IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 April 2023, accepted 21 April 2023, date of publication 1 May 2023, date of current version 8 May 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3271748*

SURVEY

# Deep Learning for Automatic Vision-Based Recognition of Industrial Surface Defects: A Survey

**MICHELA PRUNELLA[1], ROBERTO MARIA SCARDIGNO[1], (Graduate Student Member, IEEE), DOMENICO BUONGIORNO[1,2], ANTONIO BRUNETTI[1,2], NICOLA LONGO[2,3], RAFFAELE CARLI[1], (Senior Member, IEEE), MARIAGRAZIA DOTOLI[1], (Senior Member, IEEE), AND VITOANTONIO BEVILACQUA[1,2]**

[1]Department of Electrical and Information Engineering (DEI), Polytechnic University of Bari, 70126 Bari, Italy
[2]Apulian Bioengineering S.r.l., Modugno, 70026 Bari, Italy
[3]Comau S.p.A., Grugliasco, 10095 Turin, Italy

Corresponding author: Domenico Buongiorno (domenico.buongiorno@poliba.it)

This research was partially funded by the "Cognitive Diagnostics" Public-Private Laboratory between Polytechnic University of Bari and Comau® S.p.A. company. In addition, this research was partially supported by the European Union Next-Generation EU through PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR)–MISSIONE 4 COMPONENTE 2, INVESTIMENTO 3.3 – Decreto del Ministero dell'Universita e della Ricerca n.352 del 09/04/2022, within the Italian National Ph.D. Program in Autonomous Systems (DAuSy).

**ABSTRACT** Automatic vision-based inspection systems have played a key role in product quality assessment for decades through the segmentation, detection, and classification of defects. Historically, machine learning frameworks, based on hand-crafted feature extraction, selection, and validation, counted on a combined approach of parameterized image processing algorithms and explicated human knowledge. The outstanding performance of deep learning (DL) for vision systems, in automatically discovering a feature representation suitable for the corresponding task, has exponentially increased the number of scientific articles and commercial products aiming at industrial quality assessment. In such a context, this article reviews more than 220 relevant articles from the related literature published until February 2023, covering the recent consolidation and advances in the field of fully-automatic DL-based surface defects inspection systems, deployed in various industrial applications. The analyzed papers have been classified according to a bi-dimensional taxonomy, that considers both the specific defect recognition task and the employed learning paradigm. The dependency on large and high-quality labeled datasets and the different neural architectures employed to achieve an overall perception of both well-visible and subtle defects, through the supervision of fine or/and coarse data annotations have been assessed. The results of our analysis highlight a growing research interest in defect representation power enrichment, especially by transferring pre-trained layers to an optimized network and by explaining the network decisions to suggest trustworthy retention or rejection of the products being evaluated.

**INDEX TERMS** Artificial vision, auto-encoder, automatic recognition, feature attention mechanism, convolutional neural network, deep learning, explainable artificial intelligence, generative-adversarial network, industrial surface defects, transfer learning.

## I. INTRODUCTION

The automatic recognition of object defects and particularly defective surfaces from images and videos is surging in the field of industrial manufacturing, thanks to deep learning (DL)-enhanced decision-making [1]. The surface of a defective product may present anomalies referrable to the composition of employed materials [2], the inclusion of foreign objects debris [3], and disturbance affecting the production phases, such as machinery and finishing processes [4] and transportation on conveyor belt [5]. Recently, massive investments have been devoted to the development of technologies

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

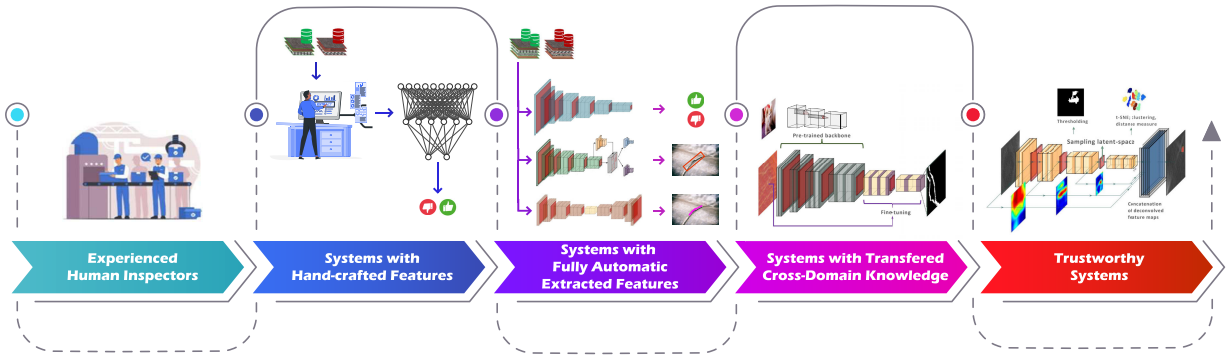



**FIGURE 1.** Evolution of defect recognition approaches in the manufacturing industry.

that are able to automatically identify and localize defects by computing and further processing a suitable representation of raw data. Historically, hand-crafted features were extracted using multiple human-aided techniques and employed into machine learning (ML) models, by feeding traditional topologies such as feed-forward artificial neural networks [6]. In particular, combined approaches of image processing algorithms and human knowledge to process, reduce, and transform initial data into meaningful concentrated representations were adopted. However, in recent years, the advent of deep neural networks, particularly convolutional neural networks (CNNs) and specialized topologies such as CNN Auto-Encoders (CNN-AEs) and Generative Adversarial Networks (GANs) embedded the feature learning step within the convolution operator. This allowed these models to effectively learn complex patterns in high-dimensional data, such as images and videos, automatically. Therefore, DL systems offer end-to-end solutions that are able to inspect surfaces without the human intervention, also cutting performance milestones [7], [8].

Although customized DL architectures are able to process images and videos remarkably accurate along the whole supply chain of any industrial application, there is a spread of specific algorithms, that tightly depend on defect types and products. Hence, they need to be fine-tuned for each particular application, which rely on the availability of massive and task-aware labeled samples to understand defective patterns [9]. This approach involves conspicuous human and economic resources with poor generalization performance when products, materials, or processes are switched [10]. Furthermore, huge computational expenses are tricky to be matched to real-time market-oriented systems, thus leading to un-competitiveness and propagation of defects [11], [12]. As a consequence, both academia and industry are willing to engage in the development of robust and softly-supervised vision systems [13].

### *A. FROM HUMAN-EXPERIENCED TO ARTIFICIAL TRUSTWORTHY KNOWLEDGE*

ML algorithms learn from experience as humans do innately. The human visual cortex is capable of judging the likelihood of a defect presence by comparing objects with the distribution of previously seen normal patterns. New types of defects can be early recognized only if skilled personnel can infer a lack of functionality related to the unseen or abnormal product appearance. Figure 1 illustrates the evolution of paradigms employed in the products defect recognition, and described in the sequel.

#### 1) EXPERIENCED HUMAN INSPECTORS

Originally, employees were involved in the entire decision-making process of manual recognition of industrial defects, which requires visual, cognitive, and physical effort. As a result, three main problems arise in the manual inspection and annotation process: first, low-contrast and hard-visible defects may not be detected, thus needing for highly experienced operators; second, the manual process consists in a subjective judgment derived from a time-consuming, tedious, and fatigue-prone task because of its repetitive nature; third, floating illumination setup, non-linear noise, and interference may act as bottleneck in the robustness of the pipeline due to the increasing recognition effort.

#### 2) SYSTEMS WITH HAND-CRAFTED FEATURES

With the introduction of ML, feature handcrafting is finalized by decisions provided by a learning network, sufficiently trained on a mid-size dataset and performing the classification task or an incomplete defect detection [14]. Manual feature extraction through traditional image processing consists in thresholding, transforming, or modeling images [15]. However, traditional ML performance are hindered by low generalization ability of hand-crafted features because they fit specific operating conditions (e.g., imaging acquisition scheme) and other changing factors in dynamic and time-varying large-scale production [16].

#### 3) SYSTEMS WITH FULLY-AUTOMATIC EXTRACTED FEATURES

DL replaces hybrid systems with end-to-end automatic learning that relies on the automatic feature extraction from hefty datasets. Features, which are extracted through convolutional filters, contain details and semantic image information, and are trained to be useful for final decision-making. The latter encloses classification and precise localization of anomalies.

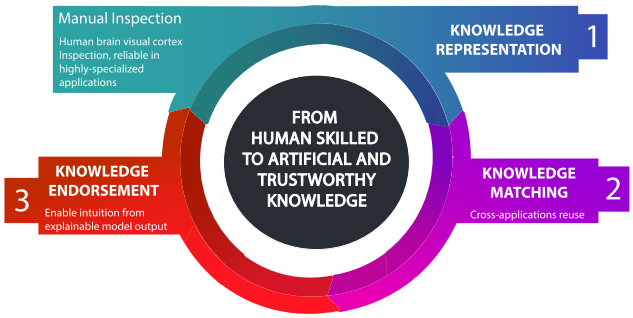



**FIGURE 2.** Cycle illustrating the collaborative connection between human and artificial knowledge.

These systems lighten human intervention that is involved almost only at the annotation stage.

#### 4) SYSTEMS WITH TRANSFERRED CROSS-DOMAIN KNOWLEDGE

Manual annotation requirements are further lowered by transferring features extracted from a large training on image data of unrelated fields, which usually improves performance and delivers a multi-domain expertise to the new network. As a consequence, such an approach has the potential to recognize new occurring defects, which are infrequent or even absent in in-house collected datasets or labeled training samples, thus reducing the false negative rate.

#### 5) TRUSTWORTHY SYSTEMS

The integration of DL with methods that aim to understand network behaviour, during the defects recognition process, or while performing post-hoc analysis, (e.g., visualizing inner states and extracted features in low-dimensional space), sheds light on the availability of new knowledge that could potentially strengthen the human experience. Increasing informativeness of feature layers allows reliable decisions and, consequently, an easier application in real-world scenarios [17].

The evolution of the above-mentioned approaches can be translated into a cyclic knowledge analysis and synthesis where human intervention is progressively lowered, as illustrated in Fig. 2. The cycle begins with problem representation and is developed to match acquired knowledge within robust models by aggregating and re-using cross-domain evidence. Finally, artificial models that make reasonable predictions, increase the valuable expertise thanks to their informativeness. The application of DL in real-world critical domains is often enabled by a trustworthy model output explaining decisions and actions to human users; interpretability can be defined as a means of finding reasonable evidence in rules of automatic feature selection.

### B. PAPER POSITIONING IN THE RELATED LITERATURE

In the last decade, several surveys and reviews focusing on image-based DL for industrial surface defects inspection have been published. However, the available works are either confined to a specialized application domain (such as metallurgical [18], [19], fabric and electronic [20], [21], [22], wood [23], and textile manufacturing [24], [25], [26]), or to a specific type of learning strategy (e.g., unsupervised methods [27]), or a specific architecture (e.g., Generative Adversarial Networks [28]), or else consider one single technique (e.g., transfer learning [29]) or requirement (e.g., lowering the number of labelled data for training [30]).

In addition to the cited works [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], the related literature counts other reviews not specialised in the inspected material, employed architecture, or supervision level used for training. For instance, Czimmermann et al. [31] explore visible and palpable defects and traditional feature extraction methods, characterizing supervised and unsupervised automatic approaches. Zheng et al. [32] provide a comprehensive review on three classes of industrial products (i.e.,, steel, fabric, and semiconductors) surface defects and respective tailored CNNs. Bhatt et al. [6] provide a taxonomy about the defect recognition tasks, inserting explainable decision-making methods and knowledge reuse into the roadmap of future goals. Mohammadi et al. [1] report well-documented image processing methods and categorize recent proposed models, with special attention on unsupervised models for anomaly detection. Ren et al. [33] compare optical illumination setups, image acquisition and deep defect detection related to Long-Short Memory, Deep-Belief Network, and Stacked Auto-Encoders. Moreover, Chen et al. [34] provide a subdivision in supervised, unsupervised, and weakly-supervised for DL solutions. Yang et al. [35] provide a classification of works in supervised and unsupervised methods, which in turn are distinguished into before DL and after DL approaches. The latest review papers of 2022 include the works of Gao et al. [36] that groups existing approaches from a feature perspective, including the traditional feature modeling, and learning through CNN, Auto-Encoder and Recurrent Neural Networks; Chen et al. [37] which compares standard DL architectures such as AlexNet, YOLO, VGG, and ResNet along with public defect datasets, and Cheng et al. [38] that explores vision-based solutions aided by personnel's knowledge for predictive maintenance in the manufacturing industry, but not including the most recent DL-based solutions. Table 1 reports a concise overview with comprehensive comparisons between our work and the previously published reviews. All considered works have been described in terms of: 1) year of publication, 2) learning methods used for the DL network training (i.e., fully-, weakly-, semi-, and un-supervised approach), 3) recognition tasks (i.e., classification, detection, and segmentation), 4) fields of specialization (e.g., regarding to material/application domain, learning setting, etc.), 5) an indicator specifying if there is a comprehensive analysis of DL methods (e.g., discussion about their strengths, weaknesses, and peculiarities), 6) an indicator specifying if all challenging aspects have been

recapitulated with respect to the industrial scenarios, 7) number of papers considered in the review of defect recognition, and 8) an indicator specifying if the entire workflow of articles identification, screening, and eligibility check is reported for the sake of reproducibility.

From this summary table, the novel contributions and advantages of our work are evident. This review focuses on DL systems for surface defect detection applicable on a variety of scenarios through the improvement and engineering of the main building-blocks, which result in flexible defect recognition systems. There is not a well-founded and effective reason to favor specialized solutions because, as proved by a great number of the reviewed works, the connection between cross-domains needs and achievements offers relevant insights. All these aspects are useful to guide the selection of the best practices in industrial vision systems, and are made clear even to those who do not have much experience. Therefore, the authors expect this review to provide a broad picture and sustain it by a substantial number of papers, which rely on cross-domain knowledge fusion or hybrid supervision to improve robustness of their frameworks.

Differently from not specialized reviews, this work starts from the presentation of problems and solutions concerning surface defects inspection on manufacturing products, and provides only the necessary theory to the comprehension of the discussion, which is corroborated by the evidence collected from real case studies. Moreover, as discussed below, the number of articles dealing with DL-based surface defect recognition systems using vision has been exponentially growing every year since 2015, thus motivating an updated revision that will include all the cutting-edge proposed solutions in the rapidly-changing industrial scenario.

In particular, this review discusses several relevant experiments that tackle key problems in the industry, such as data imbalance, multi-scale defects recognition, real-time constraint, and physical reasoning of network decisions, with the exclusive employment of DL models for defect recognition. The present work analyzes the recent relevant literature and deepens with documented discussion the future directions and developments, that were only mentioned in previous surveys. The authors are convinced that such an exhaustive review could be appreciated by practitioners as well as research community since it provides insights on tasks and methods emerging from un-compartmentalized literature analysis.

### C. PAPER OBJECTIVE AND STRUCTURE

The goal of this paper is to systematically and comprehensively review the significant trend of vision-based defect recognition among manufacturing industries with DL. For this purpose, the authors have focused on advanced systems that get accurate and timely inferences by improving deep networks architectures through knowledge transferring, and by distilling and explaining capabilities in the industrial scenario. An overview of DL-based and fully-automated defects inspection for manufacturing products is provided, highlighting innovative solutions regarding the deep architecture engineering, data generation, feature training, and visualization. These interesting strategies emerge from the exploitation of labeled samples to unravel trustworthy decisions about ambiguous or noisy samples as well as the exploration of vision-based solutions, deployed on the production line, for the continuous monitoring of products through edge-cloud collaborative systems [39].

The remainder of this paper is articulated as follows. Section II describes the research methodology, detailing the sources and explaining the inclusion and exclusion criteria used to select relevant articles from the related literature. In Section III the open issues and challenges addressed by existing vision-based DL systems are presented. Section IV proposes a bi-dimensional taxonomy of DL vision-based approaches used in industry, by grouping the defects recognition tasks alongside the different available levels of supervision. Section V develops the main methodological analysis in accordance with the proposed taxonomy. Section VI puts forward the findings and outcomes of the conducted analysis, highlighting the research gaps, trends, and promising development directions based on the synthesis of the collected evidence. Some concluding remarks are reported in Section VII. For interested readers, some useful theoretical background concepts are summarized in Appendix A.

## II. RESEARCH METHODOLOGY

This section describes how the related literature has been retrieved along with the screening process, in order to allow readers to assess the relevance of review findings. As a starting point, the search query to access the most popular scientific databases has been defined; second, for the sake of facilitating a transparent, complete, and accurate reporting, the selection process has been performed following the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines [40]. A flow chart summarizing the adopted search and screening process –and showing the number of papers retained or excluded in each phase– is given in Fig. 3.

### A. SEARCH CRITERIA

In order to provide a comprehensive and updated overview of the research topic, a large set of related publications were selected from the most popular scientific databases covering the years from 2015 to 2022.

First, an initial group of articles was retrieved through an electronic search on the Scopus® database among the articles that have been online available and accepted up to February 28, 2023. Since the research question to answer concerns the DL-based solutions used for industrial defects recognition through images and videos processing, a set of relevant keywords and keyphrases have been suitably included in the search query formulation. In particular, the query has been iteratively refined to be sensitive and specific to the research question, concatenating keyphrases with the Boolean operators 'AND', 'OR'. The precise applied query is: (anomal* OR

**TABLE 1.** Overview of existing survey papers and comparison with our work. The legend depicts the learning methods to train the DL network (i.e.,, F = fully-, W =weakly-, S = semi-, U =un-supervised) and the defect recognition tasks (C= classification, D=detection, S=segmentation) which have been encompassed.

| Year | Learning methods | Recognition tasks | Specialization field | Comprehensive DL methods analysis | Exhaustive challenges discussion | No. of reviewed papers | Review workflow description | Ref. |
|---|---|---|---|---|---|---|---|---|
| **2023** | F W S U<br>C ✓ ✓ ✓ ✓<br>D ✓ ✓ ✓ ✓<br>S ✓ ✓ ✓ ✓ | | **None** | ✓ | ✓ | **220** | ✓ | **This paper** |
| 2022 | F S | C-D-S | None | ✗ | ✓ | ca. 140 | ✗ | [36] |
| 2022 | F U | C-D-S | None | ✗ | ✗ | ca. 60 | ✗ | [37] |
| 2022 | F U | D | Electric vehicles | ✗ | ✗ | ca. 80 | ✓ | [22] |
| 2022 | F | D-S | Metals | ✓ | ✓ | 24 | ✗ | [18] |
| 2022 | F U | C | Timber | ✓ | ✗ | ca. 100 | ✗ | [23] |
| 2022 | F S U | D | Leather surfaces | ✗ | ✗ | 90 | ✓ | [24] |
| 2022 | U | D-S | Learning strategy | ✓ | ✓ | ca. 100 | ✗ | [27] |
| 2022 | F S U | C-D-S | DL Architecture | ✓ | ✗ | ca. 100 | ✗ | [28] |
| 2022 | F W S U | C-D-S | Small samples | ✗ | ✓ | ca. 60 | ✗ | [30] |
| 2022 | - | D | PdM | ✗ | ✓ | 37 | ✓ | [38] |
| 2021 | F S U | - | None | ✗ | ✓ | ca. 140 | ✗ | [32] |
| 2021 | F S U | C-D-S | None | ✗ | ✓ | ca. 110 | ✗ | [6] |
| 2021 | F S U | C | None | ✗ | ✗ | ca. 50 | ✗ | [1] |
| 2021 | F W U | C-D-S | None | ✗ | ✓ | ca. 260 | ✗ | [33] |
| 2021 | F W U | C-D-S | None | ✗ | ✓ | ca. 130 | ✗ | [34] |
| 2021 | F U | C-S | None | ✓ | ✗ | ca. 60 | ✗ | [35] |
| 2021 | - | D-S | Textile sector | ✓ | ✗ | ca. 100 | ✗ | [26] |
| 2021 | F S U | C-D-S | Silicon wafer | ✗ | ✓ | 44 | ✓ | [21] |
| 2021 | - | - | Transfer Learning | ✗ | ✓ | ca. 40 | ✗ | [29] |
| 2020 | F U | - | Fabric | ✗ | ✗ | ca. 80 | ✗ | [20] |
| 2020 | F U | C-D-S | None | ✓ | ✗ | ca. 200 | ✗ | [31] |
| 2019 | - | C-D-S | Leather | ✗ | ✓ | ca. 70 | ✗ | [25] |
| 2018 | F U | C | None | ✗ | ✗ | ca. 90 | ✗ | [19] |

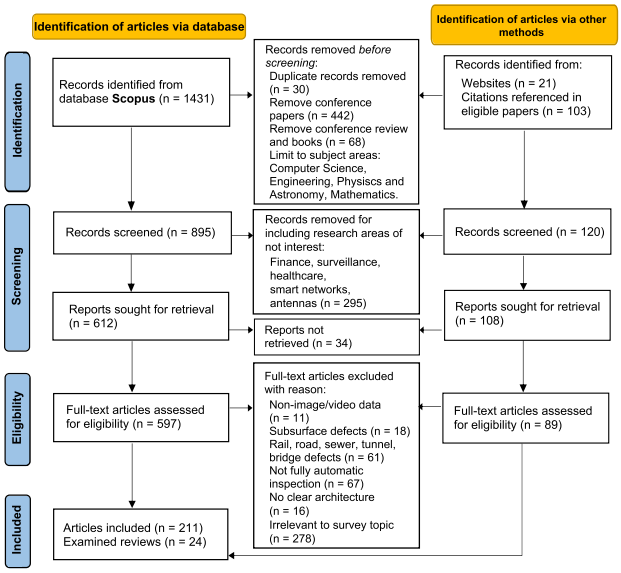



**FIGURE 3.** Flow diagram of the search and screening process adopted in the literature review: identification and selection rules following the Preferred Reporting Items for Systematic reviews and Meta-Analyses (PRISMA) guidelines [40] were used.

defect*) AND (“deep learning” OR “convolutional neural network*” OR CNN OR “deep n*”) AND (imag* OR vision OR visual OR vide*) AND (indust*). The initial query of 1,431 papers has been further constrained to journal articles only limited to the Computer Science, Engineering, Physics and Astronomy, and Mathematics research fields.

Moreover, all the studies missed by the electronic query but referenced in the lists of eligible studies, have been added by hand searching. Some articles facing the theory behind artificial learning and models, have been retrieved from Web of Science® database, for a total of additional 124 potentially relevant articles collected through other methods.

As indicated in Fig. 3, after duplicate removal the total number of selected articles at the end of the identification phase is 1015 (i.e., 895 and 120 from Scopus® and other methods, respectively).

### B. ARTICLE SCREENING AND SEARCH OUTCOMES

As a next step, the selected potentially relevant papers were submitted to a first screening process based on title and abstract. Defects recognition has applications in several different domains, such as finance, healthcare, antennas, cyber security and surveillance that were excluded accordingly; hence, in this stage, for caution, all papers whose relevance was unclear from the abstract were maintained in the list of elected papers. Conversely, all papers developing recognition algorithms for anomalies in other application than industrial manufacturing (e.g., finance, healthcare, surveillance, smart network security) were rejected.

Subsequently, the full text of all eligible papers has been analyzed and assessed in accordance with the relevance criterion. In particular, a considerable amount of papers passed

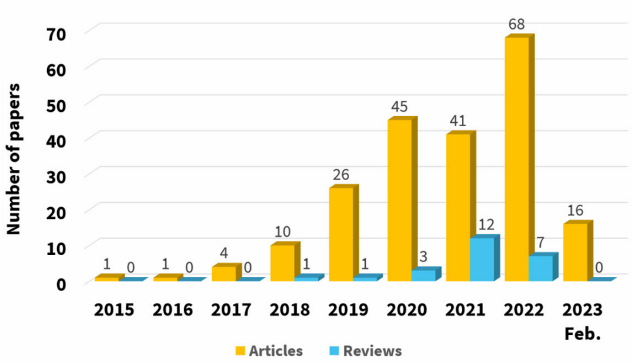



**FIGURE 4.** Number of surveyed articles and reviews as a result of the PRISMA selection process.

to the eligibility examination phase. By reading full-text, 173 papers were rejected, for the following reasons: defect recognition systems are not based on image/video data processing (e.g., mechanical destructive test); the application domains concern civil infrastructures (e.g., rail, road, sewer, tunnel, and bridge); the focus is on examining defects of the inner structure of the material, or on the parameter tuning provided by experts intervention; the study explains clearly neither the architecture nor the evaluation strategy. Furthermore, 278 articles have been rejected for being not relevant to the research question.

As a result of the final step, 211 fully compliant articles were included for the scope of this work, while 24 review papers were retrieved and considered for the paper positioning in Section I. The distribution of included articles and reviews over the years can be seen in Fig. 4.

## III. OPEN ISSUES AND CHALLENGES

This section summarizes the major challenges encountered by the recent defect recognition systems in industrial constrained scenarios. Focusing on handling each challenge separately may conflict with the remaining ones, since they are often concurrently present. Therefore, researchers are continually encouraged to find suitable trade-off solutions. These will be clearly delineated in Section VI.

### A. MULTI-SCALE AND DIFFERENTLY TARGETABLE DEFECTS

One or more categories of defects can occur on foreground objects, as well as on a framed portion of extended textures, or else can interest an object framed on a patterned background, as illustrated in Fig. 5, thus leading to an increasing recognition effort. A stable and reliable DL inspection system leverages on the self-adaptation ability of learnable layers to represent patterns within the data through feature extraction and selection [41]. The activations of CNN layers, for example, are a hierarchical translation of image characteristics into a stacked sequence of information from low-level to high or semantic level [42]. Low-level features are more easily traceable to morphological image processing

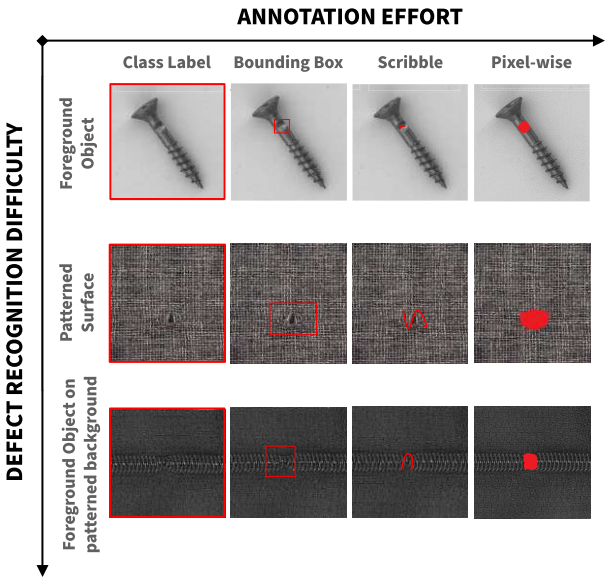



**FIGURE 5.** Illustration of four types of labeling documented in the literature and sorted from left to right with increasing detail annotation effort for three objects from the MVTec AD dataset [53]. The vertical axis, from top to bottom, denotes an increasing difficulty in defect recognition: defect occurs on a foreground object placed on a homogeneous background (screw), a patterned surface (carpet), and a foreground object placed on a defect-free and patterned background (zipper).

algorithms through gradient calculation (that are responsive to adjacent pixel values changing rapidly), such as edge and shape detection [43]. Since these features are highly sensitive to noise and background information, they should include further abstract characterization, in order to allow accurate description and lower reconstruction error, but it is at the expense of more training data and inference time, due to the exponential growth of convolution and non-linear operations in deeper architectures [44].

The information carried by feature maps measures the response to the convolutional kernel and involves spatial position. The appealing hypothesis is that the information irrelevant to defect inspection is down-weighted and finally lost while sparse defective regions are preserved during the resolution reduction. This goal is hard to achieve if the large majority of pixels belongs to non-defective background. These highly redundant portions of the feature matrix occupy the visual receptive field inefficiently, which leads to overlapping latent space representations between classes.

### B. IMBALANCED AND SMALL DATASET

In the industrial scenario, it is difficult to acquire large and balanced training data since, thanks to the overall quality of processes, anomalies rarely occur (1–5% of the amount of data [45]), even in the tune-up phases of the production lines [46]. The distribution of samples might be not only extremely skewed between classes, but also compounded by an overall small number of samples; such an *uneven dataset* hampers a stable network training unless either images related to a similar domain or pre-trained weights are available [47]. Moreover, a precise detection is hindered by a small training dataset since this latter might exclude some defect types that have not occurred yet on the inspected production process. As a first result, the lack of diversity in the training dataset, (e.g., not having complete features pertaining the defective patterns) leads supervised DL inspection systems to learn with biases towards the most represented class, hence with unreliable predictions and high miss-rate of rare defects. Second, overfitting occurs due to the high dimensionality of features, able to perfectly model the training data of the minority class by capturing irrelevant or noisy information, with a poor generalization performance on unseen samples. In fact, the number of features far exceeds the amount of extractable patterns in the few available defective samples.

The relevance of tackling data imbalance will be introduced in Section V, providing also some innovative approaches. To restore balance and cope with small datasets, both data augmentation from the existing samples and generation of new defective samples through a dedicated network (e.g., with generative networks like GAN [48]) are prolific, since they increase the size and quality of the dataset by counterbalancing the distribution and by enriching the features of the training images [49]. Besides, there exist learning methods (e.g., *one-class* approach) that leverage exclusively or prevalently non-defective (i.e., abundant) samples to construct a negative (i.e., without defects) template distribution [50], which is useful to recognize outlier samples through *contrastive learning* [45]. Other algorithm-based solutions group the approaches where the objective function is optimized to give importance to the defective samples by emphasizing the errors on the minority class [48]. In the last years, one research-related branch is focused on the *few-shot* detection challenge, in which only a bunch of defective samples can be collected [51], labeled and used [52] to develop efficient DL systems.

### C. ANNOTATION EFFORT REQUIREMENTS AND NOISY LABELS

Existing inspection systems are primarily based on the supervised learning method, which over-depends on the quality of labeled data. The annotation process requires skilled staff to register defective samples; in addition, further information may be added, starting from image-level and from region-level to pixel-based pairing with defect classes, resulting in a valuable information ready for supervision but at the cost of a quite expensive procedure. An additional cognitive effort during both annotation and network training is registered when a defective pattern has to be distinguished from the texture in the surrounding, besides when a defect occurs either on a foreground object surrounded by homogeneous background or on a textured surface or a foreground object within a defect-free and patterned background.

Image category labels, bounding box labels, fine-grain pixel-wise labels, and its more coarse version based on

scribbles, are the four classical types of labels available. Scribbles are image connected or not connected points belonging to the defective area. A thorough class mask is derived by growing the defective region having pixel similar to scribble points as a proof of knowledge during training [54]. In Fig. 5 three classes of images from the MVTec AD public dataset are used to illustrate both concepts (i.e., annotation and recognition efforts) in horizontal and vertical axes respectively [53].

Deep learning models have been showing to propagate or amplify the ambiguities introduced during the annotation phase [55]. The labeling annotation process has to face uncertainties attributed to multiple annotators besides those due to the vague boundaries of weak-contrast defects, which are hard to be accurately demarcated even for experienced workers. This could lead to incomplete demarcation of defects and inaccurate post-segmentation measurements. Moreover, some suspicious patterns can be marked as anomalous in a portion of the occurrences while as normal in the remaining ones, a problem commonly referred to as *label noise*. Such inconsistent annotations introduce a confounding pattern in the learning process due to biases in the ground truth [56]. Having a clean dataset is very complicated and recent works show that DL training is prone to overfit on corrupted labels since these latters excite more convolutional layers for the same class, thus resulting in a *memorization* effect [57], [58].

### D. UNCLEARNESS OF BLACK-BOX NETWORK DECISION

The optimization of a network involves, during training, the automatic process of feature extraction from images with the aim to minimize errors in network decision; however, this process is a sort of agnostic learning to the extent that it is unaware of the physical rules underlying faults in the defective class. Unfortunately, this process has intrinsic limitations in being transparent about how outcomes are achieved and impedes the application of powerful algorithms in fields where trusted systems are required.

A clear understanding of the rationale behind an algorithm predictions, as well as guarantees of robustness and performance, are essential steps for applications in such safety-critical areas. Heatmaps for CNNs activations are considered as a possible solution for the assessment of network reliability. Tao *et. al* through inner states visualization prove that a network can correctly predict the defective label for a sample although the focus area covers regions of the image that do not contain the defect, thus resulting in uninterpretable decision [27]. Geometrical mapping of feature in hyperspace and distance calculation, or Residual Explanation (SHAPE, LIME) together with visual explanation of feature relevance or embeddings (e.g., t-SNE) are tools to evaluate trustable behaviors [59].

### E. REAL-TIME DECISION-MAKING

Continuous and effective monitoring of the finishing products through vision inspection systems that work *on-premise* is considered an asset of the production process. A figure of merit of these *inline* defect recognition systems consists in limiting their influence on manufacturing pace. For instance, since these systems could embed high-demanding image processing on high-resolution data streams to identify defects accurately, they might require a longer response time than it would be allowed by the real deployment environment. This leads to a real-time compliance challenge along with the need to adapt to any disturbance that may occur during production. Therefore, achieving synchronization with the inspection systems requires high responsiveness and tight latency constraints, while assuring an elevated standard of accuracy. Moreover, the choice of the imaging modality can either speed up the acquisition phase, lightweight the processing, and be immune to industrial interference. An extrinsic factor exacerbating training imbalance is the utilization of a continuous stream of data coming from the production chain. Moreover, this latter is considered to disregard real-time constraints since the time elapsed for the acquisition and storage [60]. In fact, with the vast amount of data being generated in real time, besides the acquisition, even storing and managing such data can be a significant challenge; sophisticated data management systems are then required to ensure the accuracy, timeliness, and accessibility of these data [61]. Another concern is the availability of adequate number and class-balanced images within a short delayed time and their management to dynamically improve the network class representation during a new phase of training.

To tackle some of these issues, field engineers are increasingly adopting lightweight systems that provide high performance while maintaining a low inference time, by adopting, for instance, model compression and modular assembly of DL blocks [62]. In addition, cloud and edge collaborative systems are also being used to cope with the computational resources needed for more complex systems.

## IV. TAXONOMY

A believable pathway among the requirements and the constraints of defect recognition systems in industry involves understanding ''what'' and ''how'' to inspect defects and eventually measuring the corresponding extent. A defect can be measured through object localization by using a bounding box or by providing a precise pixel-wise mask. After analyzing the fully compliant papers and relevant reviews, while considering the previous key questions and previously discussed open issues and challenges, the authors propose a taxonomy based on two high-level criteria that allows an effective methodological framework to analyze paper contents: the former is the objective task, which includes segmentation, detection, and classification; the latter is the learning method or supervision level used to train the DL-based processing chain, which encloses fully-supervised, weakly-supervised, semi-supervised, and unsupervised approaches. As illustrated in Fig. 6, different pairs related to the available ground truth annotations are possible. Two objective tasks have been rarely performed in cascade.

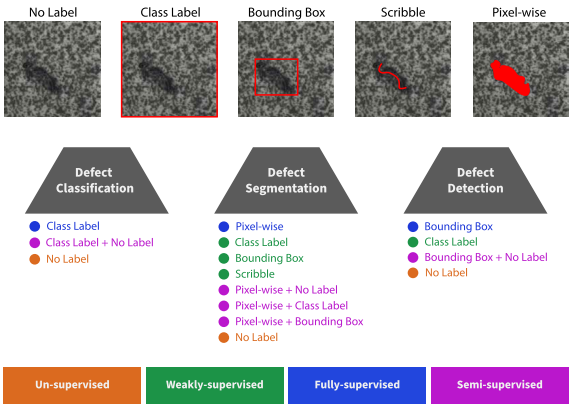



**FIGURE 6. The top of the figure illustrates the available ground truth annotations, while the center displays the usable ground truth annotations specific to each objective task, which are color-coded in accordance with the corresponding level of supervision. The color legend of the various learning methods is placed at the bottom of the figure.**

Ground truth annotations are usually apt to the objective task, owning the same level of detail with respect to the expected output from the network. For instance, the defect classification determines ''if'' a defect is present, and ''what'' kind of defect, both in a binary or multi-categorical manner; for a classification task, the same level of information is given by image-level tag encoding for ground truth. A lower level of detail matches, for example, when a defect localization task in the image is accomplished with image-level annotation: expected output from the network has a greater fine-grained result (localization of defect in the image) with respect to ground truth (overall image tag).

Remarkably, the final outputs can be achieved by training a network with a coarser information. The finer knowledge is extracted by leveraging on the network confidence level and inner attention on features useful for the final prediction. For instance, starting from a model performing classification, through saliency maps highlighting the salient (i.e., important) areas, eXplainable Artificial Intelligence (XAI) methods develop defective area segmentation having provided a single global label per image obtaining a local result, with a drastically reduced time and effort for data acquisition. Therefore, the coupling of the objective task (e.g., classification, localization or their combination) with image labels for network training has not a unique pairing, and results in ''how'' to supervise the DL system.

*Defects classification* consists in finding out ''if'' and/or ''what kind'' of defect is present, which means to perform either a binary or multi-class classification. It assigns a label (e.g., ''crack'' [63]) to the whole image or to patches in which the image is divided. *Defects localization* finds ''where'' is the defect and provides a classification score and/or defect-specific label to variable sized regions or to pixels of an image, reported as consistent with defects presence. Since Defects Localization is addressed by a huge amount of different works in the literature, the authors chose to group them more deeply, with a division in two branches called *Defect segmentation* and *Defect detection*, respectively. Then, according to the desired output, the authors have identified these three main objective tasks into which all recognized papers fall. In particular:

- **Defect classification** evaluates whether a defective pattern is present and associates to the image a prototype of concept (defect) retrieved.
- **Defect segmentation** represents the defect extension giving it precise boundaries.
- **Defect detection** delimits a rectangular region bounding the defect and classifies it.

Defect segmentation and defect detection assign to recognized defects a spatial reference and cover its extension in the image; hence, both these tasks allow to qualify the defects, and, for example, to extract morphological signature (width, length, aspect ratio, area, etc.) that can be related to the functional or production process dysregulation.

The objective task and the annotations available draw a correspondence that defines the learning method. The four main labeling techniques that a DL network uses to train defect representation for its recognition are depicted, along the horizontal axis, in Fig. 5. Alongside these four types, there exists one type of labeling that matches exactly the objective task, providing a reference to compare network output. At the same time, one or more types of annotations provide a lower level of detail with respect to the objective task requirements. This increases defect recognition effort and makes the network output not comparable with the available ground truth. The former is named ''full'' supervision, whilst the latters are ''softly'' supervised methods. These correspondences are referenced as:

- **fully-supervised** refers to a method in which the ground truth is at the same level of detail as the output (e.g., defect segmentation and pixel-wise labels attached to input images);
- **weakly-supervised** refers to a method in which the labels are simpler than in the previous case due to a lower level of detail with respect to the desired output (e.g., defect detection and class label for the whole image as input);
- **semi-supervised** provides bimodal supervision samples, which present different levels of detail (e.g., both unlabelled and labelled samples or a couple of different labels within four types mentioned in Par. III-C). The output has the same level of detail of the most detailed portion. This method enhances generalization capabilities and minimizes, at the same time, the annotation effort [64]. The unlabelled portion is useful if it carries information useful for label prediction that is not contained in the labelled data [64]. To this group belongs the so-called *active learning*, in which a partially trained supervised model makes inference on the unlabelled portion of data samples, which are further joined to labelled dataset portion with a confidence score; this process is known as *pseudo-labeling*.
- **un-supervised** methods provide no labels associated to samples and the defect recognition task generates a

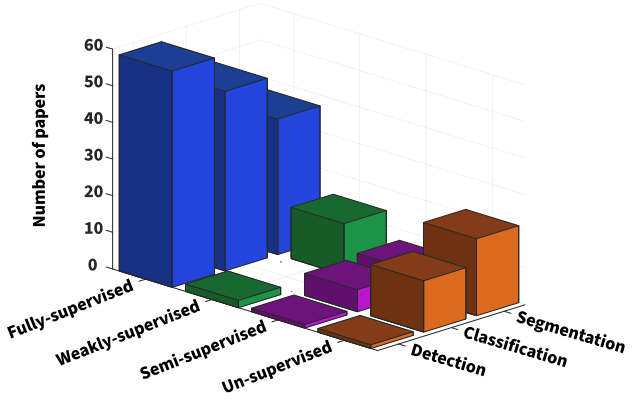



**FIGURE 7.** Distribution of collected papers per learning method and objective task.

detailed output by only considering intrinsic properties of data samples (e.g., defect classification by supplying input with no image-level labels). To this category belongs those training settings in which neither defective nor non-defective classes have labels, and the *One-class* learning approach. The latter provides for training only defect-free class samples; therefore, it deals with the accurate representation of the normal data distribution. Defects are recognized by contrast. In such a context, the *self-supervised* approach learns how to reconstruct the erased regions of the input defect-free class image through feature regression. When a defective image is to be tested, it undergoes *inpainting* and feature prediction by the network; its abnormal regions are repaired and thus a reconstruction error map overlaying the defects is got by the difference with the original image [3].

In the manufacturing production, an anomalous sample has a low probability of occurrence with respect to the defect-free class that covers the majority of samples, thus representing the *baseline behaviour*. Anomaly Detection consists in learning regularities inside data in order to recognize outliers as inconsistent with the baseline template. However, the performance of models trained only on defect-free samples is undermined by the intra-class variation in the negative class and is dependent on the distance metric used to evaluate similarity with test samples.

Summing up, all the collected papers fall into a specific learning method within the pursued objective task. An overview of the distribution of papers in these two dimensions is shown in Fig. 7. Weakly supervision does not have any correspondence for the classification task because image-level label (which means fully-supervision in the case of classification) has not a lesser informative annotation otherwise than un-supervised learning.

## V. SURFACE DEFECT RECOGNITION

This section analyzes the objective tasks oriented to defect recognition with a dedicated subsection devoted to each category. Further, within each task, the evidence from papers is presented in the light of the employed learning methods for network training. Each subsection reports a definition of the aim of the task and of the common functional parts (named ''building-blocks'') that can be identified in its reference architecture; moreover, in the ''conceptualization'' subsection the most impactful aspects for a well-posed task definition are put forward.

### A. DEFECT CLASSIFICATION

Defect classification aims at recognizing a defective pattern, by judging the ''defective'' versus ''non defective'' classes, and/or at identifying defect types occurring in the image. The correct classification of defects allows to analyze production process conditions, feeding back with information about the defect (e.g., name, type of defect) [65]. Tables 2-3-4 provide a summarised view of the surveyed articles dealing with the classification task, divided per learning method. To this purpose, for each work, the following information are reported: the application field or material, network topology, name of the dataset (when publicly available, otherwise ''custom'' stands for a in-house collected dataset), data description (regarding the imaging modality and the number of available samples used for the training phase), the performance in the test samples, the acceptation year, and the reference. As for the performance, in particular, the following metrics are used: the percent accuracy, the error rate, the True Positive Rate (TPR), and the Area Under the Curve (AUC).

#### 1) BUILDING BLOCKS

DL architectures for classification are composed of three main modules: 1) a *backbone* as feature extractor made up of convolutional layers with different widths, depths, scales and cardinality [66], [67] (e.g., Alexnet [68], GoogleNet [69], Visual Geometry Group Network (VGG) [70], ResNet [71], and DenseNet [72]; 2) an optional *neck* performs parameters selection and aggregation; finally the 3) *head* activates features towards decision.

Some ML algorithms (such as K-Nearest Neighbors, Decision Trees, Support Vector Machines, and other shallow classifiers) can constitute the classifier head as an alternative of fully-connected networks [9]. They take deep features extracted by CNN backbone as input and are trained to optimize a decision boundary to distinguish among classes.

#### 2) CONCEPTUALIZATION

The classification task with deep feature extraction enhances the ensuing decision phase: deep convolutional layers describe image content through detail and semantic information. Preserving details during downsampling operations is fundamental when the defect is a specific occurring entity that can be recognized as disruption of a subtle or visible edge pattern. High-level information, on the other hand, catches contextual information and diffformities in the result of the

semantic production process (e.g., screw fastening quality process [73]) known as *functional anomalies* [27].

Many industrial defects are likely to occur with different sizes within the same class and the encoding of these information in a pooled number of features is challenging. Apostolopoulos and Tzani [74] propose parallel feature extraction to overcome sequential downsampling loss. Then, they connect early with late feature maps, obtaining an extra feature processing path. Liu et al. [75] arrange two concurrent CNNs with different input image sizes but same functional blocks working in parallel to extract multi-scale features to enhance defects recognition, using a lightweight backbone. They compared performance of such architecture with that of other state-of-the classifiers and claimed their contribution to have a better performance even when only 20% of the training set is used.

Multilevel feature fusion is a potential aid to overcome salient information loss and improve multi-scale defects classification [76], [77], [78] but, on the other hand, it adds coarseness during higher feature maps oversampling (e.g., with bilinear interpolation) for the final summation or multiplication. Li et. al [66] proposed a boundary refinement block that restores boundaries after feature fusion by means of a residual structure [79]. Object boundaries, edges and other details are high frequency components of the image; usually defects, like ''scratch'', are recognizable among these categories. Yang et al. [80] adopted a Frequency-shifted convolutional layer to tackle high frequency information loss at the expense of semantic information prevalence in deeper layers. Other mathematical operations are being explored, such as Atrous (or Deformable) Convolution which helps combine sparse encoded information, by connecting feature related to non-adjacent image regions [81].

Adding convolutional layers may enrich feature representation ability, thus increasing network expressiveness, but it is feasible only when a balanced dataset allows to lengthen training without the overfitting side effect, which is rather an ideal condition. As a consequence, knowledge transfer, through pre-trained feature layers import, gives the advantage in the learning convergence, both in terms of prediction accuracy and learning speed. Kim *et. al* [82] express this advantage proving that transfer-learned network can be compressed up to 1/128 number of convolutional layers with only 0.48% drop in accuracy.

Learning an optimized feature space from high-dimensional data as images or video is challenging. The decision-making process for classification manages stacked information to design a decision boundary, distinguishing normal from abnormal samples or classifying the latter into defective classes. To make the final decision reliable, there are visualization tools that catches a single mid-layer class-activation, as well as a sequence of activation maps at different depths, which act as snapshots to monitor the network focusing on regions with patterns considered important. A more intelligible model could allow defective pattern interpretation in terms of its physical meaning.

#### 3) FULLY-SUPERVISED CLASSIFICATION

*Fully-supervised classification* learns with image-level annotations from input images in order to extract meaningful patterns that can be used to predict class label of unseen or test samples, supported by pairwise class category label provided during training. Currently available in literature are both binary (''defective'' vs ''non-defective'' or ''defect class A'' vs ''defect class B'') or multi-categorical (''defect class A'' vs ''defect class B'' vs... ''defect class N'' where N is the number of defect classes) classification. A simple variant in multi-class classification is to categorize N-1 of N as defect classes together with the ''Good'' or defect-free class [63], [83], [84]. Moreover, binary and multi-class classification can be combined in consecutive steps: for example Niu et al. [85] designed first a binary classification step in which non-defective products are distinguished from anomalies, and further classifies defective samples in separate types. In addition, the classification task can be deployed on sub-images or patches, when the overall classification depends on all the inferences made individually on each patch. Kamal et al. [86] used the Canny edge detector to detect and crop in eight boxes a gear image which is classified as defective or non-defective if at least one CNN box prediction is claimed as defective. Wang et al. [58] exploited the connection between the channels activation surging trend and the overfitting that occurs in presence of *noisy labels* to adjust weights of the Cross-Entropy and Mean-Absolute-Error loss function.

##### *a: OVERCOMING IMBALANCED DATA*

Having a balanced dataset between defective and non-defective class and/or among classes of defects is one of the major concerns to avoid bias during training. Mittel and Kerber [87] used oversampling of minority class to deal with imbalanced datasets. Defective images were upsampled by synthetic scaling, rotating or shifting, while the majority classes were downsampled to adjust to the number of the minority class samples. Different sized dataset are employed for training using different oversampling rates. Xu et al. [88] employ Label Dilation to extend the number of defective samples, conducting their sample number equal to the number of samples of the largest class before expansion. Sliding window approach consists in the extraction, especially from large images (such as X-Ray), of sub-images or patches [88], [89]. This gives three advantages, the first of which is to allows re-equilibrating class number representation: Mery [90] divided images whose ground truth was ''defect-free'' in numerous patches and added synthetic defects modeled with an ellipsoidal model, obtaining half of the dataset belonging to the positive class and the other half to the negative class. Secondly, patch-level classification delineates regions of higher likelihood for defects. Finally, CNN input images with lower dimension reduce the network computational load as well as the number of tunable parameters and memory occupation.

To counteract the problem of data imbalance, defect sample images augmentation as well as generation of new images with Generative Adversarial Network (GAN) are recently being performed. Data augmentation improves the generalization ability of the network when the number and diversity of available images is inadequate. A large and diversified dataset with resampled and combined representations helps the recognition network to learn new features, with an increased robustness on never-seen-before defects. Three major methods tackling hard sample mining at image-level are recurrent: 1) *image-level linear transformation*: it consists of an augmentation on the training set by scaling, rotating, translating, flipping, cropping or zooming, height and width shifting [51], [83], [87], [91], [92], [93]. Martinez et al. [73] augment their dataset of functional anomalies of screw fastening process through horizontal flipping and random rotation. Wang et al. [94] generate synthetic defect datasets using erosion, dilation, rotation or cropping on extracted defects from defective image and fuse them on defect-free images. 2) *introducing stochastic variations or modifying lighting conditions*: it consists of adding Gaussian noise, random brightness changing, enhancing contrast [74], [91], [95], [96], [97], [98], [99], using circular or elliptical templates [90]. 3) *generative models*: they include Conditional-Convolutional Variational AE [65], [100] and Deep Convolutional GAN [80].

The introduction of augmented dataset for training is effective if new features are introduced, which yield the network improvement in defect inspection. Dai et al. [101] employed geometric transformations and noise distortion + GAN to augment imbalanced datasets. Xu et al. [88] proposed a semi-supervised data augmentation using CNN. The same authors deployed a semi-supervised method to generate new defective images by improving random cropping on the original image: this latter operation was guided by the intensities feature map generated from pre-trained GoogleLeNet. Hence, the regions of interest are accurately selected, ensuring the presence of the defect even when it is proximal to the image border. A surface-Defect GAN is proposed to expand original highly uneven training datasets with the generation of defective images; Niu et al. [85] improved defect recognition by training a robust and supervised model with an expanded image dataset, thus increasing the diversity of data. Deep-Convolutional GAN is a variant introduced by Gao et al. [102] combined with traditional data enhancement (e.g., horizontal and vertical flipping, random rotation and scaling, image brightness enhancement) reinforcing original dataset to improve generalization ability. In Conditional-Convolutional Variational AE a latent vector is obtained by sampling the latent space and associating a one-hot encoded class information before decoding each defect class in the final image [65]. Prior knowledge of experts can guide image augmentation with realistic visual defects appearance; methods such as Copy-Pasting GAN [103] and addition of defects (e.g., random circle, blurring) are used; however, due to their complexity some defects cannot be efficiently managed.

#### b: TRANSFER LEARNING

Prior and effective knowledge can be infused via transfer learning. Sekhar et al. [84] verified that pre-trained models on ImageNet achieved better performance than using models trained from scratch, both in binary and multi-class classification settings. Pre-trained backbone was used as layer 0 while the following fully connected layers were trained. An improved fine tuning option is released by Hridoy et al. [104] that freeze all pre-trained weights except for the last 14 convolutional layers and the fully connected layers. Aslam et al. [15] compare three learning fine tuning strategies: on first k layers, or bottom k layers and standard fine tuning of the network. Althubiti et al. [105] provide accurate classification of products based on pre-trained CNN with VGG16 as backbone. Perri et al. [106] used SqueezeNet V1.1 for its low complexity for a four class classification using transfer learning and fine tuning customized on weld defects.

#### c: HYPERPARAMETER TUNING

Ma et al. [107] proposed Flower Pollination Algorithm to optimize learning rate by effectively searching the space where global optima exist. This improves training efficiency and training time. The learning rate hyper-parameter represents the speed with which gradient loss, calculated on the batch size (portion multiple of 2 evaluated in one iteration of network training), is used to update network weights. During performance evaluation, classification comparing different batch sizes and learning rate is made [16], [83].

#### d: ENSEMBLE LEARNING STRATEGY

Few studies were found using ensemble learning strategy: Zhang et al. [108] combine two independent CNNs inspired by notable results in the ILSVRC201 challenge, in which the first 12 winner positions used integration of models. The two learners get the same input image and contained some differences in the architecture; finally, outputs were averaged to get the final decision. Aslam et al. [15] dispose an ensemble of recent CNN architectures combining couples among DenseNet-201, $\text{ResNet}_{50}$, Xception and EfficientNet-b3. Su et al. [109] proposed an ensemble classifier made by AlexNet and GoogLeNet [69], where the classification output of each one was considered only if a reasonable higher confidence score was predicted for test image, otherwise the sample was evaluated from an experienced human. Xu et al. [88] train an ensemble model using four state-of-art models named SqueezeNet v1.1 [110], Inception v3 [111], VGG-16, and $\text{ResNet}_{18}$. Le et al. [112] augmented severely rare defective images with Wasserstein GAN (WGAN), trained ensemble and transfer-learned neural networks to make final averaged prediction of faults on decorative sheets and welding joints.

#### e: PERFORMING EVALUATION ON IMBALANCED DATASETS

The performance evaluation states the confidentiality with which a classifier recognizes defects: in fully-supervised

approach the predicted label on test image is compared with the actual label (ground truth). Among few standard metrics, accuracy quantifies the proportion of correctly classified images on the overall evaluated images. It is a fair evaluator only if a balanced dataset is given for each category: otherwise, accuracy-based metrics would be biased towards the majority class and, in this case, unconditionally high due to many network predictions on the overfitted class. For instance, Singh et al. used accuracy after having re-balanced the dataset. Buongiorno et al. [113] set a high misclassification penalty in the loss function to improve the learning ability for the minority class, thus achieving improved classification effect in deep regression and classification networks on heavily imbalanced datasets. They measure the F1-score on the positive class through five running inferences on stratified k-folds [114], [115]. Reconstruction-based methods require setting a threshold to binarize residual score (often linearized in the range of [0, 1]) and distinguish abnormal from normal samples; the Area Under the Curve of the Receiver Operating Characteristics is used to evaluate how much the classifier is robust to threshold setting and allows to choose the most suitable threshold in precision and recall trade-off.

Some studies compare the performance of their proposed solution when repeating training phase on different number of images. This allows to quantify the reduction in network detection ability when the number of samples is lowered [65], [75], [87], [116], [117]; having a little effect allows to reduce the time and cost of collecting and labeling data, which is a key factor in the industry resources management.

#### f: VISUALIZING NETWORK REASONING

The visualization of intermediate CNN activations helps understand learning representations. In correspondence of a convolutional layer depth and for each class being evaluated, a Class-Activation Map (CAM) is obtained weighting and summing feature maps of the level. As the depth of the considered layers increases, activations become increasingly abstract and less visually referable to defects [118]. Lee et al. [119] visualize activations of the last convolutional layer, comparing different significance levels assigned by the considered network to visual receptive field regions to distinguish each class; CAMs visualization prior to classification are visualized in Konovalenko et al. [120] and Yang et al. [118], where attention features highlight those regions that allow classifying an image as defective for the presence of one or more occurrences of defects. Xia et al. [121] localize network attention for decision through Guided GradCAM visual analysis. From the comparison of CAM with Guided-GradCAM, it emerges that CAM catches semantic information while the latter captures helpful details for defect localization, such as edges and texture identification. Li et al. [66] and Shih et al. [122] used Grad-CAM++ providing better visual explanations and defects location ability than using Grad-CAM and, therefore, improved faithfulness in CNN model prediction. The core perspective of Hu et al. [10] consists in developing an object-level attention to judge semantic information pertaining casting products (“defective” or “non defective”) without additional network structure to be trained; authors proposed a visualization technique named Bi-CAM which is designed for bilinear architectures. Heatmaps are obtained weighting feature maps with eigenvalues that preserve most of the information of the channel.

### 4) SEMI-SUPERVISED CLASSIFICATION

*Semi-supervised classification* utilizes a bunch of labelled (image-level) samples with the majority of data without labels. Some approaches start with a fully supervised training on the limited portion of labelled data, then this provisional knowledge is used for the estimation of categories confidence for the vast portion of unlabelled samples.

#### a: ACTIVE LEARNING

A common retrieved approach is *active learning*, which improves efficiency in the usage of available ground truth, since it assigns to unlabelled samples the same category of similar labelled sample that it is trained on. After the initialization of training in a fully-supervised manner on the available annotated data, a query strategy on unlabelled samples grades them with different levels of uncertainty and ask human to label only samples that will considerably speed up further training, while data-driven labelling is left to predict on the most confident samples. These pseudo-labeled samples are further joined to training set and will be used in the next full training phase, until a stop criterion is satisfied [123]. Liu et al. [124] design a multi-scale feature extraction CNN, apt to different sizes of defects, and implemented two independent classifiers to mutually correct pseudo labeling that may be wrong, thus improving accuracy and preventing to degenerate learning in an erroneous niche. Pseudo-labels are assigned combined with threshold selection on the confidence score.

#### b: DEALING WITH BOTH LABELLED AND UNLABELLED DATA

To keep the scope trackable even in the presence of few annotated images, the network introduced by Liu and Ye [125] is trained contextually both on labelled and unlabelled samples to avoid risk of overfitting. Pseudo-labels of unlabeled samples behave as optimization variables, and are accordingly introduced in the loss function and updated during training, while labelled samples remains fixed. This is an example of *Transductive Learning* where the focus is the continuous improvement of the network performance on already seen but unlabelled images; moreover, this semi-supervised model further minimizes human annotation effort. Di et al. [126] train with massive unlabelled data a Convolutional AE and use the encoder network as feature extractor to fed a softmax activation layer to predict among N+1 classes; N is the number of defect classes and the additional one is the binary classification of image as real or fake. Xu et al. [88]

compared feature extraction ability of VGG16 trained on a fully-supervised method with a multi-scale CNN trained only on 25% of labelled samples.

#### 5) UN-SUPERVISED CLASSIFICATION

*Un-supervised classification* works without labels and is robust against class imbalance. In addition to the *one-class* approach, one stream of studies provides training datasets that contain defective samples but are not marked; the implicit assumption is that the majority of samples belongs to defect-free class and their distribution can be accurately modeled, while the defective images are less in number and varied [143]. The objective is to effectively represent normal samples as clustered, in order to claim that abnormal patterns are inconsistent with the cluster properties. Two main approaches are found to detect defective samples, which are the *Latent Space discrimination* and the *Reconstruction-based* methods.

##### *a: LATENT SPACE DISCRIMINATION APPROACHES*

In the first approach, the so-called *Support Vector Data Description* is an encoding operation that performs feature extraction in the latent space, which is the corresponding low-dimensional representation of the image. Park and Ko [137] trained a Convolutional-AE with only 2% of anomaly images. Latent vectors extracted by the encoding path are mapped into an hypersphere. The hypersphere center reflects the average pattern; since during training the negative class is more frequently seen than the defective class, it is used as reference for the normal distribution, which on its part is employed for the inference stage. The likelihood of anomalies is evaluated according to the distance from to the center: samples outside the hypersphere decision surface are considered as defects. They adopt a pre-trained feature extractor and simultaneously endorse convolutional layers with residual addition being more robust to intra-class variation. Zhang et al. [143] during the testing phase measure the Euclidean distance between samples using their respective latent vectors and discriminate defective samples according to the centroid of obervations in the latent space. The minimization of the hypersphere volume leads to a lower false negative error rate. The *one-class* classification reduces the training dataset to containing only defect-free samples. An abnormal pattern in a test image can be highlighted by a distance measure (e.g., Wasserstein distance with GAN [100], [144]) between the latent vector and the normal prototype vector and this information is involved in the discriminator decision [138]. Lai et al. [142] train a GAN to obtain from noise a faithful latent space for detecting whether a test image is defective, by using Fréchet distance between two multivariate normal distributions in the latent space. On the basis of training on normal images, an abnormal sample is out of the reconstruction reaching with the same accuracy of defect-free images and is reflected in the anomaly score. Jiang *et. al* [131] normalized the anomaly score in the range [0, 1] and compared this with a threshold, thus obtaining a binary decision for the classification of the whole image or patch for industrial defective products. Song et al. [136] train a *one-class* GAN where the generator is an autoencoder, made up by a backbone ($ResNet_{50}$) and U-Net (Res-UNet-GAN); the aim of the generator is to maximize the reconstruction quality of the non-defective class during training. A test image undergoes encoding by means of learned normal distribution in the low-dimensional space by the discriminator: if the sample image is defective, a reconstruction error above certain threshold is an indicator of anomaly score. A pooled loss function results as the weighted sum of: 1) *Adversarial Loss*, also known as min-max optimization, minimizing generative loss and maximizing discriminative power; 2) *Structural Similarity*, which is defined on three factors that are brightness, contrast, and structure is kept high by the Adversarial Loss and 3) low *Feature Loss* defined as the norm of the vector obtained from the difference of latent vectors of input and reconstructed image.

##### *b: RECONSTRUCTION-BASED APPROACHES*

The *Reconstruction-based* approach considers a decoder path following the encoder network or a GAN, whose aim is to recover image space to tightly reproduce the input defect-free image; a considerable reconstruction error is gained, above a defined threshold, if the input image contains a defect because this cannot be reproduced due to the abnormal pattern. Generative methods based on Auto-Encoders consider only the final reconstructed image; this architectures are composed by encoder, decoder and sampling space. Input images are encoded as a parameterized distribution through the encoder path. Tang et al. [116] proposed a dual-auto-encoder-GAN in which both the generator and discriminator are AEs. The generator contains skip connections and, taken the input image, has a high reconstruction ability and generates the fake image. The discriminator can identify the difference between input image and fake image. The reconstruction error is lower for a normal sample, since during training only normal samples are provided. Li et al. [140] propose a denoising AE with constrained latent space to represent normal data. The method is built on a convolutional-GAN that helps establish a decision boundary robust to anomaly data infiltration since it has purpose of optimizing the normal image space by recovering from latent space. Yang et al. [80] introduced a reconstruction error based on residuals to quantify a pooled abnormal image pattern score, which, once compared with a threshold, generated a binary class label. Niu et al. [45] adopted a memory block to preserve historic information and to learn the group characteristics of the defect-free samples. A discriminator based on Fréchet Markov distance compares the input and reconstructed image by using a statistically-determined threshold for non-defective samples which are outliers to the gaussian normal samples distribution. In addition, the workflow can coarsely determine the location area of defect.

**TABLE 2.** Summary of the surveyed articles dealing with the classification task.

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| **Fully-supervised methods** | | | | | | |
| Tiles | $ResNet_{50}$ | Custom | 2D RGB - 30,000 | Acc.: 99.9% | 2023 | [127] |
| Steel | $ResNet_{50}$ | NEU-CLS | 2D Grayscale - N.A. | Acc.: 99.6% | 2022 | [128] |
| Steel | MobileNet-v2 | NEU-DET | 2D Grayscale - 270 | Acc.: 99.6% | 2022 | [117] |
| Welds | CNN | RIAWELC | X-ray - 15,863 | Acc.: 99.5% | 2022 | [106] |
| Welds | GAN + $ResNet_{50}$ | Custom | 2D RGB - 8,292 | Acc.: 97.8% | 2022 | [101] |
| Screws | VGG16 | Custom | 2D N.A. - 470 | Acc.: 100% | 2022 | [122] |
| PCB | VGG16 | Custom | 2D RGB - 2,159 | Acc.: 97% | 2022 | [105] |
| Gears | GAN + CNN | Custom | 2D Grayscale - 200 | Acc.: 98.4% | 2022 | [102] |
| Welds | CNN | Custom (video) | 2D RGB - 2,305 | Acc.: 98% | 2022 | [113] |
| Micro-motor armature surface | CNN | Custom | 2D RGB - 12,829 | Acc.: 98.4% | 2022 | [66] |
| Steel | EfficientNet | X-SSD | 2D RGB - Up to 20 | Acc.: 97% | 2022 | [51] |
| Welds | $DenseNet_{169}$ | SS304 TIG | 2D Grayscale - 24,204 | Acc.: 97.2% | 2022 | [84] |
| Ceramic Tiles | MobileNetV3 | Custom | 2D N.A. - 12,000 | Acc.: 98% | 2022 | [129] |
| Hex-nut | Xception | Hex-nut | 2D RGB - 4,000 | Acc.: 100% | 2022 | [104] |
| Tapered rollers | $ResNet_{101}$ | Custom | 2D Grayscale - 7,200 | Acc.: 99.7% | 2021 | [83] |
| Magnetic Tiles | VGG19 | Magnetic Tile | 2D Grayscale - 1,243 | Acc.: 92.7% | 2021 | [74] |
| Welds | CNN | GDXray | 2D Grayscale - N.A. | Acc.: 97.6% | 2021 | [63] |
| Steel | $ResNet_{152}$ | Severstal + NEU-DET | 2D Grayscale - 9,385 | Acc.: 97.1% | 2021 | [120] |
| Black line, crack, slag inclusion and gas hole | VGG16 | Custom | 2D Grayscale - 1,200 | Acc.: 74% | 2021 | [16] |
| Injection molding | VGG16 | Custom | 2D Grayscale - Up to 4,800 | Acc.: 99.5% | 2021 | [95] |
| Textures | Student-Teacher | DAGM 2007 | 2D Grayscale - Up to 5,520 | Acc.: 99.9% | 2020 | [82] |
| Leather | $EfficientNet_{B3}$ + $ResNext_{101}$ | WBLID | 2D RGB - 844 | Acc.: 82.6% | 2020 | [15] |
| Metal surface | AE | Custom | 2D Grayscale - Up to 657 | Acc.: 99.7% | 2020 | [65] |
| Steel | CNN | NEU-DET | 2D Grayscale - 360 | Acc.: 98.9% | 2020 | [75] |
| Welds | $ResNet_{18}$ | Custom | 2D Grayscale - 5,000 | Acc.: 98.4% | 2020 | [121] |
| Screws | $ResNet_{50}$ | Custom | 2D Grayscale - 378 | Acc.: 91.7% | 2020 | [73] |

**TABLE 3.** *(Continued.)* **Summary of the surveyed articles dealing with the classification task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| Commutator surfaces | GAN | Custom | 2D RGB - 950 | Err. rate: 0.74% | 2020 | [85] |
| Miscellaneous | AE + GAN | MVTec AD | 2D RGB - 3,629 | Acc.: 87.3% | 2020 | [116] |
| Metal cylindrical shell | Inception-v3 | Custom | 2D Grayscale - 1,455 | Acc.: 97.2% | 2020 | [89] |
| Aluminium castings | CNN | GDXray Castings | X-ray - 640,000 | Acc.: 96.9% | 2020 | [90] |
| Batteries | VGG16 | Custom | 2D Grayscale - 7,217 | Acc.: 99.9% | 2020 | [118] |
| Inclusion, crazing, patches, pitted surface, rolled-in scale and scratches on Steel | VGG-like | NEU-DET | 2D Grayscale - 1,260 | Acc.: 99.4% | 2019 | [119] |
| Welds | DenseNet | MINC | 2D Grayscale - 306 | Acc.: 97.2% | 2019 | [92] |
| Inclusion, crazing, patches, pitted surface, rolled-in scale and scratches on Steel | CNN | NEU-DET | 2D Grayscale - N.A. | Acc.: 98.1% | 2019 | [130] |
| Steel | CNN | Custom | 2D Grayscale - 1,600 | Acc.: 97.2% | 2019 | [99] |
| Curved surface | CNN | Custom | 2D RGB - 24,000 | Acc.: 97% | 2019 | [108] |
| Cigarette boxes | GAN | Custom | 2D RGB - 1,964 | Acc.: 100% | 2019 | [131] |
| Welds | GAN + CNN | Welding Joint | X-ray - 6,142 | Acc.: 98.6% | 2019 | [112] |
| Cutting wheels | CNN | Custom | 2D Grayscale - 400 | Acc.: 99% | 2019 | [98] |
| Cracks in metal forming processes | GoogleNet | Custom | N.A. - N.A. | Acc.: 90.4% | 2019 | [87] |
| Product surfaces | AlexNet | Custom | 2D N.A. - 4,238 | Acc.: 99.4% | 2019 | [109] |
| Blister in lithium battery | CNN | Custom | 2D N.A. - 18,860 | Acc.: 98.6% | 2019 | [107] |
| Casting | CNN | Custom | X-ray - 5,529 | Acc.: 92.8% | 2019 | [10] |
| Welds | CNN | Custom | 2D RGB - 28,809 | TPR: 98.8% | 2018 | [96] |
| Miscellaneous | CNN | Kylberg Textures 1.0 + DAGM 2007 + CIFAR100 | 2D Grayscale - 67,480 | AUC: 0.83 | 2018 | [97] |
| Product surfaces | CNN | Custom | 2D N.A. - 4,398 | Acc.: 98.2% | 2018 | [132] |
| Gears | CNN | Custom | 2D N.A. - 400 | Acc.: 96.5% | 2018 | [86] |
| Melt, plusmetal, scratch, scuff and shadowing on metals | VGG19 | AVI | 2D Grayscale - N.A. | Acc.: 75.3% | 2017 | [9] |
| Porcelain | CNN | Custom | 2D Grayscale - 346 | Acc.: 89% | 2017 | [115] |

**TABLE 4.** *(Continued.)* **Summary of the surveyed articles dealing with the classification task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| **Semi-supervised methods** | | | | | | |
| Silicon wafer | CNN | WM-811K + MixedWM38 + Custom | 2D RGB - N.A. w/o labels + 57,012 w/ image labels | Acc.: 96.3% | 2022 | [93] |
| blowhole, crack, fray, break and uneven on a surface | Student-Teacher ($ResNet_{18}$) | MT-CAS | 2D Grayscale - 439 w/o labels + 500 w/ image labels | Acc.: 98.9% | 2022 | [125] |
| Steel | VGG16 | NEU-DET | 2D Grayscale - 135 w/o labels + 45 w/ image labels | Acc.: 99% | 2021 | [124] |
| TFT-LCDs | CNN | Mura | 2D Grayscale - 5,160 w/o labels + 4,840 w/ image labels | Acc.: 94.2% | 2020 | [123] |
| Hot rolled strips | AE + GAN | Custom | 2D Grayscale - 9,000 w/o labels + N.A. w/ image labels | Acc.: 98.2% | 2019 | [126] |
| Rollers | CNN | Custom | 2D Grayscale - N.A. w/o labels + N.A. w/ image labels | Acc.: 99.6% | 2019 | [88] |
| **Un-supervised methods** | | | | | | |
| Electrical connectors | $ResNet_{152}$ | Custom | 2D RGB - 9,900 | Acc.: 97.4% | 2022 | [133] |
| Brake rotor | VGG16 | Ravirajsinh's | 2D Grayscale - 6,633 | Acc.: 100% | 2022 | [134] |
| Wood surface | AE | Custom | 2D Grayscale - 328 | Acc.: 95.2% | 2022 | [80] |
| Steel | EfficientNet | NEU-DET | 2D Grayscale - 1,440 | Acc.: 100% | 2022 | [51] |
| Miscellaneous | CNN | MVTec AD | 2D RGB - 540 | Acc.: 99.2% | 2022 | [135] |
| Steel | GAN | X-SDD | 2D Grayscale - 3,018 | Acc.: 99.2% | 2022 | [100] |
| Mura | AE + GAN | Mura | 2D Grayscale - 888 | AUC: 0.95 | 2021 | [136] |
| Micro-LED Chip | AE | Custom | 2D N.A. - 4,629 | Acc.: 95.8% | 2021 | [137] |
| Aluminium | AE | APSD | 2D RGB - 500 | AUC: 0.92 | 2020 | [138] |
| Bottles | GAN | MVTec AD | 2D RGB - N.A. | AUC: 0.89 | 2020 | [139] |
| Miscellaneous | GAN | MVTec AD | 2D RGB - 3,629 | AUC: 0.75 | 2020 | [140] |
| Screw head | CNN | Custom | 2D Grayscale - 3,000 | Acc.: 98.4% | 2018 | [141] |
| Solar panel | GAN | Custom | 2D Grayscale - N.A. | Acc.: 93.7% | 2018 | [142] |
| Electronic component | AlexNet | Custom | 2D Grayscale - 700 | AUC: 0.99 | 2017 | [143] |

#### c: DATA GENERATION

Expanding non defective image datasets for unsupervised learning has the aim to train robustly against intra-class variability. The inclusion of more samples is beneficial to enhance accuracy in the representation. For example, Ishida et al. [139] proposed to integrate data augmentation and mixing, a random based approach that combines weighting coefficient with three chains of linear image transformations of the original image finally fused with the input image. Hao et al. used a Wasserstein GAN as image data augmentation. Unlike supervised networks, in which the loss function accounts for difformities between input and ground truth, unsupervised loss considers the reconstruction error between reconstructed and original image or between their latent vectors.

### B. DEFECT SEGMENTATION

Defect segmentation gives a fine estimation of defect localization and extension providing a pixel-wise mask. Moreover, the semantic segmentation of defects classifies foreground pixels into different classes of defects without delimiting the different instances of objects, while the instance segmentation differentiates between all instances of each class, assigning a unique boundary to each one.

Defect segmentation is addressed converting the pixel-intensity correspondence in the input image into pixel-likelihood for defect in the output image. In addition, Neven et al. [145] proposed a network that outputs a probability map which estimates the severity of defects only valid for foreground pixels. Image analysis on the precise mask of defect as post-processing step leads to a detailed evaluation of the defect appearance (area [146], shape, texture and contextual feature) and could guide a multi-grades classification [25]. Tables 5-6-7-8 provide a summarised view of the articles dealing with the segmentation task, divided per learning method. The performance metrics used are: the Intersection over Union (IoU) and the mean-IoU (mIoU), obtained by the average per-class overlap between pixels, the Average Precision (AP), the F1 score, and the Dice coefficient.

#### 1) BUILDING BLOCKS

Most defect segmentation architectures typically have three key components: 1) an *encoder* with a convolutional backbone for feature extraction, 2) an optional *neck* for feature enhancement and selection (e.g., edge refinement), and 3) a *decoder* that uses up-convolutional layers to reconstruct features and produce a defect mask of the same size as the input image. The encoder is also responsible for feature extraction, a role it shares with the backbone in classification tasks.

#### 2) CONCEPTUALIZATION

Some defect recognition applications demand an accurate localization of the defect and its boundaries, separating it from the normal portion of the object or background [147], as illustrated in Fig. 5.

A challenging segmentation task consists in the pixel-wise localization of thin, small, and low-contrast defects because their features are faded by the overwhelming background pixels. In such cases, it is difficult to extract a defect from background at a glance. Ho et al. [148] proposed a segmentation task based on $\text{ResNet}_{50}$ performing feature extraction and concatenation to combine the multilevel features, followed by binary classification of image patches a little bigger than a pixel; in so doing, the system detects and locates defective pixels precisely, even if surrounded by a complex background. Chen et al. [149] proposed a multi-scale adaptive thresholding to support their GAN, highlighting potential defective pixels in the weighted difference image. More in detail, they adopt a smaller threshold to focus the inspection more on small defects in a large-scale sample and vice versa.

Boundaries restoration and refinement is usually added as final stage of the segmentation task, a process that can be found in all the learning methods. Dong et al. [147] introduced a boundaries refinement block inside their PGA-Net, thus visualizing the refined output as a residual map activated by a ReLu function. Based on the footsteps of the latter work, Yu et al. [52] proposed a novel implementation within a few-shot segmentation framework, that avoids information loss during forward propagation, and activates the query feature, where pixels share higher similarity with the support features. Lu et al. [44] used a residual structure-based boundary refinement module to help the network strengthen the details of the defect boundaries; they also performed an ablation study that revealed the usefulness of this module in return for a negligible longer time it takes for segmenting one image. Chen et al. [50] normalize the image reconstruction error map dividing by the variance estimated through a multi-layer perceptron before obtaining the segmentation map; this refines the results through a scale factor normalization.

#### 3) FULLY-SUPERVISED SEGMENTATION

In *fully-supervised segmentation*, training is accomplished by providing pixel-level ground truth annotations. Segmentation networks achieved great performance on identifying large and clear defects. Most often explored methods make use of classic convolutional architectures or GANs.

##### a: CLASSIC CONVOLUTIONAL ARCHITECTURES

Cheng et al. [150] proposed a modified version of U-Net, where several improvements were made, such as downsampling layers substitution with convolution ones and IoU loss adoption. Tabernik et al. [151] implemented a pixel-wise localization of defective surfaces through a CNN backbone optimized with pixel-wise loss, and used the extracted features for a binary classification of the image. Experiments are performed on KolecktorSDD dataset using only approximately 25–30 defective training samples. Lu et al. [152] segment the defective pixels in lace textures through rebuilding and classification from videos acquired on the industrial line. Djavadifar et al. [153] evaluated four different CNNs

(i.e., U-Net, IC-Net, DeepLab v3+ and Mask R-CNN) to perform instance segmentation, which is the distinct segmentation of all available objects of each class in the image, on a custom sheets-based dataset of 206 images. To accomplish the task, the CNNs weights were pre-trained on ImageNet and augmentation was adopted. Ouyang et al. [154] used a modified CNN that includes a dynamic activation layer, namely Pairwise Potential Activation Layer, that produces a defect probability map. Dong et al. [155] proposed a multi-stage architecture that involves U-Net for feature extraction, a Support Vector Machine to classify the type of defect present in the image and a Random Forest network for pixel-wise segmentation. Damacharla et al. [156] performed a comparison between two backbones incapsulated in U-Net, which are ResNet and DenseNet. They also compared the same backbones pre-trained on ImageNet and when using just the 50% of the available data. Results showed that pre-trained networks outperformed random initialized backbones in all the cases. Moreover, we could also identify a mixed CNN-AE-based multi-stage approach proposed by Tao et al. [157]. First, the input image is transformed into a prediction mask using their Cascaded AE, then a threshold module is used to binarize the result and obtain a detailed defect contour. After that, defect regions are extracted and classified into specific classes using a region detector and compact CNN in the classification module. It is worth mentioning the work by Luo et al. [158] since they propose a multi-learning and multi-task system that includes a memory attention feature enhancement module and a saliency detection module. This latter filters out background interference by using a human attention mechanism to measure the importance of image content, thus resulting in a heatmap that can be used to create a mask for identifying defects. The system can be trained not only with image-level annotations, thus providing a weak supervision level, but also with pixel-wise labels or a combination of both.

#### b: GAN APPROACHES

Regarding GAN approaches, Yang et al. [46] proposed a multi-stage framework to increase the generalization ability of the defect inspection model. Unlike all classic GAN approaches, this work tries to generate the defect regions and the background textures separately via mask-to-defect construction network (M2DCNet), and fake-to-real domain transformation GAN (F2RDT-GAN), respectively. In particular, M2DCNet is used as a first anomaly renderer, because the output is passed to F2RDT-GAN that transforms the primitive background in a detailed one. The generated image is then fine-tuned to an inspection model, in this case U-Net++. Niu et al. [159] proposed a GAN for defect generation: a defect mask input module and a defect direction vector module have been designed to increase the diversity of the generated defect sets by controlling the defect region and strength. They also included a defect attention loss to improve the image quality.

#### c: DEALING WITH TINY AND LOW-CONTRAST DEFECTS

Despite the great success obtained from the aforementioned approaches, the recognition of tiny and low-contrast defects is still challenging [160]. Different solutions have recently been explored to address this problem: multi-scale feature fusion, feature attention mechanisms, and a combination of both strategies. Niu et al. [161] overcome randomness in data generation by proposing a data augmentation addressed for downstream segmentation task. The CNN is forced to focus more attention to low-confidence areas for defects, which are usually inherent to tiny and stretched defect parts, once the higher-confidence regions have been occluded. This method reasons on the probability map of inner activations, with the aim to thoroughly segment defects and does not need to train any additional module.

The **fusion of multi-scale features** helps the network in the final decision by using both raw and semantic information to enhance localization accuracy [3]. Even in this task, fusing features to solve the tiny and low-contrast detection problem is one of the most commonly explored solutions. Cao et al. [162] adopted aggregation of adjacent feature layers at all depths of the encoder based on $\text{ResNet}_{50}$; this enforces all feature maps to contain both detailed and contextual information, in order to recover defect details and improve their segmentation. Yang et al. [163] proposed an efficient Fully-Convolutional AE-based (FCAE) framework called Multi-Scale-FCAE. In particular, they used different FCAE simultaneously working on the input image but at different scales. In fact, before starting, the FCAE step is preceded by a single encoding module followed by a Feature Clustering Module. Finally, the results are fused together obtaining the segmented defect. Lin et al. [160] presented EMRA-Net where three types of feature get extracted: local pyramid edge features (extracted with the help of a Laplace edge detection operator), global MSF (multi-scale fusion) features and the global convolution features. The redundancy of these features is minimized through the enhancement of different information. Dong et al. [76] developed a novel method based on multilevel deep features fusion and non-convex total variation regularized PCA (NTV-RPCA). The aim is to learn robust feature representation and to cope with the noise contamination. RPCA model is used to separate the deep feature matrix into the redundant matrix representing background and the sparse matrix representing defects.

The idea behind a **feature attention mechanism** is learning to focus on the image patterns which are relevant for the efficient recognition and, at the same time, to ignore the other irrelevant patterns. This concept changed the way a DL algorithm is seen, opening to the explanation of the inner black-box learning. Furthermore, the attention module can be used either to enhance the prediction capabilities in contexts where defects are not very clear and to attenuate irrelevant background information. The most famous DL models with an built-in self-attention mechanism are transformers. Uzen et al. [164] proposed a novel Swin transformer-based multi-scales integration network that obtained relevant results

although background similarity with low-contrast defects, and variability in defect size.

Within this learning method, the use of both **feature fusion and feature attention mechanism** is gaining relevance. Dong et al. [147] used a five resolution fusion strategy that manages to improve the efficiency of pixel-wise localization thanks to the upsampling and down-scaling of feature layers through a novel Global Context Attention Module. Tao et al. [43] proposed a Dual Attention Feature block to fuse and re-weight hierarchical features, recovering spatial information with rich context data. Yang et al. [165] proposed a bidirectional Convolutional Long-Short Memory attention module and multi-scale feature fusion through skip connections among encoder and decoder of UNet-based backbone for the improvement of microdefects segmentation. Hao et al. [166] implemented a novel version of ResNet, that already contains the Split-Attention block, including the Feature Pyramid Network that is a top-down feature fusion method. Liu et al. [167] developed a two-stage approach in which an attention-based fusion module fuses multiple scale features and attention information during the segmentation stage. They also included an adaptive scheme where learnable parameters are gradually optimized to strike dynamic balances between feature extraction and attention mechanism. Lu et al. [44] presented MRD-Net, which consists of a pre-trained MobilNetv2 backbone, a novel Reverse Attention module and a multi-scale feature enhancement fusion module. It has shown good performance on objects and textured surfaces as well. In such a context, another interesting work comes from Zhang et al. [168] since they proposed a learning-based soft template matching network that uses an innovative feature attention mechanism, by employing feature pyramid fusion. The aim of the network is first to find the image template and then output the differences between original image and reconstructed one. To detect defects in complex backgrounds, a multi-template ensemble testing module is used to further increase the accuracy. Li et al. [169] proposed a segmentation and decision multi-scale residual attention network in which the output of the U-shaped subnet and the final feature maps are used as the input of the decision subnet. This method allows precision and universality, especially in the detection of small defects by reusing shallow features. Meanwhile, Niu et al. [56] managed to train a segmentation network when a boundary suspicious region is present between defective and non-defective area thus in presence of *noisy labels*. They used a Bayesian Normalized U-Net to provide the area of a defect and to demarcate a margin region between an upper and lower boundary through a discrimination confidence weighted from multiple predictions.

#### *d: EXPLAINABILITY*

Within fully-supervised methods, authors could identify only two works that used XAI to give an idea of what the network is really doing. Ren et al. [42] operated heatmaps as part of segmentation process. In addition, they showed pixel-wise likelihood, proving that the network effectively focused on the defect regions. Otsu adaptive thresholding was applied as a binarization strategy. Wan et al. [170] performed the defect segmentation task with the support of anomaly scoring maps, which are obtained by computing the Mahalanobis distance between the features. Post-hoc they visualized the reasons behind the decision through activation maps.

### 4) WEAKLY-SUPERVISED SEGMENTATION

In *weakly-supervised segmentation*, the ground truth belongs to the image-level, bounding box or scribble category. The remainder of this paragraph groups articles according to each category of annotation provided. These methods bridge the gap between lighter supervision and pixel-level predictions, by seeking for local defective areas. Therefore, research is mainly focused on the learning process: for instance, several studies have attempted to improve existing loss functions or substitute them with some new proposition.

#### *a: IMAGE-LEVEL SUPERVISION*

Even in this learning method, CNN approaches reach significant performance. An interesting work has been done by Wu et al. [171] that improved the learning process by imitating the human eye defects recognition through the CNN and CAM. They developed an Autofocus sub-windowing, which examines progressively narrower regions in the image that differs from other normal regions in the feature distribution, meaning a higher defective potential. The loss is composed by a sum of multiple sub-loss terms: the global loss and the focused region loss. Xu et al. [172] proposed a novel cross-entropy based objective function that is per-pixel optimized when all pixels in the input image containing defects are correctly subdivided into defective and non-defective. Chen et al. [173] presented a multi-stage framework that classifies images and then segments defective areas. The CNN-based classification module was redesigned by substituting the fully connected layer with a more robust Random Forest classifier. For both the classification and segmentation modules, a spatial attention mechanism was used to reduce background interference and sharpen features tensor. The generated heatmap was thresholded by the Otsu method in the segmentation module.

According to recent studies, using GAN is becoming popular when discussing image-level annotations. Niu et al. [174] designed a defects cycle-consistency loss to properly restore defect-free patterns in the image, by adding structural similarity to the original L1 loss function, to account for structural and texture of weak defects. Subsequently, the precise defective region is segmented by thresholding defect saliency map. In a previous work, the same authors proposed the cycle-consistency loss, introduced by Zhu et al. [175], in the industrial surface defects field by using a GAN-based siamese network for training. With this kind of loss, a prototype of image (e.g., non-defective) can be obtained with translation from a different image content (e.g., with defects) giving some guidelines. Chen et al. [149] developed a multi-scale

GAN with transformer to reconstruct non-defective patches at different scales, comparing them with input patches to find pixel-level differences. In particular, the loss function of the generator involves three different loss terms: multi-scale feature loss, content loss and adversarial loss.

#### b: BOUNDING-BOX SUPERVISION

In recent years, defects bounding regions have been poorly adopted as labels for segmentation. Only Weimer et al. [41] in 2016 investigated the influence of the width and depth of the feature extractor, a typical CNN, stating that boosted performance could be obtained by deepening the architecture, but at the cost of longer inference time. Their approach is weakly-supervised because defects were coarsely-labeled with an ellipse.

#### c: SCRIBBLE SUPERVISION

When facing scribble annotations, training masks are derived, propagating category information from labelled pixels to unlabelled pixels during network training. Yao et al. [54] proposed a semantic segmentation approach that combines scribbles with super-pixels annotation to obtain training masks, named as pseudo-masks because of the labelling mistakes they can contain. Authors used a novel loss function, by aggregating several terms to counteract the simplicity of the annotation. In particular, the loss function includes different terms built on partial cross-entropy losses, one of which is the Centroid Loss.

#### d: EXPLAINABILITY

In the work proposed by Ye et al. [176] a deep comparison of the most popular XAI algorithms (Attention, CAM and Grad-CAM++) has been presented. More than explaining locally how a system ponders decisions, their use consists in enhancing localization performance, saving computations and time. Wu et al. [171] trained a CAM-based algorithm for segmentation task and improved the algorithm using a siamese network. They demonstrated that the original CAM algorithm could not produce a consistent class localization on any scale-transformed input images. Replacing Global Average pooling with LogSumExp pooling for CAM calculation, this new system surpasses other weakly-supervised state-of-art systems.

### 5) SEMI-SUPERVISED SEGMENTATION

*Semi-supervised segmentation* must be capable of using two different types of labels in a single framework. In the segmentation task overview, available semi-supervised studies are roughly 11% of the total.

#### a: PIXEL-WISE ANNOTATIONS AND NO LABELS

Typically, the training set contains a portion of images with pixel-wise annotations and another one with no labels. Shao et al. [177] used a student-teacher network that was trained with both fully- and un-supervised data. When a labeled image is sent to the student network, a segmentation result is produced by the help of a supervised loss function. Instead, when an unlabeled sample is provided, the consistency loss function ensures that the prediction result of the teacher network is similar to the prediction label of the student network. Zheng et al. [178] proposed a semi-supervised approach that requires a small quantity of labeled data based on MixMatch augmentation. It adheres to the consistency regularization principle, ensuring the class of unlabeled data remains unchanged after augmentation. They also introduced a novel residual neural network that uses a combination of supervised and un-supervised loss functions. While the supervised loss function uses common cross entropy, the un-supervised one uses a combination of the mean square error and Kullback-Leibler divergence. Lin et al. [179] presented a novel CNN based on CAM and U-Net. Their dataset was composed by 98.4% of defect-free images and the remaining 1.6% of defective pairs (sample + mask). The U-Net backbone made by VGG16 was pre-trained on ImageNet. The overall network structure consists of a single-path encoder and a multi-path decoder containing three sub-networks from which the outputs are aggregated to obtain the final segmentation mask. The CAM module uses the global average pooling to generate discriminative maps that were injected into the sub-networks, together with the other extracted features. The three decoders try to optimize different losses: an Intersection Over Union-based loss, the Binary Cross Entropy loss, and a Structural Similarity Index-based loss.

#### b: PIXEL-WISE ANNOTATIONS AND SOFT LABELS

The latest works started exploring different combinations of labels, i.e., pixel-wise labels and image-level labels [180], [181], as well as pixel-wise annotations and bounding boxes [145]. Bozic et al. [180] presented an end-to-end architecture composed of two sub-networks that employs a single parameter $\lambda$, to handle both weakly- and fully-supervised labeled samples, since the combined loss is the sum of two weighted cross-entropy functions. Hu et al. [181] designed a siamese network trained by fully- and weakly-supervised images simultaneously. The aim of this network is to produce pseudo-labels for weakly annotated samples by using an auxiliary cross-field and cross-attention network that maps features from the classification field to the segmentation field. Finally, a fully supervised segmentation model was trained.

### 6) UN-SUPERVISED SEGMENTATION

*Un-supervised segmentation* is, by definition, trained without the guidance of labels. In the classic approach both defective and non-defective images can be used while in the *one-class* approach only defect-free images are adopted. When discussing *un-supervised* methods in semantic segmentation, a relevant distinction between texture-oriented and object-oriented methods should be made, as they present varying levels of difficulty, as shown on the vertical axis in Fig. 5.

**TABLE 5.** Summary of the surveyed articles dealing with the segmentation task.

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - N. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| **Fully-supervised methods** | | | | | | |
| Steel | $ResNet_{50}$ | Severstal | 2D RGB - 8,798 | IoU: 90.3% | 2023 | [39] |
| Miscellaneous | Transformer | Fabric | 2D RGB - 1,200 | Acc.: 98.1% | 2023 | [182] |
| Cracks | CycleGAN-like | CFD | 2D Grayscale - N.A. | F1: 0.969 | 2023 | [183] |
| Miscellaneous | U-Net | KolektorSDD | 2D Grayscale - 150 | IoU: 62.0% | 2022 | [161] |
| Miscellaneous | $ResNet_{34}$ | DAGM 2007 + Magnetic tile + AITEX | 2D RGB - 5,245 | mAP: 86.3% | 2022 | [160] |
| Scratches, pit, bumps | $DenseNet_{121}$ + UNet + GAN | Custom | 2D Grayscale - 1,350 | IoU: 94% | 2022 | [46] |
| Laser beam welding | DeepLab V3+ | Custom | RGB - 169 | mIoU: 76.8% | 2022 | [184] |
| Pores in welds | U-Net | Custom | 2D Grayscale - 1,711 | N.A. | 2022 | [185] |
| Fabrics | U-Net | AITEX | 2D Grayscale - 1,360 | mIoU: 34.3% | 2022 | [150] |
| Miscellaneous | CNN | KolektorSDD2 | 2D Grayscale - Up to 246 | AP: 96.2%% | 2022 | [158] |
| Missing or broken component | CNN | Custom | CT scans - N.A. | N.A. | 2022 | [186] |
| Textures | CNN + ResNeXt | DAGM 2007 | 2D Grayscale - Up to 1,300 | AP: 100% | 2022 | [187] |
| Welds | U-Net | GDXRay | X-ray - 88 | Dice: 0.854 | 2022 | [165] |
| Cracks | U-Net | KSD | Grayscale - 300 | IoU: 71.4% | 2022 | [159] |
| Miscellaneous | CNN | MVTec | 2D RGB - 737 | IoU: 77.2% | 2022 | [44] |
| Textures | InceptionV3 | MVTeC-Textures | RGB - 891 | mIoU: 77.1% | 2022 | [164] |
| Miscellaneous | $Wide\text{-}ResNet_{50}$ | MVTec | 2D RGB - 3,629 | AUC: 0.967 | 2022 | [170] |
| Pit, edge cracks, scratches, and oxide scale on Steel | $DF\text{-}ResNeSt_{50}$ | Severstal | 2D Grayscale - 12,568 | mIoU: 65.9% | 2022 | [166] |
| Steel | GCN | Custom | 2D - 4,000 | mIoU: 85.8% | 2022 | [188] |
| Blowhole, and cracks on Magnetic material | U-Net-based | Magnetic Tile | 2D Grayscale - 407 | F1: 0.910 | 2022 | [189] |
| Plastic | $ResNet_{50}$-based | Custom | 2D - 5,747 | Recall Pixel Acc.: 73.5% | 2022 | [148] |
| Steel | U-Net-based | Severstal | 2D Grayscale - 8,798 | N.A. | 2021 | [156] |
| Scratches | $ResNet_{34}$ | CrackDataSet Magnetic tile WSCRATCH512 | 2D Grayscale - 5,248 | mIoU: 80.1% | 2021 | [43] |
| Wrinkles | CNN | Custom | 2D RGB - 7,442 | IoU: 51.6% | 2021 | [153] |

**TABLE 6.** *(Continued.)* **Summary of the surveyed articles dealing with the segmentation task. Bounding boxes are abbreviated with: bb.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| Miscellaneous | CNN | MVTec AD | 2D RGB - N.A. | Acc.: 91.6% | 2021 | [168] |
| Steel | CNN | NEU-DET | 2D Grayscale - 1,440 | Acc.: 96.9% | 2021 | [190] |
| Textures | U-Net-based | DAGM 2007 | 2D Grayscale - 10,142 | Acc.: 100% | 2020 | [167] |
| Magnetic material | $ResNet_{50}$-based | Magnetic Tiles | 2D Grayscale - 2,310 | IoU: 73.7% | 2020 | [162] |
| Frays, cracks, blowholes, and breaks on magnetic material | VGG16-based | Magnetic Tiles | 2D Grayscale - 2,840 | mIoU: 71.3% | 2020 | [147] |
| Welds | U-Net | Aerospace weld X-ray | X-rays - 411 | Acc.: 94.4% | 2020 | [155] |
| Textures | VGG16 | TILDA | 2D Grayscale - 284 | AUC: 0.936 | 2020 | [76] |
| Miscellaneous on plastic | U-Net | KolektorSDD | 2D Grayscale - N.A. | AP: 99.9% | 2019 | [151] |
| Textures | CNN | Tilda | 2D Grayscale - 1,160 | F1: 0.878 | 2019 | [154] |
| Miscellaneous | AE | Kylberg + KTH-TIPS2 | Grayscale - 400 | F1: 0.679 | 2018 | [163] |
| Welds | CNN | GDXray | Grayscale - 1,000 | N.A. | 2018 | [42] |
| Flat metal components | AE-based | Custom | 2D RGB - 3,000 | IoU: 89.6% | 2018 | [157] |
| Miscellaneous | CNN | Custom | 2D Grayscale - 1,425 | N.A. | 2017 | [191] |
| **Semi-supervised methods** | | | | | | |
| Textures | $ResNeXt_{101}$ | TILDA | 2D Grayscale - 320 w/o labels + 320 w/ pixel-wise labels | mIoU: 79.1% | 2022 | [177] |
| Miscellaneous on metal | VGG16-based | Custom | 2D Grayscale - Up to 5 pixel-wise labels | mIoU: 45.6% | 2022 | [52] |
| Pitting, inclusions, scratches, and patches on metals | U-Net-based | Severstal | 2D Grayscale - 7,544 w/ image labels + 2,512 w/ pixel-wise labels | mIoU: 57.4% | 2022 | [181] |
| Textured surfaces | CNN | DAGM 2007 | 2D Grayscale - 7,950 w/ image labels + 150 w/ pixel-wise labels | AP: 100% | 2021 | [180] |
| Sheet steel | U-Net-based | Custom | 2D Grayscale - ca. 9,000 w/ bb labels + N.A. w/ pixel-wise labels | mIoU: 43.7% | 2021 | [145] |
| Miscellaneous | VGG16 | MVTec AD | 2D RGB - 3,629 w/o label + 75 w/ pixel-wise labels | mIoU: 62.3% | 2021 | [179] |
| Textures | ResNet-based | DAGM 2007 | 2D Grayscale - 1,725 w/o labels + 1,725 w/ pixel-wise labels | Acc.: 98.8% | 2020 | [178] |

**TABLE 7.** *(Continued.)* **Summary of the surveyed articles dealing with the segmentation task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| **Weakly-supervised methods** | | | | | | |
| Miscellaneous | Student-Teacher | MVTec AD | 2D RGB - 3,629 | AUC: 0.909 | 2023 | [192] |
| Metal | GAN + Transformer | Custom | 2D RGB - 768 | Acc.: 99.3% | 2022 | [149] |
| Textures | Siamese Network | DAGM 2007 | 2D Grayscale - 4,025 | mIoU: 79.4% | 2022 | [171] |
| Textures | CNN | TILDA | 2D Grayscale - 900 | IoU: 83.1% | 2022 | [193] |
| Objects | U-Net | Custom | 2D RGB - 320 | mIoU: 76.8% | 2021 | [54] |
| Valves | CNN | Custom | 2D Grayscale - 2,699 | IoU: 73.2% | 2021 | [176] |
| Textures | GAN | DAGM 2007 | 2D Grayscale - 8,050 | IoU: 78.3% | 2021 | [174] |
| Imprinted aluminium | CNN | Custom | 2D Grayscale - 3,360 | Acc.: 92.7% | 2021 | [194] |
| Steel | GAN | NEU-DET | 2D Grayscale - N.A. | Acc.: 98.4% | 2021 | [195] |
| Miscellaneous on plastic | Faster R-CNN-like | KolektorSDD | 2D Grayscale - Up to 33 | AP: 99.5% | 2020 | [172] |
| Miscellaneous on plastic | CNN | KolektorSSD | 2D Grayscale - 250 | Acc.: 99.3% | 2020 | [196] |
| Textures | CNN | DAGM 2007 | 2D Grayscale - 3,450 | IoU: 70.6% | 2020 | [173] |
| Textures | GAN | DAGM 2007 | 2D Grayscale - 1,317 | IoU: 81.1% | 2019 | [197] |
| Textures | CNN | DAGM 2007 | 2D Grayscale - 909,440 | Acc.: 99.2% | 2016 | [41] |
| **Un-supervised methods** | | | | | | |
| Miscellaneous | $\text{ResNet}_{18}$ | MVTec AD | 2D RGB - N.A. | Pixel AUC: 0.988 | 2023 | [198] |
| Miscellaneous | Transformer | MVTec AD | 2D RGB - 1,266 | Pixel AUC: 0.983 | 2023 | [199] |
| Textile | GAN | AITEX | 2D Grayscale - N.A. | IoU: 77.8% | 2022 | [79] |
| Miscellaneous | VGG19 | MVTec AD | 2D RGB - 3,629 | AUC: 0.949 | 2022 | [3] |
| Miscellaneous | AE | MVTec AD | 2D RGB - 350 | AUC: 0.974 | 2022 | [200] |
| Miscellaneous | ViT | MVTec | 2D RGB - 3629 | Normalized AUC: 0.844 | 2022 | [201] |
| Miscellaneous | U-Net-based | MVTec AD | 2D RGB - N.A. | AUC: 0.964 | 2021 | [202] |
| Welds | U-Net-based | Aerospace weld X-ray | X-rays - N.A. | AUC: 0.986 | 2021 | [203] |
| Miscellaneous | $\text{ResNet}_{18}$ + Transformer | MVtec AD | 2D RGB - N.A. | AUC: 0.967 | 2021 | [50] |
| Miscellaneous | GAN | MVTec AD | 2D RGB - 3,629 | AUC: 0.969 | 2021 | [78] |

**TABLE 8.** *(Continued.)* **Summary of the surveyed articles dealing with the segmentation task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| Miscellaneous | CNN + SOM | MVTec AD | 2D RGB - 3,629 | AUC: 0.978 | 2021 | [204] |
| Miscellaneous | Student-Teacher ($\text{ResNet}_{18}$) | MVTec AD | 2D RGB - 3,629 | AUC: 0.970 | 2021 | [205] |
| Miscellaneous | $\text{ResNet}_{18}$ | MVTec AD | 2D RGB - 3,629 | AUC: 0.930 | 2021 | [206] |
| Textile patterns | AE | FID | 2D Grayscale - Up to 50 | Acc.: 98% | 2020 | [207] |
| Textures | AE | DAGM 2007 | 2D Grayscale - N.A. | F1: 0.616 | 2020 | [208] |
| Textures | CNN | TILDA + Custom | 2D Grayscale - N.A. | Precision: 97.2% | 2020 | [11] |
| Miscellaneous | AE | MVTec AD | 2D RGB - 3,629 | IoU: 59% | 2020 | [209] |
| Textures | GAN | TILDA | 2D Grayscale - N.A. | AUC: 0.884 | 2019 | [210] |
| Steel | CAE | NEU-DET | 2D Grayscale - 2,100 | N.A. | 2019 | [211] |
| Textures | AE | Fabrics | 2D Grayscale - N.A. | Acc.: 83.8% | 2018 | [212] |
| Capsules | DBN | Custom | 2D Grayscale - 100 | N.A. | 2015 | [213] |

Texture-oriented refers to methods trying to highlight defects within a pattern. Instead, object-oriented methods are used to segment the defective area from an entire object or a portion of it.

#### *a: TEXTURE-ORIENTED METHODS*

Texture-oriented methods can be divided into fake data generation-based and reconstruction-based methods. The former requires only normal images that are used as a baseline to obtain defective samples using a custom fake defect generator. Subsequently, a CNN is trained on pseudo-defective samples and used for testing real defective images [207]. The reconstruction-based methods can use either an AE [208], [212] or a GAN [79], [210], which is supposed to repair the input image, along with a further stage to segment the defect region by subtracting the reconstructed image from the input sample. An instance of this has been carried out by Yao et al. [208], who built a reconstruction framework based on the *one-class* strategy. It exploited the ability of an AE to build a realistic image without anomalies. A residual map is then employed to detect the defective parts, according to the pixel-wise probability for the normal texture and defective parts. Hu et al. [210] utilized a GAN in a *one-class* approach to perform a patch-based defects prediction with super-pixel segmentation by adopting a discriminator trained on defect-free samples and an inverter (encoder) to reconstruct patches in the normal image space. After that, patches are assembled and defects are segmented, due to merging of residual map with probability map. An innovative solution has been presented by Yao et al. [200] that used the one-vs-all strategy based on *Contrastive Learning*. A memory bank, containing typical normal texture mode, is used to substitute for anomalous features and obtain anomaly scores.

#### *b: OBJECT-ORIENTED METHODS*

Object-oriented methods have been extensively explored and several solutions have been proposed to determine the defective region of an object. In addition to various reconstruction-based solutions [3], [211], [213], which use the *one-class* principle [78], [214], different alternatives can be found. For instance, Yao et al. [205] developed a student-teacher network with $\text{ResNet}_{18}$ as backbone. The student network was trained from scratch while the teacher was pre-trained on ImageNet. Several anomaly maps were obtained at different feature map resolutions; hence, after up-sampling with a Gaussian receptive field, the lowest resolution maps were fused to obtain the overall anomaly map. An interesting study based on ResNet was proposed by Li et al. [204], who proposed a framework that manages to segment images using only defect-free samples for training, making use of the *one-class* approach. It is unique to the extent that, once a feature is extracted, it is organized in a bi-dimensional space through a self-organizing map used for anomaly score computation. In addition, Yoa et al. [206] developed a ResNet-based framework trained using only pairs of unlabeled images. They applied dynamic local augmentation to create pseudo-defective images from normal

ones to compute the loss function. Yang et al. [198] focus on the intra-class variance to artificially augment the dataset. They perform both textural and structural defect simulation onto the object surface. A memory module that retains normal images information from just few samples along with different spatial attention maps assists the network in the segmentation process.

Reconstruction methods rely on a class-specific threshold to evaluate anomaly score between the original and reconstructed image, which may not always be feasible. Therefore, Venkataramanan et al. [209] trained and tested Convolutional Adversarial Variational AE with Guided Attention (CAVGA) with two learning methods; in the unsupervised setting, defect-free images are used to stimulate the attention module, then concentrating on normal parts of the image. When few image-level labelled defective samples are provided, a semi-supervised setting encourages the network to expand normal attention and suppress abnormal attention on normal images. In the anomalous test images, anomalous attention and normal attention appear complementary. They scored CAVGA-generated attention maps both on object and textured images (e.g., on MVTec AD dataset [53]) with ground truth, and achieved superior results in terms of AUC and Intersection over Union, outperforming different state-of-the-art methods.

#### c: SELF-SUPERVISED NETWORKS

A segmentation network using the *self-supervised* approach is developed by Jing et al. [3], which used multi-scale deep pre-trained features to recover the multi-resolution randomly erased regions of the input image. The cosine value between the input and predicted feature maps results in the anomaly map for defects. The work of Jiang et al. [202] comprised an anomaly generation module to generate rich anomalies in the input defect-free image, which is divided into several patches that are randomly masked. An inpainting sub-network based on the Swin Transformer, which uses global context information, restores the hidden areas with anomaly-free patterns, while the discriminant U-Net sub-network is employed for anomaly segmentation by the difference between the input and the inpainted image. Nardin et al. [201] adopted the ViT-based architecture named Masked Transformer, which by being able to learn relationships between different patches of the input images, can predict the content of the masked patches from the surrounding data with the aid of an attention module. They also showed that performance were positively affected by splitting the image into patches of heterogeneous shapes capturing different scales.

#### d: EXPLOITING ACTIVATION ANOMALY MAPS

Explainability inside un-supervised methods is quite adopted, mainly based on anomaly maps [50], [199], [200], [204], [205], to perform the final segmentation through a threshold mechanism [3], [192], [208], [210], [212].

### C. DEFECT DETECTION

Defect detection delimits a rectangular region tight-fitting each defect, providing its position in the image. As a step further, classification of the boxes and/or definition of defect boundary through segmentation can be undertaken [215]. The goal is to label defects with bounding boxes; this includes a variable amount of background pixels, thus representing a crude localization of defects to coarsely measure the defects extension. Nonetheless, this task is reliable for counting the number of defects occurring in the image. The amount of defects can help to determine whether or not the product repairing would be sufficient to restore the original requirements and fulfill consumer's expectations. For example, Block et al. [216] proposed a defect detection and tracking system to reliably detect defects and use a majority voting mechanism to classify them as mild, if original quality could be achieved after reworking, or severe, if restoring quality is unreachable. Counting defects can be achieved also through instance semantic segmentation, which allows an optional morphological evaluation thanks to precise mask provided. Tables 9-10-11 provide a summarised view of the surveyed articles dealing with the detection task, divided per learning method. As for the performance, the mean-AP (mAP) and sensitivity (Recall) are considered in addition to the accuracy, AP, AUC, and F1 score, which were already used in the other tasks.

#### 1) BUILDING BLOCKS

In detector architecture three main blocks can be generally identified: 1) a *backbone*, which is a CNN for feature extraction, 2) an optional *neck* to sample feature map promoting some candidate regions for defects presence, and finally a decision 3) *head*, usually made up by two sub-networks of convolutional and dense connected layers aimed to two sub-tasks, which are bounding region regression and region classification.

Currently, two main paradigms exists: *two-stage* detectors (e.g., Regional-CNN (R-CNN) [217] introduced in 2014 and its derivatives) include a *neck* network to select regions with a confidence for defects. These sub-image portions are used as input for a further deep feature extraction and classification; additional layers can be inserted to detect the same defect occurring with different sizes, aspect ratios and shapes [18].

R-CNN generates region proposal by selective search and features are extracted from each region and input to Support Vector Classifiers. The Fast-RCNN introduced in 2015 extracts feature from a series of proposal regions and a pooling operation is applied to reduce these features to a fixed dimension vector for final classification and regression of box vertices. The Faster-RCNN in 2016 added a CNN, named Region Proposal Network (RPN), in the training path, which dynamically optimizes bounding box localization and dimension, addressing the mismatch between defect size and receptive field of detection head [218]. In fact, the presence of the RPN in a Faster R-CNN first generates $k$ anchor boxes

for each point on the feature map, then selects boxes with possible defects, and finally regresses vertices leading to proposals each with a confidence score [219].

*One-stage* detectors (e.g., Single Stage Multibox Detector (SSD) [220], You Only Look Once (YOLO) [221], RetinaNet [222]) are detectors in which box regression is applied directly over a dense sampling of possible locations, positions, scales, and sizes of candidate regions: this results in a lightweight architecture with less inference time than other approaches but at the expense of less accuracy. SSD performs worse because anchor sizes are free to cover a large amount of background, that leads to bias and confuse network.

#### 2) CONCEPTUALIZATION

A common problem to face in defect detection consists in the different sizes they can have in conjunction with different backgrounds, thus resulting in different levels of detection difficulty, as depicted along the vertical axis in Fig. 5.

CNN-based detectors are capable of effectively detecting defects on complex backgrounds and are being improved to suppress background interference for tiny, blurred, and low contrast defects recognition [223]. Moreover, such CNNs as ResNet family backbones can be equipped with feature pyramid module and deformable convolution. The former ensures an efficient representation ability of low and high-level information, thus merging higher resolution of shallower layers with stronger semantic information of deeper layers; the latter augments spatial sampling locations with better extraction of multi-shapes defects [224].

In some cases, fabric images have rich texture information and low semantic value, hence a feature aggregation module, to be accurate, should be biased towards low-level features; for this purpose, Zhou et al. [225] proposed a new module named L-shaped feature pyramid network to focus on low-level features while reducing the influence of high-level features, less important to defect detection. As a result, shorter backbones with narrower width and depth feature maps are used without a significant reduction in accuracy, but with improvement in saving sources and overall model efficiency. Liu et al. [226] proposed a R-CNN that embeds a feature enhancement and selection module to increase context complementarity and reduce confusion information at multiple scales. At the same time, to ensure optimized feature extraction for small defects in complex background, the dual attention maps (derived for channels and positions) are multiplied with the input feature maps. This shortens the information path and is more suitable for performing real-time detection [227].

On the other hand, in industrial pipelines, product images are acquired in a standardized setup and thus the variance of background appearances is very small. A network stuck on contextual information leads to over-fitting the seen background, hampering generalization ability on unseen backgrounds. As a result, defect-free texture coming from a different production process can be mis-classified as defective [228]. Secondly, defects are characterized by a distribution of sizes and positions within the image.

A basic approach consists in the subdivision of the image in adjacent or partially overlapped patches that are singularly classified. If a patch contains a defect, the fixed-sized window constitutes the defect bounding box [229]. However, it can be a partial or enlarged defect box that results in a low robustness [2]. Hence, it is necessary to accommodate sliding window size according to the characteristics of the target to be detected. Moreover, to guarantee a high recall, a great number of proposals is needed, many of which are false candidate that hamper the processing speed [230].

Handling cluttered background through a preliminary Region Of Interest (ROI) selection in which defects could be placed is a partial solution overcoming network overloading; it is what Yang et al. [231] propose with pre-processing images with Otsu thresholding for target workpiece evaluation. Secondly, the image is uniformly divided in patches that were separately classified by a CNN; finally adjacent patches belonging to the same class are merged to get a quite thorough defect detection.

Advanced approaches dynamically allocate proposals of defective region in light of extracted features. Anchors are reference locations in the image around which network evaluates a box regression and classification either to get proposals in two-stage detectors, or to predict final bounding box in one-stage detectors. In Faster-RCNNs these anchors are densely considered in the image around feature map nodes, and will be expanded in $k$ anchor boxes with different sizes and aspect ratios. Wang et al. [230] proposed Guided Anchoring, an optimized anchors selection rule that leverages semantic features, and follows two steps: first it identifies sub-regions where defects are likely to exist and then it determines the scales and aspect ratios related to different locations based on a single or multiple feature map at different levels.

#### 3) FULLY-SUPERVISED DETECTION

Considering defect detection as an object detection task, the fully-supervised approach makes use of ''box-level'' annotations bounding the defect. For training, a bounding box is provided with the coordinates of a surface defect and its corresponding class. During training, the network optimizes a learning rule to be able to predict a box bounding the defective region in test images. The objective function follows a multi-task loss composed by the classification loss and regression loss of bounding box vertices: the former is the softmax loss and quantifies how accurate the network is in recognizing an object. Furthermore, only for proposals being predicted as defect-containing (e.g., with a probability greater than 0.5), regression loss is calculated by comparing ground truth vertices with proposal vertices, through a smooth-L1 function [232].

##### *a: TWO-STAGE APPROACHES*

In a two-stage approach a list of proposals is first generated by the RPN and secondly these proposals are retained or

otherwise discarded if the classification layer does not recognize the presence of defect.

Defect detection performance mainly relies on feature extraction: a backbone with a sufficiently high recall for defect characteristics speeds up the localization phase. Complex and variable textures are common in industrial materials such as steel, fabric and leather; detection and exhaustive localization is undermined by complex background interference. Wang et al. [129] proposed a stacked network to compare a reference negative image with a test image to localize regions that are likely to contain defects; in addition, a discriminator (with ResNet as backbone) distinguishes between defects and textures within each proposed regions.

Cheng and Yu [233], to reduce information loss between backbone and neck, introduced a residual calculation (DE-block) between downsampled and upsampled feature maps. This residual map is used equivalently by the Channel Attention Module, for enhancing features. Finally, an adaptive spatial feature fusion between neck and head is used to merge scale-invariant features for steel surface defect detection. Luo et al. [234] proposed a decoupled two-stage object detection in which localization and classification sub-networks develop in parallel. A raw image is processed from a backbone and feature maps are considered ''stem features'' which undergo differentiation through aggregation and feature attention mechanism in semantic feature for classification and low-level features apt for detection and precise localization. They implement a Local-Non-Local attention module to adaptively enhance locally discriminative semantic information for the improvement of defects classification in integrated circuits (Flexible-PCB). Akhyar et al. [235] adopted a R-CNN based on $\text{ResNet}_{50}$ as baseline, improving it with a deformable RoI pooling and deformable convolutional filters that granted the model to be highly adaptive to the geometrical variation of the defect.

Data augmentation strategies recover from data imbalance and follow two groups: 1) a data-level approach introduces variability, such as brightness change, color equalization, addition of noise and geometric translation and flipping, are used directly in images [133], [236] and 2) an algorithm level (modifying loss function to avoid overfitting), and the usage of GAN as resampling method.

The transfer learning is a wide adopted strategy in defect detection task and counteracts data scarcity in the industrial field, reducing the amount of training data [237]. Guo et al. [238] peculiarly apply Conditional-GAN with contrast enhancement in discriminator in a supervised setting (in which both generator and discriminator have class label, see Section A-D) to generate new images of different classes of defects, facing two problems: feature scarcity and data imbalance. They adopted Xception with pre-trained layers as feature extractor with the first network to classify the image to locate defects and the second one to identify the specific category. Wang et al. [239] finalized a defect detector for steel surface using few shot images as target data for fine tuning in order to generate defect-specific features. Cao et al. [240] evaluated three backbone depths for fine tuning, from shallower to deeper, and tested the 1-shot and 5-shot settings; they selected $K$ samples of target data for each defect category as the training data and used the rest as the testing data, and repeated the experiments on disjoint dataset parts. The greatest accuracy and F1-score were achieved by fine tuning last stage in the 5-shot setting. Deng et al. [241] improve the CNN architecture based on pre-trained VGG16 backbone and compared the model accuracy in the case of training from scratch, of finely tuning the entire module, without freezing any layer, and of freezing the last $x$ layers. Results demonstrated the highest overfitting ratio was for the model without transfer learned feature, since it lacks enough data for feature extraction. The higher the number of frozen layers, the worse the test accuracy.

Inspired by Guided Anchor, Chen et al. [242] leveraged semantic features to yield more suitable anchor boxes for different surface defects. They proposed an Adaptive Anchor Module (AAM) that first insights on locations where surface defects are likely to exist, and then predicts the shapes at different location [230]. Following this formulation, the RPN enhanced by the AAM allowed a higher recall using a compressed backbone.

Yu et al. [243] geared towards an anchor-free fully-convolutional detector in which it is not needed to define any anchors *a priori*. Defects are located by selecting a center point and back-mapping it onto the input image to regress bounding boxes directly. Cheng and Yu [233] developed an evolutionary algorithm to iteratively optimize ratios and scales of anchor aimed at maximizing the overlap of ground truth boxes and proposed anchors. The best solution is searched starting with five ratios and three scales according to defects distribution in the image.

The most suitable depth of feature maps for anchors refinement is automatically selected during the training phase: Lv et al. [244] evaluated a set of boxes with different aspect ratios at each depth; these boxes are matched with the ground truth and the network predicts both the offsets and the confidences for each category.

Wei et al. [245] refine detection bounding boxes substituting the quantization operation with a weighted bilinear interpolation of feature. Floating vertices of boxes are mapped to corresponding floating points in feature maps, each of which is calculated as a weighted average of feature values of the closest points, named Weighted ROI pooling.

#### b: ONE-STAGE APPROACHES

In the one-stage approach, the generation of proposals is skipped but anchors are still used, like in DenseBox [246]. RetinaNet [222] improves foreground to background imbalance, adopting focal loss and feature pyramid network. YOLOv3 [247] adopts a multi-scale prediction to improve sensitivity for small defects, and deepens the network to improve accuracy being, at the same time, quicker than Faster-RCNN. Li et al. [248] optimize YOLOv4 with

channel-wise attention and feature pyramid to increase efficiency in small defects detection, and exponential moving average to stabilize network training, thus obtaining 42 frames processed for second. Chen et al. [249] proposed a network to perform the detection of low-contrast defects on blurred surfaces by using their constructed ECANet-Mobilenet SSD model. A module that combine channel and spatial attention was included to better extract discriminative features. Singh et al. [250] devised a YOLOv5-based system that is capable of processing High Dynamic Range (HDR) images, which are single images with a wider range of brightness and detail in both the shadows and highlights. More specifically, the system employs a technique that involves capturing nine images with varying exposure levels and subsequently merging them for each object. Zhang et al. [61] developed a lightweight detection model for PCB that reached almost the same mean Average Precision of different YOLO architectures but with a fraction of FLOPs, allowing it to run on a video-stream. Instead, Liang et al. [62] used a YOLOX-tiny model as baseline and included different modules to capture shallow features importance and propagate the most useful ones to the network's head. Their method requires only 3.44 gigaFLOPs while resulting the most accurate in terms of Average Precision among different detection frameworks. Lim et al. [251] improved a YOLOv5 model by adding a new feature pyramid network to better detect small-scale object by applying the feature fusion approach. The model performed slightly better than a standard YOLOv5 model while keeping an acceptable number of gigaFLOPs.

Wu et al. [252] improved the YOLOv3 architecture with K-means clustering of anchor boxes; this is used to obtain a skimming of more likely defect sizes, in correspondence of a multi-scale fusion prediction feature map, instead of considering three different feature maps at three different scales. Given a number $m$ of clusters, similar boxes are aggregated using average Intersection Over Union metric, and finally $m$ cluster centroids will be used for further evaluation. Zhang et al. [2] improved backbone MobileNetV2 [253] with K-means clustering to optimize parameter selection of candidate boxes for each of the dataset tested in the framework. They obtained real-time inference with competitive accuracy in order to deploy this architecture on edge devices. Only one work has employed two-stage detectors using K-means clustering: Zhang et al. [91] optimized the number of $m$ clusters; if they were enough, bounding boxes were more precise and accuracy was preserved, whilst if $m$ was too high, the network did not get a consistent advantage in terms of speed.

Song et al. [254] adopted a lightweight detector of the EfficientNet [255] family, which is made up by one-stage and scalable models with eight possible depths of layers performing deep separable convolution to ease computational burden. Their model EfficientDet (D0 up to D7) meets several edge device resource constraints. This example of edge computing helps overcoming problems like transmission latency between end-devices and the cloud and bandwidth demands, limiting transfer of large amount of data towards the cloud platform. In addition, Naddaf-Sh et al. [256] use a single hyperparameter that determines both width and depth of backbones (b0, b1, b6, b8) of EfficientNet family, thus holding the trade-off between accuracy and inference time.

Unlike the one-stage detection methods, Wang et al. [257] developed an anchor free end-to-end architecture based on $\text{ResNet}_{18}$ that uses only the center point of the target to generate a bounding box, allowing for faster detection speed. During the training, the network learns to output three different information regarding the defect: the center, the size, and the class.

#### c: EXPLAINABILITY

A comparison of backbone efficacy is visualized through Grad-CAM which localizes areas determining final network decisions. Providing a sequence of activated features at different depths, the gradual network convergence is explained. Nguyen et al. [258] conduce a comparison study on the feasibility to train and deploy YOLO one-stage pre-trained detectors (YOLOv5, YOLOX, YOLOv7) on GPU-enhanced embedded devices.

#### d: CLOUD-EDGE COLLABORATIVE APPROACHES

Cloud resources are employed for network parameters selection, base-training on images transferred from the edge and for performance validation. Once network performs well, it is downloaded on an edge device and deployed directly on the production chain (*on-premise*). Further refinements can be successively uploaded: for example, when new labelled images are added to the training set.

An example of two-stage detector by Faster-RCNN applied in an edge-cloud flexible setup is proposed by Wang et al. [259]. Their evolving algorithm covers several production plants and inspection lines for different products; in these lines distributed hardware sensors (cameras) are connected through edge nodes to software elaboration unit for a constant and agile upload of data sources and services. Faster-RCNN with $\text{ResNet}_{50}$ backbone, pre-trained on ImageNet, requires at most 0.1 seconds per image to detect defects [260].

Currently, smart cameras allowing to acquire, store and transmit images are being commercialized, and their resources can even allocate network inference calculation on test images. For example, Zhu et al. [261] modified the DenseNet architecture and deployed lightweight trained model on an intelligent smart camera, placed on a scalable production chain for real-time defect detection. There is no need to download the trained model on the edge device and the overall cycle includes image acquisition, image processing, and edge response.

#### e: EXPLOITING TRANSFORMERS

Considering the shortcomings related to the use of transformers in the industrial defect recognition field, the authors can find different transformer-based approaches applied

**TABLE 9.** Summary of the surveyed articles dealing with the detection task.

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| **Fully-supervised methods** | | | | | | |
| Steel | R-CNN | NEU-DET | 2D Grayscale - 1,260 | mAP: 83.4% | 2023 | [235] |
| Steel | R-CNN | NEU-DET | 2D Grayscale - 1,200 | mAP: 79.4% | 2023 | [226] |
| Metal stamping parts | YOLOv5 | Custom | 2D HDR - N.A. | mAP: 67.4% | 2023 | [250] |
| Ceramic curved surfaces | SSD | Custom | 2D RGB - 5,000 | Acc.: 96.7% | 2023 | [249] |
| PCB | CNN | PCB | 2D RGB - N.A. | mAP: 91.95% | 2023 | [61] |
| PCB | YOLOv5 | TDD-Net | 2D RGB - N.A. | mAP: 81.2% | 2023 | [251] |
| PCB | Student-Teacher-like | HRIPCB | 2D RGB - 1,500 | Acc.: 79.6% | 2023 | [262] |
| Cylindrical Shells | MobileNet-v1 | Custom | 2D RGB - N.A. | N.A. | 2023 | [263] |
| Textile | Teacher-Student | Tianchi | 2D RGB - 3,480 | mAP: 76.5% | 2022 | [264] |
| Steel strips | YOLOv4-based | NPSS | 2D RGB - 1,050 | mAP: 87.3% | 2022 | [62] |
| Sewing, sewing print, scrimp, bug, flaw, color shade, miss print, hole, and fold | YOLOv4-based | Tianchi Fabric | 2D RGB - 3,592 | mAP: 76.2% | 2022 | [77] |
| Steel | SqueezeNet | NEU-DET | 2D Grayscale - 30 | Acc: 97.7% | 2022 | [240] |
| Steel | GAN + YOLOv5 | GC10-DET | 2D Grayscale - N.A. | mAP: 85.3% | 2022 | [265] |
| Metallic surface | YOLOv4 | Custom | 2D RGB - 7,725 | mAP: 94.5% | 2022 | [248] |
| PCB | YOLOv5 | Custom | 2D RGB - 2,000 | mAP: 97.5% | 2022 | [266] |
| Steel | Faster R-CNN | NEU-DET | 2D Grayscale - 1,260 | AP: 76.4% | 2022 | [242] |
| Steel | YOLOv5 | NEU-DET | 2D Grayscale - N.A. | mAP: 75.2% | 2022 | [267] |
| Candies | YOLOv3 | Custom | 2D Grayscale - 20,000 | F1: 90% | 2022 | [268] |
| Steel | YOLOv5 | NEU-DET | 2D Grayscale - N.A. | mAP: 75.6% | 2022 | [269] |
| Textures | CNN + SSD | Custom | 2D RGB - 105 | mAP: 90.1% | 2022 | [270] |
| Textile | CNN | Tianchi | 2D RGB - N.A. | mAP: 47.1% | 2022 | [271] |
| Steel | U-Net | DAGM 2007 | 2D Grayscale - 1,046 | AP: 99.9% | 2022 | [102] |
| Steel plates | SSD | Custom | 2D RGB - 3,179 | mAP: 76.9% | 2022 | [272] |
| Steel | SSD | NEU-DET | 2D Grayscale - N.A. | AP: 89.8% | 2022 | [215] |
| Packaged chips | YOLOv4 | Custom | 2D RGB - 2,320 | mAP: 78.5% | 2022 | [94] |
| Steel | CNN | NEU-DET | 2D Grayscale - 12,568 | mAP: 97% | 2022 | [273] |

**TABLE 10.** *(Continued.)* **Summary of the surveyed articles dealing with the detection task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| Steel | $\text{ResNet}_{18}$ | NEU-DET | 2D Grayscale - 1,260 | mAP: 80.0% | 2021 | [257] |
| Aluminium castings | RetinaNet | GDXray Castings | X-ray - 2,045 | mAP: 94.2% | 2021 | [274] |
| Welds | EfficientNet | GDXray + SBD | X-ray - 17,872 | mAP: 72.4% | 2021 | [256] |
| Pitting, scratches, crazing, patches on Steel | Faster R-CNN | Severstal | 2D Grayscale - 40,200 | mAP: 87.6% | 2021 | [81] |
| Steel | Faster R-CNN | FS-ND | 2D Grayscale - 30 | mAP: 64.3% | 2021 | [239] |
| Textile textures | YOLOv2 | Custom | 2D RGB - 2,612 | mAP: 95.6% | 2021 | [254] |
| Steel | Faster R-CNN | NEU-DET | 2D Grayscale - N.A. | mAP: 82.8% | 2021 | [275] |
| Steel | CNN | NEU-DET | 2D Grayscale - 1,440 | mAP: 76.7% | 2021 | [243] |
| Welds | GAN | Custom | X-ray - 10,000 | Acc.: 92.5% | 2021 | [238] |
| FPCB | $\text{ResNet}_{101}$ | FPCB-DET | 2D RGB - 730 | mAP: 94.6% | 2021 | [234] |
| Connectors | Faster R-CNN | Custom | 2D RGB - 1,654 | Acc.: 94% | 2021 | [91] |
| Steel | $\text{ResNet}_{50}$ | NEU-DET | 2D Grayscale - 1,440 | mAP: 80.5% | 2020 | [224] |
| Metal parts | RetinaNet | Custom (videos) | 2D Grayscale - 14,432 | mAP: 76.4% | 2020 | [216] |
| Stamped metal | RetinaNet | Custom | 2D Grayscale - 5,594 | mAP: 76% | 2020 | [216] |
| Steel | RetinaNet | NEU-DET | 2D Grayscale - 1,440 | mAP: 79.1% | 2020 | [233] |
| Steel | YOLOv3 | NEU-DET | 2D Grayscale -N.A. | mAP: 82.7% | 2020 | [2] |
| USB-connector | SqueezeNet | USB-SD | 2D Grayscale - 6,000 | Acc.: 95.3% | 2020 | [231] |
| Steel | SSD | NEU-DET | 2D Grayscale - 1,440 | mAP: 72.4% | 2020 | [244] |
| Electric connectors | YOLOv3 | Custom | 2D RGB - 45,000 | mAP: 93.5% | 2020 | [252] |
| Steel | Faster R-CNN | NEU-DET | 2D RGB - N.A. | mAP: 76.5% | 2020 | [219] |
| Fabric defects | CNN | Custom | 2D Grayscale - 2,488 | AUC: 0.83 | 2020 | [261] |
| Textile | EfficientNet | Tianchi | 2D RGB - 4,730 | mAP: 20.9% | 2020 | [225] |
| Textile | $\text{ResNet}_{101}$ | Tianchi | 2D RGB - 3,022 | mAP: 51.4% | 2020 | [228] |
| Textures | GAN | DAGM 2007 | 2D Grayscale - N.A. | mIoU: 85.9% | 2020 | [214] |
| Oil Leak | Mask R-CNN | Custom | 2D Grayscale - 1,000 | AP: 91.1% | 2020 | [236] |
| Turbo blades | Faster R-CNN | Custom | 2D Grayscale - 64 | mAP: 68% | 2020 | [259] |
| Scratch, indentation, crust, and fold on Steel | Faster R-CNN | Custom | 2D Grayscale - N.A. | Recall: 97% | 2019 | [245] |
| Gears | YOLOv3 | Custom | 2D RGB - 3,600 | Acc.: 100% | 2019 | [223] |
| Steel | CNN + RPN | NEU-DET | 2D Grayscale - 1,260 | mAP: 82.3% | 2019 | [276] |

**TABLE 11.** *(Continued.)* **Summary of the surveyed articles dealing with the detection task.**

| Application field/ Materials | Network Topology | Dataset Name | Data Description [Image type - No. of images] | Performance | Year | Ref. |
|---|---|---|---|---|---|---|
| Steel | CNN + SSD | Custom | 2D - 7,400 | mAP: 75% | 2019 | [277] |
| Batteries | VGG19 | Custom | 2D RGB - 2,284 | AUC: 1 | 2019 | [278] |
| Seals in multilayer aseptic packages | Faster R-CNN | Custom | 2D RGB - 300 | Acc.: 99.2% | 2019 | [279] |
| Blisters, skid marks and scratches on aluminium | Faster R-CNN | Custom | 2D RGB - 813 | mAP: 47% | 2019 | [260] |
| **Semi-supervised method** | | | | | | |
| Patches, pitted surface, inclusion and scratches on Steel | YOLOv2 | NEU-DET | 2D Grayscale - 940 w/o labels + 260 w/ labels | mAP: 64.5% | 2020 | [280] |
| **Weakly-supervised method** | | | | | | |
| Miscellaneous | $\text{ResNet}_{101}$ | KolektorSDD 2 | 2D Grayscale - 2,331 | AP: 45% | 2023 | [281] |
| Mobile Phone Cover Glasses | Student-Teacher | MPCG | 2D Grayscale - 8,856 | mAP: 61.2% | 2020 | [282] |
| **Un-supervised method** | | | | | | |
| Magnetic Tiles | AE | Magnetic Tiles | 2D Grayscale - 952 | AUC: 0.85 | 2021 | [7] |

to the defect detection task. Both Gao et al. [283] and Zhang et al. [269] proposed a swin-transformer model. The former designed a new window-shift scheme that further strengthened the feature transfer between the windows. Therefore, a Variant-Swin transformer was used as the backbone and the extracted features are provided into a fusion module that feeds a detection framework; this latter consists of an RPN used for bounding box detection and classification, and an instance segmentation network used to highlight all the defective pixels. Cas-VSwin performed better when pre-trained. Zhang's work is a student-teacher model whose backbones include a Swin-transformer, various pre-trained YOLOv5 C3, a feature fusion system working with a dual attention module, and various decoupled detectors as the head. The dense stacking of multiple decoupled detectors helps the models to detect objects of different scales. In addition, Guo et al. [267] developed a framework based on YOLOv5, where some convolutional blocks were replaced by transformer encoders. A transformer like feature extraction stage allows the larger collection of neighborhood information related to the defect, and thus improves the accuracy of detection.

#### *f: CASCADED DETECTION AND SEGMENTATION*

The field of precision manufacturing is demanding to measure defects not only through their detection and localization, but also with pixel-wise segmentation in market-attractive solutions. The network proposed by Yang et al. [215] implements two cascaded phases, which are detection of scratches on steel based on SSD, and a growing segmentation algorithm within the selected box, whose seeds are the Principal Component points of defect. Similarly, Wu et al. [214] developed a two-stage pipeline: the first stage involves data augmentation with a novel GAN to generate realistic images with defects; the second stage aims to detect the defect areas through a light and coarse detection network, and segment them through a segmentation network. Moreover, Xiao et al. [236] proposed a pyramid CNN, which is an improvement of Mask R-CNN model. $\text{ResNet}_{101}$ was used to extract features, which are then fused and processed using a feature pyramid network. The result is sent into an RPN and a Fully Convolution Neural Network (FCNN) separately. The RPN takes care of the bounding box and classification process, whereas the FCNN performs instance segmentation for each proposal passed by the RPN.

#### 4) WEAKLY-SUPERVISED DETECTION

*Weakly-supervised detection* requires image-level annotations. It focuses on improving training efficiency while maintaining the same performance level of more labeling expensive architectures. Real-time inspection capabilities are becoming increasingly required in industry. However, small- and medium-sized companies lack sufficiently suitable performance frameworks in the production line to achieve real-time inspection [284].

Only two works found belongs to this category. Zhang et al. [282] proposed a Category-Aware defect Detection Network (CADN) that uses only image-level annotations. They used a student-teacher model to force the outputs of a lighter CADN (student) to mimic the results of a larger CADN (teacher) in the student's training process. This is done owing to knowledge distillation based on heatmaps, which helps improve both accuracy and speed. Heatmaps are chosen mainly for two reasons: they contain more explicit spatial information, and are computationally less expensive. They also contribute to trust the network once it effectively focuses on the defective areas of the image. Li et al. [281] also made use of heatmaps enhanced by spatial attention to perform detection of tiny defects. Along with a standard CNN based on $\text{ResNet}_{101}$, two modules supported by one CNN each help the training process to perform better: the former focuses on the defective part, whereas the latter focuses on the leftover portion of the image.

#### 5) SEMI-SUPERVISED DETECTION

*Semi-supervised detection* uses mixed annotations (e.g., few bounding box and few image-level labels). At this time, only one work, belonging to *active learning*, satisfies the labels requirements.

*Active learning* selects the effective data for annotation and represents a valuable alternative for reducing the labeling efforts. Since the process starts from unlabeled data, the annotator needs to work only on uncertain images, which are used to retrain the entire system. Only one work is found: Lv et al. [280] proposed a framework based on YOLOv2, pretrained on ImageNet, which consists in a loop of three main modules, that are detection model, active strategy to sample uncertain images, and annotations update.

#### 6) UN-SUPERVISED DETECTION

*Un-supervised detection* trains a model without labeled data. Currently, the authors have found *one-class* approach that uses only defect-free images to locate and delimit the defective area with a bounding box. Dong et al. [7] developed a multitask learning method with an AE and a *one-class* classifier. The total loss takes into account losses in both image reconstruction and minimum hypersphere volume estimation. Since the encoder included in the *one-class* classifier comprises a fully connected layer, it can only be used to classify the entire input image. Then, they used a moving window strategy that cuts out patches from the input image, feds the network, and gets an anomaly score for each patch. Using a threshold, patches that scored higher were selected as anomalous. Arima et al. [134] adopt a CAE to reconstruct, learning from defect-free samples, the input image, and take the absolute value of the difference between images to retrieve a defect localization.

## VI. DISCUSSION

The ever-rising throughput in the manufacturing industry is striving to enhance products quality evaluation and precise repairing decision-making even in complex scenarios; therefore image- or video-based defect inspection systems boosted by accurate, fast, and explainable DL have a pivotal role. The large diversity of defects encourages cross-domain research to cope with data scarcity and over-dependency from specific operating conditions.

The reviewed publications have addressed surface defects recognition with three specific issues (segmentation, detection, and classification), together with a range of learning methods (including fully-supervised, semi-supervised, unsupervised and, except for classification, weakly-supervised approaches) depending on the available data and research objectives. The following section summarizes the DL methods which endeavour to recognize surface defects while counteracting some of the challenges presented in Section III, whose relationships are illustrated in the Venn diagram of Fig. 10. We review the approaches crossing the overlapping areas with a devoted subsection and in the Table 12.

### A. DISTRIBUTION OF LITERATURE

The barplot in Fig. 7 reveals that some implementations have been employed more than others in the related literature. The defect classification task is the widest adopted objective task in the reviewed works even because its miniaturization is applied to the prediction of image sub-regions (e.g., image patches). Moreover, defects classification and detection are the most pursued in the fully-supervised setting. Classification can be used to recognize defects, their severity or functional anomalies in the focused portion of products: chronologically, the classification task was the first to be explored in the field of defect recognition, followed in second instance by segmentation and, finally by detection, as shown in Fig. 8a. In particular, the latter saw a slow development, surging only since 2019, because it was subject to topologies advances to accommodate real-time and adequate inference demands on the production lines. In fact, although R-CNN dates back to 2015, it produced satisfactory results at the expense of a relatively low speed (about 10 fps). Only with the introduction of YOLOv3, and subsequent versions, the performance in terms of inference time (more than 30 fps) and accuracy have started to be suitable for defect recognition in the industrial sector. Defect detection gives a more flexible localization with regression of bounding boxes. A reason is the feasibility of annotations because both image-level and bounding box annotations require less expensive human effort even when the dataset is acquired directly from the

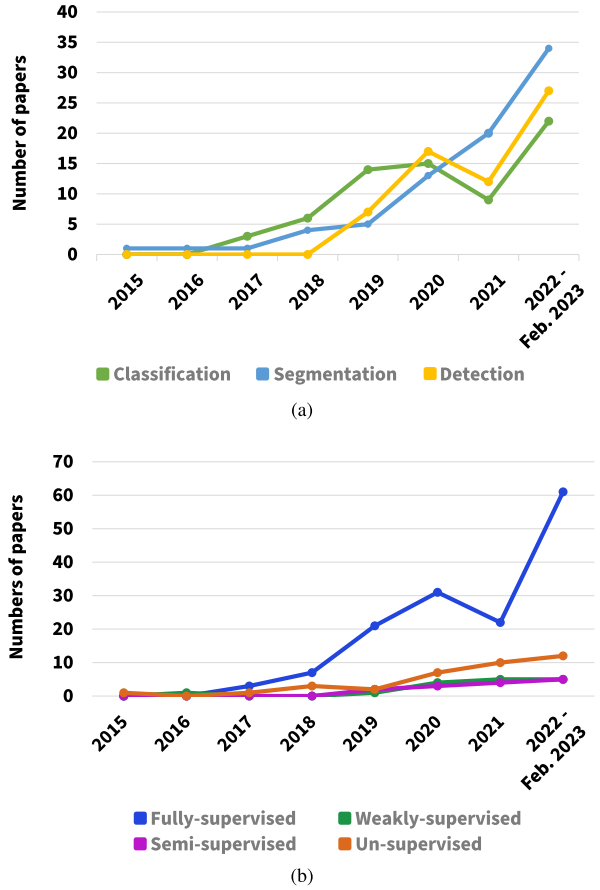



**FIGURE 8.** Line charts displaying the number of paper publications over the years for (a) objective tasks and (b) learning methods.

production line. Conversely, the segmentation task is the less affordable in the fully-supervised setting, since dense and pixel-wise annotations are required: actually, weakly supervision for segmentation task employs the same labels used for fully-supervised classification or detection (image-level and box-level labels respectively) in addition to scribbles; hence, after the fully-supervised approach, it is the widest adopted learning method for segmentation. Semi-supervised and unsupervised approaches are roughly equally used for classification and segmentation, and more than for detection.

Furthermore, grouping the articles by learning methods, it can be noted that fully-supervised setting is still the most explored in the industrial field, since it is compliant with easy-to-use software with estimated performance and additional implementation costs that are lower than the annotation ones. On the other hand, softly-supervised approaches are progressively gaining ground, as can be seen in Fig. 8b, because they allow more flexible solutions in dynamic environments and are keeping pace of state-of-art performance. However, they are at the forefront of the current research, and involve less costs for the annotation phase but higher framework development costs, with a delayed but effective return-on-investment. Considering this, the authors investigate whether these distributions depend on the annotations provided along with public and large-scale datasets.

#### 1) OPEN SOURCE DATASETS FOR BENCHMARK

Large-scale datasets lead to advancement in many areas of image-based DL research, and provide a common benchmark for fair comparisons and quantification of performance. At a first glance, an objective task can be performed if the network output can be directly compared with the available ground truth. In addition, a range of different ground truths can address an objective task, which is consistent with authors' initial hypothesis.

Among the reviewed papers, even if several works are trained on in-house labelled datasets, the most referenced public datasets include the *MVTec AD* dataset [53] which collects 5,354 RGB images of 15 categories with pixel-precise ground truth for textured and defects on foreground objects. Defect types mimic real-world industrial occurrences and are subdivided into training set, which contains only defect-free images, and test sets with both defects and defect-free images. *DAGM 2007* is a synthetic grayscale dataset containing 10 defect classes in 575 training images and 575 test images on various textured backgrounds. It provides dense ellipses coarsely overlapping defective areas both on the training and test set. The *NEU* dataset collects 6 surface defect classes on metal workpieces, each entailing 300 grayscale images and provide ground truth bounding box [37]. The *Severstal* dataset contains 12,568 training and 5,506 test images roughly balanced between defective and non-defective classes of four types of strip steel surface defects provided with pixel-wise masks.

The authors remark that, up to date, no public datasets provides scribble-based image annotations; probably because it might require some strategies to enrich annotations with finer details, with the aim of reducing defects scale uncertainty [285].

The authors have found the upgrade of one type of class label to a more specific one to allow fully-supervised learning [202], [276]. For instance, MVTec AD is tailored for unsupervised learning, but is also found in the fully- and semi-supervised setting by undertaking manual additional labelling. On the other hand, each of the four datasets in Fig. 9 would provide a fully-supervised setting for classification. In this case, a labelled image takes the tag associated with a defect mask, bounding box or delimiting ellipse. Figure 9 reproduces the number of papers per each objective task, grouped by the learning methods explained in Section IV and using these public datasets. NEU-Det is the widest adopted for defect detection with full-supervision. Bounding boxes represent weak supervision for the segmentation task, while a fully supervised setting can be available for this task by adding precise-pixel mask for defects.

### B. IMPROVING THE GENERALIZATION ABILITY OF MODELS

#### 1) TACKLING IMBALANCE AND SMALL DATA PROBLEM

Training a supervised DL system heavily rely on the availability of big data regarding the application field of interest

**TABLE 12. Comparison of advantages and disadvantages of the most common DL methods addressing the challenges in the industry defect recognition field.**

| Methods | Advantages | Disadvantages |
|---|---|---|
| Data generation<br>Data augmentation | Relieve from a small or imbalanced dataset;<br>Reduce data collection costs;<br>Feature enrichment. | Can propagate biases coming from the original dataset;<br>Quality evaluation of new samples may be huge. |
| Transfer learning | Acquires source knowledge from either close or unrelated domains;<br>Addresses domain-shift problem;<br>Faster model convergence;<br>Requires less training samples. | Requires optimal domain adaptation. |
| Feature enhancement (Pyramid Feature Fusion, Boundary refinement, Deformable convolution) | Improves the receptive ability of tiny, low-contrast, and multiscale defects. | Suffers from intra- and inter-class variability;<br>Additional layers to be trained. |
| Semi-, weakly-, un-supervised training | Leverages the majority class samples;<br>Decreases manual annotation efforts. | Mild or absent guidance to pattern recognition. |
| Knowledge distillation | Fosters salient knowledge learned by a mentor model;<br>Lightweights model complexity;<br>Compliant to real-time inspection needs. | Convergence in the student model is suboptimal in reference to the original model. |
| Perceptive methods | Reflect the contribution of input pixels for final decision;<br>Localize and/or segment defects. | An expert evaluation is required due to qualitative results. |
| Mathematical methods | Can help in deciding the optimal network hyperparameters;<br>Quantitative measures can be further derived (i.e. clustering/similarity). | Rely on plots to be deciphered. |
| Thresholding CAM | Exploits inner feature maps to obtain pixel-wise prediction. | Coarse defect segmentation and threshold dependent. |
| Lightweight models | Inference time is lowered with respect to traditional methods. | Generally less accuracy is achieved. |
| Few-shot | Uses just a bunch of defective images. | Support set needs to be constructed carefully. |
| Incremental learning | Can dynamically update the decision rule according to new injected samples;<br>Detects new defects without forgetting the previous knowledge. | May struggle with stability. |
| Active learning | Select only uncertain data for expert annotation;<br>Can handle noisy labels with an evaluation of the confidence level. | Hybrid processing due to human intervention. |
| Confident learning<br>Meta learning<br>Label smoothing<br>Sample bootstrapping | Finds out mislabelled samples. Retrieve and repair noisy labels. | Can reject samples due to intra-class variability;<br>Can exacerbate small data problem. |

to avoid overfitting. Although the recent advancements in the generation of images by sampling the manufacturing line products, the assembly of huge datasets inevitably requires time and human effort during collection and labeling, that leads to the small sample problem [49]. The data imbalance problem escalates the recognition effort when dealing with fine-grained defects, since they require a dense visual receptivity [39]. On the other hand, there exist intrinsic factors of imbalance in the application itself since industrial processes are continuously improved to avoid anomalies. Real-world datasets suffer from several forms of imbalance: an *image-level imbalance* for a skewed ratio between image classes; an *object-level imbalance* when the distribution of object occurrences is skewed (e.g., due to occlusions) and *pixel-wise imbalance* between background and foreground pixels. However, even in the case of an almost balanced dataset, there

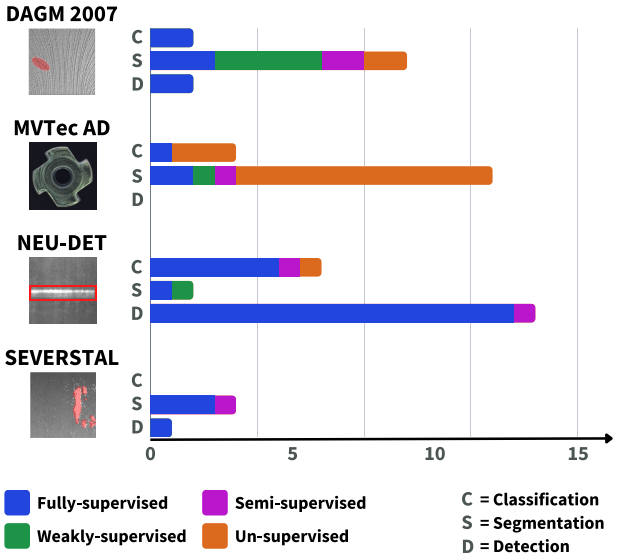



**FIGURE 9.** Papers publications per each objective task (Classification, Segmentation and Detection grouped by the learning methods explained in Section IV (colors blue for fully-supervised, green for weakly-supervised, fuchsia for semi-supervised and orange for un-supervised), using each public dataset (DAGM2007, MVTec AD, NEU-DET, and Severstal).

could be imbalance between subgroups of defect classes as well as scale imbalance of defects due to intra-class variability. In the sequel we summarize the methods to relieve from a small or imbalanced dataset with the aim to allow robust training and fair testing of the DL recognition systems; indeed, the performance in case of overfitting must be sensitive to data distribution. They aims at:

- **Rebalancing the dataset** by operating from the root of the problem, hence directly on the dataset. The *data augmentation* methods encounter image wrapping based on several possible manipulations and oversampling approaches that preserve the label. *Data generation* consists in synthetic data generation with AE and adversarial networks (i.e., GAN based).
- **Enhancing the feature extraction** to overcome overfitting, acting on the feature engineering (e.g., feature pyramid fusion and feature weighting or selection through a feature attention mechanism [39]), or on the information flow like residual networks (e.g., ResNet backbone), dropout and batch normalization.
- **Strengthening the training** through *semi-* and *unsupervised* learning methods, which circumvent the class imbalance by leveraging the majority class samples. An optimized loss function should introduce a cost-sensitive penalty or a regularization term to penalize missed defects recognition. Furthermore, the transfer learning helps in reducing the amount of defective training images. It is often adopted to generate well trained networks when *few-shot* of target defects are collected and labelled [51]. Otherwise, the few-shot algorithms adopt a base training (e.g., on augmented data) and fine-tuning resorting on the few available samples to resolve the imbalance and the volume of annotated data required [239].

### 2) TACKLING DATA ANNOTATIONS WITH NOISE

DL-based visual pattern recognition relies on a high-quality data preparation but in real conditions it may be impaired by annotations often inadequate. In situations where a small training dataset is available, the collection of additional samples of different sources may lead to varying levels of label quality. The *noisy or inconsistent labels* can largely degrade the model training by confusing the model, which worsen the decision boundaries fitted on the *clean* data samples [286]. The overfitting is more likely to occur since, in a situation where labels are noisy, the patterns to learn raise and more channels will be activated [58]. The methods used to overcome the issue of noisy labels towards a robust training includes:

- **Architecture strengthening** by adding a *noise adaptation layer* which consists in weighting the model's prediction by the label transition matrix which is learnt during training. Moreover, *dedicated architectures* are used to tackle more complex noises. As instance, Yu et al. [262] designed an *auxiliary inference model* that compares the mapping functions developed within a labelled dataset having label noise and unlabelled dataset. The consistency of the sample predictions between the two datasets is used to handle inaccurate annotations.
- **Modifying loss function** to automatically ignore or to weak the emphasis on mislabeled samples along with the model training [287].
- **Label smoothing and refurbishment**: the former transforms the hard label y (e.g., one-hot encoded for a K-class classification) to a soft target $y_s$ acting as a regularization technique [288]: $y_s = (1-\alpha)y + \frac{\alpha}{K}$, where $\alpha$ parameter modulates the level of confidence during training thus avoiding over-fitting predictions. The latter consists in replacing the original given noisy label with a refurbished one; Gao et al. [289] adopted a plug-and-play additional module with Bayesian statistic and a time-weighting module for optimal label selection.
- **Sample bootstrapping** which provides a selection method for clean samples to update the model by recognizing clean samples as the small-loss training samples [287]:
- **Active learning** corrects noisy labels graded by evaluating the confidence level expressed by the network, refining labeling at each iteration with pseudo-labeling; it may rely on processing engineers as oracle to verify the annotations given for some uncertain samples that will help the model to better generalize [280], [290].
- **Confident learning** improves the training by estimating sample confidence levels to characterize the suspected wrong-labeled pixels, pruning the mislabeled samples

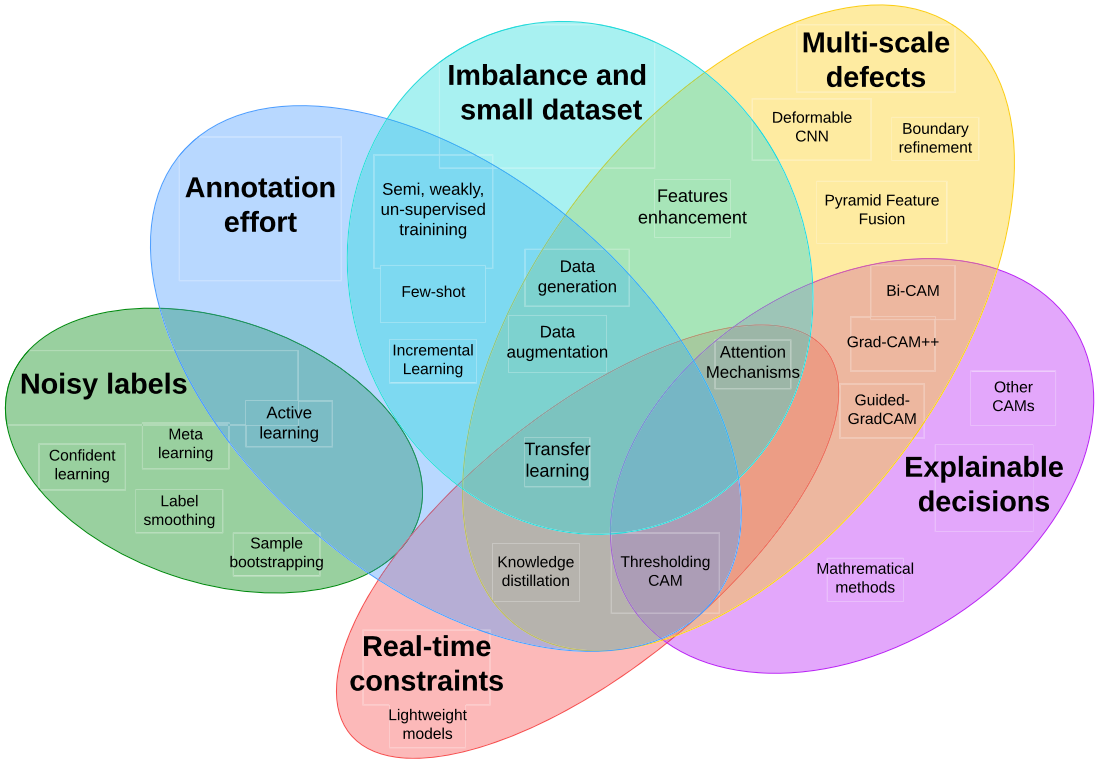



**FIGURE 10.** Venn diagram illustrating the relationships between the DL methods employed to recognize industrial surface defects and the counteracted challenges presented in Section III. Some methods fall into more than one category.

to choose the clean data, and re-training models on the purified dataset [55].

- **Meta-learning** regresses an implicit rule to update the learning process aimed to consistently tolerate the noise presence and get the underlying knowledge from data. For instance, Li et al. attach to training images different synthetic generated labels as perturbations and enforce the network to be consistent with the prediction of a teacher network trained only on clean samples, by updating the gradient before the conventional update [57].

### C. KNOWLEDGE TRANSFER, REUSE AND DISTILLATION

#### 1) TRANSFER LEARNING

Throughout the analysis of works applied to various DL surface defects applications, the efficacy of the backbone has proved to have a leading role in the comprehensive understanding of training data distribution. Designing and engineering the architecture – regarding depth, width, and cardinality of layers and additional modules, both trainable or not (e.g., multi-scale convolution, implementing feature fusion, channel and position attention maps, feature pyramid concatenation and residuals additions) – enable defects recognition because it shrinks the receptivity for the over-stuck or noisy irrelevant areas. However, the model is made obsolete by any change in the test distribution due to *"domain shift"*, such as changing operating conditions, inclusion of new production units, and new defects [291]. Hence, a new training phase to update the model is required, but overfitting can incur unless a sufficient number of novel samples has been acquired. Moreover, this constraint delays the point at which the model is put in operation [190]. Transfer learning encompasses these shortcomings and can help boost training, which becomes more robust against image perturbations of various kinds [84]. The literature contains several contributions using transfer learning paradigm to acquire source knowledge from either close or unrelated domains, and optionally fine-tuning it with a bunch of target images [292]. Given a powerful pre-trained deep network on huge available dataset (e.g., ImageNet), the knowledge reused across domain-specific industrial plants relieves human labor costs and supervision. The most appropriate pre-trained network for a given application depends on the adaptation of features in the same latent space. In addition, several alternatives exist to import pre-trained layers in the context of the model, like freezing and re-training from different network checkpoints [83], [95]. The authors summarize the key findings listing that pre-trained networks are used in the encoder part instead of training from scratch mainly for: coping with difficulty in capturing enough information due to few samples availability; speeding-up the training process, and thus reaching earlier the optimum convergence for the same benchmark problem. Moreover, transfer learning is a widely used technique for model compression [82], [151], [293].

#### 2) INCREMENTAL LEARNING

The generalization ability of a DL defect inspector is currently verified upon a dynamically assembled test set, which belongs to the real-world industrial application. The DL model struggles with the recognition of new defect types that were absent in the training and validation set and that occur along with the production process. Therefore, it is asked to expand the acquired capabilities and update the decision rule by perceiving the alarming samples as those foreign to the familiar ones and by modifying the feature extractor, eventually after a labelling phase, without overwriting or hindering the prior knowledge. Such a *"catastrophic forgetting"* must be avoided in order to preserve the recognition accuracy on old classes while being able to recognize new defects as well [294]. Specifically, this feat is to be achieved without using the data of the previous training or in the hypothesis to have just few prototypical samples of the new classes. The incremental learning approach, thus, endows the DL network with the ability to constructively merge the expertise in different defects and tasks emerged in several stages, without the need to retrain it from scratch [295]. Therefore, the *incremental* or *continual learning* updates the model under the *stability-plasticity* trade-off constraint [296]; remarkably, it copes with the risk of obsolescence of the trained networks and counteracts the consequent tuning for tightly specific applications. Two main approaches hold the goal by:

- **Replaying prior samples** with a memory-auxiliary generative network that reconstructs prior training samples distribution to add them to the novel data when retraining [297].
- **Introducing a regularization term** to avoid a noisy or hasty update of weights, thus consolidating previous knowledge while learning on the new task [296].

Rosenfeld et al. [298] managed to reuse the existing weights forcing to fine tuning on the new defects data through a linear combination of the original filters in the corresponding layer. They achieved better results than simple transfer learning or learning from scratch.

#### 3) KNOWLEDGE DISTILLATION

The *knowledge distillation* enables to ameliorate the performance of a lightweight model by using as supervisory signals the salient teachings learnt by a bigger network [82], [264]. The low-complexity model (i.e., the *student*) exploits both the pseudo-labels generated from the bigger model (i.e., the *teacher*) and the real labelled data. Therefore, it has strong recognition capability even though the number of its parameters is reduced, and its performance are comparable or even better than those of the teacher. In so doing, the knowledge distillation is a class of algorithms devoted to model compression and fusion; among these, there exist also methods that make usage of low-rank factorization, parameters pruning and sharing or pre-trained convolutional filters, as reported by Cheng et al. in their survey article [299].

The work of Hinton et al. [300] sheds light on the distillation of knowledge alongside the transfer of weights from the teacher to the student network to compensate for the lack of supervisory annotation data for training. The student training process is supervised by the teacher distilled knowledge that allows to obtain high performance with less annotation effort. For instance, when dealing with unlabelled data, the teacher helps the student network with different approaches:

- **Employing several channel-level losses** to capture the normal feature distribution of intermediate layers [205], [301];
- **Using a multi-task loss** to check segmentation, contour, and distance map performance [177];
- **Providing pseudo-labels** in order to train the student network along with real labelled samples [125].

The pseudo-labels are the output probabilities from the teacher and these concur to guide the student learning process along with the hard (i.e., real) labels. During training, the student model endeavours to match the teacher output probabilities regulated by the shared weights; thus, a ''distillation loss'', sometimes assisted by an attention module [269], steers the probability distribution generated by the student model towards that provided by the teacher model. The loss minimization process can be related to a *network-level*, whenever the knowledge is optimized only by the last layers or to a *channel-level* if knowledge is optimized at different levels of feature maps, de facto computing multiple losses at time.

As a result of gathering the attention to a compressed model, the frameworks that use knowledge distillation approach can meet real-time predictions in the order of milliseconds: Kim et al. [82] proved their student network has inference time reduced by 97.43% with respect to the teacher ones.

### D. EXPLAINABLE ARTIFICIAL INTELLIGENCE

End-to-end systems leverage self-adaptation of DL but, do not use hand-crafted knowledge. Although several studies have innovated systems and performance of surface defects detection, their black-box behaviour hampers the technology transfer to industry. Therefore, a DL network should be intelligible and describe the connections between inputs and outputs or a mapping function for the model [119].

Authors investigated how XAI methods are used for industrial defect recognition and how they could endorse trust and knowledge; Fig. 11 illustrates the undertaken workflows in the field. An input image is processed by a backbone that extracts local and semantic features during forward information flow. During information stacking, pictures of the intermediate convolutional activations describe how the model is being trained and how straight it goes towards the discrimination of a defective pattern. For example, a backbone can perform feature extraction differently, according to the number of layers, kernel size and depth, and number of training epochs: CAM images can help optimize architecture based on what the network has learnt, and these maps can confirm whether they delineate progressively defective

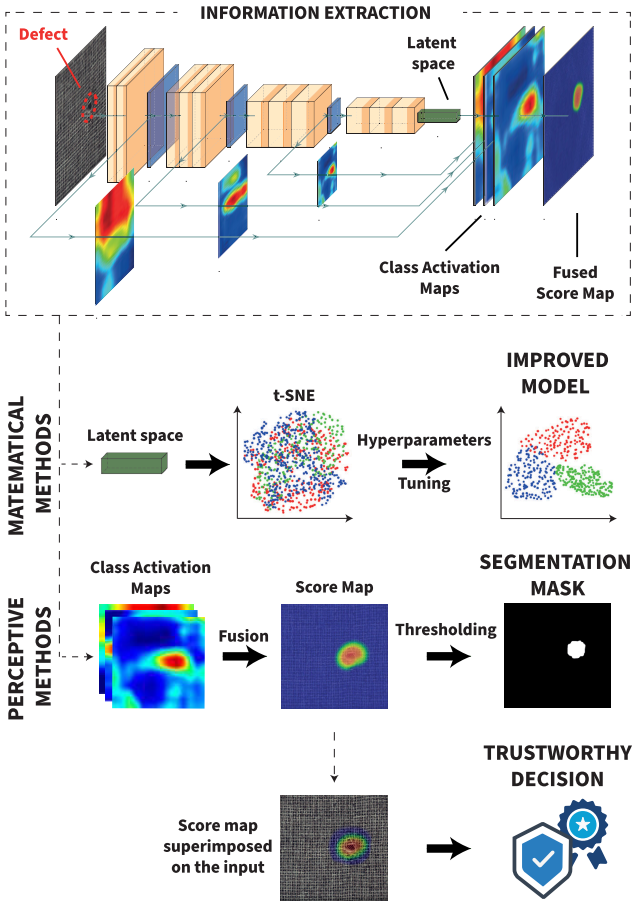



**FIGURE 11.** Use of XAI in DL-based industrial defect recognition. After an information extraction process, the knowledge can be used mainly via mathematical structures to improve a model by hyperparameter tuning and visualizing the latent space distribution (i.e., with t-SNE), alternatively perceptive methods can be used to segment defective areas through thresholding of the fused CAM, and to trust a network decision by overlapping the defective score map with the input.

areas [239], [254]. A new loss function term, supporting a faster training convergence, can prove to be useful by comparing the activation maps extracted from the same architecture when the baseline loss implementation is employed [121]. A parametric evaluation to compare models consists in the visualization of inner states when different sized bunches of target training images or an augmented dataset are provided for fine-tuning the defect recognition. Furthermore, visualization is used to prove an enhanced discriminative ability that can be ascribed to spatial or attention module [270], multi-scale feature fusion [205] or feature regularization [179]. Visualizing feature response maps, which are generated when pre-trained weights are transferred, qualitatively emphasizes the advantages in terms of convergence with lower number of parameters [82] and training images required [95]. Response maps at different depths can be upsampled and averaged to obtain the aggregate defects score map, whose dimensions are the same as input image [3]. The last step in the encoding path contains latent decomposition of the image, which is summarized in vectors that have lost the spatial reference.

Following the taxonomy provided by Tjoa et al. [302], the authors identified works belonging to the *perceptive explanation* and *explanation via mathematical structures*. *Perceptive explanation* consists in saliency maps reflecting the contribution of input pixels for final decision. These weights are mapped into probabilities or super-pixels magnitude importance, such as heatmaps. The class activation map (CAM) is widely adopted to generate heat/saliency/relevance-maps. *Explanation via mathematical structures* analyzes the representability of concepts provided by the extracted features whether separate or similar, through clustering metrics that attempt to show similarities or distance in the low-dimensional latent space. Such algorithms as t-SNE arrange the latent space in two or three-dimensions to intuitively figure out feature embeddings.

Furthermore, the authors found that perceptive explanations methods could be deeper divided according to their linking with defect recognition result: a direct link consists in achieving the segmentation output consisting in a dense and pixel-wise mask for defects when the saliency map covers defects with accurate localization and fine boundaries, instead of demanding network to make a prediction for each pixel. Several segmentation works use feature mapping in conjunction with adaptive thresholds to segment the defective pattern [173], [174]. This step is found in several segmentation networks guided by weak annotations; remarkably, 7/10 unsupervised studies that used XAI also inferred masks thanks to anomalies or residual scores [205]. Secondly, superimposing heatmaps on the initial images shows how successfully the model converts the input image in hierarchical and meaningful features; it is an auxiliary result which is a proof of trustworthiness when detecting defects.

Latent space embeddings can be managed to develop a disentangled samples representations and t-SNE is the correspondent most used method to render the improvements [142], [238]. The latent space encodes image prototypes and several studies using AEs consider the distance between vectors of two distinct classes to maximize the inter-class margin. Visualizing the t-SNE plot for latent features has been employed in the ablation studies of several articles while testing a number of training conditions; these tests have to prove that the proposed implementation leads to better a performance with respect to other options. Moreover, t-SNE has been employed to choose from the available datasets the most suitable one for base learning, in order to subsequently transfer response maps into the target domain with an optimal adaptation by selecting the nearby points to target data [15]. Graphically, t-SNE renders similar samples with short distance in some measurement space, hence an improved model has clustered latent vectors with low intra-class dispersion and high inter-class sparsity.

Exposing most relevant features and weights is becoming a straightforward method. Besides telling "where" the network looks at during the inference stage, a post-hoc explanation can

help to understand ''why'' the network fails if there are some correlation patterns in the dataset acting as confounding factors. For example, a dataset in which an occurring defect-class appears in combination with some constant characteristics could be prone to mis-classify during testing. Consequently, visualizing internal states of a CNN can effectively diagnose reasons behind a biased learning. Addressing low-contrast and tiny defects, visualization methods have been compared visually according to their catching finer details in constrained architectures (Grad-CAM++ [66], Spatial Attention CAM [176], bi-CAM [10], Score-CAM [303]).

Explaining and debugging the DL model can lead to an easier and faster adoption on the production line [304]. Activation maps are used to denote defects and can assist in the reverification process [122]. When false negative errors must be prevented with a special care, some studies could relax the fully-automatic evaluation for an operator-assisted post-evaluation in which unclear predictions are further examined [305], [151].

### E. REAL-TIME AND DECENTRALIZED RESOURCES

Real-time is one of the most crucial challenges that researchers are attempting to achieve. Usually it is difficult to deploy DL vision systems on resource-constrained devices such as the Internet of Things (IoT) and smart devices. Currently, two main solutions can be found in the literature, which are use of lightweight and efficient networks beside the use of decentralized and scalable resources. Increasing efficiency meets compatibility criteria with the constrained computational availability of edge devices, which are usually deployed directly on the production line. Heavier and more estabilished networks can achieve very good performance at the expenses of longer computational time. Cloud resources are quite affordable and provide a good workload collaboration with the edge of inspection systems; moreover, they are almost indefinitely scalable. However, relying totally on cloud platforms on which upload databases and networks for training and inference on new acquired images or videos may overwhelm the bandwidth, create traffic jams with increased latency, and is exposed to security breaches.

Furthermore, DL networks that utilize images for quality inspection tasks require intermittent stopping of the industrial chain, such as a conveyor belt, to enable cameras to capture images of the object being inspected. This intermittent process necessitates a synchronization mechanism to ensure that images are captured at the right moment. Conversely, video-based networks capture a continuous stream of data, allowing for greater speed and accuracy in detecting defects, albeit at the expense of increased computational and data management requirements.

In this context, high-speed cameras play an important role into constraint strengthening since most of current localization models can not run over 60 frame per second [147], [176], [182], [224], [248]. Moreover, not all papers focus on this aspect in a comprehensive manner (e.g., reporting both FPS performances and full hardware settings [62]), making a fair comparison more difficult. Some quality inspection tasks require to check the object from multiple sides; for instance, this can be achieved by letting a cylindrical object rotate on a conveyor chain, leaving the camera fixed in the same position [263]. However, beside the acquisition speed, a performing data management and processing is required. The authors compared a desktop-based solution with respect to a web-based ones and showed how handling data with javascript was more efficient with respect to OpenCV, allowing them to use a 120 FPS camera on a MobileNet smoothly.

In the defect detection task, 80% of works compliant to data processing and inspection in real-time makes use of one-stage flexible architectures: EfficientNet [77], [254], [256], YOLOv3 [268], [272], SSD [270], DenseNet [261], instead of two-stage detectors with additional RPN network. This latter are rarely employed for real-time inspection, although their backbone can be modified and compressed due to pre-trained layers benefit [259], [275] and can be fine-tuned with few shots [240]. In addition, having compressed models in result of knowledge transfer and distillation may lead to a faster and better training. A growing interest is within detection and tracking systems in edge-cloud collaborative resources towards the continuous monitoring of products and processes directly *inline*.

### F. FUTURE OUTLOOK

This section stems from the literature analysis and aims to propose current and outlook research synergies. Despite several progresses have been traced, the main feasible avenues in the field consist in the following. 1) Coping with the mismatch between convolutional kernel dimensions and defect scales, in addition with saving computational resources for edge devices capability. 2) Improving strategies for dynamic inter-domain alignments of pre-trained layers through fine tuning, combining complementary training datasets. While benefitting from using pre-learnt deep features, it is challenging to fine-tune layers using small datasets due to overfitting. 3) Registration and fusion of RGB with depth cross-modal information to enrich the differences between defective and normal patterns [306]. 4) Increasing reasonableness of the system by inserting parameters coming from the process and proving the collaborative interplay of physics-based features with deep features extraction [42]. In some critical application scenarios it can be a step toward the evidence-based prediction which could increase the interpretability of inner states [15]. 5) Continue research on latent space disentanglement factors and saliency map generation, for qualified and accurate defect predictions strengthened in ''softly'' supervised setting [307]. 6) Creation of public available dataset for algorithms benchmarking.

In the near future, the authors believe time-saving, flexible and explainable solutions would exert a turning force towards best practices for competitive computer vision systems, especially for the industrial field. These technologies

are expanding the scales and scopes progressively, advancing cross-domains knowledge aggregation and distillation within wide and public datasets to reach a robust validation.

## VII. CONCLUSION

This survey has evaluated promising deep learning (DL) frameworks addressing surface defects recognition on industrial manufactured products and components. The three main objective tasks (segmentation, detection, and classification) for products quality assessment count a plethora of works. In this article, the authors emphasize the different learning methods to train DL systems depending on the available knowledge listed in the training dataset. To explore this hypothesis, the selected publications were firstly grouped into different objective tasks to further analyze how they were achieved with different training supervision. A detailed description of each publication was provided, extracting the required benchmarking to highlight relevant research trends towards the improvement of surface inspection systems in the rapidly-changing industry 4.0 revolution. The reader was gradually involved into the main challenges during the exploration of targeted solutions alongside with their strengths and weaknesses. Thanks to this analysis the authors have discussed how some possible solutions are being realized and are more explored in some learning methods and/or objective tasks than in others. Common vision architectures were studied to provide interested readers with an effective guide to approach both academic and industry research starting from a compound and recent overview. Inspection tasks based on Convolutional Neural Network (CNN) are extensively used due to their effectiveness in capturing not only detailed and semantic, but also either local or long range patterns. In the tables CNN-based frameworks processing various image data types and patch sizes, sources (RGB cameras, X-Rays, thermal, IR thermography, CT-scan) and number of channels (RGB, grayscale) are reported. Deployed architectures such as CNNs, as well as data pre-processing, augmentation, feature engineering, and loss functions have been constantly improved considering class imbalance and subtle differences between classes, in order to catch multi-level defects appearance, detect new defects, and become more efficient by making full use of features and training on a reduced number of images.

This work is aimed to overcome narrowed conclusions concerning a specific vision DL-based defect diagnosis, and to encourage a synergistic further research.

## APPENDIX A
## BACKGROUND ON DEFECT RECOGNITION METHODS

The aim of this Appendix is to provide useful theoretical concepts on defect recognition methods to the reader, thus allowing a comprehensive understanding of the in-depth analysis reported throughout the paper.

### A. MULTI-SCALE IMAGE REPRESENTATIONS

In defect recognition tasks, many researchers explored the aggregation of multi-level features, which is apt to enrich local information and boost the performance in defects localization [308], [309]. Performing convolution on the image realizes several downsampling operations, which are defined by the stride of convolution and pooling layers. As a result, this significantly reduces the resolution of feature maps while augmenting abstraction and feature depth layer by layer [308]. Convolutional Neural Networks (CNNs) recover full resolution and original image size through Atrous or Deformable convolution, which is followed by bilinear interpolation as upsampling filter. Instead of using deconvolutional layers, this approach limits the number of learning parameters as well as the computational burden.

A Spatial Pyramid Pooling (SPP) module absolves the mismatch between input image size and dense-connected neurons. This module generates fixed-size vector from arbitrary sized input image by pooling within spatial bins, whose resolution is proportional to the image size. SPP module enables to compare pooled features from locally-connected regions. The existence of objects with different sizes makes defects recognition an even more challenging task. A standard way to tackle this issue is to extract different CNNs intermediate representations of images, to recovers the original image resolution, and fuse them. As shown in Fig. 12, multi-scale feature are resized to target dimensions and concatenated to realize an overall perception of context through feature pyramid fusion.

However, fusion does not take into account feature importance; to help propagating effective information throughout layers, a trainable layer called feature attention mechanism weights the extracted features and enhances simpler defects discrimination before features fusion. Deeper models with high-level or semantic features are proved to be useful for classification task and less for localization of defects in images. In fact, a smoother feature response can help in the coarse localization but not for delineate boundaries. To this purpose, boundary refinement module is a residual structure that inserts shortcut connection between shallower (fine-grained or edge contrast) features and deeper (contextual) features, thus helping in preserving salient details.

### B. AUTO-ENCODERS

An overall structure of an Auto-Encoder (AE) consists of three main components, which are an encoder, a bottleneck configured with latent space embeddings and a decoder, as shown in Fig. 13. Thanks to this configuration, AE can first learn a low-dimensional encoded representation of the input image, and then use this information to reconstruct back the original input. The encoder path reduces the input dimension and extract data informativeness into a restricted latent vector containing abstracted knowledge. The aim of the bottleneck is to let only the most essential information pass from the encoder to the decoder. Finally, the decoder consists of upsampling and up-convolutional blocks that reconstruct the output from the bottleneck.

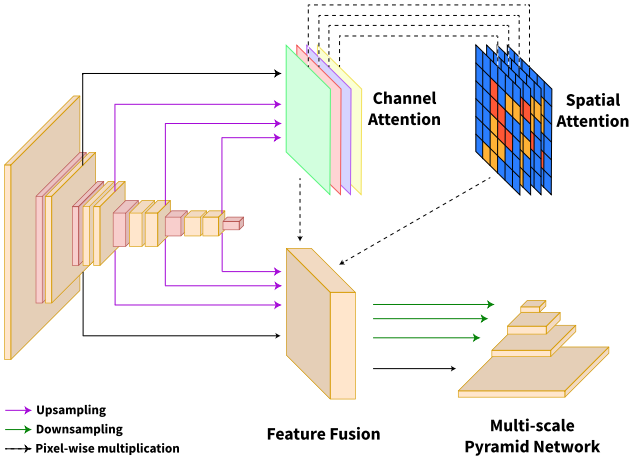



**FIGURE 12.** Multi-scale feature fusion technique with optional channel and spatial feature attention mechanism for information enrichment.

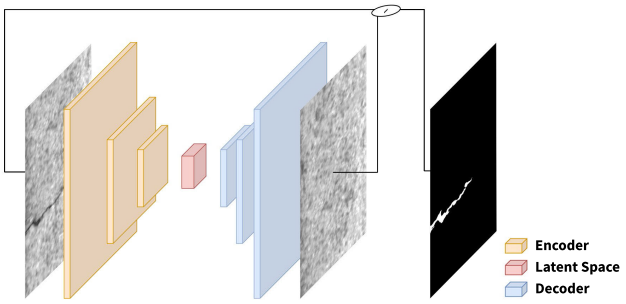



**FIGURE 13.** Auto-Encoder sample scheme for defect segmentation task (input, output, and defect mask from KolektorSDD [151]).

In defect recognition tasks, AEs can be trained in supervised learning (Variational-AE) to generate images containing a specific type of defect, or in un-supervised learning with only defect-free images whose possible applications can be grouped in: 1) reconstruction of input image and 2) its denoising; both tasks are possible if an accurate description of salient normal class features is pursued. During inference, an input defective image is poorly reconstructed because defects disappear; consequently, subtracting the input image, defects can be perceived and quantified through residuals. The anomaly score is often linearized in a standard range (e.g., [0, 1]) and compared with a threshold to obtain the final prediction. To enlarge the decision margin with a remarkable reconstruction error, AEs need to capture only those patterns present in the normal class and not in the anomalous class, otherwise they could fail in distinguishing two different types of samples. A topic of interest is to balance reconstructive power of an AE with latent space dimension.

### C. VISION TRANSFORMERS

Transformers are ever more used in image defect recognition tasks. Basically, they are composed by an encoder–decoder architecture. The encoder represents input data in a latent space, while the decoder takes all the encodings and their

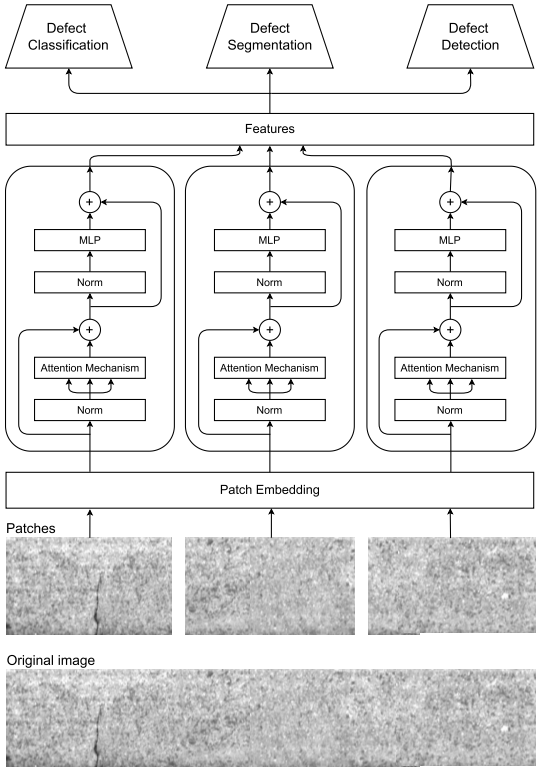



**FIGURE 14.** Vision Transformer sample scheme with 3 patches (input image from KolektorSDD [151]).

enclosed contextual information to generate the output sequence. Both these components comprise a variable number of blocks with the same composition: a multi-head attention layer, a shortcut connection, a feed-forward neural network, and a layer normalization block.

More specifically, in the context of Computer Vision it is worth to mention Vision Transformers (ViT) [310], which is a pure transformer that directly acts on the sequences of image patches. It follows the original design of the transformer as much as possible. The number and dimensions of patches can be easily modified. The self-attention mechanism owned by transformers can perfectly address the different tasks in the industrial defect recognition field. In fact, after being applied originally in the Natural Language Processing field, researchers demonstrated how transformer-based models show excellent performance on a wide range of visual tasks, including high/mid-level vision, low-level vision, and video processing [311]. The head of a ViT can differ depending on the objective task to be accomplished. For instance, a decoder can be present in a segmentation task, whereas in a classification task a Multi Layer Perceptron could provide the desired image-level tag. A sample configuration of a ViT, with 3 equal-size patches, is shown in Fig. 14.

Transformers for object detection can be used in several ways [312]: transformer backbones for feature extraction, with a R-CNN-based head for detection; CNN backbone for visual features and a Transformer-based decoder for detection; a purely transformer-based design for end to-end object

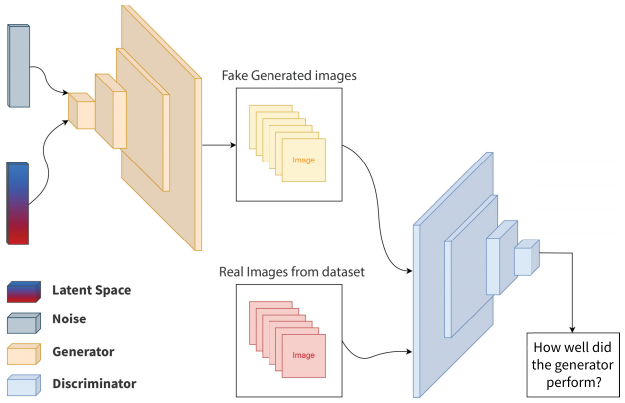



**FIGURE 15.** Generative Adversarial Network (GAN) sample scheme.

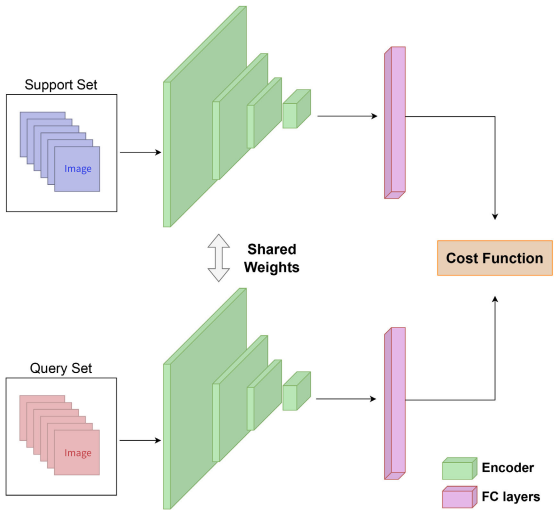



**FIGURE 16.** Siamese Networks sample scheme.

detection. However, transformers have some disadvantages. For example, if the image resolution is high, the transformer requires significant computational power; the computational complexity of its self-attention is quadratic to the image size. Swin-transformers [313] reduce the computational burden by shifting window partitions to calculate self-attention, thus making the complexity linear with the image size.

The original vision transformer is good at capturing long-range dependencies between patches but disregards the local feature extraction, as the 2D patch is projected onto a vector with a simple linear layer [311]. Compared to CNNs, pure transformers lack inductive biases and rely heavily on massive datasets for large-scale training [310]. Consequently, the quality of data has a significant influence on the generalization and robustness of transformers [311].

### D. GENERATIVE ADVERSARIAL NETWORK

The original Generative Adversarial Network (GAN) architecture by Goodfellow et al. [314] in 2014 is shown in Fig. 15, and comprises two main components, which are a generator network and a discriminator network. The former network takes as input random noise and generates new examples, while the latter takes both real examples from the dataset and examples generated by the generator, and tries to determine which are real and which are fake in an un-supervised way. The two networks are trained together in an adversarial manner, according to which the generator tries to output an undistinguishable image from the one given as input, by decoding the latent-space in order to fool the discriminator, while the discriminator tries to correctly identify the fake images. A basic GAN architecture. Over training epochs, the generator becomes better at producing realistic examples, and the discriminator becomes better at identifying the fake examples, thus generating high-quality, expanded data.

Training a GAN does not require a balanced dataset, and it is often trained only on anomaly-free images. The information flow through the GAN can be sampled in a mid stage, meaning that the latent space contains the generative and reconstructive potential that is skimmed from the input and will be developed by the decoder. Conditional Generative Adversarial Network (cGAN) is trained in a supervised setting to guide the generation of specific categories of defective images; thus performing augmentation to tackle the imbalance of training data [315].

### E. SIAMESE NETWORKS

Siamese networks are a class of architectures designed to compare two inputs to determine their similarity, and is showing sterling performance in the visual *few-shot learning* [52]. They consist of two identical branches, each one composed by several convolutional filters, that are trained on distinct inputs coming from the same or different classes, and are called the *query set* and the *support set*; the Fig. 16 illustrates that the weights of the two branches are shared in order to represent similar samples in contiguous vectors in the latent space while maximizing the distance between disjoint patterns, as objective function [240]. At inference time, the defect is recognized by querying the most similar example among those previously stored and labelled. In so doing, the siamese network allows the recognition of novel classes of defect without retraining and by only storing *few* representative (i.e labelled) examples to unravel decisions for test samples.

### F. TRANSFER LEARNING

In the actual production process, labeling high volume and high quality images for DL training is difficult and costly. Besides the scarcity of defective samples, the operating conditions change and even a well-trained model can have a poor performance when deployed on test data (real-world, target data), whose distribution is different from that of data on which it was trained. Hence, the distribution change or domain-shift problem hampers reusability of existing methods [316].

Transfer learning can address domain-shift problem and improve the performance of models by converging the

knowledge acquired from auxiliary systems, which are trained with a large availability of images (from one or more dataset) [317]. Transfer learning is an effective optimization method for trainable parameters especially when ambiguous edge and low contrast defects occur [43]. This transferable knowledge enhances a disentangled representation with a reduced overlap among concepts and classes in deep feature extraction and weighting [29]. Deep features are hierarchically organised and shallower features are easier to transfer with an optimal domain adaptation than those with higher semantic content [318]. Heterogeneus transfer learning projects source and target features in a common space; the number of dimensions is adapted, as well as, other parameters (e.g., the number of classes). Shi et al. [128] recently improve the projection of source and target features with a Center-based Transfer Feature Learning, in which both mean value of distributions (location parameter) and variance (scale parameter) are considered in order to reduce distribution difference and improve robustness of classification adaptation.

Transfer learning is commonly deployed with supervision and fine-tuning, but there exists unsupervised transfer learning in which source data is labelled and target data is unlabelled. Knowledge is transferred into domain specific applications with usually few categories (than ImageNet dataset [319] with 1000 classes, which is widely adopted in many image-based defect recognition tasks [81], [224], [242], [275], [276], [277], [278]) by importing trained weights as warm or frozen checkpoints in the new backbone. In the first case network re-weights all layers back-propagating the error on the handful target images; in the second case, freezes shallower layers and fine tunes only deeper ones. Pre-trained feature transfer is equivalent to taking the outcome of convolutional learners as a shortcut towards a well-posed DL system with a leap towards convergence in feature representation (more intra-class compactness and inter-class discrimination than training network from scratch). The fully connected neurons, as well as the input and output image size have to be adapted to the model requirements and defect recognition classes.

### G. MAKING THE NETWORK DECISION HUMAN-INTERPRETABLE

In contrast to linear models, in which decision boundaries seem transparently determined from updated learning weights, deep neural networks are black-box decisors both in feature selection and class representation [17]. Supervised learning lies in finding patterns inside given correspondence of data with ground truth, whilst softly supervised systems are more free to represent data. Nevertheless, in both cases, but especially the latter, they should be interpretable suggesting explanations based on causal relationships. In defect recognition tasks, the explanation of the network output consists in extracting information from a learned model; a post-hoc analysis is developed by visualizing latent representations

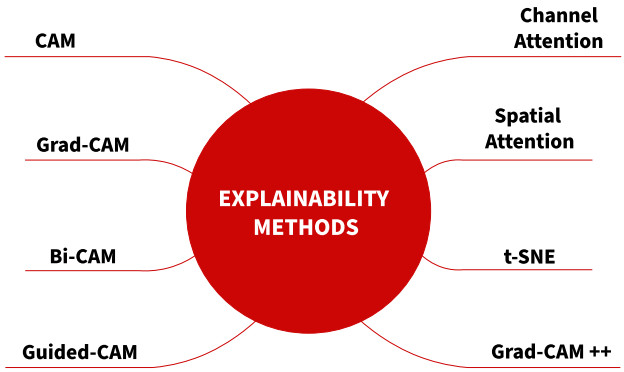



**FIGURE 17.** Methods used for explanation in defect recognition tasks.

that are conducive for the prediction [320]. An overview of the methods found in the surface defects literature is proposed in Fig. 17 to elucidate the functioning of the model.

To qualitatively assess the effectiveness of training rules, the visualization of high-dimensional features in a manifold reduced latent space is provided from T-distributed Stochastic Neighbor Embedding (t-SNE). This algorithm projects similar features as spatial clustered points and viceversa. A good feature extractor (e.g., having good discrimination ability) has a t-SNE plot, in which nearby features are for those samples that not only belong to the same class, but are also sufficiently separated from sample features belonging to other classes.

Another popular post-hoc qualitative analysis computes saliency maps and visualizes areas locally relevant for network decision: the so-called Class Activation Mapping (CAM) method. It performs global average pooling on the last feature map of the network before the softmax decision function. However, this procedure is only applicable if the architecture has a suitable layer. To expand bilinear architectures for CAM definition, bi-CAM gets weights from the eigen decomposition approach [10]. Gradient-based methods use backpropagation to compute the derivative of class score with respect to the input image, acting as weighting coefficients for internal feature state, thus allowing to quantify the importance of each pixel for the output. Grad-CAM has a good trade-off between semantic and spatial information since it results from the linear combination of weighted sum of convolutional feature maps, followed by ReLU function usually fed by the last convolutional layer. Grad-CAM does not weight average partial derivatives, which leads to representing only partially objects and defects where network looks on, lowering trust in the output [321]. Grad-CAM++ is an effective generalization to cope with poor object localization, which computes the CAM-weights as a weighted average instead of a global average. When Grad-CAM is pointwise multiplicated by Guided back-propagation, Guided-GradCAM is obtained, which presents some finer details that are useful for both localization and texture description.

The attention mechanism is a module aimed to emphasize or suppress data representations in correspondence of any convolutional layer depth, as illustrated in Fig. 12. It is extensively studied in many vision tasks [52], [66], [270] since it gives twofold advantages in features of interest representation. In fact, while focusing on the discriminative inter-channel or inter-spatial relationships among features, it performs adaptive features refinement [67], thus suppressing noisy information and boosting the performance. Channel attention is a 1-dimensional vector weighting each channel in correspondence of a feature map, whilst spatial attention module is a 2-dimensional vector weighting (enhancing or suppressing) regions of a single channel according to their representation power. In these attention mechanisms, which can be performed sequentially through element-wise multiplication, channel attention weights are broadcasted along spatial dimensions, and viceversa. Attention weights are trained to let the network focus on features that are important for the application scenario, thus exploiting well latent space dimensions. These salient features can be visualized through such model as Grad-CAM and compared with the baseline feature representation, which are commonly shaped by the expressiveness of the loss function.

### H. PERFORMANCE METRICS

The performance metrics are used to evaluate the mapping function between the feature space and the ground-truth label learnt during either the supervised and the un-supervised settings. In fact, even in this latter case the test samples are labelled and used to evaluate the quality of the model prediction. The four main outcome measures are: true positive (TP), which stands for a correct classification or a thorough localization of defect, while a wrongly recognized defect where it is absent results in a false positive (FP); a true negative (TN) is derived when a defect is not present and, concordantly, not recorded; lastly, a false negative (FN) stands for a missed defect recognition, when defect is actually present, or for an incomplete localization of its extent. According to the level of detail of the network decision, these metrics are suitable to compare image-level, region-level, and pixel-wise predictions with the ground truth. Specifically, defined TP, FP, FN, and TN, further indicators like accuracy, recall, precision, and $F1_{score}$ (i.e., the harmonic mean of recall and precision) follow, and are described in (1):

$$\begin{aligned} Recall\ (True\ Positive\ Rate, TPR) &= \frac{TP}{TP+FN} \\ Precision &= \frac{TP}{TP+FP} \\ False\ Positive\ Rate, FPR &= \frac{FP}{FP+TN} \\ Error\ Rate &= \frac{FP+FN}{TP+FP+TN+FN} \\ F1_{score} &= \frac{2*TP}{2*TP+FP+FN} \end{aligned}$$

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

In the defect detection and segmentation, the Intersection over Union (IoU) measures the degree of overlap between the ground truth and the predicted bounding box or mask.

$$IoU = \frac{TP+TN}{TP+FP+FN} \quad (2)$$

The mean IoU (mIoU) is calculated as the average IoU across the K classes: $\frac{1}{K}\sum_{i=1}^{K} IoU_i$. The model binarizes the prediction probability (i.e. likelihood) for a defect presence through a threshold that balances the trade-off between recall and precision. Therefore, the threshold becomes an independent variable for the model decision and a curve can be drawn by calculating the corresponding metrics for each threshold value considered. In this context, the Average Precision (AP) represents the area under the precision-recall curve.

$$AP = \int_0^1 p\,(r)\ dr \quad (3)$$

The mean AP (mAP) is calculated as the average across the K classes: $\frac{1}{K}\sum_{i=1}^{K} AP_i$. By scanning the trade-off between TPR and FPR, the *Receiver Operating Characteristic Curve* (ROC) is determined. The AUROC/AUC indicator corresponds to the *area under* the ROC curve; high AUROC value indicates that the model performs accurately in identifying the anomalies whenever the selected threshold. It is worth mentioning that in defect recognition tasks, the high number of background pixels dominates on FPR, thus is frequent to have a high AUROC value despite many false positive detections [27]. Additionally, the Dice score is another commonly used performance indicator in segmentation tasks. It is computed by dividing the intersection between the predicted and true segmentations by their union, and ranges from 0 to 1, with higher values indicating better segmentation performance.

The evaluation of the efficiency in performing real-time recognition makes use of different metrics that measure model performances in terms of speed; we count the number of FLoating point Operations Per Second (FLOPs), number of evaluated Frames Per Second (FPS), and the inference time. While FLOPs measurement is independent of the hardware technology, the FPS and inference time are strictly correlated to the computational power of a calculator.

## ACKNOWLEDGMENT

The authors would like to thank the Comau S.p.A., Staff, specifically to the Developer Engineers Simone Panicucci and Enrico Civitelli, for their valuable and constructive comments for the preparation of this article and also would like to thank Vladimiro Suglia for his writing contribution in terms of grammatical revision.

*(Michela Prunella and Roberto Maria Scardigno contributed equally to this work.)*

## REFERENCES


[1] B. Mohammadi, M. Fathy, and M. Sabokrou, "Image/video deep anomaly detection: A survey," 2021, *arXiv:2103.01739*.

[2] J. Zhang, X. Kang, H. Ni, and F. Ren, "Surface defect detection of steel strips based on classification priority YOLOv3-dense network," *Ironmaking Steelmaking*, vol. 48, no. 5, pp. 547–558, May 2021, doi: 10.1080/03019233.2020.1816806.

[3] Y. Jing, H. Zheng, W. Zheng, and K. Dong, "A pixel-wise foreign object debris detection method based on multi-scale feature inpainting," *Aerospace*, vol. 9, no. 9, p. 480, Aug. 2022.

[4] B.-L. Jian, J.-P. Hung, C.-C. Wang, and C.-C. Liu, "Deep learning model for determining defects of vision inspection machine using only a few samples," *Sensors Mater.*, vol. 32, no. 12, pp. 4217–4231, 2020, doi: 10.18494/SAM.2020.3101.

[5] B. Wang, D. Dou, and N. Shen, "An intelligent belt wear fault diagnosis method based on deep learning," *Int. J. Coal Preparation Utilization*, vol. 43, no. 4, pp. 1–18, 2022, doi: 10.1080/19392699.2022.2072306.

[6] P. M. Bhatt, R. K. Malhan, P. Rajendran, B. C. Shah, S. Thakar, Y. J. Yoon, and S. K. Gupta, "Image-based surface defect detection using deep learning: A review," *J. Comput. Inf. Sci. Eng.*, vol. 21, no. 4, pp. 1–23, Aug. 2021.

[7] X. Dong, C. J. Taylor, and T. F. Cootes, "Defect classification and detection using a multitask deep one-class CNN," *IEEE Trans. Automat. Sci. Eng.*, vol. 19, no. 3, pp. 1719–1730, Jul. 2022.

[8] S.-J. Oh, M.-J. Jung, C. Lim, and S.-C. Shin, "Automatic detection of welding defects using faster R-CNN," *Appl. Sci.*, vol. 10, no. 23, p. 8629, Dec. 2020. [Online]. Available: https://www.mdpi.com/journal/applsci

[9] V. Natarajan, T.-Y. Hung, S. Vaikundam, and L.-T. Chia, "Convolutional networks for voting-based anomaly classification in metal surface inspection," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Mar. 2017, pp. 986–991.

[10] C. Hu and Y. Wang, "An efficient convolutional neural network model based on object-level attention mechanism for casting defect detection on radiography images," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10922–10930, Jan. 2020. [Online]. Available: https://www.ieee.org/publications/rights/index.html

[11] W. Wei, D. Deng, L. Zeng, and C. Zhang, "Real-time implementation of fabric defect detection based on variational automatic encoder with structure similarity," *J. Real-Time Image Process.*, vol. 18, no. 3, pp. 807–823, Jun. 2021, doi: 10.1007/s11554-020-01023-5.

[12] T. Ueno, Q. Zhao, and S. Nakada, "Deep learning-based industry product defect detection with low false negative error tolerance," in *Proc. 11th Int. Conf. Awareness Sci. Technol. (iCAST)*, 2020, pp. 1–6.

[13] M. Khanafer and S. Shirmohammadi, "Applied AI in instrumentation and measurement: The deep learning revolution," *IEEE Instrum. Meas. Mag.*, vol. 23, no. 6, pp. 10–17, Sep. 2020.

[14] M. Chu, R. Gong, S. Gao, and J. Zhao, "Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine," *Chemom. Intell. Lab. Syst.*, vol. 171, pp. 140–150, Sep. 2017.

[15] M. Aslam, T. M. Khan, S. S. Naqvi, G. Holmes, and R. Naffa, "Ensemble convolutional neural networks with knowledge transfer for leather defect classification in industrial settings," *IEEE Access*, vol. 8, pp. 198600–198614, 2020.

[16] D. Wang, Y. Xu, B. Duan, Y. Wang, M. Song, H. Yu, and H. Liu, "Intelligent recognition model of hot rolling strip edge defects based on deep learning," *Metals*, vol. 11, no. 2, pp. 1–17, 2021.

[17] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[18] R. Usamentiaga, D. G. Lema, O. D. Pedrayes, and D. F. Garcia, "Automated surface defect detection in metals: A comparative review of object detection and semantic segmentation using deep learning," *IEEE Trans. Ind. Appl.*, vol. 58, no. 3, pp. 4203–4213, May 2022.

[19] X. Sun, J. Gu, S. Tang, and J. Li, "Research progress of visual inspection technology of steel products—A review," *Appl. Sci.*, vol. 8, no. 11, p. 2195, Nov. 2018.

[20] A. Rasheed, B. Zafar, A. Rasheed, N. Ali, M. Sajid, S. H. Dar, U. Habib, T. Shehryar, and M. T. Mahmood, "Fabric defect detection using computer vision techniques: A comprehensive review," *Math. Problems Eng.*, vol. 2020, pp. 1–24, Nov. 2020.

[21] U. Batool, M. I. Shapiai, M. Tahir, Z. H. Ismail, N. J. Zakaria, and A. Elfakharany, "A systematic review of deep learning for silicon wafer defect recognition," *IEEE Access*, vol. 9, pp. 116572–116593, 2021.

[22] P. Dixit, P. Bhattacharya, S. Tanwar, and R. Gupta, "Anomaly detection in autonomous electric vehicles using AI techniques: A comprehensive survey," *Exp. Syst.*, vol. 39, no. 5, Jun. 2022, Art. no. e12754.

[23] T. Hong Chun, U. R. Hashim, S. Ahmad, L. Salahuddin, N. H. Choon, and K. Kanchymalay, "A review of the automated timber defect identification approach," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 13, no. 2, p. 2156, Apr. 2023.

[24] Z. Chen, J. Deng, Q. Zhu, H. Wang, and Y. Chen, "A systematic review of machine-vision-based leather surface defect inspection," *Electronics*, vol. 11, no. 15, pp. 1–27, 2022.

[25] M. Aslam, T. M. Khan, S. S. Naqvi, G. Holmes, and R. Naffa, "On the application of automated machine vision for leather defect inspection and grading: A survey," *IEEE Access*, vol. 7, pp. 176065–176086, 2019.

[26] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, "Fabric defect detection in textile manufacturing: A survey of the state of the art," *Secur. Commun. Netw.*, vol. 2021, pp. 1–13, May 2021.

[27] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep learning for unsupervised anomaly localization in industrial images: A survey," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–21, 2022.

[28] X. He, Z. Chang, L. Zhang, H. Xu, H. Chen, and Z. Luo, "A survey of defect detection applications based on generative adversarial networks," *IEEE Access*, vol. 10, pp. 113493–113512, 2022.

[29] B. Maschler and M. Weyrich, "Deep transfer learning for industrial automation: A review and discussion of new techniques for data-driven machine learning," *IEEE Ind. Electron. Mag.*, vol. 15, no. 2, pp. 65–75, Jun. 2021.

[30] Q. Jin and L. Chen, "A survey of surface defect detection of industrial products based on a small number of labeled data," 2022, *arXiv:2203.05733*.

[31] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. M. Oddo, and P. Dario, "Visual-based defect detection and classification approaches for industrial applications—A survey," *Sensors*, vol. 20, no. 5, pp. 1–25, 2020.

[32] X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *Int. J. Adv. Manuf. Technol.*, vol. 113, nos. 1–2, pp. 35–58, Mar. 2021.

[33] Z. Ren, F. Fang, N. Yan, and Y. Wu, "State of the art in defect detection based on machine vision," *Int. J. Precis. Eng. Manuf.-Green Technol.*, vol. 9, no. 2, pp. 661–691, 2021, doi: 10.1007/s40684-021-00343-6.

[34] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao, "Surface defect detection methods for industrial products: A review," *Appl. Sci.*, vol. 11, no. 16, p. 7657, Aug. 2021.

[35] J. Yang, R. Xu, Z. Qi, and Y. Shi, "Visual anomaly detection for images: A survey," 2021, *arXiv:2109.13157*.

[36] Y. Gao, X. Li, X. V. Wang, L. Wang, and L. Gao, "A review on recent advances in vision-based defect recognition towards industrial intelligence," *J. Manuf. Syst.*, vol. 62, pp. 753–766, Jan. 2022.

[37] C. Chen, A. Abdullah, S. H. Kok, and D. T. K. Tien, "Review of industry workpiece classification and defect detection using deep learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 329–340, 2022.

[38] X. Cheng, J. K. Chaw, K. M. Goh, T. T. Ting, S. Sahrani, M. N. Ahmad, R. A. Kadir, and M. C. Ang, "Systematic literature review on visual analytics of predictive maintenance in the manufacturing industry," *Sensors*, vol. 22, no. 17, pp. 1–16, 2022.

[39] V. Sampath, I. Maurtua, J. J. A. Martin, A. Rivera, J. Molina, and A. Gutierrez, "Attention guided multi-task learning for surface defect identification," *IEEE Trans. Ind. Informat.*, early access, Jan. 4, 2023, doi: 10.1109/TII.2023.3234030.

[40] M. J. Page, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, no. 1, pp. 1–10, Dec. 2021. [Online]. Available: https://www.bmj.com/content/372/bmj.n71

[41] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Ann.-Manuf. Technol.*, vol. 65, no. 1, pp. 417–420, 2016.

[42] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 929–940, Mar. 2018.

[43] X. Tao, D. Zhang, W. Hou, W. Ma, and D. Xu, "Industrial weak scratches inspection based on multifeature fusion network," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.

[44] P. Lu, J. Jing, and Y. Huang, "MRD-Net: An effective CNN-based segmentation network for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
[45] T. Niu, B. Li, W. Li, Y. Qiu, and S. Niu, "Positive-sample-based surface defect detection using memory-augmented adversarial autoencoders," *IEEE/ASME Trans. Mechtron.*, vol. 27, pp. 46–57, 2021.
[46] B. Yang, Z. Liu, G. Duan, and J. Tan, "Mask2Defect: A prior knowledge-based data augmentation method for metal surface defect inspection," *IEEE Trans. Ind. Informat.*, vol. 18, no. 10, pp. 6743–6755, Oct. 2022, doi: 10.1109/TII.2021.3126098.
[47] S. Yang, W. Xiao, M. Zhang, S. Guo, J. Zhao, and F. Shen, "Image data augmentation for deep learning: A survey," 2022, *arXiv:2204.08610*.
[48] V. Sampath, I. Maurtua, J. J. A. Martín, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," *J. Big Data*, vol. 8, no. 1, pp. 1–59, Dec. 2021.
[49] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, Dec. 2019.
[50] L. Chen, Z. You, N. Zhang, J. Xi, and X. Le, "UTRAD: Anomaly detection and localization with U-Transformer," *Neural Netw.*, vol. 147, pp. 53–62, Mar. 2022.
[51] A. M. Nagy and L. Czúni, "Classification and fast few-shot learning of steel surface defects with randomized network," *Appl. Sci.*, vol. 12, no. 8, p. 3967, Apr. 2022.
[52] R. Yu, B. Guo, and K. Yang, "Selective prototype network for few-shot metal surface defect segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[53] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1038–1059, Apr. 2021.
[54] K. Yao, A. Ortiz, and F. Bonnin-Pascual, "A weakly-supervised semantic segmentation approach based on the centroid loss: Application to quality control and inspection," *IEEE Access*, vol. 9, pp. 69010–69026, 2021.
[55] Z. Xu, D. Lu, J. Luo, Y. Wang, J. Yan, K. Ma, Y. Zheng, and R. K.-Y. Tong, "Anti-interference from noisy labels: Mean-teacher-assisted confident learning for medical image segmentation," *IEEE Trans. Med. Imag.*, vol. 41, no. 11, pp. 3062–3073, Nov. 2022.
[56] T. Niu, B. Li, K. Li, Y. Lin, Y. Li, W. Li, and Z. Wang, "Learning trustworthy model from noisy labels based on rough set for surface defect detection," 2023, *arXiv:2301.10441*.
[57] J. Li, Y. Wong, Q. Zhao, and M. Kankanhalli, "Learning to learn from noisy labeled data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Mar. 2018, pp. 5046–5054.
[58] Y. Wang, Y. Zhang, Z. Jiang, L. Zheng, J. Chen, and J. Lu, "Robust learning against label noise based on activation trend tracking," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
[59] D. F. N. Oliveira, L. F. Vismari, A. M. Nascimento, J. R. de Almeida, P. S. Cugnasca, J. B. Camargo, L. Almeida, R. Gripp, and M. Neves, "A new interpretable unsupervised anomaly detection method based on residual explanation," *IEEE Access*, vol. 10, pp. 1401–1409, 2022.
[60] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya, "A survey on addressing high-class imbalance in big data," *J. Big Data*, vol. 5, no. 1, pp. 1–30, 2018.
[61] D. Zhang, X. Hao, D. Wang, C. Qin, B. Zhao, L. Liang, and W. Liu, "An efficient lightweight convolutional neural network for industrial surface defect detection," *Artif. Intell. Rev.*, Mar. 2023, Art. no. 0123456789, doi: 10.1007/s10462-023-10438-y.
[62] Y. Liang, J. Li, J. Zhu, R. Du, X. Wu, and B. Chen, "A lightweight network for defect detection in nickel-plated punched steel strip images," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–15, 2023.
[63] L. Yang, J. Fan, B. Huo, and Y. Liu, "Inspection of welding defect based on multi-feature fusion and a convolutional network," *J. Nondestruct. Eval.*, vol. 40, no. 4, pp. 1–11, Dec. 2021, doi: 10.1007/s10921-021-00823-4.
[64] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.
[65] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated defect inspection system for metal surfaces based on deep learning and data augmentation," *J. Manuf. Syst.*, vol. 55, pp. 317–324, Apr. 2020, doi: 10.1016/j.jmsy.2020.03.009.
[66] Y. Li, M. Yang, J. Hua, Z. Xu, J. Wang, and X. Fang, "A channel attention-based method for micro-motor armature surface defect detection," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8672–8684, May 2022. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[67] S. Woo, J. Park, J.-Y. Lee, and I.-S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.
[68] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2017.
[69] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[70] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[72] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.
[73] P. Martinez, M. Al-Hussein, and R. Ahmad, "Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing," *Int. J. Adv. Manuf. Technol.*, vol. 109, nos. 3–4, pp. 645–657, Jul. 2020.
[74] I. D. Apostolopoulos and M. A. Tzani, "Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with deep learning approach," *J. Ambient Intell. Humanized Comput.*, vol. 2022, pp. 1–14, Jan. 2022, doi: 10.1007/s12652-021-03688-7.
[75] Y. Liu, Y. Yuan, C. Balta, and J. Liu, "A light-weight deep-learning model with multi-scale features for steel surface defect classification," *Materials*, vol. 13, no. 20, p. 4629, Oct. 2020. [Online]. Available: https://www.mdpi.com/journal/materials
[76] Y. Dong, J. Wang, C. Li, Z. Liu, J. Xi, and A. Zhang, "Fusing multi-level deep features for fabric defect detection based NTV-RPCA," *IEEE Access*, vol. 8, pp. 161872–161883, 2020.
[77] X. Yu, S. Member, W. Lyu, D. Zhou, C. Wang, W. Xu, S. Member, and W. L. X. Yu, "ES-Net: Efficient scale-aware network for tiny defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[78] Y. Yan, D. Wang, G. Zhou, Q. Chen, and S. Member, "Unsupervised anomaly segmentation via multilevel image reconstruction and adaptive attention-level transition," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5015712. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[79] B. Li, Y. Zou, R. Zhu, W. Yao, J. Wang, and S. Wan, "Fabric defect segmentation system based on a lightweight GAN for industrial Internet of Things," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–17, May 2022, doi: 10.1155/2022/9680519.
[80] Z. Yang, M. Zhang, Y. Chen, N. Hu, L. Gao, L. Liu, E. Ping, and J. I. Song, "Surface defect detection method for air rudder based on positive samples," *J. Intell. Manuf.*, vol. 2022, pp. 1–19, Oct. 2022, doi: 10.1007/s10845-022-02034-8.
[81] S. Wang, X. Xia, L. Ye, and B. Yang, "Automatic detection and classification of steel surface defect using deep convolutional neural networks," *Metals*, vol. 11, no. 3, pp. 1–23, 2021.
[82] S. Kim, Y.-K. Noh, and F. C. Park, "Efficient neural network compression via transfer learning for machine vision inspection," *Neurocomputing*, vol. 413, pp. 294–304, Nov. 2020.
[83] S. A. Singh and K. A. Desai, "Automated surface defect detection framework using machine vision and convolutional neural networks," *J. Intell. Manuf.*, vol. 34, pp. 1–17, 2022, doi: 10.1007/s10845-021-01878-w.
[84] R. Sekhar, D. Sharma, and P. Shah, "Intelligent classification of tungsten inert gas welding defects: A transfer learning approach," *Frontiers Mech. Eng.*, vol. 8, p. 20, Mar. 2022.
[85] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with GAN for improving defect recognition," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 13, pp. 1–12, 2020.
[86] I. M. Kamal, R. A. Sutrisnowati, H. Bae, and T. Lim, "Gear classification for defect detection in vision inspection system using deep convolutional neural networks," *ICIC Exp. Lett., B, Appl.*, vol. 9, no. 12, pp. 1279–1286, 2018.

[87] D. Mittel and F. Kerber, "Vision-based crack detection using transfer learning in metal forming processes," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, Sep. 2019, pp. 544–551.

[88] X. Xu, H. Zheng, Z. Guo, X. Wu, and Z. Zheng, "SDD-CNN: Small data-driven convolution neural networks for subtle roller defect inspection," *Appl. Sci.*, vol. 9, no. 7, p. 1364, Mar. 2019.

[89] Y. Gong, J. Luo, H. Shao, K. He, and W. Zeng, "Automatic defect detection for small metal cylindrical shell using transfer learning and logistic regression," *J. Nondestruct. Eval.*, vol. 39, no. 1, p. 24, Mar. 2020, doi: 10.1007/s10921-020-0668-4.

[90] D. Mery, "Aluminum casting inspection using deep learning: A method based on convolutional neural networks," *J. Nondestruct. Eval.*, vol. 39, no. 1, pp. 1–12, Mar. 2020, doi: 10.1007/s10921-020-0655-9.

[91] K. Zhang and H. Shen, "Solder joint defect detection in the connectors using improved faster-RCNN algorithm," *Appl. Sci.*, vol. 11, no. 2, p. 576, Jan. 2021, doi: 10.3390/app11020576.

[92] P. Sassi, P. Tripicchio, and C. A. Avizzano, "A smart monitoring system for automatic welding defect detection," *IEEE Trans. Ind. Electron.*, vol. 66, no. 12, pp. 9641–9650, Dec. 2019. [Online]. Available: http://ieeexplore.ieee.org

[93] R. E. Sarpietro, C. Pino, S. Coffa, A. Messina, S. Palazzo, S. Battiato, C. Spampinato, and F. Rundo, "Explainable deep learning system for advanced silicon and silicon carbide electrical wafer defect map assessment," *IEEE Access*, vol. 4, pp. 99102–99128, 2016.

[94] S. Wang, H. Wang, F. Yang, F. Liu, and L. Zeng, "Attention-based deep learning for chip-surface-defect detection," *Int. J. Adv. Manuf. Technol.*, vol. 121, nos. 3–4, pp. 1957–1971, Jul. 2022, doi: 10.1007/s00170-022-09425-4.

[95] J. Liu, F. Guo, H. Gao, M. Li, Y. Zhang, and H. Zhou, "Defect detection of injection molding products on small datasets using transfer learning," *J. Manuf. Processes*, vol. 70, pp. 400–413, Oct. 2021.

[96] J.-K. Park, W.-H. An, and D.-J. Kang, "Convolutional neural network based surface inspection system for non-patterned welding defects," *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 3, pp. 363–374, 2019, doi: 10.1007/s12541-019-00074-4.

[97] B. Staar, M. Lütjen, and M. Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Proc. CIRP*, vol. 79, pp. 484–489, Jan. 2019, doi: 10.1016/j.procir.2019.02.123.

[98] S. Yang, X. Li, X. Jia, Y. Wang, H. Zhao, and J. Lee, "Deep learning-based intelligent defect detection of cutting wheels with industrial images in manufacturing," *Proc. Manuf.*, vol. 48, pp. 902–907, Jan. 2020, doi: 10.1016/j.promfg.2020.05.128.

[99] Q. Lv and Y. Song, "Few-shot learning combine attention mechanism-based defect detection in bar surface," *ISIJ Int.*, vol. 59, no. 6, pp. 1089–1097, 2019.

[100] Z. Hao, Z. Li, F. Ren, S. Lv, and H. Ni, "Strip steel surface defects classification based on generative adversarial network and attention mechanism," *Metals*, vol. 12, no. 2, p. 311, Feb. 2022. [Online]. Available: https://www.mdpi.com/2075-4701/12/2/311/htm

[101] W. Dai, D. Li, D. Tang, H. Wang, and Y. Peng, "Deep learning approach for defective spot welds classification using small and class-imbalanced datasets," *Neurocomputing*, vol. 477, pp. 46–60, Mar. 2022.

[102] H. Gao, Y. Zhang, W. Lv, J. Yin, T. Qasim, and D. Wang, "A deep convolutional generative adversarial networks-based method for defect detection in small sample industrial parts images," *Appl. Sci.*, vol. 12, no. 13, p. 6569, Jun. 2022.

[103] R. Arandjelović and A. Zisserman, "Object discovery with a copy-pasting GAN," 2019, *arXiv:1905.11369*.

[104] M. W. Hridoy, Mohammad, M. Rahman, S. Sakib, M. M. Rahman, and S. Sakib, "A framework for industrial inspection system using deep learning," *Ann. Data Sci.*, doi: 10.1007/s40745-022-00437-1.

[105] S. A. Althubiti, F. Alenezi, S. Shitharth, K. Sangeetha, and C. V. S. Reddy, "Circuit manufacturing defect detection using VGG16 convolutional neural networks," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–10, Apr. 2022.

[106] S. Perri, F. Spagnolo, F. Frustaci, and P. Corsonello, "Welding defects classification through a convolutional neural network," *Manuf. Lett.*, vol. 35, pp. 29–32, Jan. 2023, doi: 10.1016/j.mfglet.2022.11.006.

[107] L. Ma, W. Xie, and Y. Zhang, "Blister defect detection based on convolutional neural network for polymer lithium-ion battery," *Appl. Sci.*, vol. 9, no. 6, p. 1085, Mar. 2019.

[108] Z. Zhang, B. Zhang, T. Akiduki, T. Mashimo, and T. Yu, "Research on surface defects detection of reflected curved surface based on convolutional neural networks," *ICIC Exp. Lett., B, Appl.*, vol. 10, no. 7, pp. 627–634, 2019.

[109] K. Su, Q. Zhao, and P. C. Lien, "Product surface defect detection based on CNN ensemble with rejection," in *Proc. IEEE 17th Int. Conf. Dependable, Autonomic Secure Comput., IEEE 17th Int. Conf. Pervasive Intell. Comput., IEEE 5th Int. Conf. Cloud Big Data Comput., 4th Cyber Scienc*, Aug. 2019, pp. 326–331.

[110] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: AlexNet-level accuracy with 50x fewer parameters and < 1 MB model size," 2016, *arXiv:1602.07360*.

[111] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. And Pattern Recognit.*, Oct. 2015, pp. 2818–2826.

[112] X. Le, J. Mei, H. Zhang, B. Zhou, and J. Xi, "A learning-based approach for surface defect detection using small image datasets," *Neurocomputing*, vol. 408, pp. 112–120, Sep. 2020.

[113] D. Buongiorno, M. Prunella, S. Grossi, S. M. Hussain, A. Rennola, N. Longo, G. Di Stefano, V. Bevilacqua, and A. Brunetti, "Inline defective laser weld identification by processing thermal image sequences with machine and deep learning techniques," *Appl. Sci.*, vol. 12, no. 13, p. 6455, Jun. 2022.

[114] M. Bhagat and B. Bakariya, "Implementation of logistic regression on diabetic dataset using train-test-split, K-fold and stratified K-fold approach," *Nat. Acad. Sci. Lett.*, vol. 45, no. 5, pp. 401–404, Oct. 2022.

[115] A. Birlutiu, A. Burlacu, M. Kadar, and D. Onita, "Defect detection in porcelain industry based on deep learning techniques," in *Proc. 19th Int. Symp. Symbolic Numeric Algorithms Sci. Comput. (SYNASC)*, Sep. 2017, pp. 263–270.

[116] T.-W. Tang, W.-H. Kuo, J.-H. Lan, C.-F. Ding, H. Hsu, and H.-T. Young, "Anomaly detection neural network with dual auto-encoders GAN and its industrial inspection applications," *Sensors*, vol. 20, no. 12, p. 3336, Jun. 2020.

[117] V. Nath, C. Chattopadhyay, and K. A. Desai, "NSLNet: An improved deep learning model for steel surface defect classification utilizing small training datasets," *Manuf. Lett.*, vol. 35, pp. 39–42, Jan. 2023, doi: 10.1016/j.mfglet.2022.10.001.

[118] Y. Yang, L. Pan, J. Ma, R. Yang, Y. Zhu, Y. Yang, and L. Zhang, "A high-performance deep learning algorithm for the automated optical inspection of laser welding," *Appl. Sci.*, vol. 10, no. 3, p. 933, Jan. 2020. [Online]. Available: https://www.mdpi.com/journal/applsci

[119] S. Y. Lee, B. A. Tama, S. J. Moon, and S. Lee, "Steel surface defect diagnostics using deep convolutional neural network and class activation map," *Appl. Sci.*, vol. 9, no. 24, p. 5449, Dec. 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/24/5449/htm

[120] I. Konovalenko, P. Maruschak, V. Brevus, and O. Prentkovskis, "Recognition of scratches and abrasions on metal surfaces using a classifier based on a convolutional neural network," *Metals*, vol. 11, no. 4, p. 549, Mar. 2021. [Online]. Available: https://www.mdpi.com/2075-4701/11/4/549/htm

[121] C. Xia, Z. Pan, Z. Fei, S. Zhang, and H. Li, "Vision based defects detection for keyhole TIG welding using deep learning with visual explanation," *J. Manuf. Processes*, vol. 56, pp. 845–855, Aug. 2020, doi: 10.1016/j.jmapro.2020.05.033.

[122] Y. Shih, C.-C. Kuo, and C.-H. Lee, "Low-cost real-time automated optical inspection using deep learning and attention map," *Intell. Autom. Soft Comput.*, vol. 35, no. 2, pp. 2087–2099, 2023.

[123] H. Yang, K. Song, F. Mao, and Z. Yin, "Autolabeling-enhanced active learning for cost-efficient surface defect visual classification," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–15, 2021.

[124] J. Liu, F. Guo, Y. Zhang, B. Hou, and H. Zhou, "Defect classification on limited labeled samples with multiscale feature fusion and semi-supervised learning," *Int. J. Speech Technol.*, vol. 52, no. 7, pp. 8243–8258, May 2022, doi: 10.1007/s10489-021-02917-y.

[125] T. Liu and W. Ye, "A semi-supervised learning method for surface defect classification of magnetic tiles," *Mach. Vis. Appl.*, vol. 33, no. 2, Mar. 2022, doi: 10.1007/s00138-022-01286-x.

[126] H. Di, X. Ke, Z. Peng, and Z. Dongdong, "Surface defect classification of steels with a new semi-supervised learning method," *Opt. Lasers Eng.*, vol. 117, pp. 40–48, Jun. 2019.

[127] J. J. A. Kovilpillai and S. Jayanthy, "An optimized deep learning approach to detect and classify defective tiles in production line for efficient industrial quality control," *Neural Comput. Appl.*, vol. 35, no. 15, pp. 11089–11108, May 2023.
[128] Y. Shi, L. Li, J. Yang, Y. Wang, and S. Hao, "Center-based transfer feature learning with classifier adaptation for surface defect recognition," *Mech. Syst. Signal Process.*, vol. 188, Apr. 2023, Art. no. 110001.
[129] K. Wang, Z. Li, and X. Wang, "Concatenated network fusion algorithm (CNFA) based on deep learning: Improving the detection accuracy of surface defects for ceramic tile," *Appl. Sci.*, vol. 12, no. 3, p. 1249, Jan. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1249/htm
[130] P. Kostenetskiy, R. Alkapov, N. Vetoshkin, R. Chulkevich, I. Napolskikh, and O. Poponin, "Real-time system for automatic cold strip surface defect detection," *FME Trans.*, vol. 47, no. 4, pp. 765–774, 2019.
[131] Y. Jiang, W. Wang, and C. Zhao, "A machine vision-based realtime anomaly detection method for industrial products using deep learning," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 4842–4847.
[132] P. C. Lien and Q. Zhao, "Product surface defect detection based on deep learning," in *Proc. IEEE 16th Int. Conf Dependable, Autonomic Secure Comput., 16th Int. Conf Pervasive Intell. Comput., 4th Int. Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Oct. 2018, pp. 256–261.
[133] Y. Zhao, J. Li, Q. Zhang, C. Lian, P. Shan, C. Yu, Z. Jiang, and Z. Qiu, "Simultaneous detection of defects in electrical connectors based on improved convolutional neural network," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 3511710. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[134] K. Arima, F. Nagata, T. Shimizu, K. Miki, H. Kato, A. Otuka, and K. Watanabe, "Visualization and location estimation of defective parts of industrial products using convolutional autoencoder," *Artif. Life Robot.*, vol. 27, no. 4, pp. 804–811, Nov. 2022, doi: 10.1007/s10015-022-00797-0.
[135] B. A. U. Olimov, K. C. Veluvolu, A. Paul, and J. Kim, "UzADL: Anomaly detection and localization using graph Laplacian matrix-based unsupervised learning method," *Comput. Ind. Eng.*, vol. 171, Sep. 2022, Art. no. 108313.
[136] S. Song, K. Yang, A. Wang, S. Zhang, and M. Xia, "A Mura detection model based on unsupervised adversarial learning," *IEEE Access*, vol. 9, pp. 49920–49928, 2021.
[137] S. Park and J. H. Ko, "Robust inspection of micro-LED chip defects using unsupervised anomaly detection," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2021, pp. 1841–1843.
[138] J. Liu, K. Song, M. Feng, Y. Yan, Z. Tu, and L. Zhu, "Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection," *Opt. Lasers Eng.*, vol. 136, Jan. 2021, Art. no. 106324.
[139] N. Ishida, Y. Nagatsu, and H. Hashimoto, "Unsupervised anomaly detection based on data augmentation and mixing," in *Proc. IECON 46th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2020, pp. 529–533.
[140] J. Li, X. Xu, L. Gao, Z. Wang, and J. Shao, "Cognitive visual anomaly detection with constrained latent representations for industrial inspection robot," *Appl. Soft Comput.*, vol. 95, Oct. 2020, Art. no. 106539.
[141] L. Song, X. Li, Y. Yang, X. Zhu, Q. Guo, and H. Yang, "Detection of micro-defects on metal screw surfaces based on deep convolutional neural networks," *Sensors*, vol. 18, no. 11, p. 3709, Oct. 2018.
[142] Y. T. Lai, J. S. Hu, Y. H. Tsai, and W. Y. Chiu, "Industrial anomaly detection and one-class classification using generative adversarial networks," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechtron. (AIM)*, Jul. 2018, pp. 1444–1449.
[143] M. Zhang, J. Wu, H. Lin, P. Yuan, and Y. Song, "The application of one-class classifier based on CNN in image defect detection," *Proc. Comput. Sci.*, vol. 114, pp. 341–348, Jan. 2017, doi: 10.1016/j.procs.2017.09.040.
[144] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
[145] R. Neven and T. Goedemé, "A multi-branch U-Net for steel surface defect type and severity segmentation," *Metals*, vol. 11, no. 6, p. 870, May 2021.
[146] Y. Yang, Y. He, H. Guo, Z. Chen, and L. Zhang, "Semantic segmentation supervised deep-learning algorithm for welding-defect detection of new energy batteries," *Neural Comput. Appl.*, vol. 34, no. 22, pp. 19471–19484, Nov. 2022, doi: 10.1007/s00521-022-07474-0.
[147] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, "PGA-Net: Pyramid feature fusion and global context attention network for automated surface defect detection," *IEEE Trans. Ind. Informat.*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020.
[148] C.-C. Ho, M. A. B. Hernández, Y.-F. Chen, C.-J. Lin, and C.-S. Chen, "Deep residual neural network-based defect detection on complex backgrounds," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[149] K. Chen, N. Cai, Z. Wu, H. Xia, S. Zhou, and H. Wang, "Multi-scale GAN with transformer for surface defect inspection of IC metal packages," *Exp. Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118788, doi: 10.1016/j.eswa.2022.118788.
[150] L. Cheng, J. Yi, A. Chen, and Y. Zhang, "Fabric defect detection based on separate convolutional UNet," *Multimedia Tools Appl.*, vol. 82, no. 2, pp. 3101–3122, Jan. 2022.
[151] D. Tabernik, S. Šela, J. Skvarč, and D. Skocaj, "Segmentation-based deep-learning approach for surface-defect detection," *J. Intell. Manuf.*, vol. 31, no. 3, pp. 759–776, Jun. 2019.
[152] B. Lu, D. Xu, and B. Huang, "Deep-learning-based anomaly detection for lace defect inspection employing videos in production line," *Adv. Eng. Informat.*, vol. 51, Jan. 2022, Art. no. 101471.
[153] A. Djavadifar, J. B. Graham-Knight, M. Korber, P. Lasserre, and H. Najjaran, "Automated visual detection of geometrical defects in composite manufacturing processes using deep convolutional neural networks," *J. Intell. Manuf.*, vol. 33, no. 8, pp. 2257–2275, Dec. 2022.
[154] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.
[155] X. Dong, C. J. Taylor, and T. F. Cootes, "Defect detection and classification by training a generic convolutional neural network encoder," *IEEE Trans. Signal Process.*, vol. 68, pp. 6055–6069, 2020.
[156] P. Damacharla, A. Rao, J. Ringenberg, and A. Y. Javaid, "TLU-Net: A deep learning approach for automatic steel surface defect detection," 2021, *arXiv:2101.06915*.
[157] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Appl. Sci.-Basel*, vol. 8, no. 9, pp. 1–15, Sep. 2018.
[158] X. Luo, S. Li, Y. Wang, T. Zhan, X. Shi, and B. Liu, "MaMiNet: Memory-attended multi-inference network for surface-defect detection," *Comput. Ind.*, vol. 145, Feb. 2023, Art. no. 103834, doi: 10.1016/j.compind.2022.103834.
[159] S. Niu, B. Li, X. Wang, and Y. Peng, "Region- and strength-controllable GAN for defect generation and segmentation in industrial images," *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4531–4541, Jul. 2022, doi: 10.1109/TII.2021.3127188.
[160] Q. Lin, J. Zhou, Q. Ma, Y. Ma, L. Kang, and J. Wang, "EMRA-Net: A pixel-wise network fusing local and global features for tiny and low-contrast surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[161] S. Niu, Y. Peng, B. Li, Y. Qiu, T. Niu, and W. Li, "A novel deep learning motivated data augmentation system based on defect segmentation requirements," *J. Intell. Manuf.*, Jan. 2023.
[162] J. Cao, G. Yang, and X. Yang, "A pixel-level segmentation convolutional neural network based on deep feature fusion for surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.
[163] H. Yang, Y. Chen, K. Song, and Z. Yin, "Multiscale feature-clustering-based fully convolutional autoencoder for fast accurate visual inspection of texture surface defects," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1450–1467, Jul. 2019.
[164] H. Üzen, M. TürkoE8lu, B. Yanikoglu, and D. Hanbay, "Swin-MFINet: Swin transformer based multi-feature integration network for detection of pixel-level surface defects," *Exp. Syst. Appl.*, vol. 209, Dec. 2022, Art. no. 118269.
[165] L. Yang, S. Song, J. Fan, B. Huo, E. Li, and Y. Liu, "An automatic deep segmentation network for pixel-level welding defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[166] Z. Hao, Z. Wang, D. Bai, B. Tao, X. Tong, and B. Chen, "Intelligent detection of steel defects based on improved split attention networks," *Frontiers Bioeng. Biotechnol.*, vol. 9, p. 1478, Jan. 2022.
[167] G. Liu, N. Yang, and L. Guo, "An attention-based network for textured surface anomaly detection," *Appl. Sci.*, vol. 10, no. 18, p. 6215, Sep. 2020.
[168] H. Zhang, Y. Chen, B. Liu, X. Guan, and X. Le, "Soft matching network with application to defect inspection," *Knowl.-Based Syst.*, vol. 225, Aug. 2021, Art. no. 107045.
[169] Z. Li, J. Li, and W. Dai, "A two-stage multiscale residual attention network for light guide plate defect detection," *IEEE Access*, vol. 9, pp. 2780–2792, 2021.

[170] Q. Wan, L. Gao, X. Li, and L. Wen, "Industrial image anomaly localization based on Gaussian clustering of pretrained feature," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6182–6192, Jun. 2022, doi: 10.1109/TIE.2021.3094452.

[171] X. Wu, T. Wang, Y. Li, P. Li, and Y. Liu, "A CAM-based weakly supervised method for surface defect inspection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[172] L. Xu, S. Lv, Y. Deng, and X. Li, "A weakly supervised surface defect detection based on convolutional neural network," *IEEE Access*, vol. 8, pp. 42285–42296, 2020.

[173] H. Chen, Q. Hu, B. Zhai, H. Chen, and K. Liu, "A robust weakly supervised learning of deep Conv-Nets for surface defect inspection," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11229–11244, Aug. 2020, doi: 10.1007/s00521-020-04819-5.

[174] S. Niu, B. Li, X. Wang, S. He, and Y. Peng, "Defect attention template generation cycleGAN for weakly supervised surface defect segmentation," *Pattern Recognit.*, vol. 123, Mar. 2022, Art. no. 108396.

[175] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017, *arXiv:1703.10593*.

[176] M. Ye, W. Zhang, G. Cui, and X. Wang, "Surface defects inspection of cylindrical metal workpieces based on weakly supervised learning," *Int. J. Adv. Manuf. Technol.*, vol. 119, nos. 3–4, pp. 1933–1949, Mar. 2022. [Online]. Available: https://www.researchgate.net/publication/352337746SurfaceDefectsInspectionofCylindricalMetalWorkpiecesBasedonWeaklySupervisedLearning

[177] L. Shao, E. Zhang, Q. Ma, and M. Li, "Pixel-wise semisupervised fabric defect detection method combined with multitask mean teacher," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[178] X. Zheng, H. Wang, J. Chen, Y. Kong, and S. Zheng, "A generic semi-supervised deep learning-based approach for automated surface inspection," *IEEE Access*, vol. 8, pp. 114088–114099, 2020.

[179] D. Lin, Y. Li, S. Prasad, T. L. Nwe, S. Dong, and Z. M. Oo, "CAM-guided multi-path decoding U-Net with triplet feature regularization for defect detection and segmentation," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107272.

[180] J. Božič, D. Tabernik, and D. Skočaj, "Mixed supervision for surface-defect detection: From weakly to fully supervised learning," *Comput. Ind.*, vol. 129, Aug. 2021, Art. no. 103459.

[181] B. Hu, X. Wang, and W. Yu, "Joint weakly and fully supervised learning for surface defect segmentation from images," *Signal Process., Image Commun.*, vol. 107, Sep. 2022, Art. no. 116807, doi: 10.1016/j.image.2022.116807.

[182] J. Wang, G. Xu, F. Yan, J. Wang, and Z. Wang, "Defect transformer: An efficient hybrid transformer architecture for surface defect detection," *Measurement*, vol. 211, Apr. 2023, Art. no. 112614, doi: 10.1016/j.measurement.2023.112614.

[183] E. Branikas, P. Murray, and G. West, "A novel data augmentation method for improved visual crack detection using generative adversarial networks," *IEEE Access*, vol. 11, pp. 22051–22059, 2023.

[184] C. Nowroth, T. Gu, J. Grajczak, S. Nothdurft, J. Twiefel, J. Hermsdorf, S. Kaierle, and J. Wallaschek, "Deep learning-based weld contour and defect detection from micrographs of laser beam welded semi-finished products," *Appl. Sci.*, vol. 12, no. 9, p. 4645, May 2022.

[185] T. Tyystjärvi, I. Virkkunen, P. Fridolf, A. Rosell, and Z. Barsoum, "Automated defect detection in digital radiography of aerospace welds using deep learning," *Weld. World*, vol. 66, no. 4, pp. 643–671, Apr. 2022.

[186] H. A. Gabbar, A. Chahid, M. J. A. Khan, O. G. Adegboro, and M. I. Samson, "CTIMS: Automated defect detection framework using computed tomography," *Appl. Sci.*, vol. 12, no. 4, p. 2175, Feb. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/4/2175/htm

[187] Y. Meng, H. Xu, Z. Ma, J. Zhou, and D. Hui, "Detail-semantic guide network based on spatial attention for surface defect detection with fewer samples," *Int. J. Speech Technol.*, vol. 53, no. 6, pp. 7022–7040, Mar. 2022, doi: 10.1007/s10489-022-03671-5.

[188] C. Zhang, J. Cui, and W. Liu, "Multilayer feature extraction of AGCN on surface defect detection of steel plates," *Comput. Intell. Neurosci.*, vol. 2022, Oct. 2022, Art. no. 2549683.

[189] X. Xie, R. Zhang, L. Peng, and S. Peng, "A four-stage product appearance defect detection method with small samples," *IEEE Access*, vol. 10, pp. 83740–83754, 2022.

[190] M. Abu, A. Amir, Y. H. Lean, N. A. H. Zahri, and S. A. Azemi, "The performance analysis of transfer learning for steel defect detection by using deep learning," *J. Phys., Conf.*, vol. 1755, no. 1, Feb. 2021, Art. no. 012041.

[191] C. Lile and L. Yiqun, "Anomaly detection in thermal images using deep neural networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2299–2303.

[192] Q. Zhou, H. Wang, Y. Tang, and Y. Wang, "Defect detection method based on knowledge distillation," *IEEE Access*, vol. 11, pp. 35866–35873, 2023.

[193] Y. Huang and Z. Xiang, "RPDNet: Automatic fabric defect detection based on a convolutional neural network and repeated pattern analysis," *Sensors*, vol. 22, no. 16, p. 6226, Aug. 2022. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/36015986/

[194] I. Y. Moon, H. W. Lee, S.-J. Kim, Y.-S. Oh, J. Jung, and S.-H. Kang, "Analysis of the region of interest according to CNN structure in hierarchical pattern surface inspection using CAM," *Materials*, vol. 14, no. 9, p. 2095, Apr. 2021, doi: 10.3390/ma14092095.

[195] L. Zhu, D. Baolin, Z. Xiaomeng, F. Shaoliang, C. Zhen, Z. Junjie, and C. Shumin, "Surface defect detection method based on improved semisupervised multitask generative adversarial network," *Scientific Program.*, vol. 2022, pp. 1–17, Jan. 2022, doi: 10.1155/2022/4481495.

[196] Z. Zhang, C. Lv, M. Sun, and Z. Wang, "Reliable and robust weakly supervised attention networks for surface defect detection," in *Proc. 7th Int. Conf. Dependable Syst. Their Appl. (DSA)*, Nov. 2020, pp. 407–414.

[197] S. Niu, H. Lin, T. Niu, B. Li, and X. Wang, "DefectGAN: Weakly-supervised defect detection using generative adversarial network," in *Proc. IEEE 15th Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2019, pp. 127–132.

[198] M. Yang, P. Wu, and H. Feng, "MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105835.

[199] X. Tao, C. Adak, P.-J. Chun, S. Yan, and H. Liu, "ViTALnet: Anomaly on industrial textured surfaces with hybrid transformer," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[200] H. Yao, W. Yu, and X. Wang, "A feature memory rearrangement network for visual inspection of textured surface defects toward edge intelligent manufacturing," 2022, *arXiv:2206.10830*.

[201] A. De Nardin, P. Mishra, G. L. Foresti, and C. Piciarelli, "Masked transformer for image anomaly localization," *Int. J. Neural Syst.*, vol. 32, no. 7, Jul. 2022, Art. no. 2250030.

[202] J. Jiang, J. Zhu, M. Bilal, Y. Cui, N. Kumar, R. Dou, F. Su, and X. Xu, "Masked Swin transformer Unet for industrial anomaly detection," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 2200–2209, Feb. 2022.

[203] X. Dong, C. J. Taylor, and T. F. Cootes, "Automatic aerospace weld inspection using unsupervised local deep feature learning," *Knowl.-Based Syst.*, vol. 221, Jun. 2021, Art. no. 106892.

[204] N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, "Anomaly detection via self-organizing map," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 974–978.

[205] X. Yao, R. Li, C. Zhang, K. Huang, and K. Sun, "Multi-scale feature distillation for anomaly detection," in *Proc. 27th Int. Conf. Mechtron. Mach. Vis. Pract. (M2VIP)*, 2021, pp. 486–491.

[206] S. Yoa, S. Lee, C. Kim, and H. J. Kim, "Self-supervised learning for anomaly detection with dynamic local augmentation," *IEEE Access*, vol. 9, pp. 147201–147211, 2021.

[207] Z. Wang and J. Jing, "Pixel-wise fabric defect detection by CNNs without labeled training data," *IEEE Access*, vol. 8, pp. 161317–161325, 2020.

[208] H. Yao, D. Li, Y. Zhu, and W. Yu, "PM-AE: Pyramid memory autoencoder for unsupervised textured surface defect detection," in *Proc. 5th Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Dec. 2020, pp. 1324–1328.

[209] S. Venkataramanan, K. C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII*. Cham, Switzerland: Springer, Nov. 2020, pp. 485–503.

[210] G. Hu, J. Huang, Q. Wang, J. Li, Z. Xu, and X. Huang, "Unsupervised fabric defect detection based on a deep convolutional generative adversarial network," *Textile Res. J.*, vol. 90, nos. 3–4, pp. 247–270, Feb. 2020.

[211] S. Youkachen, M. Ruchanurucks, T. Phatrapomnant, and H. Kaneko, "Defect segmentation of hot-rolled steel strip surface by using convolutional auto-encoder and conventional image processing," in *Proc. 10th Int. Conf. Inf. Commun. Technol. Embedded Syst. (IC-ICTES)*, Mar. 2019, pp. 1–5.

[212] S. Mei, Y. Wang, and G. Wen, "Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model," *Sensors*, vol. 18, no. 4, pp. 1–18, 2018.
[213] R. Liu, M. Yao, and X. Wang, "Defects detection based on deep learning and transfer learning," *Metall. Mining Ind.*, vol. 7, pp. 312–321, Jul. 2015.
[214] X. Wu, L. Qiu, X. Gu, and Z. Long, "Deep learning-based generic automatic surface defect inspection (ASDI) with pixelwise segmentation," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–10, 2021.
[215] L. Yang, F. Zhou, and L. Wang, "A scratch detection method based on deep learning and image segmentation," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5015012. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[216] S. B. Block, R. D. Da Silva, L. B. Dorini, and R. Minetto, "Inspection of imprint defects in stamped metal surfaces using deep learning and tracking," *IEEE Trans. Ind. Electron.*, vol. 68, no. 5, pp. 4498–4507, May 2021.
[217] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.
[218] J. H. Bappy and A. K. Roy-Chowdhury, "CNN based region proposals for efficient object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3658–3662.
[219] B. Zhao, M. Dai, P. Li, R. Xue, and X. Ma, "Defect detection method for electric multiple units key components based on deep learning," *IEEE Access*, vol. 8, pp. 136808–136818, 2020.
[220] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2015, pp. 1–10.
[221] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2015, pp. 779–788.
[222] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2999–3007.
[223] L. Yu, Z. Wang, and Z. Duan, "Detecting gear surface defects using background-weakening method and convolutional neural network," *J. Sensors*, vol. 2019, pp. 1–13, Nov. 2019, doi: 10.1155/2019/3140980.
[224] R. Hao, B. Lu, Y. Cheng, X. Li, and B. Huang, "A steel surface defect inspection approach towards smart industrial monitoring," *J. Intell. Manuf.*, vol. 32, no. 7, pp. 1833–1843, 2020, doi: 10.1007/s10845-020-01670-2.
[225] T. Zhou, J. Zhang, H. Su, W. Zou, and B. Zhang, "EDDs: A series of efficient defect detectors for fabric quality inspection," *Measurement*, vol. 172, Feb. 2021, Art. no. 108885.
[226] R. Liu, M. Huang, Z. Gao, Z. Cao, and P. Cao, "MSC-DNet: An efficient detector with multi-scale context for defect detection on strip steel surface," *Measurement*, vol. 209, Mar. 2023, Art. no. 112467, doi: 10.1016/j.measurement.2023.112467.
[227] Y. Zhang, F. Xie, L. Huang, J. Shi, J. Yang, and Z. Li, "A lightweight one-stage defect detection network for small object based on dual attention mechanism and PAFPN," *Frontiers Phys.*, vol. 9, Oct. 2021, Art. no. 708097.
[228] Z. Zeng, B. Liu, J. Fu, and H. Chao, "Reference-based defect detection network," *IEEE Trans. Image Process.*, vol. 30, pp. 6637–6647, 2021.
[229] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 9–12, pp. 3465–3471, 2018.
[230] J. Wang, K. Chen, S. Yang, C. C. Loy, and D. Lin, "Region proposal by guided anchoring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2960–2969.
[231] J. Yang, G. Fu, W. Zhu, Y. Y. Cao, Y. Y. Cao, M. Y. Yang, and M. Y. Yang, "A deep learning-based surface defect inspection system using multiscale and channel-compressed features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8032–8042, Apr. 2020. [Online]. Available: https://www.ieee.org/publications/rights/index.html
[232] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[233] X. Cheng and J. Yu, "RetinaNet with difference channel attention and adaptively spatial feature fusion for steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[234] J. Luo, Z. Yang, S. Li, and Y. Wu, "FPCB surface defect detection: A decoupled two-stage object detection framework," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.
[235] F. Akhyar, Y. Liu, C.-Y. Hsu, T. K. Shih, and C.-Y. Lin, "FDD: A deep learning–based steel defect detectors," *Int. J. Adv. Manuf. Technol.*, vol. 126, nos. 3–4, pp. 1093–1107, May 2023, doi: 10.1007/s00170-023-11087-9.
[236] L. Xiao, B. Wu, and Y. Hu, "Surface defect detection using image pyramid," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7181–7188, Jul. 2020.
[237] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
[238] R. Guo, H. Liu, G. Xie, and Y. Zhang, "Weld defect detection from imbalanced radiographic images based on contrast enhancement conditional generative adversarial network and transfer learning," *IEEE Sensors J.*, vol. 21, no. 9, pp. 10844–10853, May 2021.
[239] H. Wang, Z. Li, and H. Wang, "Few-shot steel surface defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, 2022, Art. no. 5003912, [Online]. Available: https://www.ieee.org/publications/rights/index.html
[240] Y. Cao, W. Zhu, J. Yang, G. Fu, D. Lin, and Y. Cao, "An effective industrial defect classification method under the few-shot setting via two-stream training," *Opt. Lasers Eng.*, vol. 161, Feb. 2023, Art. no. 107294.
[241] H. Deng, Y. Cheng, Y. Feng, and J. Xiang, "Industrial laser welding defect detection and image defect recognition based on deep learning model developed," *Symmetry*, vol. 13, no. 9, p. 1731, Sep. 2021.
[242] F. Chen, M. Deng, H. Gao, X. Yang, and D. Zhang, "ACA-Net: An adaptive convolution and anchor network for metallic surface defect detection," *Appl. Sci.*, vol. 12, no. 16, p. 8070, Aug. 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/16/8070/htm
[243] J. Yu, X. Cheng, and Q. Li, "Surface defect detection of steel strips based on anchor-free network with channel attention and bidirectional feature fusion," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.
[244] X. Lv, F. Duan, J.-J. Jiang, X. Fu, and L. Gan, "Deep metallic surface defect detection: The new benchmark and detection network," *Sensors*, vol. 20, no. 6, p. 1562, Mar. 2020.
[245] R. Wei, Y. Song, and Y. Zhang, "Enhanced faster region convolutional neural networks for steel surface defect detection," *ISIJ Int.*, vol. 60, no. 3, pp. 539–545, 2020.
[246] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "DenseBox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*.
[247] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[248] W. Li, H. Zhang, G. Wang, G. Xiong, M. Zhao, G. Li, and R. Li, "Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 80, Apr. 2023, Art. no. 102470, doi: 10.1016/j.rcim.2022.102470.
[249] W. Chen, B. Zou, C. Huang, J. Yang, L. Li, J. Liu, and X. Wang, "The defect detection of 3D-printed ceramic curved surface parts with low contrast based on deep learning," *Ceram. Int.*, vol. 49, no. 2, pp. 2881–2893, Jan. 2023, doi: 10.1016/j.ceramint.2022.09.272.
[250] A. R. Singh, T. Bashford-Rogers, D. Marnerides, K. Debattista, and S. Hazra, "HDR image-based deep learning approach for automatic detection of split defects on sheet metal stamping parts," *Int. J. Adv. Manuf. Technol.*, vol. 125, nos. 5–6, pp. 2393–2408, Mar. 2023.
[251] J. Lim, J. Lim, V. M. Baskaran, and X. Wang, "A deep context learning based PCB defect detection model with anomalous trend alarming system," *Results Eng.*, vol. 17, Mar. 2023, Art. no. 100968, doi: 10.1016/j.rineng.2023.100968.
[252] W. Wu and Q. Li, "Machine vision inspection of electrical connectors based on improved Yolo v3," *IEEE Access*, vol. 8, pp. 166184–166196, 2020.
[253] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.
[254] S. Song, J. Jing, Y. Huang, and M. Shi, "EfficientDet for fabric defect detection based on edge computing," *J. Engineered Fibers Fabrics*, vol. 16, Jan. 2021, Art. no. 155892502110083.
[255] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," 2019, *arXiv:1905.11946*.
[256] S. Naddaf-Sh, M.-M. Naddaf-Sh, H. Zargarzadeh, M. Dalton, S. Ramezani, G. Elpers, V. S. Baburao, and A. R. Kashani, "Real-time explainable multiclass object detection for quality assessment in 2-Dimensional radiography images," *Complexity*, vol. 2022, pp. 1–17, Aug. 2022.

[257] W. Wang, C. Mi, Z. Wu, K. Lu, H. Long, B. Pan, D. Li, J. Zhang, P. Chen, and B. Wang, "A real-time steel surface defect detection approach with high accuracy," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–10, 2022.

[258] H.-V. Nguyen, J.-H. Bae, Y.-E. Lee, H.-S. Lee, and K.-R. Kwon, "Comparison of pre-trained Yolo models on steel surface defects detector based on transfer learning with GPU-based embedded devices," *Sensors*, vol. 22, no. 24, p. 9926, Dec. 2022.

[259] Y. Wang, M. Liu, P. Zheng, H. Yang, and J. Zou, "A smart surface inspection system using faster R-CNN in cloud-edge computing environment," *Adv. Eng. Informat.*, vol. 43, Jan. 2020, Art. no. 101037.

[260] F. M. Neuhauser, G. Bachmann, and P. Hora, "Surface defect classification and detection on extruded aluminum profiles using convolutional neural networks," *Int. J. Mater. Forming*, vol. 13, no. 4, pp. 591–603, Jul. 2020.

[261] Z. Zhu, G. Han, G. Jia, and L. Shu, "Modified DenseNet for automatic fabric defect detection with edge computing for minimizing latency," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9623–9636, Oct. 2020.

[262] X. Yu, L. Han-Xiong, and H. Yang, "Collaborative learning classification model for PCBs defect detection against image and label uncertainty," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, 2023.

[263] A. Wibowo, J. D. Setiawan, H. Afrisal, A. A. S. M. M. J. Mertha, S. P. Santosa, K. B. Wisnu, A. Mardiyoto, H. Nurrakhman, B. Kartiwa, and W. Caesarendra, "Optimization of computational resources for real-time product quality assessment using deep learning and multiple high frame rate camera sensors," *Appl. Syst. Innov.*, vol. 6, no. 1, pp. 1–15, 2023.

[264] Z. Liu, W. Lyu, C. Wang, Q. Guo, D. Zhou, and W. Xu, "D-CenterNet: An anchor-free detector with knowledge distillation for industrial defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[265] H. Zhang, D. Pan, J. Liu, and Z. Jiang, "A novel MAS-GAN-based data synthesis method for object surface defect detection," *Neurocomputing*, vol. 499, pp. 106–114, Aug. 2022.

[266] J.-T. Huang and C.-H. Ting, "Deep learning object detection applied to defect recognition of memory modules," *Int. J. Adv. Manuf. Technol.*, vol. 121, nos. 11–12, pp. 8433–8445, Aug. 2022, doi: 10.1007/s00170-022-09716-w.

[267] Z. Guo, C. Wang, G. Yang, Z. Huang, and G. Li, "MSFT-YOLO: Improved YOLOv5 based on transformer for detecting defects of steel surface," *Sensors*, vol. 22, no. 9, p. 3467, May 2022.

[268] C.-H. Liu, S.-W. Chen, C.-J. Tsai, W. C.-C. Chu, and C.-T. Tsai, "Development of an intelligent defect detection system for gummy candy under edge computing," *J. Internet Technol.*, vol. 23, no. 5, pp. 981–988, 2022.

[269] Z.-K. Zhang, M.-L. Zhou, R. Shao, M. Li, and G. Li, "A defect detection model for industrial products based on attention and knowledge distillation," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–18, Oct. 2022.

[270] J. Zhang, J. Jing, P. Lu, and S. Song, "Improved MobileNetV2-SSDLite for automatic fabric defect detection system based on cloud-edge computing," *Measurement*, vol. 201, Sep. 2022, Art. no. 111665.

[271] Y. Zheng and L. Cui, "Defect detection on new samples with Siamese defect-aware attention network," *Int. J. Speech Technol.*, vol. 53, no. 4, pp. 4563–4578, Feb. 2023, doi: 10.1007/s10489-022-03595-0.

[272] H. Ma and S. Lee, "Smart system to detect painting defects in shipyards: Vision AI and a deep-learning approach," *Appl. Sci.*, vol. 12, no. 5, p. 2412, Feb. 2022.

[273] K. R. Ahmed, "DSTEELNet: A real-time parallel dilated CNN with atrous spatial pyramid pooling for detecting and classifying defects in surface steel strips," *Sensors*, vol. 23, no. 1, p. 544, Jan. 2023.

[274] A. G. Pérez, M. J. G. Silva, and A. De La Escalera Hueso, "Automated defect recognition of castings defects using neural networks," *J. Nondestruct. Eval.*, vol. 41, no. 1, pp. 1–15, Mar. 2022, doi: 10.1007/s10921-021-00842-1.

[275] Q. Luo, W. Jiang, J. Su, J. Ai, and C. Yang, "Smoothing complete feature pyramid networks for roll mark detection of steel strips," *Sensors*, vol. 21, no. 21, p. 7264, Oct. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/21/7264/htm

[276] Y. He, K. Song, Q. Meng, and Y. Yan, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1493–1504, Apr. 2020.

[277] W. Zeng, Z. You, M. Huang, Z. Kong, Y. Yu, and X. Le, "Steel sheet defect detection based on deep learning method," in *Proc. 10th Int. Conf. Intell. Control Inf. Process. (ICICIP)*, 2019, pp. 152–157.

[278] O. Badmos, A. Kopp, T. Bernthaler, and G. Schneider, "Image-based defect detection in lithium-ion battery electrode using convolutional neural networks," *J. Intell. Manuf.*, vol. 31, no. 4, pp. 885–897, Apr. 2020.

[279] K. Adem and C. Közkurt, "Defect detection of seals in multilayer aseptic packages using deep learning," *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 27, no. 6, pp. 4220–4230, Nov. 2019.

[280] X. Lv, F. Duan, J.-J. Jiang, X. Fu, and L. Gan, "Deep active learning for surface defect detection," *Sensors*, vol. 20, no. 6, p. 1650, Mar. 2020.

[281] Y. Li, X. Wu, P. Li, and Y. Liu, "Ferrite beads surface defect detection based on spatial attention under weakly supervised learning," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–12, 2023.

[282] J. Zhang, H. Su, W. Zou, X. Gong, Z. Zhang, and F. Shen, "CADN: A weakly supervised learning-based category-aware object detection network for surface defect detection," *Pattern Recognit.*, vol. 109, Jan. 2021, Art. no. 107571.

[283] L. Gao, J. Zhang, C. Yang, and Y. Zhou, "Cas-VSwin transformer: A variant Swin transformer for surface-defect detection," *Comput. Ind.*, vol. 140, Sep. 2022, Art. no. 103689.

[284] A. G. Passos, T. Cousseau, and M. A. Luersen, "A smart deep convolutional neural network for real-time surface inspection," *Comput. Syst. Sci. Eng.*, vol. 41, no. 2, pp. 583–593, 2022.

[285] Y. Luo, Z. Wang, Z. Huang, Y. Yang, and C. Zhao, "Coarse-to-fine annotation enrichment for semantic segmentation learning," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manag.*, 2018, pp. 237–246.

[286] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 7, 2022, doi: 10.1109/TNNLS.2022.3152527.

[287] D. Patel and P. S. Sastry, "Adaptive sample selection for robust learning under label noise," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3921–3931.

[288] C. Wang, Z. Zhou, and Z. Chen, "An enhanced YOLOv4 model with self-dependent attentive fusion and component randomized mosaic augmentation for metal surface defect detection," *IEEE Access*, vol. 10, pp. 97758–97766, 2022.

[289] M. Gao, X. Feng, M. Geng, Z. Jiang, L. Zhu, X. Meng, C. Zhou, Q. Ren, and Y. Lu, "Bayesian statistics-guided label refurbishment mechanism: Mitigating label noise in medical image classification," *Med. Phys.*, vol. 49, no. 9, pp. 5899–5913, Sep. 2022.

[290] G. Koutroulis, T. Santos, M. Wiedemann, C. Faistauer, R. Kern, and S. Thalmann, "Enhanced active learning of convolutional neural networks: A case study for defect classification in the semiconductor industry," in *Proc. 12th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, 2020, pp. 269–276.

[291] G. Michau and O. Fink, "Unsupervised transfer learning for anomaly detection: Application to complementary operating condition transfer," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106816.

[292] M. Raghu, C. Zhang, J. M. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, p. 1–11.

[293] K. Zhao, Y. Chen, and M. Zhao, "A contrastive knowledge transfer framework for model compression and transfer learning," 2023, *arXiv:2303.07599*.

[294] C.-H. Chen, C.-H. Tu, J.-D. Li, and C.-S. Chen, "Defect detection using deep lifelong learning," in *Proc. 19th Int. Conf. Ind. Inform. (INDIN)*, 2021, pp. 1–6.

[295] W. Sun, R. Al Kontar, J. Jin, and T.-S. Chang, "A continual learning framework for adaptive defect classification and inspection," 2022, *arXiv:2203.08796*.

[296] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, pp. 3366–3385, 2019.

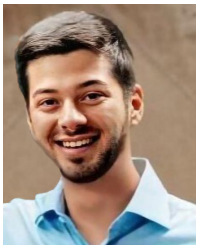
[297] G. M. Van De Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature Commun.*, vol. 11, no. 1, p. 4069, Aug. 2020.

[298] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, Mar. 2017.

[299] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2020, *arXiv:1710.09282*.
[300] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[301] Q. Cheng, S. Qu, and J. Lee, "72-3: Deep learning based visual defect detection in noisy and imbalanced data," *SID Symp. Dig. Tech. Papers*, vol. 53, no. 1, pp. 971–974, Jun. 2022, doi: 10.1002/sdtp.15658.
[302] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, 2021.
[303] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. P. Mardziel, and X. Hu, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2019, pp. 111–119.
[304] J. A. Whittaker and H. H. Thompson, "Black box debugging," *Queue*, vol. 1, no. 9, pp. 68–74, Dec. 2003.
[305] T. Almeida, F. Moutinho, and J. P. Matos-Carvalho, "Fabric defect detection with deep learning and false negative reduction," *IEEE Access*, vol. 9, pp. 81936–81945, 2021.
[306] J. Wang, K. Song, D. Zhang, M. Niu, and Y. Yan, "Collaborative learning attention network based on RGB image and depth image for surface defect inspection of no-service rail," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 4874–4884, Dec. 2022.
[307] A. Harsh Jha, S. Anand, M. Singh, and V. S. R. Veeravasarapu, "Disentangling factors of variation with cycle-consistent variational auto-encoders," 2018, *arXiv:1804.10469*.
[308] L. C. Chen, G. Papandreou, and I. Kokkinos, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Jun. 2017.
[309] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
[310] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[311] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, and Z. Yang, "A survey on vision transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.
[312] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surveys*, vol. 54, no. 10s, pp. 1–41, 2022, doi: 10.1145/3505244.
[313] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical vision transformer using shifted windows," 2021, *arXiv:2103.14030*.
[314] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[315] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Exp. Syst. Appl.*, vol. 91, pp. 464–471, Jan. 2018.
[316] Y. Zhang, Y. Wang, Z. Jiang, L. Zheng, J. Chen, and J. Lu, "Tire defect detection by dual-domain adaptation-based transfer learning strategy," *IEEE Sensors J.*, vol. 22, no. 19, pp. 18804–18814, Oct. 2022.
[317] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.
[318] W. Mei and D. Weihong, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Jul. 2018.
[319] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[320] J. Lorentz, T. Hartmann, A. Moawad, F. Fouquet, and D. Aouada, "Explaining defect detection with saliency maps," in *Proc. IEA/AIE*, 2021, pp. 506–518.
[321] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.

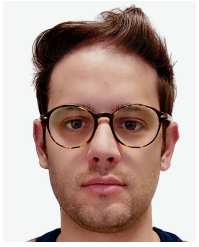
**MICHELA PRUNELLA** received the B.Sc. and M.Sc. degrees (cum laude) in medical systems engineering from the Polytechnic University of Bari, Italy, in 2020 and 2022, respectively, where she is currently pursuing the Ph.D. degree in autonomous systems with the Industrial Laboratory, Department of Electrical and Information Engineering.

Her master's thesis was focused on the definition of an algorithm for the automatic 3-D segmentation of patient-specific renal blood vessels and parenchyma from CTA scans. In 2022, she was a Research Fellow with the "Cognitive Diagnostics" Public Private Laboratory in cooperation between the Polytechnic University of Bari and Comau S.p.A., developing defect detection deep neural networks, feature engineering, and computer vision algorithms. Her research interests include the development of intelligent systems for diagnosis and precision surgery by creating an adaptive biological digital twin from the elaboration of biomedical data, signals, and images and videos based on machine and deep learning frameworks.



**ROBERTO MARIA SCARDIGNO** (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees (cum laude) in medical systems engineering from the Polytechnic University of Bari, Italy, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree in autonomous systems with the Industrial Laboratory, Department of Electrical and Information Engineering.

During his master's thesis, he developed a serious game to assess the sense of agency in children with cerebral palsy. His research interests include intelligent systems for automated diagnosis in the industrial and biomedical sectors.



**DOMENICO BUONGIORNO** received the B.Sc. and M.Sc. degrees (cum laude) in automation and control theory engineering from the Polytechnic University of Bari, Bari, Italy, in 2011 and 2014, respectively, and the Ph.D. degree in emerging digital technologies from the PERCeptual RObotics Laboratory (PERCRO), Scuola Superiore Sant'Anna, Pisa, Italy, in 2017. He is currently an Assistant Professor with the Department of Electrical and Information Engineering, Polytechnic University of Bari. His bachelor's and master's thesis were supervised by Prof. Vitoantonio Bevilacqua, which concerned machine learning-based optimization techniques for energy consumption optimization and human neuromusculoskeletal modeling, respectively. His Ph.D. thesis titled "Advanced Control Strategies for Natural Human-Exoskeleton Interaction" concerns the control of robotic interfaces for interaction with virtual environments, robot-aided neurorehabilitation (EMG-based control-myocontrol), and bilateral multi-DoF teleoperation in a disaster scenario. Since 2017, he has been a visiting Ph.D. student with the Biorobotics Laboratory, UCI Irvine, CA, USA, under the supervision of Prof. David Reinkensmeyer. His research interests include robotic-assisted surgery, medical data processing, and biomedical engineering. He teaches biomedical instrumentation with the Polytechnic University of Bari and electronic and information bioengineering with the Medical School of the University of Bari. In March 2019, he founded Apulian Bioengineering s.r.l., a spin-off company of the Polytechnic University of Bari. In 2019, he received the Italian National Bioengineering Group Award for the best doctoral thesis.

**ANTONIO BRUNETTI** received the bachelor's degree in computer science and automation engineering and the master's degree (cum laude) in computer science engineering from the Polytechnic University of Bari and the Ph.D. degree in electrical and information engineering from the Doctoral School, Polytechnic University of Bari, in February 2020, under the supervision of Prof. Vitoantonio Bevilacqua. He is currently an Assistant Professor with the Department of Electrical and Information Engineering, Polytechnic University of Bari. His final dissertation in human–computer interaction concerned the design and implementation of an intelligent system able to adapt the living environment of subjects with mild cognitive impairment based on cognitive analyses based on event-related potentials measured on the EEG signal stimulated via an innovative protocol based on virtual reality. From December 2015 to October 2016, he was a Research Fellow with the Polytechnic University of Bari for the GCESYS—Green Community Efficiency Systems Project, where he worked on designing and developing optimization algorithms applied to energy consumption scheduling in smart buildings with systems fueled by mixed traditional and renewable sources. His Ph.D. thesis dealt with the design and implementation of intelligent frameworks based on image processing, machine learning, and deep learning algorithms, for supporting clinical diagnosis in the Precision Medicine Era. From April 2020 to December 2021, he was a Postdoctoral Research Fellow working on research activities in the fields of electronic and informatics bioengineering and bioinformatics. In March 2019, he founded Apulian Bioengineering s.r.l., a spin-off of the Polytechnic University of Bari, as a shareholder.

**NICOLA LONGO** received the B.Sc. and M.Sc. degrees (cum laude) in automation and control theory engineering from the Polytechnic University of Bari, Bari, Italy, in 2011 and 2014, respectively, and the second-level master's degree in industrial automation from the Polytechnic University of Turin, Turin, Italy. After a research experience in distributed control of robotic swarms with New York University, he started working on machine vision applied to railway inspection. Since 2016, he has been with Comau S.p.A., an Italian robotic company (a leader in the automotive market), where he is responsible for the cognitive automation team, dedicated to robot perception, and the IoT.

**RAFFAELE CARLI** (Senior Member, IEEE) received the Laurea degree (Hons.) in electronic engineering and the Ph.D. degree in electrical and information engineering from the Polytechnic University of Bari, Italy, in 2002 and 2016, respectively.

From 2003 to 2004, he was a Reserve Officer with Italian Navy. From 2004 to 2012, he was a System and Control Engineer and the Technical Manager for a space and defense multinational company. He is currently an Assistant Professor in automatic control with the Polytechnic University of Bari. He is the author of more than 80 printed international publications. His research interests include formalization, simulation, the implementation of decision and control systems, and modeling and optimization of complex systems.

Dr. Carli was a member of the international program committee of more than 30 international conferences and a guest editor of special issues in international journals. He is an Associate Editor of IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS.

**MARIAGRAZIA DOTOLI** (Senior Member, IEEE) received the Laurea degree (Hons.) in electronic engineering and the Ph.D. degree in electrical engineering from the Polytechnic University of Bari, Italy, in 1995 and 1999, respectively.

She has been a Visiting Scholar with Paris 6 University and the Technical University of Denmark. She is currently an Expert Evaluator of the European Commission since the sixth Framework Programme. She is also a Full Professor in automatic control with the Polytechnic University of Bari, since 1999. She has been a Vice-Rector for research of the Polytechnic University of Bari and a Member Elect of the Academic Senate. She is the author of more than 200 publications, including one textbook (in Italian) and more than 80 international journal articles. Her research interests include modeling, identification, management, control and diagnosis of discrete event systems, manufacturing systems, logistics systems, traffic networks, smart grids, and networked systems.

Prof. Dotoli was a member of the international program committee of more than 80 international conferences. She was the Co-Chairperson of the Training and Education Committee of ERUDIT, the European Commission Network of Excellence for fuzzy logic and uncertainty modeling in information technology, a Key Node Representative of EUNITE, and the EUropean Network of Excellence on Intelligent Technologies. She is a Senior Editor of IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. She is the General Chair of the 2024 IEEE Conference on Automation Science and Engineering. She was the General Chair of the 2021 Mediterranean Conference on Control and Automation, the Program Chair of the 2020 IEEE Conference on Automation Science and Engineering, the Program Co-Chair of the 2017 IEEE Conference on Automation Science and Engineering, the Workshop and Tutorial Chair of the 2015 IEEE Conference on Automation Science and Engineering, the Special Session Co-Chair of the 2013 IEEE Conference on Emerging Technology and Factory Automation, and the Chair of the National Committee of the 2009 IFAC Workshop on Dependable Control of Discrete Systems.

**VITOANTONIO BEVILACQUA** received the Laurea degree in electronic engineering, the Ph.D. degree in electrical engineering, and the Postdoctoral degree in industrial informatics from the Polytechnic University of Bari.

In March 2019, he founded Apulian Bioengineering s.r.l., a start-up and spin-off company of the Polytechnic University of Bari, where he is currently the Chief Executive Officer. In September 2019, he joined as an affiliate Professor with the BioRobotics Institute, Scuola Superiore Sant'Anna di Pisa. He is currently a Full Professor in electronic and information bioengineering with the Electrical and Information Engineering Department, Polytechnic University of Bari, where he is the Head of the Industrial Informatics Laboratory. He is also the Unique Representative and Scientific Coordinator of the Public-Private Laboratory "Cognitive Diagnostics" founded in a partnership between the Polytechnic University of Bari and Comau S.p.A. Since 1996, he has been working and investigating in the field of computer vision and image processing, bioengineering, human–machine interaction based on machine learning, and soft computing techniques (neural networks, evolutionary algorithms, hybrid expert systems, and deep learning). The main applications of his research are in medicine, biometry, bioinformatics, ambient assisted living, and industry. In July 2000, he was involved as a Visiting Researcher in an EC-funded Trans-Mobility of Researchers (TMR) network (ERB FMRX-CT97-0127) called CAd Modeling Environment from Range Images (CAMERA) and U.K. Robotics Ltd., Manchester, U.K., in the field of geometric feature extraction and 3-D objects reconstruction. He has published more than 220 scientific articles.

Open Access funding provided by 'Politecnico di Bari' within the CRUI CARE Agreement