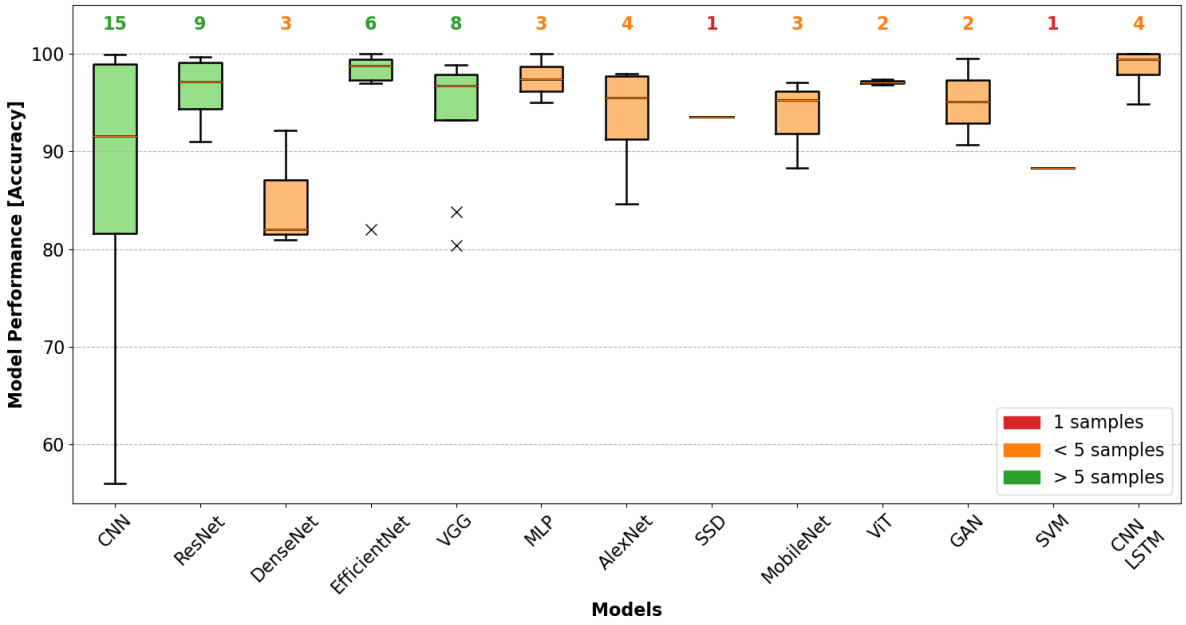

Figure 10. Distribution of model accuracy for all classification models reporting it as a metric in our publication corpus. × marks data points that have a distance of more than 1.5 times the interquartile range to the first or third quartile.

The same applies to MLPs and ViT, which show a higher median accuracy than all other models with two to four occurrences. Despite their good performance, no general recommendation can be given for MLPs because all of them employ hand-engineered input features. EfficientNet, on the contrary, is very close performance-wise but was applied to five datasets by four different authors. ResNet and VGG are the best two architectures, with considerable sample sizes of nine and eight, respectively. While the accuracy distribution for VGG shows less spread, meaning more consistent performance over different tasks, ResNet's median performance is almost as good. Custom CNNs have the third-worst median accuracy and a very large spread, as well. As this category contains very different architectures, this is to be expected. So, all in all, the best-performing choice for solving classification tasks based on accuracy as the metric from our publication corpus is EfficientNet.

When looking at the classification F1-scores in Figure 11, we generally see similar median values and a higher spread compared to accuracy.

This is unexpected but could be an indicator that most of the datasets do not show strong imbalances. As for accuracy, EfficientNet has the highest median F1-score, closely followed by ViT. MLPs have the third-highest median F1-score and the third-narrowest interquartile range. ResNet has the fourth-highest median F1-score as well as the third-largest interquartile range, but it is still the best model, with at least five occurrences. It is closely followed by custom CNNs and MobileNet with very similar median F1-scores but

also the largest spread for the former. VGG-type models show a much weaker performance when evaluated with the F1-score compared to accuracy. Based on our publication corpus, we would recommend using Efficient or ResNet models for classification tasks if one does not want or does not have the experience or expertise to create their own custom CNN model. ViT looks promising, as well, but has to be investigated more thoroughly to make a general statement.

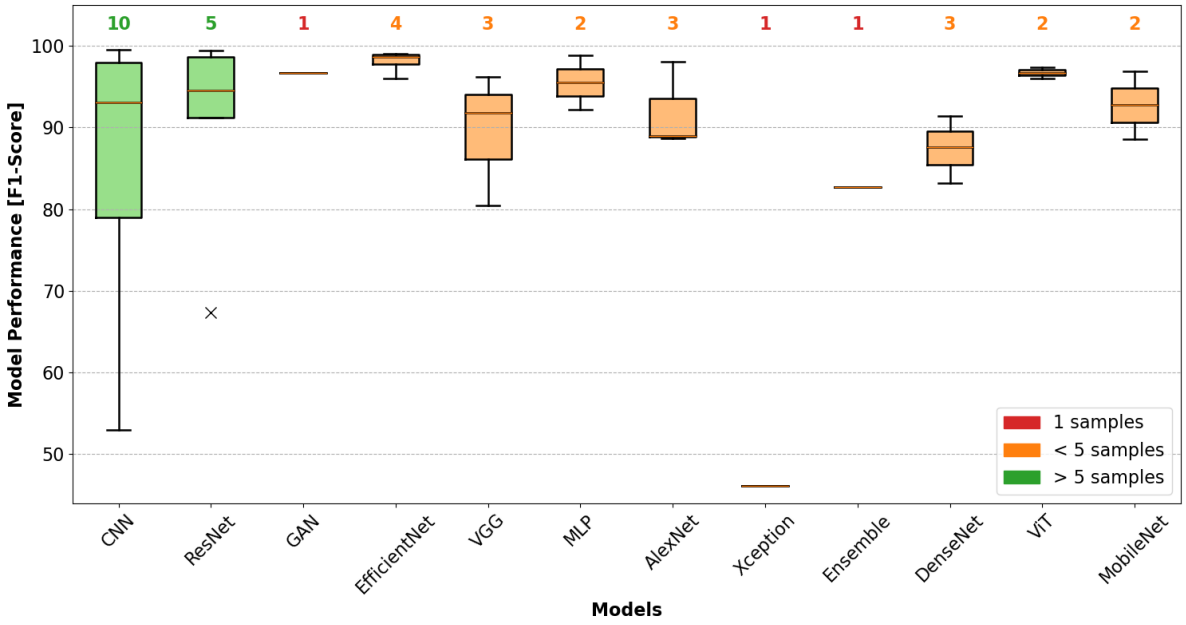

Figure 11. Distribution of model F1-scores for all classification models reporting it (or precision and recall) as a metric in our publication corpus. × marks data points that have a distance of more than 1.5 times the interquartile range to the first or third quartile.

The localization performance represented by mAP is visualized in Figure 12.

Mask RCNN models show the highest median mAP and the lowest interquartile range and spread among models with at least five occurrences. The two other models with a high mAP performance and at least five samples are YOLO and Faster RCNN. Faster RCNNs have a similar median mAP but a much larger interquartile range with a slightly higher min–max spread. As YOLO has more than twice as many occurrences as Faster RCNN, it seems reasonable to say it is the better model, especially the newest versions. It is interesting to see that the spread and interquartile range of Mask RCNN are so much smaller than those of Faster RCNN, as it is actually a Faster RCNN architecture with an added segmentation head. Multi-task training seems to show a benefit regarding better generalization. Custom CNN architectures perform much worse median-wise compared to classification tasks, even though they have a smaller performance spread than YOLO and Faster RCNN. SSD, UNet, DeepLab, FCN, ensemble models, AlexNet, VGG, DETR, and the fully convolutional one-stage object detection model (FCOS) have only two or fewer publications reporting mAP. Except for DETR and FCOS, the performance is within the mAP range of Mask RCNN or even above it, which indicates promise. However, the data basis is insufficient for a valid assessment.

When looking at the localization F1-scores in Figure 13, there are fewer models with only one reported performance and also one more with a sample size of five or above compared to mAP (Mask RCNN, YOLO, UNet, DeepLab, and CNN).

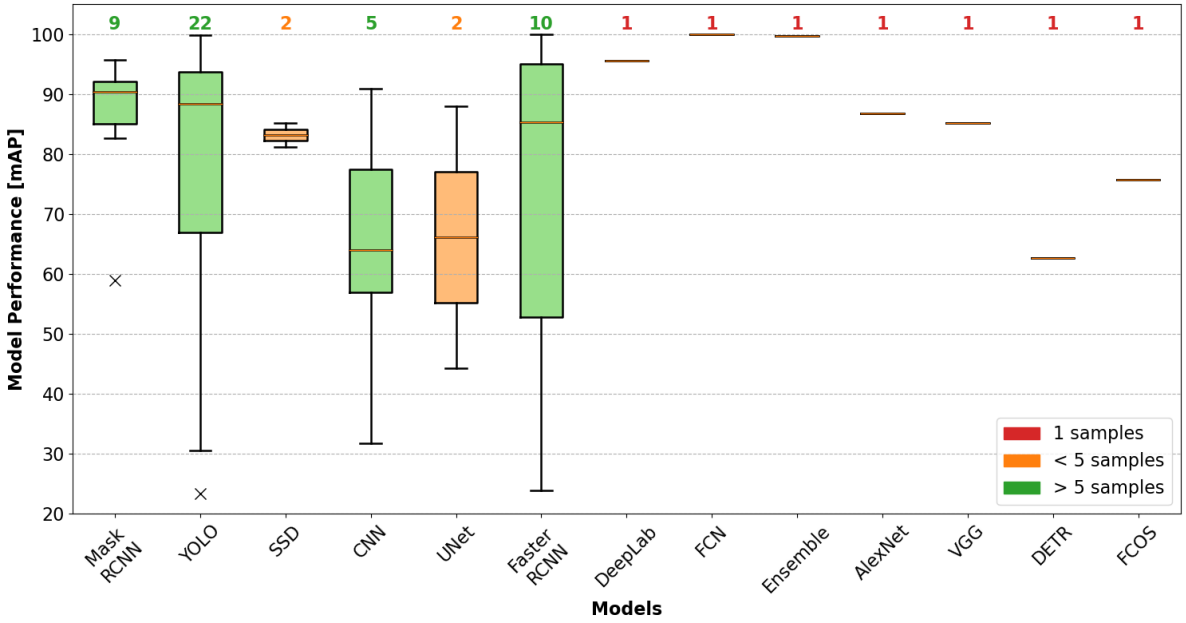

Figure 12. Distribution of model mAP for all localization models reporting it as a metric in our publication corpus. × marks data points that have a distance of more than 1.5 times the interquartile range to the first or third quartile.

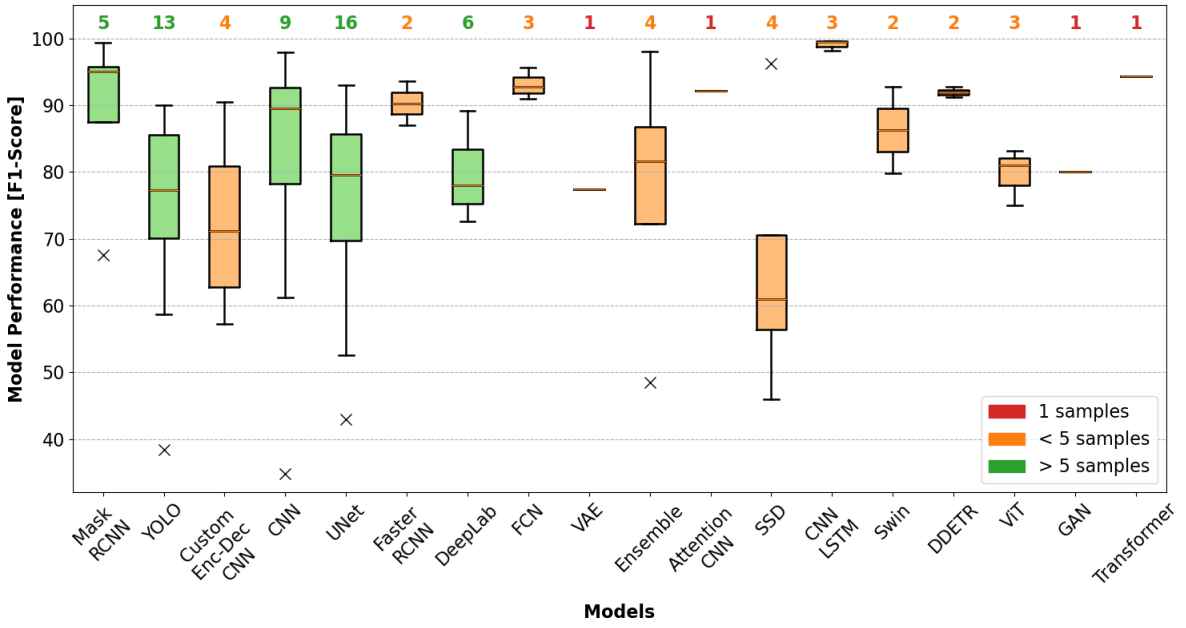

Figure 13. Distribution of model F1-scores for all localization models reporting it (or precision and recall) as a metric in our publication corpus. × marks data points that have a distance of more than 1.5 times the interquartile range to the first or third quartile.

Mask RCNN has the highest median F1-score as well as the lowest spread among those five and also the second-highest median overall. Custom CNNs show the second-highest median F1-score, as well as a smaller interquartile range and min–max spread, compared to the mAP metric. The next three models, UNet, DeepLab, and YOLO, are within 5% of their median F1-score, around 78%. UNet shows the highest median of the three, but also the largest interquartile range and min–max spread. YOLO's median performance is the lowest, with an interquartile range very similar to that of UNet but a slightly smaller min–max range. So, DeepLab seems to be the best of those three models, as

it features the second-highest median F1-score and the narrowest spread. Faster RCNN, FCN, attention CNN, CNN LSTM, DDETR, and custom transformer models have between 1 and 3 reported performances within the range of the best-performing model, Mask RCNN. Further investigations on those models will yield more general insight into their promising performance.

There are mostly segmentation models found in the mIoU boxplots in Figure 14, as it was reported as a metric for other CV tasks less than five times.

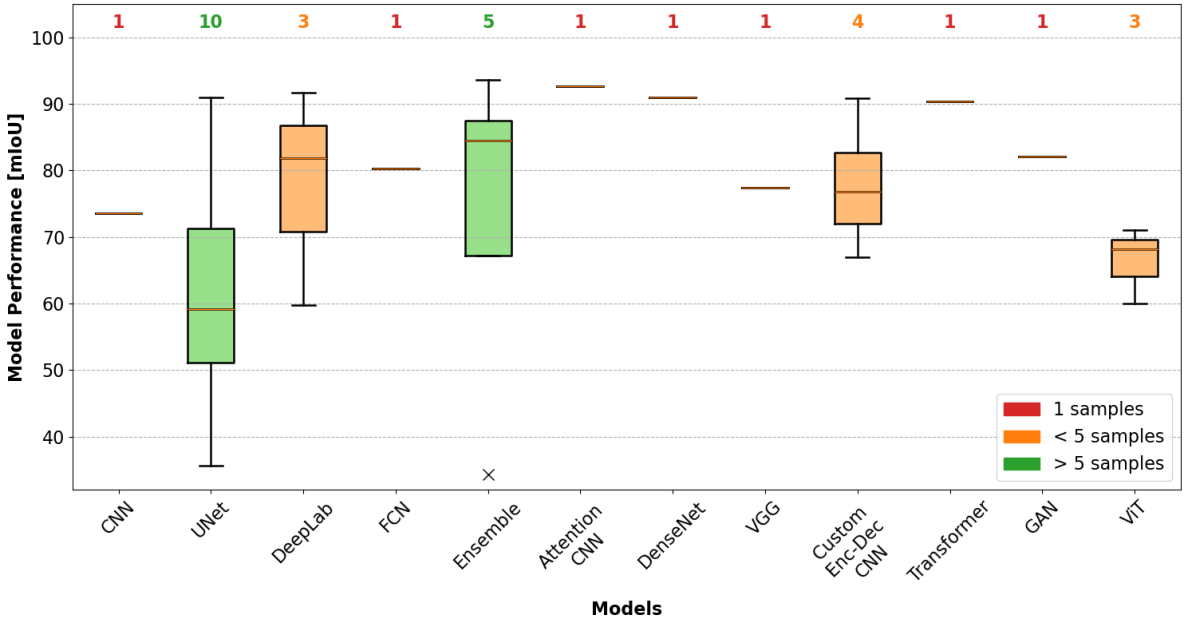

Figure 14. Distribution of model mIoU for all localization models reporting it as a metric in our publication corpus. × marks data points that have a distance of more than 1.5 times the interquartile range to the first or third quartile.

The reported values are generally lower than mAP and F1-score due to the higher difficulty of the segmentation task compared to object detection. There are also only five models with more than one sample, which are UNet, ensemble models, DeepLab, ViT, and custom encoder–decoder-CNNs. The best of those five regarding the median mIoU are ensemble models. DeepLab shows the second-highest median mIoU with a slightly larger min–max spread compared to ensemble models. Custom encoder–decoder CNNs achieve a median mIoU 5%p lower but with a slightly lower spread. The UNet model, which occurred the most, also has the worst performance regarding the median, as well as the interquartile range and min–max spread. Of the seven models with only one sample, an attention-enhanced CNN, DenseNet, and a custom transformer yield a performance that is more than 10%p better than the median of DeepLab. Further investigation of these architectures on different datasets seems promising.

Based on our surveyed literature, we would recommend using Mask RCNN or YOLO models for localization tasks solved with object detection models and DeepLab, Mask RCNN, or UNet for segmentation. Despite a higher time expenditure for implementation and training, the use of customized CNN models can be justified and yield similar performance to the aforementioned models. However, it should be noted that, on the one hand, the necessary expertise must be available and, on the other hand, the amount of data must be sufficient. CNN LSTM, DDETR, custom transformer, and FCN look promising, as well, but have to be investigated more thoroughly to make a more informed statement.

4.4. Comparison with Academic Development in Deep Learning Computer Vision Models

Figure 15 illustrates how often a certain DL model was applied to VI each year between 2012 and 2023 based on our publication corpus. In addition, we highlight the

progress of SOTA DL CV models by marking in which year the models were first proposed. There were very few publications from 2012 to 2017, but 2018 onward, there has been a strong increase every following year, reaching a maximum to date of almost 100 publications in 2022.

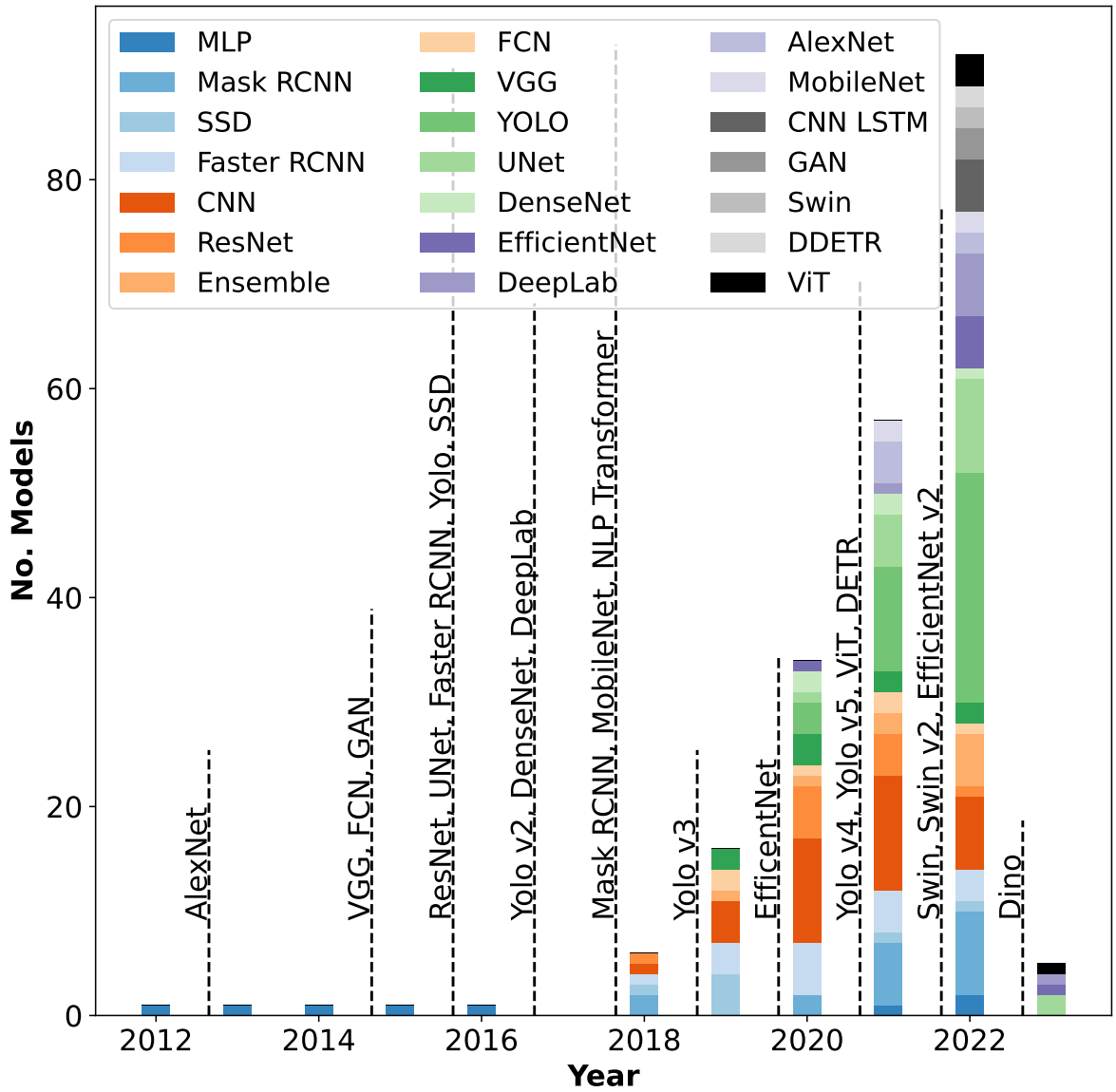

Figure 15. Timeline showing the number of occurrences of models in a certain year in our publication corpus from 2012 to 2023. Only models that occurred at least two times in total were included. Vertical dashed lines mark the proposal of models considered to be research milestones.

In 2012, a CNN received significant attention for its classification ability. This CNN is called AlexNet and took first place at the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC 2012) with an error rate 10.9%p lower than the second-place model [249]. Thereafter, the field of CV was dominated by CNN architectures. In the following years, the models became deeper and more complex. In 2014, VGG [250] and FCN [251] were introduced, but with deeper models, the vanishing gradient problem occurs more intensively. One solution was introduced in 2015 with Microsoft's ResNet, which uses residual connections [67]. Due to the residual connections, the gradient does not decrease arbitrarily. This year also marks the proposal of the one-stage object detection architectures YOLO [68] and SSD [252], which are optimized for inference speed; the third iteration of the two-stage detector (Faster) RCNN [253]; and the segmentation architecture

UNet [254]. In 2016, DenseNet was published, and it uses a similar approach to ResNet's residual connections, called dense connections [255]. In the same year, the second version of YOLO [256] was published, adopting the concept of anchor boxes from SDD and using a new backbone network, called Darknet-19, specifically designed for object detection. In 2017, an RCNN version for semantic segmentation, called Mask RCNN [257], as well as the first transformer model in natural language processing [258], was proposed. The third version of YOLO [259],proposed in 2018, adds multi-scale detection in addition to a deeper backbone network to further improve the detection of small objects. In 2019, Google proposed EfficientNets [260], a series of models, where architectural scaling parameters like width, depth, and resolution were not chosen by the authors but determined by a learning-based approach called neural architecture search. YOLO v4 [261] and v5 [262] were published in very short succession by different authors implementing many similar improvements, like a new data augmentation strategy called mosaic augmentation, where different images are combined for training, the mish activation function [263], and cross-stage partial connections (CSP) [264], to name a few.

Recently, the impressive results of transformer models in natural language processing (NLP) have attracted the attention of CV researchers to adapt them to their domain. Two different approaches have been established: either they are utilized as backbone networks, such as the vision transformer (ViT) [265] or the shifted windows transformer (Swin) [266], or they function as detection heads, such as the detection transformer (DETR) [267] and the dynamic head transformer (DyHead) [268], which work on features extracted by an upstream (CNN) backbone. Nowadays, transformer models outperform convolution-based models on CV benchmark datasets like MS COCO [269] or Pascal visual object classes (VOC) by up to 7.5% in mAP [270].

Another advantage of transformers is that they do not require a lot of labeled data, since they can be pretrained for object detection in a self-supervised manner before being fine-tuned on small labeled datasets. Chen et al. proposed a pretraining task specifically for the DETR model that improves on its supervised performance, while Bar et al. even achieved highly accurate zero-shot with the same model [271,272]. Xu et al. developed a student–teacher-based training procedure for object detection that is independent of the model architecture, but transformers show very good performance when trained with it [273]. These aforementioned methods are limited to the task of object detection, but transformers are also able to learn more general representations from self-supervised training that can be adapted to different CV tasks, such as classification, object detection, segmentation, key point detection, or pose estimation. These methods can be grouped into three different categories: masked image modeling (MIM), contrastive learning, and hybrids of those two. In masked image modeling, parts of the input images are masked out, usually with the average color values of the image, and the task is to reconstruct these masked-out sections [274,275]. Contrastive learning tries to achieve similar representations for similar input images. This is achieved by augmenting the input in two different ways and forcing the model to represent both views close to each other, while augmented views of other images are pushed away in the representation space [276–281]. Fang et al. combined both paradigms, conditioning their model to reconstruct the masked-out features of a contrastive image language pretraining (CLIP) model [282] to achieve even better results. In AVI, properties like strong generalization and data efficiency are highly desirable, since labeled data are scarce. Especially in tasks where errors or damages need to be detected, there is a lack of examples for training. When looking at the timeline in Figure 15, it is noticeable that there is a time gap of two to three years between the invention of a model and its transfer to an AVI application. Consequently, we can argue that the performance in AVI can be improved by the increased application of newer CNN models, such as Efficient v1/2 [260,283], FCOS [284], or attention-enhanced CNN models, which showed promise in localization tasks with regard to their F1-score and mIoU. The introduction of vision transformer models such as DETR, Swin v1, or the Retina-Swin [266,267,285] started in 2022, which is already faster than what we observed with most CNN models in the past.

Most of these publications yielded results within the top range of their AVI use case, as has been shown in Section 4 [70,123,152,164,187]; therefore, we expect and recommend the acceleration of the application of vision transformers to the domain of AVI, especially even newer, very parameter-efficient models like Dino and its variants [286–289]. All publications utilizing transformer models applied them via supervised transfer learning, disregarding their excellent semi-supervised learning capabilities with regard to generalization and data efficiency. This could potentially reduce the amount of labeled data required to train them, which ultimately leads to a lower investment of time for labeling and thus cost. Nevertheless, it must be noted that transformers need more memory than CNNs; therefore, the benefits may be limited on edge devices. Further research into the model miniaturization of vision transformers may lead to an improvement on this front.

5. Conclusions

In this review, we provide a comprehensive overview of 196 scientific publications published between 2012 and March 2023 dealing with the application of DL-based models for AVI in manufacturing and maintenance.

The publications were categorized and critically evaluated based on six guiding questions, which helped us throughout the literature review. By answering these questions, we can report several findings.

Based on the literature, we derived key requirements for AVI use cases. These requirements were subsequently organized into two distinct dimensions, namely, general versus specific requirements and hard technical versus soft human factors. As a result, we identified several essential aspects that transcend particular use cases and are of broad significance. These encompass performance, explainability, robustness/generalization, the ability to handle sparse data, real-time capability, and adherence to hardware constraints.

The use cases in which DL-based AVI is applied can be structured hierarchically. All of them fall under quality inspection, but they can be subdivided into damage detection, completeness check, and others, where the latest contains use cases like plant disease detection, food contamination detection, and indirect wear and vibration estimation. The damage detection category features one additional sub-use-case that occurs frequently: crack detection.

We analyzed the datasets utilized in AVI regarding size distributions as well as the uniformity of their class distributions. This is measured by a balance metric that we propose. In many cases, the dataset parameters required to compute it are incompletely or imprecisely reported. On the one hand, this limited our analysis, but on the other hand, it also makes the results less comparable between publications and harder to reproduce and hinders further development of existing research results by others, which is detrimental to the progress of AVI overall. In addition, we could identify 14 datasets that were utilized more than once to benchmark tasks, out of the wide selection of 77 open datasets provided by authors. So, we want to encourage AVI researchers to give more detailed dataset descriptions to improve comparability and use and improve open datasets to advance the field as a whole.

We found that AVI can be addressed with classification and localization tasks, of which both come in two forms: binary and multi-class. These four AVI tasks are closely related to the three most common CV tasks, which means that classification AVI tasks are performed by image classification models, while localization tasks are performed by either object detection or segmentation models. In the domain of maintenance-oriented AVI use cases, we observed that localization methodologies are prevalent, as they are utilized approximately 77.2% of the time, while classification methods are employed in about 22.8% of instances. In contrast, within the manufacturing-focused AVI applications, the distribution of localization and classification strategies is more evenly balanced, with localization accounting for 49.2% and classification accounting for 50.8% of use cases. These discernible discrepancies in the adoption of localization and classification approaches can be attributed to various environmental factors that govern the distinct operational requirements and objectives of maintenance and manufacturing scenarios. Especially for maintenance, we

recommend prioritizing localization-capable methodologies such as YOLO and Mask R-CNNs. The inherent capability of these methods to precisely localize and delineate objects of interest is deemed crucial to enhancing the effectiveness of AVI for maintenance purposes. If the task can be performed through classification, ResNet and EfficientNet are the models we recommend using.

We also compared the SOTA in AVI applications with the advance in academic CV research. We identified a time gap of two to three years between the first proposal of a model and its transfer to an industrial AVI application. The first publications utilizing vision transformer models occurred at the lower bound of this gap in the first quarter of 2022, while more followed over the course of the year. Since their results are in the upper performance range of their respective AVI use cases, and much effort is being put into reducing their hardware requirements, we expect them to become the standard in the future. A more thorough investigation of vision transformers for classification and attention-enhanced CNNs, DDETR, and custom transformer models for localization in more different use cases could yield insights if these models live up to the potential they have shown in their low number of applications.

This review paper is not without limitations, which should be acknowledged in order to fully assess its scope and implications. The first limitation concerns the use of exclusively 2D image data. While this choice was motivated by broad applicability, it inadvertently limits the integration of more sophisticated imaging technologies. Nevertheless, it is important to emphasize that input signals other than traditional RGB images limit the effectiveness of domain adaptation techniques, e.g., transfer learning, due to a more pronounced domain gap. Another limitation of our literature review is the concise treatment of the models and training methods used in the reviewed publications. Given the extensive nature of the publication corpus considered, it could distort the research scope that we wanted to address by providing a detailed exposition of each model and training methodology. It should be noted that all of the included literature sources are open-access, allowing interested readers unrestricted access to the publications themselves, where more detailed descriptions of these aspects can be found.

These limitations provide avenues for future research to delve deeper into the topics addressed herein and extend the knowledge base in the field. Therefore, a future addition to our work will be looking at other input sensors for AVI. For example, an interesting research question to investigate is, To what extent can approaches used for 2D images be transferred to 3D images, containing depth information, or other sensor outputs, such as point clouds?

Another direction for future research is to investigate the time gap between academic research and industrial applications, which we assume is not limited to AVI. The reasons are probably manifold, but is it possible to reduce the identified gap of two to three years? And what is necessary to reduce the gap?

We expect vision transformers to become the models of choice for AVI applications. However, most publications focused on fine-tuning pretrained models in a supervised manner, ignoring their strong self-supervised learning capabilities. These have been demonstrated on benchmark datasets for specific tasks, such as object detection, as well as for learning more general representations transferable to a multitude of tasks. From our point of view, the transfer of self-supervised learning to AVI applications seems to be a research path worth spending more attention on in the future, especially because the available datasets for industrial applications are much smaller than the benchmark datasets.

Author Contributions: Conceptualization, N.H. and M.A.G.; Methodology, N.H. and M.A.G.; Resources, T.M.; Data Curation, N.H., M.A.G., K.A. and F.H.; Writing—Original Draft Preparation, N.H., M.A.G., K.A. and F.H.; Writing—Review, R.M. and T.M.; Writing—Editing, N.H. and M.A.G.; Visualization, N.H. and M.A.G.; Supervision, T.M.; Project Administration, R.M.; Funding Acquisition, R.M. and T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the German Federal Ministry for Digital and Transport in the program "future rail freight transport" under grant number 53T20011UW.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AVI	Automated visual inspection
DL	Deep learning
CNN	Convolutional neural network
MS COCO	Microsoft common objects in context (object detection) dataset
CV	Computer vision
DETR	Detection transformer [267]
DDETR	Deformable detection transformer [290]
FCN	Fully convolutional neural network (semantic segmentation model) [251]
FCOS	Fully convolutional one-stage object detection (model) [284]
FLOPS	Floating-point operations per second
FN(R)	False-negative rate
FP(R)	False-positive rate
FPS	Frames per second
GAN	Generative adversarial network
ILSVRC 2012/ImageNet	ImageNet Large Scale Visual Recognition Challenge 2012
LiDAR	Light detection and ranging, 3D laser scanning method
LSTM	Long short-term memory cell (recurrent neural network variant)
mAP	Mean average precision (common object detection performance metric)
MIM	Masked image modeling
MLP	Multi-layer perceptron (network)
NLP	Natural language processing
Pascal VOC	Pascal visual object classes (object detection dataset)
ResNet	Residual Network [67]
RCNN	Regional convolutional neural network [253]
SOTA	State of the art
SSD	Single-shot detector [252]
SVM	Support vector machine
Swin	Shifted windows transformer [266]
TN(R)	True-negative rate
TP(R)	True-positive rate
VI	Visual inspection
ViT	Specific architecture of a vision transformer model published in [265]
VGG	Visual geometry group (model) [250]
WoS	Web of Science
YOLO	You only look once (object detection model) [68]

Appendix A

Table A1. Detailed listing of included and excluded Web of science categories.

Web of Science Category	Exc.	Web of Science Category	Exc.
Engineering Electrical Electronic		Food Science Technology	
Instruments Instrumentation		Mathematics Applied	
Computer Science Information Systems		Medical Informatics	x
Engineering Multidisciplinary		Nanoscience Nanotechnology	
Materials Science Multidisciplinary		Nuclear Science Technology	x
Chemistry Analytical		Oceanography	x
Telecommunications		Operations Research Management Science	x
Physics Applied		Psychology Experimental	x
Engineering Civil		Thermodynamics	x
Chemistry Multidisciplinary		Agricultural Engineering	
Imaging Science Photographic Technology		Agriculture Dairy Animal Science	x
Remote Sensing	x	Audiology Speech Language Pathology	x
Environmental Sciences		Behavioral Sciences	x
Computer Science Interdisciplinary Applications		Biochemistry Molecular Biology	x
Geosciences Multidisciplinary	x	Ecology	x
Construction Building Technology		Engineering Industrial	
Engineering Mechanical		Health Care Sciences Services	x
Multidisciplinary Sciences		Materials Science Textiles	
Radiology Nuclear Medicine Medical Imaging	x	Medicine Research Experimental	x
Engineering Biomedical	x	Pathology	x
Astronomy Astrophysics	x	Physics Mathematical	x
Computer Science Artificial Intelligence		Physics Multidisciplinary	
Mechanics		Physics Particles Fields	x
Transportation Science Technology		Physiology	x
Neurosciences	x	Quantum Science Technology	x
Energy Fuels		Respiratory System	x
Acoustics	x	Robotics	
Oncology	x	Surgery	x
Engineering Manufacturing		Architecture	
Mathematical Computational Biology	x	Chemistry Medicinal	x
Mathematics Interdisciplinary Applications		Dentistry Oral Surgery Medicine	x
Metallurgy Metallurgical Engineering		Dermatology	x
Optics	x	Developmental Biology	x
Green Sustainable Science Technology		Engineering Environmental	
Biochemical Research Methods	x	Fisheries	x
Computer Science Software Engineering		Forestry	x
Automation Control Systems		Gastroenterology Hepatology	x
Computer Science Theory Methods	x	Genetics Heredity	x
Computer Science Hardware Architecture	x	Geriatrics Gerontology	x
Geography Physical	x	Immunology	x
Agriculture Multidisciplinary		Infectious Diseases	x
Chemistry Physical	x	Marine Freshwater Biology	x
Engineering Aerospace		Materials Science Biomaterials	
Environmental Studies	x	Medical Laboratory Technology	x
Materials Science Composites		Obstetrics Gynecology	x
Medicine General Internal	x	Otorhinolaryngology	x
Physics Condensed Matter	x	Paleontology	x
Rehabilitation	x	Parasitology	x
Biotechnology Applied Microbiology	x	Peripheral Vascular Disease	x
Clinical Neurology	x	Pharmacology Pharmacy	x
Engineering Ocean		Physics Fluids Plasmas	x

Table A1. *Cont.*

Web of Science Category	Exc.	Web of Science Category	Exc.
Materials Science Characterization Testing		Physics Nuclear	x
Meteorology Atmospheric Sciences	x	Plant Sciences	x
Water Resources	x	Psychiatry	x
Geochemistry Geophysics	x	Public Environmental Occupational Health	x
Mathematics		Sport Sciences	x
Neuroimaging	x	Transportation	
Agronomy		Tropical Medicine	x
Cell Biology	x	Veterinary Sciences	x
Engineering Marine			

References

1. Drury, C.G.; Watson, J. Good practices in visual inspection. In *Human Factors in Aviation Maintenance-Phase Nine, Progress Report, FAA/Human Factors in Aviation Maintenance*; 2002.
2. Steger, C.; Ulrich, M.; Wiedemann, C. *Machine Vision Algorithms and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2018.
3. Sheehan, J.J.; Drury, C.G. The analysis of industrial inspection. *Appl. Ergon.* **1971**, *2*, 74–78. [CrossRef] [PubMed]
4. Chiang, H.Q.; Hwang, S.L. Human performance in visual inspection and defect diagnosis tasks: A case study. *Int. J. Ind. Ergon.* **1988**, *2*, 235–241. [CrossRef]
5. Swain, A.D.; Guttmann, H.E. *Handbook of Human-Reliability Analysis with Emphasis on Nuclear Power Plant Applications, Final Report*; Sandia National Lab.: Albuquerque, NM, USA, 1983. [CrossRef]
6. Drury, C.; Fox, J. The imperfect inspector. *Human Reliability in Quality Control*; Taylor & Francis: London, UK, 1975; pp. 11–16.
7. Jiang, X.; Gramopadhye, A.K.; Melloy, B.J.; Grimes, L.W. Evaluation of best system performance: Human, automated, and hybrid inspection systems. *Hum. Factors Ergon. Manuf. Serv. Ind.* **2003**, *13*, 137–152. [CrossRef]
8. Voulodimos, A.; Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep Learning for Computer Vision: A Brief Review. *Comput. Intell. Neurosci.* **2018**, *2018*, 1–13. [CrossRef]
9. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]
10. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
11. Vom Brocke, J.; Simons, A.; Riemer, K.; Niehaves, B.; Plattfaut, R.; Cleven, A. Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. *Commun. Assoc. Inf. Syst.* **2015**, *37*, 9. [CrossRef]
12. Zheng, X.; Zheng, S.; Kong, Y.; Chen, J. Recent advances in surface defect inspection of industrial products using deep learning techniques. *Int. J. Adv. Manuf. Technol.* **2021**, *113*, *35–58.* [CrossRef]
13. Jenssen, R.; Roverso, D. Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 107–120.
14. Sun, X.; Gu, J.; Tang, S.; Li, J. Research progress of visual inspection technology of steel products—A review. *Appl. Sci.* **2018**, *8*, 2195. [CrossRef]
15. Yang, J.; Li, S.; Wang, Z.; Dong, H.; Wang, J.; Tang, S. Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges. *Materials* **2020**, *13*, 5755. [CrossRef]
16. Nash, W.; Drummond, T.; Birbilis, N. A review of deep learning in the study of materials degradation. *NPJ Mater. Degrad.* **2018**, *2*, 1–12. [CrossRef]
17. Liu, S.; Wang, Q.; Luo, Y. A review of applications of visual inspection technology based on image processing in the railway industry. *Transp. Saf. Environ.* **2019**, *1*, 185–204. [CrossRef]
18. De Donato, L.; Flammini, F.; Marrone, S.; Mazzariello, C.; Nardone, R.; Sansone, C.; Vittorini, V. A Survey on Audio-Video Based Defect Detection Through Deep Learning in Railway Maintenance. *IEEE Access* **2022**, *10*, 65376–65400. [CrossRef]
19. Ali, L.; Alnajjar, F.; Khan, W.; Serhani, M.A.; Al Jassmi, H. Bibliometric Analysis and Review of Deep Learning-Based Crack Detection Literature Published between 2010 and 2022. *Buildings* **2022**, *12*, 432. [CrossRef]
20. Ali, R.; Chuah, J.H.; Abu Talip, M.S.; Mokhtar, N.; Shoaib, M.A. Structural crack detection using deep convolutional neural networks. *Autom. Constr.* **2022**, *133*, 103989. [CrossRef]
21. Hamishebahar, Y.; Guan, H.; So, S.; Jo, J. A Comprehensive Review of Deep Learning-Based Crack Detection Approaches. *Appl.-Sci.-Basel* **2022**, *12*, 1374. [CrossRef]
22. Omar, I.; Khan, M.; Starr, A. Compatibility and challenges in machine learning approach for structural crack assessment. *Struct. Health Monit. Int. J.* **2022**, *21*, 2481–2502. [CrossRef]
23. Chu, C.; Wang, L.; Xiong, H. A review on pavement distress and structural defects detection and quantification technologies using imaging approaches. *J. Traffic Transp. Eng. Engl. Ed.* **2022**, *9*, 135–150. [CrossRef]
24. Qureshi, W.S.; Hassan, S.I.; McKeever, S.; Power, D.; Mulry, B.; Feighan, K.; O'Sullivan, D. An Exploration of Recent Intelligent Image Analysis Techniques for Visual Pavement Surface Condition Assessment. *Sensors* **2022**, *22*, 9019. [CrossRef] [PubMed]

25. Ranyal, E.; Sadhu, A.; Jain, K. Road Condition Monitoring Using Smart Sensing and Artificial Intelligence: A Review. *Sensors* **2022**, *22*, 3044. [CrossRef] [PubMed]
26. Kim, Y.M.; Kim, Y.G.; Son, S.Y.; Lim, S.Y.; Choi, B.Y.; Choi, D.H. Review of Recent Automated Pothole-Detection Methods. *Appl. Sci.* **2022**, *12*, 5320. [CrossRef]
27. Zhou, H.; Xu, C.; Tang, X.; Wang, S.; Zhang, Z. A Review of Vision-Laser-Based Civil Infrastructure Inspection and Monitoring. *Sensors* **2022**, *22*, 5882. [CrossRef] [PubMed]
28. Hassani, S.; Mousavi, M.; Gandomi, A.H. Structural Health Monitoring in Composite Structures: A Comprehensive Review. *Sensors* **2022**, *22*, 153. [CrossRef]
29. Chew, M.Y.L.; Gan, V.J.L. Long-Standing Themes and Future Prospects for the Inspection and Maintenance of Facade Falling Objects from Tall Buildings. *Sensors* **2022**, *22*, 6070. [CrossRef] [PubMed]
30. Luleci, F.; Catbas, F.N.; Avci, O. A literature review: Generative adversarial networks for civil structural health monitoring. *Front. Built Environ.* **2022**, *8*. [CrossRef]
31. Mera, C.; Branch, J.W. A survey on class imbalance learning on automatic visual inspection. *IEEE Lat. Am. Trans.* **2014**, *12*, 657–667. [CrossRef]
32. Tao, X.; Gong, X.; Zhang, X.; Yan, S.; Adak, C. Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5018021. [CrossRef]
33. Rippel, O.; Merhof, D. Anomaly Detection for Automated Visual Inspection: A Review. In *Bildverarbeitung in der Automation: Ausgewählte Beiträge des Jahreskolloquiums BVAu 2022*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 1–13.
34. Newman, T.S.; Jain, A.K. A survey of automated visual inspection. *Comput. Vis. Image Underst.* **1995**, *61*, 231–262. [CrossRef]
35. Li, K.; Rollins, J.; Yan, E. Web of Science use in published research and review papers 1997–2017: A selective, dynamic, cross-domain, content-based analysis. *Scientometrics* **2018**, *115*, 1–20. [CrossRef]
36. Chin, R.T. Automated visual inspection: 1981 to 1987. *Comput. Vision Graph. Image Process.* **1988**, *41*, 346–381. [CrossRef]
37. See, J.E.; Drury, C.G.; Speed, A.; Williams, A.; Khalandi, N. The Role of Visual Inspection in the 21 st Century. In *Foundations of Augmented Cognition*; Schmorrow, D.D., Fidopiastis, C.M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2015; Volume 61, pp. 262–266. [CrossRef]
38. Brandoli, B.; de Geus, A.R.; Souza, J.R.; Spadon, G.; Soares, A.; Rodrigues, J.F., Jr., ; Komorowski, J.; Matwin, S. Aircraft Fuselage Corrosion Detection Using Artificial Intelligence. *Sensors* **2021**, *21*, 4026. [CrossRef] [PubMed]
39. Wang, Z.; Gao, J.; Zeng, Q.; Sun, Y. Multitype Damage Detection of Container Using CNN Based on Transfer Learning. *Math. Probl. Eng.* **2021**, *2021*, 5395494. [CrossRef]
40. Chen, Y.W.; Shiu, J.M. An implementation of YOLO-family algorithms in classifying the product quality for the acrylonitrile butadiene styrene metallization. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 8257–8269. [CrossRef] [PubMed]
41. Zhang, M.; Zhang, Y.; Zhou, M.; Jiang, K.; Shi, H.; Yu, Y.; Hao, N. Application of Lightweight Convolutional Neural Network for Damage Detection of Conveyor Belt. *Appl. Sci.* **2021**, *11*, 7282. [CrossRef]
42. Wei, X.; Wei, D.; Da S.; Jia, L.; Li, Y. Multi-Target Defect Identification for Railway Track Line Based on Image Processing and Improved YOLOv3 Model. *IEEE Access* **2020**, *8*, 61973–61988. [CrossRef]
43. Kin, N.W.; Asaari, M.S.M.; Rosdi, B.A.; Akbar, M.F. Fpga Implementation of CNN for Defect Classification on CMP Ring. *J. Teknol.-Sci. Eng.* **2021**, *83*, 101–108. [CrossRef]
44. Smith, A.G.; Petersen, J.; Selvan, R.; Rasmussen, C.R. Segmentation of roots in soil with U-Net. *Plant Methods* **2020**, *16*, 1. [CrossRef]
45. Kuric, I.; Klarak, J.; Bulej, V.; Saga, M.; Kandera, M.; Hajducik, A.; Tucki, K. Approach to Automated Visual Inspection of Objects Based on Artificial Intelligence. *Appl. Sci.* **2022**, *12*, 864. [CrossRef]
46. Selmaier, A.; Kunz, D.; Kisskalt, D.; Benaziz, M.; Fuerst, J.; Franke, J. Artificial Intelligence-Based Assistance System for Visual Inspection of X-ray Scatter Grids. *Sensors* **2022**, *22*, 811. [CrossRef]
47. Fan, Z.; Li, C.; Chen, Y.; Wei, J.; Loprencipe, G.; Chen, X.; Di Mascio, P. Automatic Crack Detection on Road Pavements Using Encoder-Decoder Architecture. *Materials* **2020**, *13*, 2960. [CrossRef]
48. Napoletano, P.; Piccoli, F.; Schettini, R. Anomaly Detection in Nanofibrous Materials by CNN-Based Self-Similarity. *Sensors* **2018**, *18*, 209. [CrossRef] [PubMed]
49. Ulger, F.; Yuksel, S.E.; Yilmaz, A. Anomaly Detection for Solder Joints Using beta-VAE. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2021**, *11*, 2214–2221. [CrossRef]
50. Adibhatla, V.A.; Huang, Y.C.; Chang, M.C.; Kuo, H.C.; Utekar, A.; Chih, H.C.; Abbod, M.F.; Shieh, J.S. Unsupervised Anomaly Detection in Printed Circuit Boards through Student-Teacher Feature Pyramid Matching. *Electronics* **2021**, *10*, 3177. [CrossRef]
51. Chandran, P.; Asber, J.; Thiery, F.; Odelius, J.; Rantatalo, M. An Investigation of Railway Fastener Detection Using Image Processing and Augmented Deep Learning. *Sustainability* **2021**, *13*, 12051. [CrossRef]
52. Wang, T.; Yang, F.; Tsui, K.L. Real-Time Detection of Railway Track Component via One-Stage Deep Learning Networks. *Sensors* **2020**, *20*, 4325. [CrossRef] [PubMed]
53. Ferguson, M.; Ronay, A.; Lee, Y.T.T.; Law, K.H. Detection and Segmentation of Manufacturing Defects with Convolutional Neural Networks and Transfer Learning. *Smart Sustain. Manuf. Syst.* **2018**, *2*, 137–164. [CrossRef]
54. Wang, P.; Tseng, H.W.; Chen, T.C.; Hsia, C.H. Deep Convolutional Neural Network for Coffee Bean Inspection. *Sens. Mater.* **2021**, *33*, 2299–2310. [CrossRef]

55. Hussain, M.A.I.; Khan, B.; Wang, Z.; Ding, S. Woven Fabric Pattern Recognition and Classification Based on Deep Convolutional Neural Networks. *Electronics* **2020**, *9*, 1048. [CrossRef]
56. Aslam, M.; Khan, T.M.; Naqvi, S.S.; Holmes, G.; Naffa, R. Ensemble Convolutional Neural Networks With Knowledge Transfer for Leather Defect Classification in Industrial Settings. *IEEE Access* **2020**, *8*, 198600–198614. [CrossRef]
57. Chen, Y.; Fu, Q.; Wang, G. Surface Defect Detection of Nonburr Cylinder Liner Based on Improved YOLOv4. *Mob. Inf. Syst.* **2021**, *2021*. [CrossRef]
58. Neven, R.; Goedeme, T. A Multi-Branch U-Net for Steel Surface Defect Type and Severity Segmentation. *Metals* **2021**, *11*, 870. [CrossRef]
59. Qu, Z.; Mei, J.; Liu, L.; Zhou, D.Y. Crack Detection of Concrete Pavement With Cross-Entropy Loss Function and Improved VGG16 Network Model. *IEEE Access* **2020**, *8*, 54564–54573. [CrossRef]
60. Samma, H.; Suandi, S.A.; Ismail, N.A.; Sulaiman, S.; Ping, L.L. Evolving Pre-Trained CNN Using Two-Layers Optimizer for Road Damage Detection From Drone Images. *IEEE Access* **2021**, *9*, 158215–158226. [CrossRef]
61. Sun, Y.; Yang, Y.; Yao, G.; Wei, F.; Wong, M. Autonomous Crack and Bughole Detection for Concrete Surface Image Based on Deep Learning. *IEEE Access* **2021**, *9*, 85709–85720. [CrossRef]
62. Wang, D.; Cheng, J.; Cai, H. Detection Based on Crack Key Point and Deep Convolutional Neural Network. *Appl. Sci.* **2021**, *11*, 11321. [CrossRef]
63. O'Byrne, M.; Ghosh, B.; Schoefs, F.; Pakrashi, V. Applications of Virtual Data in Subsea Inspections. *J. Mar. Sci. Eng.* **2020**, *8*, 328. [CrossRef]
64. Ahmad, N.; Asif, H.M.S.; Saleem, G.; Younus, M.U.; Anwar, S.; Anjum, M.R. Leaf Image-Based Plant Disease Identification Using Color and Texture Features. *Wirel. Pers. Commun.* **2021**, *121*, 1139–1168. [CrossRef]
65. Velasquez, D.; Sanchez, A.; Sarmiento, S.; Toro, M.; Maiza, M.; Sierra, B. A Method for Detecting Coffee Leaf Rust through Wireless Sensor Networks, Remote Sensing, and Deep Learning: Case Study of the Caturra Variety in Colombia. *Appl. Sci.* **2020**, *10*, 697. [CrossRef]
66. Pagani, L.; Parenti, P.; Cataldo, S.; Scott, P.J.; Annoni, M. Indirect cutting tool wear classification using deep learning and chip colour analysis. *Int. J. Adv. Manuf. Technol.* **2020**, *111*, 1099–1114. [CrossRef]
67. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
68. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
69. Jian, B.l.; Hung, J.P.; Wang, C.C.; Liu, C.C. Deep Learning Model for Determining Defects of Vision Inspection Machine Using Only a Few Samples. *Sens. Mater.* **2020**, *32*, 4217–4231. [CrossRef]
70. Ali, L.; Jassmi, H.A.; Khan, W.; Alnajjar, F. Crack45K: Integration of Vision Transformer with Tubularity Flow Field (TuFF) and Sliding-Window Approach for Crack-Segmentation in Pavement Structures. *Buildings* **2023**, *13*, 55. [CrossRef]
71. Rajadurai, R.S.; Kang, S.T. Automated Vision-Based Crack Detection on Concrete Surfaces Using Deep Learning. *Appl. Sci.* **2021**, *11*, 5229. [CrossRef]
72. Hallee, M.J.; Napolitano, R.K.; Reinhart, W.F.; Glisic, B. Crack Detection in Images of Masonry Using CNNs. *Sensors* **2021**, *21*, 4929. [CrossRef] [PubMed]
73. Mohammed, M.A.; Han, Z.; Li, Y. Exploring the Detection Accuracy of Concrete Cracks Using Various CNN Models. *Adv. Mater. Sci. Eng.* **2021**, *2021*, 9923704. [CrossRef]
74. Stephen, O.; Maduh, U.J.; Sain, M. A Machine Learning Method for Detection of Surface Defects on Ceramic Tiles Using Convolutional Neural Networks. *Electronics* **2022**, *11*, 55. [CrossRef]
75. Chaiyasarn, K.; Sharma, M.; Ali, L.; Khan, W.; Poovarodom, N. Crack detection in historical structures based on convolutional neural network. *Int. J. Geomate* **2018**, *15*, 240–251. [CrossRef]
76. Ali, L.; Alnajjar, F.; Al Jassmi, H.; Gocho, M.; Khan, W.; Serhani, M.A. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors* **2021**, *21*, 1688. [CrossRef]
77. Santos, R.; Ribeiro, D.; Lopes, P.; Cabral, R.; Calcada, R. Detection of exposed steel rebars based on deep-learning techniques and unmanned aerial vehicles. *Autom. Constr.* **2022**, *139*, 104324. [CrossRef]
78. Woo, J.; Lee, H. Nonlinear and Dotted Defect Detection with CNN for Multi-Vision-Based Mask Inspection. *Sensors* **2022**, *22*, 8945. [CrossRef]
79. Avdelidis, N.P.; Tsourdos, A.; Lafiosca, P.; Plaster, R.; Plaster, A.; Droznika, M. Defects Recognition Algorithm Development from Visual UAV Inspections. *Sensors* **2022**, *22*, 4682. [CrossRef] [PubMed]
80. Stephen, O.; Madanian, S.; Nguyen, M. A Hard Voting Policy-Driven Deep Learning Architectural Ensemble Strategy for Industrial Products Defect Recognition and Classification. *Sensors* **2022**, *22*, 7846. [CrossRef] [PubMed]
81. Ortiz, A.; Bonnin-Pascual, F.; Garcia-Fidalgo, E.; Company-Corcoles, J.P. Vision-Based Corrosion Detection Assisted by a Micro-Aerial Vehicle in a Vessel Inspection Application. *Sensors* **2016**, *16*, 2118. [CrossRef]
82. Jin, W.W.; Chen, G.H.; Chen, Z.; Sun, Y.L.; Ni, J.; Huang, H.; Ip, W.H.; Yung, K.L. Road Pavement Damage Detection Based on Local Minimum of Grayscale and Feature Fusion. *Appl. Sci.* **2022**, *12*, 13006. [CrossRef]
83. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.

84. Walther, D.; Schmidt, L.; Schricker, K.; Junger, C.; Bergmann, J.P.; Notni, G.; Maeder, P. Automatic detection and prediction of discontinuities in laser beam butt welding utilizing deep learning. *J. Adv. Join. Processes* **2022**, *6*, 100119. [CrossRef]
85. Kumar, G.S.; Natarajan, U.; Ananthan, S.S. Vision inspection system for the identification and classification of defects in MIG welding joints. *Int. J. Adv. Manuf. Technol.* **2012**, *61*, 923–933. [CrossRef]
86. Alqahtani, H.; Bharadwaj, S.; Ray, A. Classification of fatigue crack damage in polycrystalline alloy structures using convolutional neural networks. *Eng. Fail. Anal.* **2021**, *119*, 104908. [CrossRef]
87. Elhariri, E.; El-Bendary, N.; Taie, S.A. Using Hybrid Filter-Wrapper Feature Selection With Multi-Objective Improved-Salp Optimization for Crack Severity Recognition. *IEEE Access* **2020**, *8*, 84290–84315. [CrossRef]
88. Kim, B.; Choi, S.W.; Hu, G.; Lee, D.E.; Juan, R.O.S. An Automated Image-Based Multivariant Concrete Defect Recognition Using a Convolutional Neural Network with an Integrated Pooling Module. *Sensors* **2022**, *22*, 3118. [CrossRef]
89. Yang, J.; Li, S.; Wang, Z.; Yang, G. Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning. *IEEE Access* **2019**, *7*, 89278–89291. [CrossRef]
90. Dang, X.; Shang, X.; Hao, Z.; Su, L. Collaborative Road Damage Classification and Recognition Based on Edge Computing. *Electronics* **2022**, *11*, 3304. [CrossRef]
91. Alqethami, S.; Alghamdi, S.; Alsubait, T.; Alhakami, H. RoadNet: Efficient Model to Detect and Classify Road Damages. *Appl. Sci.* **2022**, *12*, 11529. [CrossRef]
92. Chandra, S.; AlMansoor, K.; Chen, C.; Shi, Y.; Seo, H. Deep Learning Based Infrared Thermal Image Analysis of Complex Pavement Defect Conditions Considering Seasonal Effect. *Sensors* **2022**, *22*, 9365. [CrossRef]
93. Wang, D.; Xu, Y.; Duan, B.; Wang, Y.; Song, M.; Yu, H.; Liu, H. Intelligent Recognition Model of Hot Rolling Strip Edge Defects Based on Deep Learning. *Metals* **2021**, *11*, 223. [CrossRef]
94. Schlosser, T.; Friedrich, M.; Beuth, F.; Kowerko, D. Improving automated visual fault inspection for semiconductor manufacturing using a hybrid multistage system of deep neural networks. *J. Intell. Manuf.* **2022**, *33*, 1099–1123. [CrossRef]
95. Maeda, K.; Takahashi, S.; Ogawa, T.; Haseyama, M. Convolutional sparse coding-based deep random vector functional link network for distress classification of road structures. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 654–676. [CrossRef]
96. Ahmad, A.; Jin, Y.; Zhu, C.; Javed, I.; Maqsood, A.; Akram, M.W. Photovoltaic cell defect classification using convolutional neural network and support vector machine. *Iet Renew. Power Gener.* **2020**, *14*, 2693–2702. [CrossRef]
97. Shin, H.K.; Ahn, Y.H.; Lee, S.H.; Kim, H.Y. Automatic Concrete Damage Recognition Using Multi-Level Attention Convolutional Neural Network. *Materials* **2020**, *13*, 5549. [CrossRef]
98. Dunphy, K.; Fekri, M.N.; Grolinger, K.; Sadhu, A. Data Augmentation for Deep-Learning-Based Multiclass Structural Damage Detection Using Limited Information. *Sensors* **2022**, *22*, 6193. [CrossRef]
99. Stephen, O.; Madanian, S.; Nguyen, M. A Robust Deep Learning Ensemble-Driven Model for Defect and Non-Defect Recognition and Classification Using a Weighted Averaging Sequence-Based Meta-Learning Ensembler. *Sensors* **2022**, *22*, 9971. [CrossRef] [PubMed]
100. Chen, C.; Chandra, S.; Han, Y.; Seo, H. Deep Learning-Based Thermal Image Analysis for Pavement Defect Detection and Classification Considering Complex Pavement Conditions. *Remote Sens.* **2022**, *14*, 106. [CrossRef]
101. Nagy, A.M.; Czuni, L. Classification and Fast Few-Shot Learning of Steel Surface Defects with Randomized Network. *Appl. Sci.* **2022**, *12*, 3967. [CrossRef]
102. Dunphy, K.; Sadhu, A.; Wang, J. Multiclass damage detection in concrete structures using a transfer learning-based generative adversarial networks. *Struct. Control Health Monit.* **2022**, *29*. [CrossRef]
103. Guo, X.; Liu, X.; Krolczyk, G.; Sulowicz, M.; Glowacz, A.; Gardoni, P.; Li, Z. Damage Detection for Conveyor Belt Surface Based on Conditional Cycle Generative Adversarial Network. *Sensors* **2022**, *22*, 3485. [CrossRef]
104. Ogunjinmi, P.D.; Park, S.S.; Kim, B.; Lee, D.E. Rapid Post-Earthquake Structural Damage Assessment Using Convolutional Neural Networks and Transfer Learning. *Sensors* **2022**, *22*, 3471. [CrossRef]
105. Chen, H.C. Automated Detection and Classification of Defective and Abnormal Dies in Wafer Images. *Appl. Sci.* **2020**, *10*, 3423. [CrossRef]
106. Wu, X.; Xu, H.; Wei, X.; Wu, Q.; Zhang, W.; Han, X. Damage Identification of Low Emissivity Coating Based on Convolution Neural Network. *IEEE Access* **2020**, *8*, 156792–156800. [CrossRef]
107. Stamoulakatos, A.; Cardona, J.; McCaig, C.; Murray, D.; Filius, H.; Atkinson, R.; Bellekens, X.; Michie, C.; Andonovic, I.; Lazaridis, P.; et al. Automatic Annotation of Subsea Pipelines Using Deep Learning. *Sensors* **2020**, *20*, 674. [CrossRef]
108. Konovalenko, I.; Maruschak, P.; Brevus, V.; Prentkovskis, O. Recognition of Scratches and Abrasions on Metal Surfaces Using a Classifier Based on a Convolutional Neural Network. *Metals* **2021**, *11*, 549. [CrossRef]
109. Xiang, S.; Jiang, S.; Liu, X.; Zhang, T.; Yu, L. Spiking VGG7: Deep Convolutional Spiking Neural Network with Direct Training for Object Recognition. *Electronics* **2022**, *11*, 2097. [CrossRef]
110. Meister, S.; Wermes, M.; Stueve, J.; Groves, R.M. Cross-evaluation of a parallel operating SVM-CNN classifier for reliable internal decision-making processes in composite inspection. *J. Manuf. Syst.* **2021**, *60*, 620–639. [CrossRef]
111. Meister, S.; Moeller, N.; Stueve, J.; Groves, R.M. Synthetic image data augmentation for fibre layup inspection processes: Techniques to enhance the data set. *J. Intell. Manuf.* **2021**, *32*, 1767–1789. [CrossRef]

112. Al-Waisy, A.S.; Ibrahim, D.; Zebari, D.A.; Hammadi, S.; Mohammed, H.; Mohammed, M.A.; Damasevicius, R. Identifying defective solar cells in electroluminescence images using deep feature representations. *PeerJ Comput. Sci.* **2022**, *8*, e992. [CrossRef] [PubMed]
113. Maeda, K.; Takahashi, S.; Ogawa, T.; Haseyama, M. Deterioration level estimation via neural network maximizing category-based ordinally supervised multi-view canonical correlation. *Multimed. Tools Appl.* **2021**, *80*, 23091–23112. [CrossRef]
114. Konovalenko, I.; Maruschak, P.; Brezinova, J.; Vinas, J.; Brezina, J. Steel Surface Defect Classification Using Deep Residual Neural Network. *Metals* **2020**, *10*, 846. [CrossRef]
115. Liu, Z.; Zhang, C.; Li, C.; Ding, S.; Dong, Y.; Huang, Y. Fabric defect recognition using optimized neural networks. *J. Eng. Fibers Fabr.* **2019**, *14*, 1558925019897396. [CrossRef]
116. Mushabab Alqahtani, M.; Kumar Dutta, A.; Almotairi, S.; Ilayaraja, M.; Abdulrahman Albraikan, A.; Al-Wesabi, F.N.; Al Duhayyim, M. Sailfish Optimizer with EfficientNet Model for Apple Leaf Disease Detection. *Comput. Mater. Contin.* **2023**, *74*, 217–233. [CrossRef]
117. Barman, U.; Pathak, C.; Mazumder, N.K. Comparative assessment of Pest damage identification of coconut plant using damage texture and color analysis. *Multimed. Tools Appl.* **2023**, *82*, 25083–25105. [CrossRef]
118. Ksibi, A.; Ayadi, M.; Soufiene, B.O.; Jamjoom, M.M.; Ullah, Z. MobiRes-Net: A Hybrid Deep Learning Model for Detecting and Classifying Olive Leaf Diseases. *Appl. Sci.* **2022**, *12*, 10278. [CrossRef]
119. Wu, L.; Liu, Z.; Bera, T.; Ding, H.; Langley, D.A.; Jenkins-Barnes, A.; Furlanello, C.; Maggio, V.; Tong, W.; Xu, J. A deep learning model to recognize food contaminating beetle species based on elytra fragments. *Comput. Electron. Agric.* **2019**, *166*, 105002. [CrossRef]
120. Kang, D.H.; Cha, Y.J. Efficient attention-based deep encoder and decoder for automatic crack segmentation. *Struct. Health Monit. Int. J.* **2022**, *21*, 2190–2205. [CrossRef] [PubMed]
121. Yuan, G.; Li, J.; Meng, X.; Li, Y. CurSeg: A pavement crack detector based on a deep hierarchical feature learning segmentation framework. *IET Intell. Transp. Syst.* **2022**, *16*, *782–799.* [CrossRef]
122. Andrushia, D.; Anand, N.; Neebha, T.M.; Naser, M.Z.; Lubloy, E. Autonomous detection of concrete damage under fire conditions. *Autom. Constr.* **2022**, *140*, 104364. [CrossRef]
123. Wan, C.; Ma, S.; Song, K. TSSTNet: A Two-Stream Swin Transformer Network for Salient Object Detection of No-Service Rail Surface Defects. *Coatings* **2022**, *12*, 1730. [CrossRef]
124. Su, L.; Huang, H.; Qin, L.; Zhao, W. Transformer Vibration Detection Based on YOLOv4 and Optical Flow in Background of High Proportion of Renewable Energy Access. *Front. Energy Res.* **2022**, *10*, 764903. [CrossRef]
125. Oishi, Y.; Habaragamuwa, H.; Zhang, Y.; Sugiura, R.; Asano, K.; Akai, K.; Shibata, H.; Fujimoto, T. Automated abnormal potato plant detection system using deep learning models and portable video cameras. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *104*, 102509. [CrossRef]
126. Naddaf-Sh, M.M.; Hosseini, S.; Zhang, J.; Brake, N.A.; Zargarzadeh, H. Real-Time Road Crack Mapping Using an Optimized Convolutional Neural Network. *Complexity* **2019**, *2019*, 2470735. [CrossRef]
127. Song, W.; Jia, G.; Zhu, H.; Jia, D.; Gao, L. Automated Pavement Crack Damage Detection Using Deep Multiscale Convolutional Features. *J. Adv. Transp.* **2020**, *2020*, 6412562. [CrossRef]
128. Saleem, M.R.; Park, J.W.; Lee, J.H.; Jung, H.J.; Sarwar, M.Z. Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network. *Struct. Health Monit. Int. J.* **2021**, *20*, 1760–1777. [CrossRef]
129. Chen, R. Migration Learning-Based Bridge Structure Damage Detection Algorithm. *Sci. Program.* **2021**, *2021*, 1102521. [CrossRef]
130. Chun, C.; Ryu, S.K. Road Surface Damage Detection Using Fully Convolutional Neural Networks and Semi-Supervised Learning. *Sensors* **2019**, *19*, 5501. [CrossRef] [PubMed]
131. Shen, Y.; Yu, Z.; Li, C.; Zhao, C.; Sun, Z. Automated Detection for Concrete Surface Cracks Based on Deeplabv3+BDF. *Buildings* **2023**, *13*, 118. [CrossRef]
132. Kou, L.; Sysyn, M.; Fischer, S.; Liu, J.; Nabochenko, O. Optical Rail Surface Crack Detection Method Based on Semantic Segmentation Replacement for Magnetic Particle Inspection. *Sensors* **2022**, *22*, 8214. [CrossRef] [PubMed]
133. Siriborvornratanakul, T. Downstream Semantic Segmentation Model for Low-Level Surface Crack Detection. *Adv. Multimed.* **2022**, *2022*, 3712289 . [CrossRef]
134. Chen, H.; Lin, H.; Yao, M. Improving the Efficiency of Encoder-Decoder Architecture for Pixel-Level Crack Detection. *IEEE Access* **2019**, *7*, 186657–186670. [CrossRef]
135. Li, S.; Zhao, X. A Performance Improvement Strategy for Concrete Damage Detection Using Stacking Ensemble Learning of Multiple Semantic Segmentation Networks. *Sensors* **2022**, *22*, 3341. [CrossRef]
136. Shim, S.; Kim, J.; Cho, G.C.; Lee, S.W. Stereo-vision-based 3D concrete crack detection using adversarial learning with balanced ensemble discriminator networks. *Struct. Health Monit. Int. J.* **2023**, 22, 1353–1375. [CrossRef]
137. Meng, M.; Zhu, K.; Chen, K.; Qu, H. A Modified Fully Convolutional Network for Crack Damage Identification Compared with Conventional Methods. *Model. Simul. Eng.* **2021**, *2021*, 5298882. [CrossRef]
138. Wu, D.; Zhang, H.; Yang, Y. Deep Learning-Based Crack Monitoring for Ultra-High Performance Concrete (UHPC). *J. Adv. Transp.* **2022**, *2022*, 4117957. [CrossRef]

139. Ali, L.; Khan, W.; Chaiyasarn, K. Damage detection and localization in masonry structure using faster region convolutional networks. *Int. J. Geomate* **2019**, *17*, 98–105. [CrossRef]
140. Dong, C.; Li, L.; Yan, J.; Zhang, Z.; Pan, H.; Catbas, F.N. Pixel-Level Fatigue Crack Segmentation in Large-Scale Images of Steel Structures Using an Encoder-Decoder Network. *Sensors* **2021**, *21*, 4135. [CrossRef] [PubMed]
141. Jamshidi, M.; El-Badry, M.; Nourian, N. Improving Concrete Crack Segmentation Networks through CutMix Data Synthesis and Temporal Data Fusion. *Sensors* **2023**, *23*, 504. [CrossRef] [PubMed]
142. Yu, G.; Dong, J.; Wang, Y.; Zhou, X. RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation. *Sensors* **2023**, *23*, 53. [CrossRef]
143. Loverdos, D.; Sarhosis, V. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Autom. Constr.* **2022**, *140*, 104389. [CrossRef]
144. Pantoja-Rosero, B.G.; Oner, D.; Kozinski, M.; Achanta, R.; Fua, P.; Perez-Cruz, F.; Beyer, K. TOPO-Loss for continuity-preserving crack detection using deep learning. *Constr. Build. Mater.* **2022**, *344*. [CrossRef]
145. Zhao, S.; Kang, F.; Li, J. Non-Contact Crack Visual Measurement System Combining Improved U-Net Algorithm and Canny Edge Detection Method with Laser Rangefinder and Camera. *Appl. Sci.* **2022**, *12*, 10651. [CrossRef]
146. Shim, S.; Cho, G.C. Lightweight Semantic Segmentation for Road-Surface Damage Recognition Based on Multiscale Learning. *IEEE Access* **2020**, *8*, 102680–102690. [CrossRef]
147. Ji, H.; Cui, X.; Ren, W.; Liu, L.; Wang, W. Visual inspection for transformer insulation defects by a patrol robot fish based on deep learning. *IET Sci. Meas. Technol.* **2021**, *15*, 606–618. [CrossRef]
148. Shim, S.; Kim, J.; Lee, S.W.; Cho, G.C. Road damage detection using super-resolution and semi-supervised learning with generative adversarial network. *Autom. Constr.* **2022**, *135*, 104139. [CrossRef]
149. Dong, J.; Li, Z.; Wang, Z.; Wang, N.; Guo, W.; Ma, D.; Hu, H.; Zhong, S. Pixel-Level Intelligent Segmentation and Measurement Method for Pavement Multiple Damages Based on Mobile Deep Learning. *IEEE Access* **2021**, *9*, 143860–143876. [CrossRef]
150. Li, T.; Hao, T. Damage Detection of Insulators in Catenary Based on Deep Learning and Zernike Moment Algorithms. *Appl. Sci.* **2022**, *12*, 5004. [CrossRef]
151. Chen, S.; Zhang, Y.; Zhang, Y.; Yu, J.; Zhu, Y. Embedded system for road damage detection by deep convolutional neural network. *Math. Biosci. Eng.* **2019**, *16*, 7982–7994. [CrossRef] [PubMed]
152. Luo, Q.; Su, J.; Yang, C.; Gui, W.; Silven, O.; Liu, L. CAT-EDNet: Cross-Attention Transformer-Based Encoder-Decoder Network for Salient Defect Detection of Strip Steel Surface. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5009813. [CrossRef]
153. Liu, W.; Zhang, J.; Su, Z.; Zhou, Z.; Liu, L. Binary Neural Network for Automated Visual Surface Defect Detection. *Sensors* **2021**, *21*, 6868. [CrossRef]
154. Konovalenko, I.; Maruschak, P.; Brezinova, J.; Prentkovskis, O.; Brezina, J. Research of U-Net-Based CNN Architectures for Metal Surface Defect Detection. *Machines* **2022**, *10*, 327. [CrossRef]
155. Konovalenko, I.; Maruschak, P.; Kozbur, H.; Brezinova, J.; Brezina, J.; Nazarevich, B.; Shkira, Y. Influence of Uneven Lighting on Quantitative Indicators of Surface Defects. *Machines* **2022**, *10*, 194. [CrossRef]
156. Wang, Z.; Zhang, Y.; Luo, L.; Wang, N. AnoDFDNet: A Deep Feature Difference Network for Anomaly Detection. *J. Sens.* **2022**, *2022*, 3538541. [CrossRef]
157. Park, S.S.; Tran, V.T.; Lee, D.E. Application of Various YOLO Models for Computer Vision-Based Real-Time Pothole Detection. *Appl. Sci.* **2021**, *11*, 11229. [CrossRef]
158. van Ruitenbeek, R.E.; Bhulai, S. Multi-view damage inspection using single-view damage projection. *Mach. Vis. Appl.* **2022**, *33*, 46. [CrossRef]
159. Zhao, G.; Hu, J.; Xiao, W.; Zou, J. A mask R-CNN based method for inspecting cable brackets in aircraft. *Chin. J. Aeronaut.* **2021**, *34*, 214–226. [CrossRef]
160. Pan, X.; Yang, T.Y. Image-based monitoring of bolt loosening through deep-learning-based integrated detection and tracking. *Comput. Aided Civ. Infrastruct. Eng.* **2022**, *37*, 1207–1222. [CrossRef]
161. Brion, D.A.J.; Shen, M.; Pattinson, S.W. Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Addit. Manuf.* **2022**, *56*, 102838. [CrossRef]
162. Salcedo, E.; Jaber, M.; Carrion, J.R. A Novel Road Maintenance Prioritisation System Based on Computer Vision and Crowdsourced Reporting. *J. Sens. Actuator Netw.* **2022**, *11*, 15. [CrossRef]
163. Zhao, J.; Zhang, X.; Yan, J.; Qiu, X.; Yao, X.; Tian, Y.; Zhu, Y.; Cao, W. A Wheat Spike Detection Method in UAV Images Based on Improved YOLOv5. *Remote Sens.* **2021**, *13*, 3095. [CrossRef]
164. Huetten, N.; Meyes, R.; Meisen, T. Vision Transformer in Industrial Visual Inspection. *Appl. Sci.* **2022**, *12*, 11981. [CrossRef]
165. Wang, C.; Zhao, J.; Yu, Z.; Xie, S.; Ji, X.; Wan, Z. Real-Time Foreign Object and Production Status Detection of Tobacco Cabinets Based on Deep Learning. *Appl. Sci.* **2022**, *12*, 10347. [CrossRef]
166. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [CrossRef]
167. Tanveer, M.; Kim, B.; Hong, J.; Sim, S.H.; Cho, S. Comparative Study of Lightweight Deep Semantic Segmentation Models for Concrete Damage Detection. *Appl. Sci.* **2022**, *12*, 12786. [CrossRef]
168. Islam, M.M.M.; Kim, J.M. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder-Decoder Network. *Sensors* **2019**, *19*, 4251. [CrossRef]

169. Kumar, P.; Sharma, A.; Kota, S.R. Automatic Multiclass Instance Segmentation of Concrete Damage Using Deep Learning Model. *IEEE Access* **2021**, *9*, 90330–90345. [CrossRef]
170. He, Y.; Jin, Z.; Zhang, J.; Teng, S.; Chen, G.; Sun, X.; Cui, F. Pavement Surface Defect Detection Using Mask Region-Based Convolutional Neural Networks and Transfer Learning. *Appl. Sci.* **2022**, *12*, 7364. [CrossRef]
171. Kulambayev, B.; Beissenova, G.; Katayev, N.; Abduraimova, B.; Zhaidakbayeva, L.; Sarbassova, A.; Akhmetova, O.; Issayev, S.; Suleimenova, L.; Kasenov, S.; et al. A Deep Learning-Based Approach for Road Surface Damage Detection. *Comput. Mater. Contin.* **2022**, *73*, 3403–3418. [CrossRef]
172. Zhou, S.; Pan, Y.; Huang, X.; Yang, D.; Ding, Y.; Duan, R. Crack Texture Feature Identification of Fiber Reinforced Concrete Based on Deep Learning. *Materials* **2022**, *15*, 3940. [CrossRef] [PubMed]
173. Bai, Y.; Zha, B.; Sezen, H.; Yilmaz, A. Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events. *Struct. Health Monit. Int. J.* **2023**, *22*, 338–352. [CrossRef]
174. Dais, D.; Bal, I.E.; Smyrou, E.; Sarhosis, V. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Autom. Constr.* **2021**, *125*, 103606. [CrossRef]
175. Hu, G.X.; Hu, B.L.; Yang, Z.; Huang, L.; Li, P. Pavement Crack Detection Method Based on Deep Learning Models. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5573590. [CrossRef]
176. Du, F.J.; Jiao, S.J. Improvement of Lightweight Convolutional Neural Network Model Based on YOLO Algorithm and Its Research in Pavement Defect Detection. *Sensors* **2022**, *22*, 3537. [CrossRef]
177. Li, L.; Fang, B.; Zhu, J. Performance Analysis of the YOLOv4 Algorithm for Pavement Damage Image Detection with Different Embedding Positions of CBAM Modules. *Appl. Sci.* **2022**, *12*, 10180. [CrossRef]
178. Wang, L.; Li, J.; Kang, F. Crack Location and Degree Detection Method Based on YOLOX Model. *Appl. Sci.* **2022**, *12*, 12572. [CrossRef]
179. Yang, Z.; Ni, C.; Li, L.; Luo, W.; Qin, Y. Three-Stage Pavement Crack Localization and Segmentation Algorithm Based on Digital Image Processing and Deep Learning Techniques. *Sensors* **2022**, *22*, 8459. [CrossRef] [PubMed]
180. Yin, J.; Qu, J.; Huang, W.; Chen, Q. Road Damage Detection and Classification based on Multi-level Feature Pyramids. *Ksii Trans. Internet Inf. Syst.* **2021**, *15*, 786–799. [CrossRef]
181. Xu, H.; Chen, B.; Qin, J. A CNN-Based Length-Aware Cascade Road Damage Detection Approach. *Sensors* **2021**, *21*, 689. [CrossRef]
182. Mallaiyan Sathiaseelan, M.A.; Paradis, O.P.; Taheri, S.; Asadizanjani, N. Why Is Deep Learning Challenging for Printed Circuit Board (PCB) Component Recognition and How Can We Address It? *Cryptography* **2021**, *5*, 9. [CrossRef]
183. Schwebig, A.I.M.; Tutsch, R. Intelligent fault detection of electrical assemblies using hierarchical convolutional networks for supporting automatic optical inspection systems. *J. Sens. Sens. Syst.* **2020**, *9*, 363–374. [CrossRef]
184. Yan, S.; Song, X.; Liu, G. Deeper and Mixed Supervision for Salient Object Detection in Automated Surface Inspection. *Math. Probl. Eng.* **2020**, *2020*, 3751053. [CrossRef]
185. Liang, H.; Lee, S.C.; Seo, S. Automatic Recognition of Road Damage Based on Lightweight Attentional Convolutional Neural Network. *Sensors* **2022**, *22*, 9599. [CrossRef]
186. Zhang, H.; Wu, Z.; Qiu, Y.; Zhai, X.; Wang, Z.; Xu, P.; Liu, Z.; Li, X.; Jiang, N. A New Road Damage Detection Baseline with Attention Learning. *Appl. Sci.* **2022**, *12*, 7594. [CrossRef]
187. Lin, C.S.; Hsieh, H.Y. An Automatic Defect Detection System for Synthetic Shuttlecocks Using Transformer Model. *IEEE Access* **2022**, *10*, 37412–37421. [CrossRef]
188. Abedini, F.; Bahaghighat, M.; S'hoyan, M. Wind turbine tower detection using feature descriptors and deep learning. *Facta Univ. Ser. Electron. Energetics* **2020**, *33*, 133–153. [CrossRef]
189. Kim, H.; Lee, S.; Han, S. Railroad Surface Defect Segmentation Using a Modified Fully Convolutional Network. *Ksii Trans. Internet Inf. Syst.* **2020**, *14*, 4763–4775. [CrossRef]
190. Zhang, Z.; Liang, M.; Wang, Z. A Deep Extractor for Visual Rail Surface Inspection. *IEEE Access* **2021**, *9*, 21798–21809. [CrossRef]
191. Tabernik, D.; Sela, S.; Skvarc, J.; Skocaj, D. Segmentation-based deep-learning approach for surface-defect detection. *J. Intell. Manuf.* **2020**, *31*, 759–776. [CrossRef]
192. Shi, H.; Lai, R.; Li, G.; Yu, W. Visual inspection of surface defects of extreme size based on an advanced FCOS. *Appl. Artif. Intell.* **2022**, *36*, 2122222. [CrossRef]
193. Zhou, Q.; Situ, Z.; Teng, S.; Chen, W.; Chen, G.; Su, J. Comparison of classic object-detection techniques for automated sewer defect detection. *J. Hydroinform.* **2022**, *24*, 406–419. [CrossRef]
194. Shin, H.K.; Lee, S.W.; Hong, G.P.; Sael, L.; Lee, S.H.; Kim, H.Y. Defect-Detection Model for Underground Parking Lots Using Image Object-Detection Method. *Comput. Mater. Contin.* **2021**, *66*, 2493–2507. [CrossRef]
195. Urbonas, A.; Raudonis, V.; Maskeliunas, R.; Damasevicius, R. Automated Identification of Wood Veneer Surface Defects Using Faster Region-Based Convolutional Neural Network with Data Augmentation and Transfer Learning. *Appl. Sci.* **2019**, *9*, 4898. [CrossRef]
196. Roberts, R.; Giancontieri, G.; Inzerillo, L.; Di Mino, G. Towards Low-Cost Pavement Condition Health Monitoring and Analysis Using Deep Learning. *Appl. Sci.* **2020**, *10*, 319. [CrossRef]
197. Shihavuddin, A.S.M.; Chen, X.; Fedorov, V.; Christensen, A.N.; Riis, N.A.B.; Branner, K.; Dahl, A.B.; Paulsen, R.R. Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis. *Energies* **2019**, *12*, 676. [CrossRef]

198. Allam, A.; Moussa, M.; Tarry, C.; Veres, M. Detecting Teeth Defects on Automotive Gears Using Deep Learning. *Sensors* **2021**, *21*, 8480. [CrossRef]
199. Lee, K.; Hong, G.; Sael, L.; Lee, S.; Kim, H.Y. MultiDefectNet: Multi-Class Defect Detection of Building Facade Based on Deep Convolutional Neural Network. *Sustainability* **2020**, *12*, 9785. [CrossRef]
200. Wei, R.; Bi, Y. Research on Recognition Technology of Aluminum Profile Surface Defects Based on Deep Learning. *Materials* **2019**, *12*, 1681. [CrossRef] [PubMed]
201. Palanisamy, P.; Mohan, R.E.; Semwal, A.; Melivin, L.M.J.; Gomez, B.F.; Balakrishnan, S.; Elangovan, K.; Ramalingam, B.; Terntzer, D.N. Drain Structural Defect Detection and Mapping Using AI-Enabled Reconfigurable Robot Raptor and IoRT Framework. *Sensors* **2021**, *21*, 7287. [CrossRef] [PubMed]
202. Siu, C.; Wang, M.; Cheng, J.C.P. A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection. *Autom. Constr.* **2022**, *137*, 104213. [CrossRef]
203. Chen, Q.; Gan, X.; Huang, W.; Feng, J.; Shim, H. Road Damage Detection and Classification Using Mask R-CNN with DenseNet Backbone. *Comput. Mater. Contin.* **2020**, *65*, 2201–2215. [CrossRef]
204. Zhang, J.; Cosma, G.; Watkins, J. Image Enhanced Mask R-CNN: A Deep Learning Pipeline with New Evaluation Measures for Wind Turbine Blade Defect Detection and Classification. *J. Imaging* **2021**, *7*, 46. [CrossRef] [PubMed]
205. Dogru, A.; Bouarfa, S.; Arizar, R.; Aydogan, R. Using Convolutional Neural Networks to Automate Aircraft Maintenance Visual Inspection. *Aerospace* **2020**, *7*, 171. [CrossRef]
206. Kim, B.; Cho, S. Automated Multiple Concrete Damage Detection Using Instance Segmentation Deep Learning Model. *Appl. Sci.* **2020**, *10*, 8008. [CrossRef]
207. Kim, A.; Lee, K.; Lee, S.; Song, J.; Kwon, S.; Chung, S. Synthetic Data and Computer-Vision-Based Automated Quality Inspection System for Reused Scaffolding. *Appl. Sci.* **2022**, *12*, 10097. [CrossRef]
208. Yan, K.; Zhang, Z. Automated Asphalt Highway Pavement Crack Detection Based on Deformable Single Shot Multi-Box Detector Under a Complex Environment. *IEEE Access* **2021**, *9*, 150925–150938. [CrossRef]
209. Jang, J.; Shin, M.; Lim, S.; Park, J.; Kim, J.; Paik, J. Intelligent Image-Based Railway Inspection System Using Deep Learning-Based Object Detection and Weber Contrast-Based Image Comparison. *Sensors* **2019**, *19*, 4738. [CrossRef] [PubMed]
210. Ramalingam, B.; Manuel, V.H.; Elara, M.R.; Vengadesh, A.; Lakshmanan, A.K.; Ilyas, M.; James, T.J.Y. Visual Inspection of the Aircraft Surface Using a Teleoperated Reconfigurable Climbing Robot and Enhanced Deep Learning Technique. *Int. J. Aerosp. Eng.* **2019**, *2019*, 5137139. [CrossRef]
211. Maeda, H.; Sekimoto, Y.; Seto, T.; Kashiyama, T.; Omata, H. Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 1127–1141. [CrossRef]
212. Lv, L.; Yao, Z.; Wang, E.; Ren, X.; Pang, R.; Wang, H.; Zhang, Y.; Wu, H. Efficient and Accurate Damage Detector for Wind Turbine Blade Images. *IEEE Access* **2022**, *10*, 123378–123386. [CrossRef]
213. Wei, Z.; Fernandes, H.; Herrmann, H.G.; Tarpani, J.R.; Osman, A. A Deep Learning Method for the Impact Damage Segmentation of Curve-Shaped CFRP Specimens Inspected by Infrared Thermography. *Sensors* **2021**, *21*, 395. [CrossRef] [PubMed]
214. Munawar, H.S.; Ullah, F.; Shahzad, D.; Heravi, A.; Qayyum, S.; Akram, J. Civil Infrastructure Damage and Corrosion Detection: An Application of Machine Learning. *Buildings* **2022**, *12*, 156. [CrossRef]
215. Wang, A.; Togo, R.; Ogawa, T.; Haseyama, M. Defect Detection of Subway Tunnels Using Advanced U-Net Network. *Sensors* **2022**, *22*, 2330. [CrossRef]
216. Zheng, Z.; Zhang, S.; Yu, B.; Li, Q.; Zhang, Y. Defect Inspection in Tire Radiographic Image Using Concise Semantic Segmentation. *IEEE Access* **2020**, *8*, 112674–112687. [CrossRef]
217. Wu, W.; Li, Q. Machine Vision Inspection of Electrical Connectors Based on Improved Yolo v3. *IEEE Access* **2020**, *8*, 166184–166196. [CrossRef]
218. Kumar, P.; Batchu, S.; Swamy S., N.; Kota, S.R. Real-Time Concrete Damage Detection Using Deep Learning for High Rise Structures. *IEEE Access* **2021**, *9*, 112312–112331. [CrossRef]
219. Lin, H.I.; Wibowo, F.S. Image Data Assessment Approach for Deep Learning-Based Metal Surface Defect-Detection Systems. *IEEE Access* **2021**, *9*, 47621–47638. [CrossRef]
220. Shihavuddin, A.S.M.; Rashid, M.R.A.; Maruf, M.H.; Abul Hasan, M.; ul Haq, M.A.; Ashique, R.H.; Al Mansur, A. Image based surface damage detection of renewable energy installations using a unified deep learning approach. *Energy Rep.* **2021**, *7*, 4566–4576. [CrossRef]
221. Yu, L.; Yang, E.; Luo, C.; Ren, P. AMCD: An accurate deep learning-based metallic corrosion detector for MAV-based real-time visual inspection. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *14*, 8087–8098. [CrossRef]
222. Du, F.; Jiao, S.; Chu, K. Application Research of Bridge Damage Detection Based on the Improved Lightweight Convolutional Neural Network Model. *Appl. Sci.* **2022**, *12*, 6225. [CrossRef]
223. Guo, Y.; Zeng, Y.; Gao, F.; Qiu, Y.; Zhou, X.; Zhong, L.; Zhan, C. Improved YOLOV4-CSP Algorithm for Detection of Bamboo Surface Sliver Defects With Extreme Aspect Ratio. *IEEE Access* **2022**, *10*, 29810–29820. [CrossRef]
224. Huang, H.; Luo, X. A Holistic Approach to IGBT Board Surface Fractal Object Detection Based on the Multi-Head Model. *Machines* **2022**, *10*, 713. [CrossRef]
225. Li, Y.; Fan, Y.; Wang, S.; Bai, J.; Li, K. Application of YOLOv5 Based on Attention Mechanism and Receptive Field in Identifying Defects of Thangka Images. *IEEE Access* **2022**, *10*, 81597–81611. [CrossRef]

226. Ma, H.; Lee, S. Smart System to Detect Painting Defects in Shipyards: Vision AI and a Deep-Learning Approach. *Appl. Sci.* **2022**, *12*, 2412. [CrossRef]
227. Teng, S.; Liu, Z.; Li, X. Improved YOLOv3-Based Bridge Surface Defect Detection by Combining High- and Low-Resolution Feature Images. *Buildings* **2022**, *12*, 1225. [CrossRef]
228. Wan, F.; Sun, C.; He, H.; Lei, G.; Xu, L.; Xiao, T. YOLO-LRDD: A lightweight method for road damage detection based on improved YOLOv5s. *Eurasip J. Adv. Signal Process.* **2022**, *2022*, 98. [CrossRef]
229. Zhang, C.; Yang, T.; Yang, J. Image Recognition of Wind Turbine Blade Defects Using Attention-Based MobileNetv1-YOLOv4 and Transfer Learning. *Sensors* **2022**, *22*, 6009. [CrossRef] [PubMed]
230. Wang, C.; Zhou, Z.; Chen, Z. An Enhanced YOLOv4 Model With Self-Dependent Attentive Fusion and Component Randomized Mosaic Augmentation for Metal Surface Defect Detection. *IEEE Access* **2022**, *10*, 97758–97766. [CrossRef]
231. Du, X.; Cheng, Y.; Gu, Z. Change Detection: The Framework of Visual Inspection System for Railway Plug Defects. *IEEE Access* **2020**, *8*, 152161–152172. [CrossRef]
232. Zheng, D.; Li, L.; Zheng, S.; Chai, X.; Zhao, S.; Tong, Q.; Wang, J.; Guo, L. A Defect Detection Method for Rail Surface and Fasteners Based on Deep Convolutional Neural Network. *Comput. Intell. Neurosci.* **2021**, *2021*, 2565500. [CrossRef] [PubMed]
233. Zhang, Y.; Sun, X.; Loh, K.J.; Su, W.; Xue, Z.; Zhao, X. Autonomous bolt loosening detection using deep learning. *Struct. Health Monit. Int. J.* **2020**, *19*, 105–122. [CrossRef]
234. Lei, T.; Lv, F.; Liu, J.; Zhang, L.; Zhou, T. Research on Fault Detection Algorithm of Electrical Equipment Based on Neural Network. *Math. Probl. Eng.* **2022**, *2022*, 9015796. [CrossRef]
235. An, Y.; Lu, Y.N.; Wu, T.R. Segmentation Method of Magnetic Tile Surface Defects Based on Deep Learning. *Int. J. Comput. Commun. Control* **2022**, *17*. [CrossRef]
236. Chen, S.W.; Tsai, C.J.; Liu, C.H.; Chu, W.C.C.; Tsai, C.T. Development of an Intelligent Defect Detection System for Gummy Candy under Edge Computing. *J. Internet Technol.* **2022**, *23*, 981–988. [CrossRef]
237. Song, K.; Yan, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Appl. Surf. Sci.* **2013**, *285*, 858–864. [CrossRef]
238. Dorafshan, S.; Thomas, R.J.; Maguire, M. SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data Brief* **2018**, *21*, 1664–1668. [CrossRef]
239. Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [CrossRef]
240. Arya, D.; Maeda, H.; Ghosh, S.K.; Toshniwal, D.; Mraz, A.; Kashiyama, T.; Sekimoto, Y. Transfer Learning-based Road Damage Detection for Multiple Countries. *arXiv* **2020**, arXiv:2008.13101. [CrossRef]
241. Gan, J.; Li, Q.; Wang, J.; Yu, H. A Hierarchical Extractor-Based Visual Rail Surface Inspection System. *IEEE Sens. J.* **2017**, *17*, 7935–7944. [CrossRef]
242. Grishin, A.; Boris V.I.; Inversion, O. Severstal: Steel Defect Detection Dataset. 2019. Available online: https://kaggle.com/competitions/severstal-steel-defect-detection (accessed on 17 January 2023).
243. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans. Image Processing: Publ. IEEE Signal Process. Soc.* **2018**, *28*, 1498–1512. [CrossRef]
244. Zou, Q.; Cao, Y.; Li, Q.; Mao, Q.; Wang, S. CrackTree: Automatic crack detection from pavement images. *Pattern Recognit. Lett.* **2012**, *33*, 227–238. [CrossRef]
245. Özgenel, Ç.F. *Concrete Crack Images for Classification*; Mendeley: London, UK, 2018. [CrossRef]
246. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1525–1535. [CrossRef]
247. Amhaz, R.; Chambon, S.; Idier, J.; Baltazart, V. Automatic Crack Detection on Two-Dimensional Pavement Images: An Algorithm Based on Minimal Path Selection. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2718–2729. [CrossRef]
248. Huang, Y.; Qiu, C.; Yuan, K. Surface defect saliency of magnetic tile. *Vis. Comput.* **2020**, *36*, 85–96. [CrossRef]
249. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
250. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
251. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *arXiv* **2014**, arXiv:1411.4038v2. [CrossRef].
252. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision–ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
253. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
254. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015.

255. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
256. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 30 June 2016.
257. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
258. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
259. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
260. Tan, M.; Le V, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *PMLR* **2019**, 97, 6105–6114.
261. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
262. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012.; Kwon, Y.; Michael, K.; Tao, X.; Fang, J.; imyhxy.; et al. *ultralytics/yolov5: V7.0—YOLOv5 SOTA Realtime Instance Segmentation*; Zenodo: Geneva, Switzerland, 2022. [CrossRef]
263. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. *arXiv* **2019**, arXiv:1908.08681. [CrossRef].
264. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 15–20 June 2019. [CrossRef]
265. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
266. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
267. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2020.
268. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
269. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755.
270. Everingham, M.; Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
271. Dai, Z.; Cai, B.; Lin, Y.; Chen, J. Up-detr: Unsupervised pre-training for object detection with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 1601–1610.
272. Bar, A.; Wang, X.; Kantorov, V.; Reed, C.J.; Herzig, R.; Chechik, G.; Rohrbach, A.; Darrell, T.; Globerson, A. DETReg: Unsupervised Pretraining with Region Priors for Object Detection. *arXiv* **2021**, arXiv:2106.04550.
273. Xu, M.; Zhang, Z.; Hu, H.; Wang, J.; Wang, L.; Wei, F.; Bai, X.; Liu, Z. End-to-End Semi-Supervised Object Detection with Soft Teacher. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
274. Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. SimMIM: A Simple Framework for Masked Image Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
275. Bao, H.; Dong, L.; Wei, F. BEiT: BERT Pre-Training of Image Transformers. *arXiv* **2021**, arxiv:2106.08254.
276. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
277. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *PMLR* **2020**, *119*, 1597–1607.
278. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved Baselines with Momentum Contrastive Learning. *arXiv* **2020**, arXiv:2003.04297.
279. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow Twins: Self-Supervised Learning via Redundancy Reduction. *PMLR* **2021**, *139*, 12310–12320.
280. Chen, X.; Xie, S.; He, K. An Empirical Study of Training Self-Supervised Vision Transformers. In Proceedings of the CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021.
281. Bardes, A.; Ponce, J.; LeCun, Y. VICRegL: Self-Supervised Learning of Local Visual Features. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 8799–8810.
282. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. *PMLR* **2021**, *139*, 8748–8763.
283. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 18–24 July 2021; pp. 10096–10106.

284. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
285. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019.
286. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection. *arXiv* **2022**, arXiv:2203.03605. [CrossRef]
287. Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Li, C.; Yang, J.; Su, H.; Zhu, J.; et al. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. *arXiv* **2023**, arXiv:2303.05499.
288. Cai, Y.; Zhou, Y.; Han, Q.; Sun, J.; Kong, X.; Li, J.; Zhang, X. Reversible Column Networks. *arXiv* **2023**, arXiv:2212.11696.
289. Ren, T.; Yang, J.; Liu, S.; Zeng, A.; Li, F.; Zhang, H.; Li, H.; Zeng, Z.; Zhang, L. A Strong and Reproducible Object Detector with Only Public Datasets. *arXiv* **2023**, arXiv:2304.13027.
290. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2020**, arXiv:2010.04159.