

## Review

# Image Processing Techniques for Concrete Crack Detection: A Scientometrics Literature Review

Md. Al-Masrur Khan <sup>1</sup>, Seong-Hoon Kee <sup>1,\*</sup>, Al-Sakib Khan Pathan <sup>2</sup> and Abdullah-Al Nahid <sup>3</sup>

<sup>1</sup> Department of ICT Integrated Ocean Smart Cities Engineering, Dong-A University, Busan 49315, Republic of Korea; almasrurkhan@donga.ac.kr

<sup>2</sup> Department of Computer Science and Engineering, United International University (UIU), Dhaka 1212, Bangladesh

<sup>3</sup> Electronics and Communication Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

\* Correspondence: shkee@dau.ac.kr

**Abstract:** Cracks in concrete surfaces are one of the most prominent causes of the degradation of concrete structures such as bridges, roads, buildings, etc. Hence, it is very crucial to detect cracks at an early stage to inspect the structural health of the concrete structure. To solve the drawbacks of manual inspection, Image Processing Techniques (IPTs), especially those based on Deep Learning (DL) methods, have been investigated for the past few years. Due to the groundbreaking development of this field, researchers have devoted their endeavors to detecting cracks using DL-based IPTs and as a result, the techniques have given answers to many challenging problems. However, to the best of our knowledge, a state-of-the-art systematic review paper is lacking in this field that would present a scientometric analysis as well as a critical survey of the existing works to document the research trends and summarize the prominent IPTs for detecting cracks in concrete structures. Therefore, this article comes forward to spur researchers with a systematic review of the relevant literature, which will present both scientometric and critical analysis of the papers published in this research area. The scientometric data that are brought out from the articles are analyzed and visualized by using VOSviewer and CiteSpace text mining tools in terms of some parameters. Furthermore, this article elucidates research from all over the world by highlighting and critically analyzing the incarnated essence of some of the most influential papers. Moreover, this research raises some common questions as well as extracts answers from the analyzed papers to highlight various features of the utilized methods.

**Keywords:** crack detection; concrete structures; deep learning; image processing techniques; scientometric analysis



**Citation:** Khan, M.A.-M.; Kee, S.-H.; Pathan, A.-S.K.; Nahid, A.-A. Image Processing Techniques for Concrete Crack Detection: A Scientometrics Literature Review. *Remote Sens.* **2023**, *15*, 2400. <https://doi.org/10.3390/rs15092400>

Academic Editor: Johannes R. Sveinsson

Received: 8 March 2023

Revised: 27 April 2023

Accepted: 29 April 2023

Published: 4 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A crack in a concrete surface (e.g., bridge, road, wall) is a very narrow gap between two sides of the surface that generally appears when the surface is slightly damaged. Cracking in concrete surfaces is quite inevitable and concrete surfaces can be cracked due to various reasons, such as deformation of the concrete structures, reaction of salts contained in the earth with concrete surfaces, thermal shrinkage of the concrete structures, overloading in the concrete surfaces, and so on. Concrete infrastructure, especially in South Korea, is quite likely to be cracked, as the percentage of ancient (more than 30 years old) reinforced concrete structures was inferred to be about 3.8% in 2014, and this is predicted to jump up to 13.8% and 33.7% in 2024 and 2029, respectively [1]. These cracks can cause deadly accidents as well as the expenditure of a huge amount of money for the maintenance and repair of concrete structures. So, crack detection at an early stage is very essential; this includes inspecting as well as evaluating the structural health and serviceability of the concrete structures.

For many years, manual inspection was a very common and traditional method for detecting cracks in concrete structures. However, manual inspection lacks both efficiency and accuracy. Moreover, this technique is so time-consuming, more arduous, and expensive because, in this method, the inspectors detect the cracks with only their human vision by roaming along the concrete structures. Therefore, realizing the drawbacks of manual inspection and the advancement of automation technologies, Ho et al. in 1990 [2] introduced the usage of image-based methods for detecting cracks automatically in concrete structures. Due to the advantages of vision-based algorithms over manual inspection techniques, the algorithms have gained vast popularity among both engineers and researchers in recent years. Hence, we see that nowadays, researchers from all over the world are devoting their efforts to developing and utilizing vision-based automated crack detection algorithms.

The primary steps for detecting cracks include acquiring the images, image preprocessing, and finally detecting or classifying the images. The literature shows that different types of images, such as camera images [3], Infrared Ray (IR) images [4], Ground Penetrating Radar (GPR) images [5], ultrasonic images [6], etc., are being utilized for detecting cracks. To extract necessary features from the acquired images as well as to remove noise due to shadows, poor illumination conditions, and thin cracks, researchers are developing and utilizing different IPTs such as wavelet transformation [7], Digital Image Correlation [8], Percolation methods [9], Ostu's method [10], Morphological approach [11], Canny edge detector [12], Sobel operator [13], Hough Transformation methods [14], and so on. After extracting the features, it is essential to detect and classify the cracks by using different classifier algorithms. For further improvement in crack detection, researchers nowadays are more willing to use Machine Learning (ML)- and Deep Learning (DL)-based classifier algorithms such as Support Vector Machine (SVM) [15], Random Forest [16], Convolutional Neural Networks (CNNs) [17], Recurrent Neural Networks (RNNs) [18], etc., as Neural Networks can extract necessary features automatically from concrete images and detect cracks more accurately.

With the developments of these image processing and classifier algorithms, vision-based crack detection methods are becoming more popular than ever before. As a result, a few technical articles have already been published in this research field. However, the field still lacks a reasonable amount of relevant systematic review papers presenting a scientometric analysis as well as a critical analysis of the existing works to show the research trends and summarize the prominent IPTs and classifier algorithms for detecting cracks in concrete structures. This gap in the existing literature and huge research scope motivate us to present a systematic review by analyzing the notable papers published between the years 2010 and 2020 which would focus on image-based crack detection algorithms to facilitate new researchers with useful information about this research field. In fact, Deep Learning (DL) started gaining popularity starting in 2012 with the advancement of the AlexNet model and consequently, researchers thought of utilizing DL for crack detection after that time. This one decade (2010–2020) has been specifically taken into consideration for this work because that time period sets the basis for work for this area. We have opted not to include the years 2021 and 2022, as that would be beyond our research objective (which is to analyze the very first decade of this particular research domain).

The main contributions of this survey paper are as follows:

- It presents a scientometric analysis of a few selected papers on image-based crack detection algorithms using data mining techniques to find out the current research trends, important research terms, influential publications, journals, and collaboration patterns of this research field.
- It presents a critical analysis of the papers related to image-based crack detection methods.
- Finally, it provides a summary of prominent image processing techniques and classifier algorithms for detecting cracks.

## 2. Literature Review

Computer vision, or image processing-based technology, has revealed itself to be a prominent research field for crack detection over the last decade. As a result, nowadays, it has become a great contributor to automating the crack detection process. Researchers from all over the world are devoting their efforts to developing and improving image-based crack detection methods. As a whole, now these methods have become an engrossing research interest both for researchers and engineers. Along with the continuous endeavors of improving the algorithms, researchers also enlisted the existing methodologies in the theme of survey papers to accelerate the research work in this area. This section briefly summarizes a few primary aspects from the preceding review papers and discusses the prominence of modern articles that establish themselves as some remarkable inclusions to the research field.

The earliest review paper that this work analyzes in this section was authored by John et al. in 1994 [19]. The authors discussed the usage of ultrasonic imaging techniques for detecting cracks in concrete structures. They also highlighted that severe improvement is needed in ultrasonic imaging techniques. After that, McCan et al. and Jahanshahi et al. presented a deep analysis of nondestructive testing (NDT) methods and image processing-based technologies like wavelet transform, Haar transforms, and the Digital Image co-relation technique for detecting cracks in [20] and [21], respectively. In 2014, Yao et al. [22] provided an overview of crack types and sources of cracks. In addition to this, the authors categorized the crack detection approaches into direct sensing and indirect sensing approaches. At a later time, works like [20,23–25] were published in 2016. The authors discussed various computer vision methods for detecting cracks and presented several platforms for image acquisition. In [24], Mohan et al. remarked that researchers would be more willing to use camera images for detecting cracks. Another notable survey was carried out by Gopalakrishnan et al. [26] in 2018, where the researchers gave a review of recently published articles (at that time) that used Deep Convolutional Neural Networks (DCNNs) for pavement crack detection. The authors also discussed and compared existing DL frameworks and network architectures for detecting cracks.

Vijayan et al. in [19] provided an overview of a few DL algorithms along with other processing techniques and suggested DL algorithms as the most preferable methods by analyzing previous works. In 2020, Sharma et al. [27] highlighted crack propagation over time and the depth and severity of cracks, which need to be determined. The authors also mentioned that there is still a huge research scope for developing a crack detection technique that is fast and accurate at the same time. In another paper of 2020, Hsieh et al. [28] presented ML and DL algorithms and available public datasets for crack segmentation in pavement images. The authors determined that Fully Connected Networks (FCNs) and U-Net produce an improved performance in the case of crack segmentation. Table 1 recapitulates the survey papers published on image-based crack detection algorithms. This table presents the publication year, source, major contributions, and limitations of the papers. The table is ordered based on the publication year of the papers. However, these survey papers neither collected articles systematically nor presented a bibliometric analysis to discuss the research trend, extract the most influential articles and countries, and present the collaboration pattern of this research field. In addition, many research papers did not categorize the articles according to their utilized image processing techniques and also did not analyze the articles (accurately) so that future researchers can have a clear vision of the research field of image-based crack detection techniques. As there is still a huge literature gap and research scope, in this work we are going to delineate the existing papers in this domain in a systematic manner; we will present a bibliometric analysis as well as a critical analysis of the works in an effort to lessen the difficulties for new researchers to understand the research trends, hot topics, and methodologies of image-based crack detection algorithms.

**Table 1.** Summary of previous review on image processing-based crack detection.

Ref	Year of Publication	Name of the Journal/Conference	Major Contributions	Limitations
[20]	2001	NDT & E International	<ul style="list-style-type: none"> <li>Discussing the necessity of NDT methods for structural health monitoring of concrete bridges.</li> <li>Presenting several NDT methods along with their measurable parameters, advantage, disadvantage, and cost.</li> <li>Pointing out the future research areas of NDT methods for bridge inspection.</li> </ul>	<ul style="list-style-type: none"> <li>Discussed several NDT methods but did not analyze papers based on the utilized methodology and achieved results that use these NDT methods for bridge inspection.</li> </ul>
[21]	2009	Structure and Infrastructure Engineering	<ul style="list-style-type: none"> <li>Presenting a deep analysis of the underlying computational models of IPTs to detect cracks in concrete bridge structures.</li> <li>Reviewing several papers which utilize IPTs for crack detection.</li> </ul>	<ul style="list-style-type: none"> <li>The papers are not collected in a systematic way.</li> <li>Did not categorize the papers based on their corresponding image processing techniques.</li> </ul>
[22]	2014	Structural Control and Health Monitoring	<ul style="list-style-type: none"> <li>Providing an overview of crack types and sources of cracks.</li> <li>Categorizing the crack detection approaches into direct sensing and indirect sensing approaches.</li> <li>Analyzing the articles related to each approach.</li> <li>Mentioning the accuracy and low computation power as the advantages of direct and indirect sensing approaches, respectively.</li> <li>Highlighting the reduction in consumed power and data loss as the research challenge for emerging crack detection approaches.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> <li>The articles are not summarized by the architecture, accuracy, and other parameters of the approaches.</li> <li>The articles are categorized in a broader way (e.g., direct, indirect approach) rather than in a more precise way (e.g., wavelet transform, ML, DL, etc., approaches)</li> </ul>
[29]	2016	Advanced Engineering Informatics	<ul style="list-style-type: none"> <li>Presenting a comprehensive analysis and synthesis of computer vision techniques to detect cracks on concrete and asphalt pavement structures.</li> <li>Reviewing relevant articles and categorizing the reviewed articles as per their image processing techniques or computer vision methods.</li> <li>Highlighting that visual inspection is necessary for structural health monitoring of concrete bridges.</li> <li>Mentioning that the automatic retrieval and assessment of defect properties are a future research challenge</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> </ul>

Table 1. Cont.

Ref	Year of Publication	Name of the Journal/Conference	Major Contributions	Limitations
[23]	2016	Computational Methods in Engineering	<ul style="list-style-type: none"> <li>Presenting various image acquisition platforms and IPTs for pavement crack detection.</li> <li>Presenting a general framework for automatic pavement crack detection.</li> <li>Mentioning the papers which focus on the existing IPTs for detecting pavement cracks.</li> <li>Comparing among the evaluation metrics for crack detection in the pavement.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> <li>The articles are not briefly analyzed based on the utilized methodology and achieved results.</li> </ul>
[24]	2016	Alexandria Engineering Journal	<ul style="list-style-type: none"> <li>Presenting several IPTs to detect cracks and analyzing 50 articles based on the techniques, dataset, and accuracy level.</li> <li>Identifying the steps in IPTs and the research challenges in this field.</li> <li>Highlighting that researchers are more willing to use camera images for detecting cracks.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> </ul>
[25]	2016	Journal of Imaging	<ul style="list-style-type: none"> <li>Reviewing the literature that discusses IR thermography with natural excitation for detecting cracks in reinforced concrete structures.</li> <li>Summarizing the utilized equipment and physical background to analyze thermographs.</li> <li>Presenting the application area of IRT as well as both advantages and disadvantages of IRT methods.</li> <li>Pointing out the usage of combined thermography approaches as the future research trend for crack detection.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> <li>Did not mention any numerical value to show the accuracy and efficiency of the used IR thermography methods.</li> </ul>
[30]	2017	Arabic Journal of Science & Engineering	<ul style="list-style-type: none"> <li>Giving a survey on the articles which use NDT methods for concrete damage detection.</li> <li>Identifying the observable parameters for each algorithm or method in case of damage detection.</li> <li>Highlighting both the advantages and disadvantages of each method.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> <li>No performance measure metrics or numerical values have been mentioned to evaluate the efficiency of the methods utilized in the articles.</li> </ul>

Table 1. Cont.

Ref	Year of Publication	Name of the Journal/Conference	Major Contributions	Limitations
[31]	2017	ISPRS Journal of Photogrammetry and Remote Sensing	<ul style="list-style-type: none"> <li>Constituted a database by systematic analysis.</li> <li>Highlighting the articles which utilize existing ultrasonic NDE methods to detect cracks in concrete infrastructures.</li> <li>Mentioning that the existing ultrasonic NDE techniques are not enough; improvement, as well as new approaches, are needed for better accuracy.</li> </ul>	<ul style="list-style-type: none"> <li>Mentioned the articles which utilized ultrasonic NDE techniques but did not analyze the papers based on the utilized methodology and achieved results.</li> </ul>
[26]	2018	Data	<ul style="list-style-type: none"> <li>Giving a review of the recently published articles which use DCNN algorithms for pavement distress detection.</li> <li>Pointing out both the achievements and the complexities of DCNN-based crack detection algorithms.</li> <li>Discussing and comparing DL frameworks, hyperparameters, and network architectures that are relevant for crack detections deployed by each article.</li> <li>Pointing the future research direction as finding out the crack type, severity of distress along with detecting cracks.</li> </ul>	<ul style="list-style-type: none"> <li>The articles are not collected in a systematic way.</li> </ul>
[19]	2018	International Journal of Pure and Applied Mathematics	<ul style="list-style-type: none"> <li>Reviewing the literature that detects cracks using IPTs.</li> <li>Providing an overview of a few DL algorithms which are being used for detecting cracks on building walls and concrete surfaces.</li> <li>Claiming that DL algorithms are the best for detecting cracks.</li> </ul>	<ul style="list-style-type: none"> <li>The authors only presented an overview of DL algorithms but did not analyze any paper based on the utilized methodology and achieved results that utilize DL algorithms for detecting cracks.</li> <li>Discussed only 10 papers that utilize image processing techniques. Furthermore, these papers were not categorized as per their corresponding IPTs.</li> <li>The articles are not collected in a systematic manner.</li> <li>Claimed that DL is the best method but did not present any evaluation metrics or numerical values to prove the efficiency of DL methods.</li> </ul>

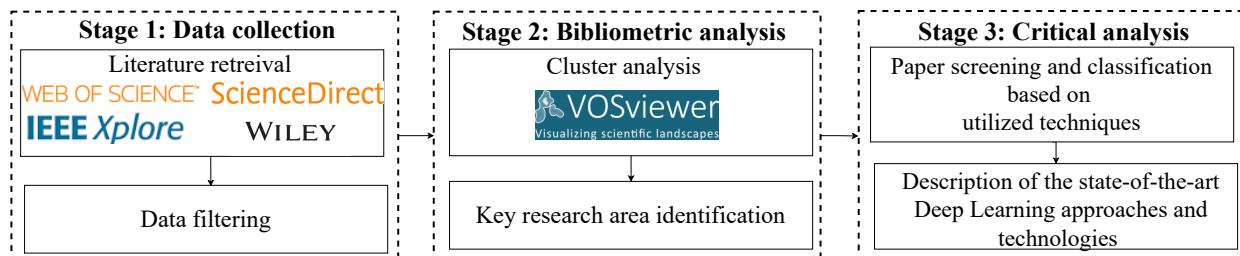
Table 1. Cont.

Ref	Year of Publication	Name of the Journal/Conference	Major Contributions	Limitations
[27]	2020	EasyChair Preprint	<ul style="list-style-type: none"> <li>• Surveying articles related to image-based crack detection.</li> <li>• Presenting several crack types and suitable algorithms both for detecting and classifying cracks of a particular type.</li> <li>• Highlighting the determination of crack propagation over time as a current research trend.</li> <li>• Mentioning that there is still a huge research scope for developing a crack detection technique that is fast and accurate at the same time.</li> </ul>	<ul style="list-style-type: none"> <li>• The articles are not collected in a systematic way.</li> <li>• Presented an overview of DL algorithms but did not analyze any paper based on the utilized methodology and achieved results corresponding to those algorithms.</li> <li>• Analyzed image-based articles but did not categorize those papers based on image processing techniques.</li> </ul>
[32]	2020	2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-EITTE)	<ul style="list-style-type: none"> <li>• Reviewing papers that focus on the classifiers used for crack detection on concrete roads.</li> <li>• Presenting comparison among the classifier algorithms.</li> <li>• Claiming CNN as the most preferable algorithm for crack detection on roads.</li> </ul>	<ul style="list-style-type: none"> <li>• The articles are not collected in a systematic way.</li> <li>• Number of analyzed papers that utilize the DL algorithms is too low.</li> </ul>
[28]	2020	American Society of Civil Engineers	<ul style="list-style-type: none"> <li>• Presenting the research trend of ML- and DL-based crack detection algorithms in terms of annual publication rate and appearance time of algorithms.</li> <li>• Categorizing the crack detection algorithms into three classes: classification, object detection, and segmentation.</li> <li>• Presenting a survey of 68 publications that focus on the ML and DL algorithms to detect cracks in pavement images and also presenting the publicly available dataset for crack detection.</li> <li>• Finding out the algorithms (FCN and U-Net) which produce an improved performance in the case of crack segmentation by conducting some performance evaluation techniques.</li> </ul>	<ul style="list-style-type: none"> <li>• The articles are not collected in a systematic way.</li> <li>• Only mentioned the papers under their corresponding algorithms but did not analyze the papers one by one based on their features or characteristics.</li> </ul>



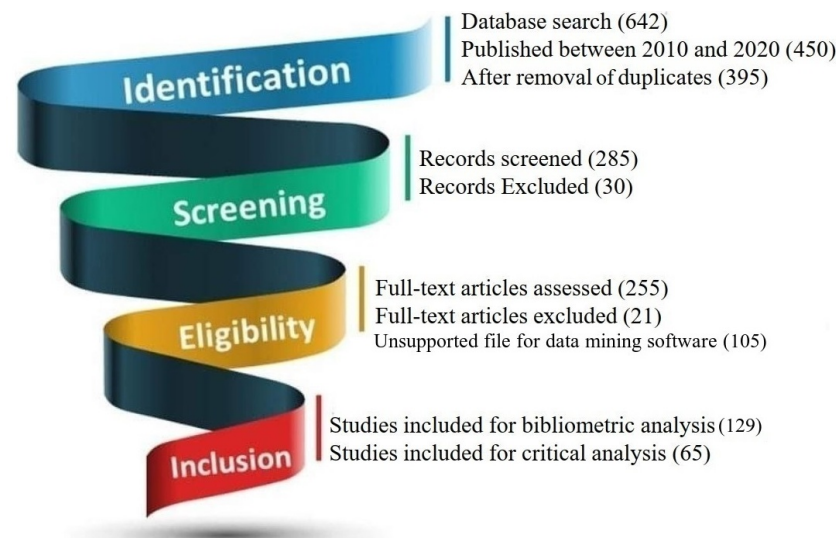
### 3. Research Methodology

This work has been designed using a mixed method for presenting a bibliometric analysis and critical analysis of the papers which focus on the algorithms utilized for crack detection on concrete bridges, buildings, and roads. Figure 1 presents the overall methodology of this study.



**Figure 1.** Overview of research methodology.

As seen in Figure 1, the first stage is about the data collection for this systematic review. The second stage is related to the bibliometric analysis for identifying the key research areas. Stage 3 presents the critical review of the papers based on the abstract, methodology, and results for giving a brief overview of the development of the algorithms utilized for crack detection. For conducting the literature review in a systematic manner, we have followed a set of guidelines to include the most relevant articles. The overall process of the literature retrieval and data filtering technique (the first stage of Figure 1) can be visualized in Figure 2.



**Figure 2.** Overview of the literature retrieval and filtering process.

The data collection (literature retrieval and data filtering) process was divided into a total of four phases based on the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) method: identification, screening, eligibility, and inclusion.

- Phase 1:** The authors searched for the papers in four different online digital libraries in November 2020 including Web of Science (WOS), Sciencedirect, IEEE Xplore, and Wiley online library using the search string “crack detection” AND (“bridge” OR “road” OR “concrete”) AND (“vision” OR “image”). In this way, the authors were able to download 642 papers initially. However, they limited the search string to a time span of ten years (2010 to 2020) for discussing the latest technologies. After removing the duplicate records, the authors identified a total of 395 papers at the end of Phase 1.



- **Phase 2:** In this stage, the authors screened 285 papers among the 395 papers extracted in Phase 1 by title and abstract which were published in the peer-reviewed journals. To avoid the inclusion of irrelevant articles in a systematic fashion, the authors developed some exclusion criteria and discarded the papers if (a) the research focus of any particular article is on non-image-based crack detection algorithms, (b) the paper discusses crack detection on reinforced plastic, beam, or steel structures, (c) the article is a review article instead of an original study. By employing the exclusion criteria, a total of 30 articles were excluded in this phase.
- **Phase 3:** In this stage, the remaining 255 papers were assessed by investigating the full text of the articles. The authors excluded an article from the systematic review if the article (a) was not closely related to the research focus of this study, (b) did not have a novel as well as efficient contribution to the research domain of image-based crack detection algorithms, (c) did not provide detailed information about the design or the implementation of the proposed idea. As a consequence, 21 papers were excluded. So, the number of extracted papers becomes 234. After that, an additional 105 papers were also excluded from these 234 papers for bibliometric analysis as they were not supported by the data mining software utilized in this work.
- **Phase 4:** After completing all the previous phases, 129 papers were finally included in this systematic review for scientometric analysis, and of the 234 papers, 65 DL-based papers were selected for critical analysis.

#### 4. Bibliometric Analysis

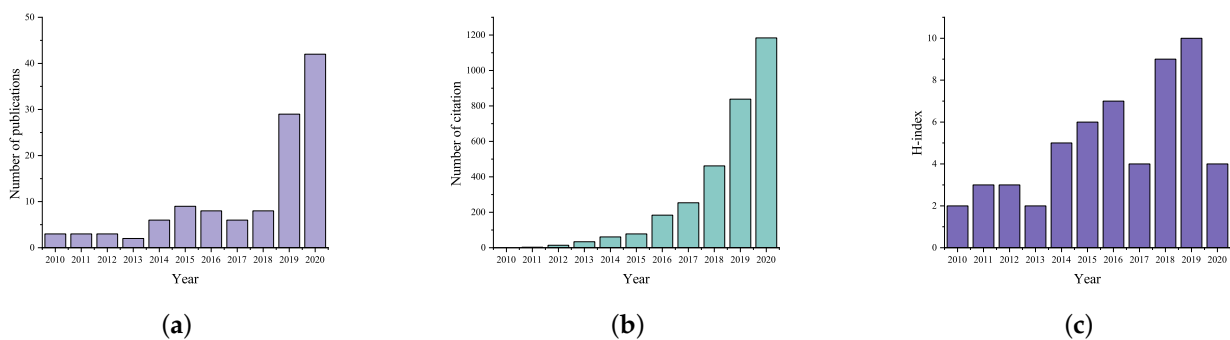
The scientometric analysis is a technique to assess the academic quality of publications, sources, and authors and determine the research trends of a particular research topic by several statistical methods, such as publication rate, citation rate, collaboration pattern, keyword occurrences, etc. This work utilized two prominent visualization tools, VOSviewer [33] and CiteSpace [34], to provide a bibliometric analysis of the papers collected from the databases chosen in this work. In the following section, this work will extract the most productive publications, authors, and publication sources in the research field of image-based crack detection algorithms. In addition, this work will also present some scientific mapping analysis. This work considered co-citation analysis, co-authorship analysis, and occurrences of the keyword and timeline view analysis as the subsections of scientific mapping analysis. Co-citation analysis can elicit the relatedness and measure the proximity degree of the sources and authors. Co-authorship analysis can determine the collaboration pattern among the countries and institutions. Again, keyword occurrences can extract the research trends and important terms of a particular research topic.

##### 4.1. Overview of the Publications

###### 4.1.1. Annual Analysis of the Publications

From the online databases, this work was capable of extracting a total of 129 papers for bibliometric analysis within the year range 2010–2020. Figure 3a shows the number of publications per year. As can be observed from the figure, the publication rate in the earlier years (2010–2013) of this decade was too low; less than five papers were published each year. After 2013, the number of published articles per year begins to accelerate and fluctuates in the range of six to nine during the years 2014–2019. However, the number of published articles increases dramatically in 2019. The number of publications jumps to 29, which is about 22.48% of the total published papers (by that time). In 2020, the publication rate also follows an upward trajectory. As a result, until November 2020, forty-two (42) papers were published, which clearly indicates that by that time, the researchers started devoting their efforts more towards this research area, and from the analysis, it can be said that this research field would then undergo a huge increase in publication rate in the upcoming days. As in our analysis, we modeled the first crucial decade of this research trend, i.e., up to 2020 would be our range; however, checking the most recent works, we

also find a similar pattern in 2021 (so far at the time of writing this article, even in 2022); the increase in the number of published papers is continuing.



**Figure 3.** Overview of annual characteristics of the publications. (a) Number of IPT-based publications over the years (2010–2020). (b) Number of received citations (2010–2020). (c) The H-index of the publications (2010–2020).

This work has also analyzed the citation rate of publications per year. Figure 3b illustrates the distribution of citations achieved by the publications each year. The 129 papers were cited 3112 times during that period of ten years. The figure shows that the citation rate follows a continuously increasing trend with the passage of time. If this work divides the time span from 2010 to 2020 into three phases, then it can be seen from the figure that in the first phase (2010–2013), the number of citations in each year was less than 50 and the total number of citations was 51, which covers only 1.54% of the total citations. In the next phase (2014–2017), the distribution of citations also follows an increasing trend. The highest number of received citations in this phase is 254 in 2017, followed by 78 and 84 citations in 2015 and 2016, respectively. This phase consists of 18.54% of total received citations.

In the last phase (2018–2020), the number of citations increases significantly. In 2018, the publications were cited 462 times, and this number jumps to 838 in 2019. This is the highest increment of received citations by the publications between two particular years. Finally, in 2020, the publications received 1184 citations, which is the topmost among all the years in the decade. The last phase covers about 75% of the total citations, which implies that in recent years, impactful contributions are being made to this chosen research field.

This work also uses an author-level metric named the H-index, which ensures productivity and citation impact, to conduct the annual analysis of the publications. Figure 3c depicts the H-index distribution of the papers over the years. From the figure, it can be seen that the topmost H-index is 10 in 2019. The years 2018 and 2017 hold the second and third positions with H-index 9 and 8, respectively. It can also be seen that the H-index fluctuates over the years over the whole decade. The total H-index for 129 publications is 23, which means that among the 129 publications, only 23 publications have at least 23 citations. Furthermore, as this H-index is greater in the later part of the decade than in the earlier part, it can be inferred that the number of influential and productive papers is increasing in recent years.

#### 4.1.2. The Most Cited Publications

In this work, we have found and analyzed the most influential and popular articles among the 129 articles based on the received citations by the publications. As a consequence, we set a threshold of a minimum of 50 citations and were able to extract 15 papers. These 15 papers were cited 2090 times, which is about 67.16% of the total citations received by all of the publications. As the lion's share of the citations comes from these papers, the productivity and influence of these papers in the research domain of image-based crack detection algorithms are evident.

These top-cited papers are summarized in Table 2 by their title, publication year, publication source, corresponding author's name, corresponding author's country, received

citations, and average citations per year. This table is ordered based on the number of citations received by the publications. The highest citation (574) received by any single article is for the paper entitled “*Deep Learning-based Crack Damage Detection Using Convolutional Neural Networks*”, which was published in 2017 in “*Computer-Aided Civil and Infrastructure Engineering*”. This paper is so influential and popular among researchers that it received 574 citations within only 4 years, with a citation rate of 134.50 per year. The second highest on the list, “*Crack Tree: Automatic Crack Detection From Pavement Images*”, received 242 citations. This paper was published in 2012 and its citation rate per year is 26.89.

A deeper analysis reveals that (see Table 2) most of the papers are receiving citations over the years in a linear manner, but [35–37] are receiving citations at an increasing rate. Though [36,37] were published in 2019 and 2018, respectively, they received 80 citations, each with a citation rate of 40 and 26.67 per year, which clearly indicates that along with [35], these papers are going to contribute significantly in the research field. It also indicates that these papers are highly influential and receiving attention from the researchers within a relatively short time. In addition, refs. [38–40] have also maintained a good citation rate over the years. On the other hand, refs. [41,42] have the least citations (56 and 52), and their low citation rates (5.60 and 7.43 citations per year) indicate that these papers are not receiving enough attention from researchers.

**Table 2.** Summary of the top cited papers.

Reference	Journal	Corresponding Author	Country of Corresponding Author	Publication Year	Citation	Average Citation per Year
[35]	Computer-aided Civil and Infrastructure Engineering	Young-Jin Cha	Canada	2017	575	143.50
[38]	Pattern Recognition Letters	Qin Zou	China	2012	242	26.89
[43]	Machine Vision and Applications	Tomoyuki Yamaguchi	Japan	2010	176	16
[39]	Computer-aided Civil and Infrastructure Engineering	Shirley Dyke	USA	2015	142	23.67
[44]	IEEE Transactions On Intelligent Transportation Systems	Henrique Oliveira	Portugal	2013	139	17.38
[45]	Computer-aided Civil and Infrastructure Engineering	Takafumi Nishikawa	Japan	2012	136	15.11
[40]	IEEE Transactions on Automation Science And Engineering	Kristin J. Dana	USA	2016	118	23.60
[46]	Sensors	David F. Llorca	Spain	2011	115	11.50
[47]	Computer-aided Civil and Infrastructure Engineering	Eduardo Zalama	Spain	2014	102	14.57
[48]	Machine Vision And Applications	Yusuke Fujita	Japan	2011	101	10.10
[36]	Automation In Construction	Cao Vu Dung	Japan	2019	80	40
[37]	Construction And Building Materials	Sattar Dorafshan	USA	2018	80	26.67
[10]	Optik	Ahmed Mahgoub Ahmed Talab	China	2016	57	11.40
[41]	Image And Vision Computing	Qin Zou	China	2011	56	5.60
[42]	Journal Of Computing In Civil Engineering	Matthew M. Torok	Japan	2014	52	7.43

## 4.2. Influential Journals, Authors, and Countries

### 4.2.1. The Most Productive Journals

In this subsection, this work describes the most productive publication sources in the field of image-based crack detection algorithms. The 129 collected papers were published in 64 different journals. However, this work extracted the top 10 publication sources based on the number of publications. These 10 journals published 68 (52.71%) articles in total among the 129 papers. The other 54 journals are responsible for the other 61 (47.29%) papers. Table 3 summarizes these most productive journals by their name, total publications, total citations, average citations per year, Impact Factor, 5-year Impact Factor, and H-index. The table is ordered based on the number of publications. From Table 3, it can be seen that the journal “Computer-aided Civil Infrastructure Engineering” holds the first position with a total of 11 publications and 1100 citations. The IF (8.552) of this journal is also quite high. This journal has an H-index of 8, which clearly indicates the popularity of this journal among researchers. The “Sensors” of “MDPI” is in second place because of its 10 publications and 193 citations.

**Table 3.** Summary of the most productive journals.

Journal Name	Total Publications	Total Citations	Average Citations	Impact Factor	5 Years Impact Factor	Publisher	H-Index
Computer-aided Civil Infrastructure and Engineering	11	1100	122.22	8.552	6.212	Wiley	8
Sensors	10	193	21.44	3.275	3.427	MDPI	6
Journal of Computing in Civil Engineering	9	136	19.43	2.979	2.943	ASCE-AMER SOC Civil Engineers	6
Automation in Construction	8	143	71.50	5.669	6.121	Elsevier	4
Construction and Building Materials	7	128	21.33	4.419	5.0396	Elsevier	4
IEEE Access	7	15	7.50	3.745	4.076	IEEE	2
Applied Sciences Basel	5	20	10.00	2.474	2.458	MDPI	2
Structural Health Monitoring an International Journal	4	12	2.00	4.87	4.922	SAGE	2
Advances in Civil Engineering	3	23	11.50	1.176	-	Hindawi	2
Machine Vision and Applications	3	277	27.70	1.605	2	Springer	2

The top 5 journals on this list have the higher number of citations. However, it is quite strange for the other journals. For example, IEEE Access has published seven papers until now but received only 15 citations. Interestingly, the number of published articles in other journals would fluctuate from three to five, with a low citation number, which is a clear indication that the researchers are not paying attention to the journals at the bottom part of Table 3 for the papers related to the chosen research field of this work. However,

it is notable that the journal “Machine Vision And Applications”, which has an IF of only 1.605, published just 3 articles, but the papers were cited (already) 277 times. The number of received citations of this journal implies that the published papers in this journal are playing a significant role in the research field.

To understand the trend of citations and impact, we were interested in those journals which published the least number of papers but received a higher citation, so we searched in the dataset to check for the existence of these journals. We were able to find a few journals, such as “IEEE Transactions on Intelligent Transportation System” (2 papers, 257 citations), “Pattern Recognition Letters” (1 paper, 242 citations), and “IEEE Transaction on Automation Science and Engineering” (1 paper, 118 citations).

To understand the historical development of the top publication sources in terms of publications and citations, we have summarized the information in Table 4. From Table 4 it can be seen that all of the journals started publishing image-based crack detection algorithms-related papers regularly in around 2018. Before that period, i.e., 2010–2017, these journals published merely three to four papers per year except in 2013. In fact, in 2013, these journals did not publish a single paper in this research field (in accordance with our set criteria). In the case of citations, the understanding is that only three journals, “Computer-aided Civil Infrastructure Engineering”, “Sensors”, and “Machine Vision and Applications”, are receiving citations in all of the years, and the rest are receiving citations from 2015 onwards. Among the journals, “IEEE Access” and “Applied Sciences Basel” received citations only in 2019 and 2020. It is notable that the journals are following an upward trend in the case of receiving citations over the years and, as a result, all of the journals received the maximum number of citations in 2020.

**Table 4.** Historical development of the journals in terms of the publications and citations.

Journal Name	2010		2011		2012		2013		2014		2015		2016		2017		2018		2019		2020	
	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C	P	C
Computer-aided Civil Infrastructure and Engineering	0	0	0	0	1	2	0	12	2	17	1	7	0	39	1	67	1	196	3	331	2	429
Sensors	0	0	1	0	0	2	0	2	0	3	0	9	0	20	1	22	3	28	3	36	2	71
Journal of Computing in Civil Engineering	0	0	0	0	0	0	0	0	1	1	0	4	3	13	0	12	0	23	2	34	3	49
Automation in Construction	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	25	3	116
Construction and Building Materials	0	0	0	0	0	0	0	0	0	0	1	3	0	5	0	0	2	6	1	35	2	79
IEEE Access	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	2	3	13
Applied Sciences Basel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	8	2	12
Structural Health Monitoring and International Journal	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	3	0	0	0	0	3	7
Advances in Civil Engineering	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	1	18
Machine Vision And Applications	1	0	1	3	0	7	0	10	0	16	0	15	0	32	0	34	0	47	0	50	1	63

P denotes the number of published papers; C denotes the number of citations received by the paper.

#### 4.2.2. The Most Productive Authors

In this subsection, this work will discuss the most productive authors in the research area of image-based crack detection algorithms. From the dataset collected from WoS, we found that 425 authors are responsible for 129 papers. We have extracted the top ten authors according to the number of publications (five authors) and the received citations

(five authors). Table 5 summarizes the most productive authors by their name, number of total publications, total citations, average citations per year, number of published papers as first author, H-index, and country.

**Table 5.** Summary of the most productive authors.

	Author's Name	Total Publications	Total Citations	Average Citations	As 1st Author	H-Index	Country
Based on Publications	Ying Chen	5	49	12.25	0	3	China
	Zhong Qu	5	44	8.80	5	3	China
	Weigang Zou	4	18	9.00	0	3	China
	Wei Li	4	9	4.50	1	2	China
	Qingquan Li	3	338	37.56	1	3	China
Based on Citations	Young-zin Cha	1	575	143.75	1	1	Canada
	Choi Wooram	1	575	143.75	0	1	Canada
	Oral Buyukozturk	1	575	143.75	1	1	Canada
	Qingquan Li	3	338	37.56	0	3	China
	Mao Qin ZOu	2	298	33.11	1	2	China

The first part of Table 5 presents the top five authors who have the highest number of publications. From the table, it can be seen that the highest number of publications by any single author is five. “Ying Chen” and “Zhong Qu” both have published five papers, but “Ying Chen” has received more citations than “Zhong Qu”. However, the interesting thing is that “Zhong Qu” has published all of his five papers as the first author, while no other author has published more than one paper as the first author. “Weigang Zou” and “Wei Li” both have published four papers and received eighteen and nine citations, respectively. “Qingquan Li” holds the fifth position on this list with three publications. However, with only 3 publications, “Qingquan Li” received 388 citations, which is a clear indication of both productivity and the high influence of this author. One more noticeable thing from the first part of this table is that all of the productive authors in terms of the number of publications are from China, which implies the significance of Chinese researchers in the chosen research field (of this work).

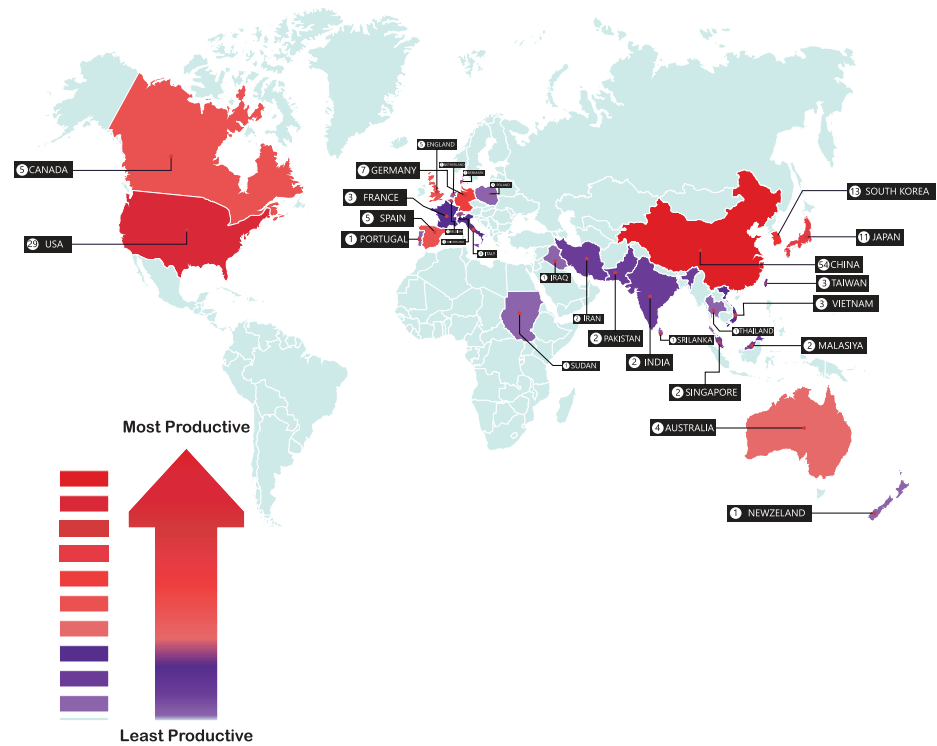
The second part of this table presents the top five most cited authors. It is visible from the table that the first three authors have the same values for all of the measurement parameters used in this table. They are the authors of the highest cited paper of the dataset entitled “*Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks*”. They published only one paper in the chosen research field and received 575 citations, with an average of 143.75 citations per year, and proved their excellence in this field. “Qingquan Li” is the only author among the top ten authors who places in both the first part and the second part of Table 5. Finally, “Mao Qin Zou” holds 5th place, with 2 publications and 298 citations. Among the top five authors extracted based on received citations, two authors are from China and the other three are from Canada. However, “Young-zin Cha” is also a Chinese researcher who was working at the University of Manitoba, Canada during the publication of his paper.

#### 4.2.3. The Most Productive Countries

Let us now see the most productive countries in the research domain of image-based crack detection algorithms. From the dataset collected from WoS, it is found that 29 countries are responsible for 129 papers. Figure 4 presents the geographical distribution of the papers around the world. In this figure, crimson-colored countries published more papers than lavender-colored ones. If we analyze the contributions of the continents in the case of publishing papers, then it can be observed that Asia is the supreme continent,



publishing 58.43% of all the publications. America, Europe, and Australia are responsible for 20.48%, 15%, and 3.61% of the published papers, respectively, while the remaining 2.48% comes from the other parts of the globe.



**Figure 4.** Geographical distribution of the publications.

After presenting the geographical distribution of the publications, our work extracted the top 10 countries based on the number of publications. Table 6 summarizes the most productive countries by their names, total publications, total citations, average citations per year, the number of cited papers greater than or equal to 30/20/10/5, and the H-index. This table is ordered based on the number of published papers by each country. From Table 6, it can be seen that China is the leading country, with 54 publications and 722 citations. The H-index (14) of China is also the highest among these countries, along with the USA. The next country on the list is the USA. The number of published papers (29) from the USA is less than China, but the total number of citations (1450) is not only greater than China, but also the highest among all the countries. Again, the USA has the highest number of publications (8) and received greater than or equal to 30 citations. The USA claims the highest rate of citations (161.11) per year. South Korea holds the 3rd position on the list, with 13 articles and 104 citations. Following South Korea, Japan published 11 articles with 551 citations. However, the citation rate of South Korea (52) is quite close to Japan (55.10), though there is a huge difference between their citations.

With a deeper understanding, if we check Table 6, then it can be found that despite being number 5 on the list with 7 publications, Germany has received only 21 citations, while Canada and Spain received 584 and 252 citations, respectively, for only 5 publications, which clearly denotes the influence of the papers published from these countries. England, Australia, and Vietnam published 5, 4, and 3 papers, respectively, and received 21, 16, and 90 citations. However, there are also a few other countries in the dataset that are not present in the top 10 countries, but they received greater citations with the least number of publications. This fact indicates the significance of the papers published in those countries, e.g., Portugal (1 publication, 139 citations) and India (2 publications, 98 citations).



**Table 7.** Co-citation indices of the sources.

Cited Source	Citation	Total Link Strength
Computer-aided Civil Infrastructure and Engineering (compute-aided civ inf)	363	5201
Automation in Construction (automat constr)	183	3053
Journal of Computer in Civil Engineering (j comput civil eng)	178	2943
Proceeding CVPR IEEE (proc cvpr ieee)	112	1905
IEEE Transactions on Pattern Analysis and Machine Intelligence (ieee t pattern anal)	98	1541
IEEE Transactions on Intelligent Transportation Systems (ieee t intell transp)	78	1364
Lecture Notes in Computer Science (lect notes comput sc)	63	1105
Construction and Building Materials (constr build mater)	72	1010
Machine Vision and Applications (mach vision appl)	63	933
Sensors (sensors-basel)	70	924
Advanced Engineering Informatics (adv eng inform)	48	881
IEEE Conference on Computer Vision and Pattern Recognition (ieee i conf comp vis)	39	806
Structural Health Monitoring (struct health monit)	36	707
Structural Control and Health Monitoring (struct control hlth)	44	677
Pattern Recognition Letters (patetrn recogn lett)	45	623
Transportation Research Record (transport res rec)	39	602
NDT & E International (ndt & e int)	34	463
IEEE Transactions on Image Processing (ieee t image process)	30	432
Smart Materials and Structures (smart mater struct)	30	360

From Figure 5, it can be seen that the journals have been divided into a total of 3 clusters (red, green, blue) with 190 links and 13,484 link strength in total. Each node in Figure 5 represents the corresponding journal. The bigger the node, the higher the weight. For the co-citation analysis, total link strength has been selected as the weight in this work; that means the sources which have higher weight have higher link strength. The connection line between two consecutive journals illustrates that these two sources have been cited together in a publication. The thicker the line, the more frequently they have been cited together. The interesting issue about the clusters is that all the sources of each cluster have been cited along with the sources of all other clusters.

With a deeper perspective, it can be noticed that the red cluster contains a total of 10 sources (50%). The prominent journal in the red cluster as well as all the clusters is “Computer-aided Civil Infrastructure and Engineering”, with 19 links and a total link strength of 5201. The journal has been cited with all the other journals; however, it has been cited the most times (819) along with the journal “Automation in Construction”, which clearly indicates the high relatedness of these two journals in this specific research field. “Automation in Construction” achieves the second position, with a total link strength of 3053 in the red cluster. The other influential sources in this cluster are “Journal of Computer in Civil Engineering” (2943), “Construction and Building Materials” (1010), “Machine Vision and Applications” (933).

The second cluster (green) consists of seven sources. The most influential source in this cluster is “Proceeding CVPR IEEE”, with a total link strength of 1905. This source has been cited the most times (476) with the journal “Computer-aided Civil Infrastructure and Engineering”. The other influential journals of this cluster are “IEEE Transactions on Pattern Analysis and Machine Intelligence” (1541) and “IEEE Transaction on Intelligent Transportation System” (1438). The interesting point about the green cluster is that the sources of this cluster have been cited more times with a few sources of the red cluster

(“Computer-aided Civil Infrastructure and Engineering”, “Automation in Construction”, “Journal of Computer in Civil Engineering”) than the sources of its own cluster, which undoubtedly indicates the high influence of those three sources of the red cluster in the case of co-citation. Finally, the blue cluster consists of only three sources. The highly influential source of this cluster is “Advanced Engineering Information”, with total link strength of 881. The other two sources of this cluster are situated closely in Figure 5, but “Advanced Engineering Information” is situated far away from the other two journals of the blue cluster. The sources of the blue cluster have been cited more times with the highest sources of the red and green clusters than the sources of the blue cluster. It implies that though the sources of the blue cluster have relatedness based on their research topic, over the years, these sources have not been cited too many times together.

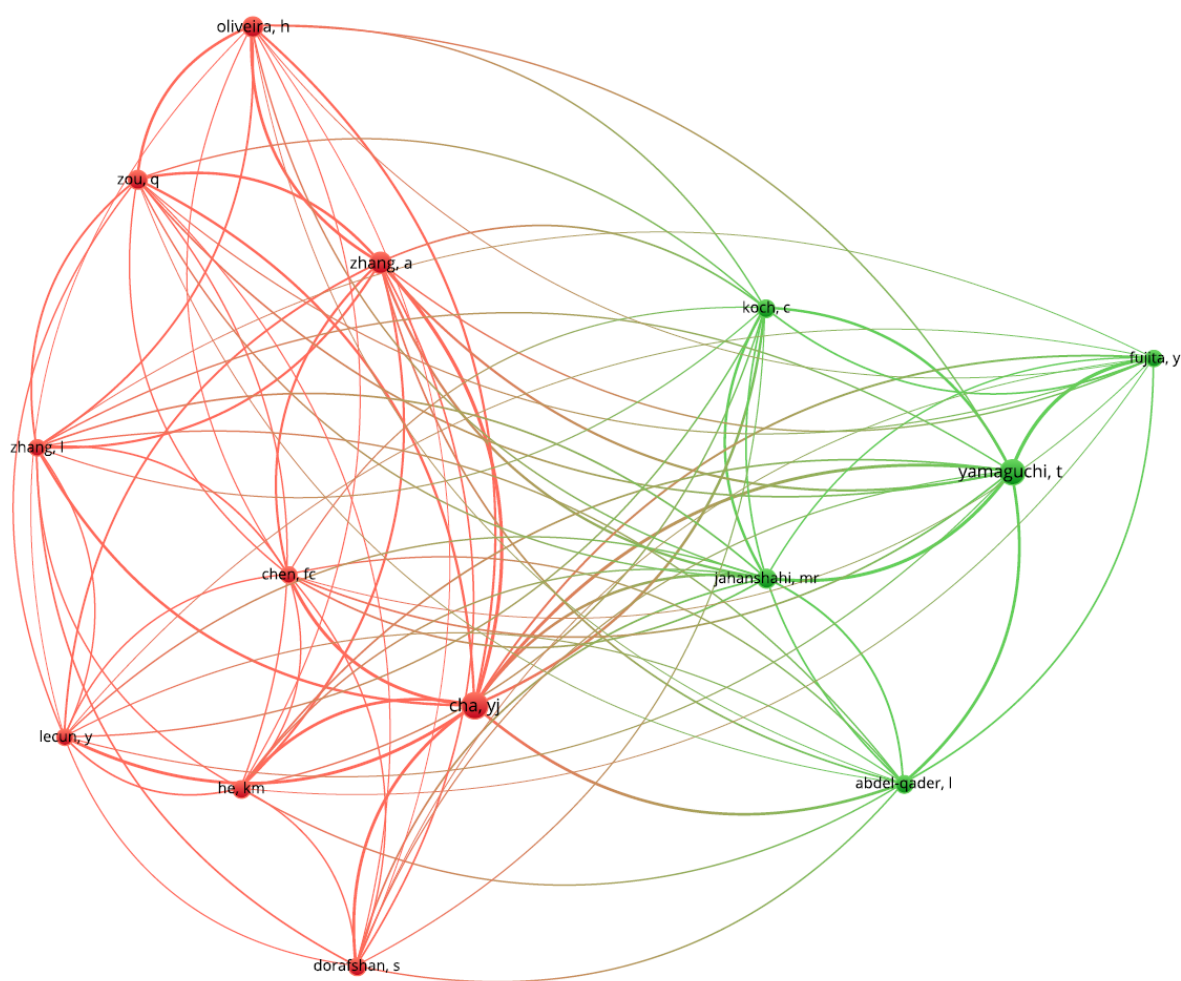
After analyzing the co-citation network of cited sources, this work analyzes the co-citation network of cited authors. As a consequence, this work set a threshold of at least 30 citations, and among 2361 cited authors, only 14 authors met the threshold. Table 8 presents these cited authors with the total link strength of co-citation. This table is ordered by the total link strength of the authors.

**Table 8.** Co-citation indices of the authors.

Cited Author	Citation	Total Link Strength
Young-Jin Cha	363	5201
Tomoyuki Yamaguchi	77	387
Qinayun Zhang	52	372
Mohammad R. Jahanshahi	44	314
Qiang Zou	44	255
Christian Koch	37	254
Abed Abdel Qader	39	253
Fu-Chen Chen	31	250
Lei Zhang	33	245
KM Liew	35	240
Yann LeCun	33	220
Henrique Oliveira	49	219
Yusuke Fujita	32	208
Sattar Dorafshan	38	202

For better understanding, we have generated the scientific landscape of the co-citation network of the cited authors using VOSviewer software, as shown in Figure 6.

From Figure 6, it can be seen that the authors have been divided into a total of 2 clusters (red, green) with 91 links and a total link strength of 1973. Like the cited sources in the case of cited authors, total link strength has been selected as the weights of the nodes. If there is a connection line between two authors, then it indicates that the authors have been cited together in any other publications. The thicker the line, the more frequently the authors have been cited together. It can be seen that the red cluster contains a total of nine authors. The leading author in this cluster is “Young-jin Cha”, with a link strength of 527. The author has been cited the most times (59) with “Qinayun Zhang”, which clearly indicates that their publication focus is on a similar type of research topic. Young-Jin Cha is also cited many times along with other authors, such as “Fu-Chen Chen” (50), “Yann LeCun” (49), and “Tomoyuki Yamaguchi” (49). The other influential authors according to total link strength in this cluster are “Qinayun Zhang” (372), “Qiang Zou” (255), and “Fu-Chen Chen” (250). The cited authors of the red cluster have higher citation linkage with the authors of the red cluster rather than the authors of the green cluster.



**Figure 6.** Co-citation analysis of the cited authors.

The green cluster consists of five authors. The most prominent cited author of this cluster is “Tomoyuki Yamaguchi”, with 387 link strength. This author has the maximum link strength with “Yu Fujita” (64), and he has also a higher link strength with “Mohammad R. Jahanshahi” (56). The other influential authors of this cluster include “Mohammad R. Jahanshahi” (314) and “Christian Koch” (255). Like the red cluster, the cited authors of the green cluster have higher citation linkage with the authors of the green cluster rather than the authors of the red cluster. So, this thing is different from the co-citation network of cited sources. In case of cited sources, few sources are so influential that they have a higher citation linkage with the sources of all the clusters.

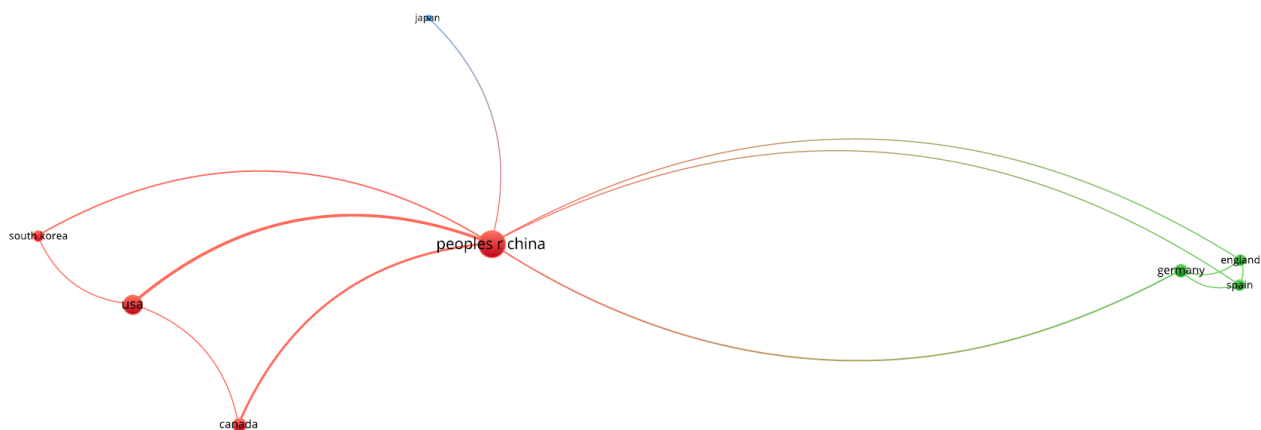
#### 4.3.2. Co-Authorship Analysis

Collaboration in research works is very important to produce creative ideas and implement them in an easier and smarter way, as one individual can find it too difficult to complete a research task. Co-authorship analysis is another technique that has been used in this work as a bibliometric measurement. In this section, this work will present a co-authorship analysis by using countries and institutions as the units of analysis to show the collaboration pattern among the authors of different countries and institutions. In the case of the co-authorship analysis of the countries, this work set a threshold of a minimum of 5 documents per country and found 8 countries among the 29 countries which satisfied the threshold. Table 9 presents the countries by total link strength. This table is ordered by the total link strength of the countries.

**Table 9.** Co-authorship indices of the countries.

Country	Documents	Total Link Strength
China	54	18
USA	29	9
Canada	5	5
Germany	7	4
England	5	3
South Korea	13	3
Spain	5	3
Japan	11	1

For a better analysis, we have generated the scientific landscape of the co-authorship network of the countries using the VOSviewer software (Figure 7).

**Figure 7.** Network visualization of co-authorship of the countries.

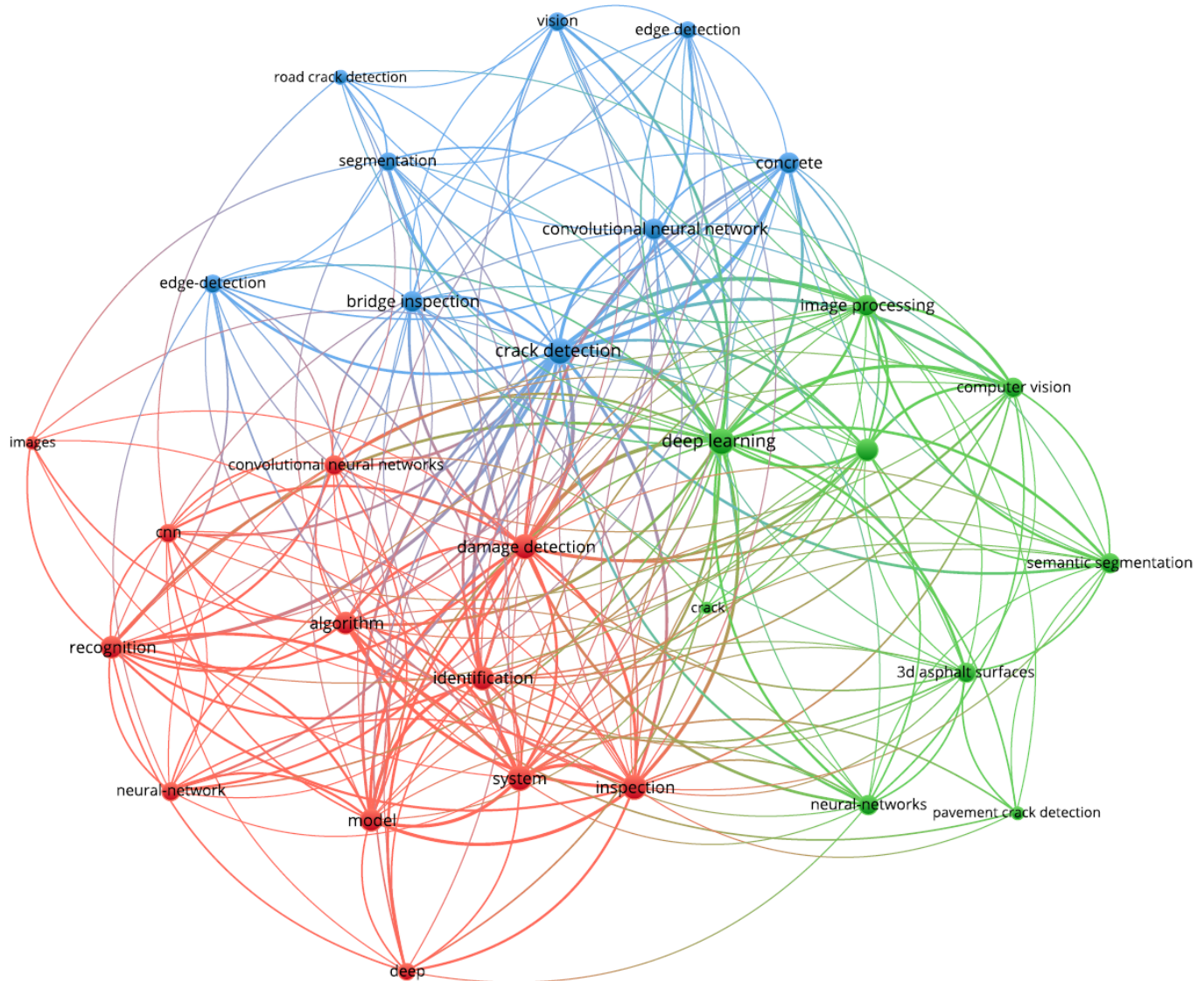
From Figure 7, it can be seen that the countries have been divided into a total of three clusters (red, green, blue). Each node in Figure 7 represents a country. The bigger the node, the more the country has collaborated with other countries. The connection line between the two countries reveals the presence of collaboration between the countries. The thicker the line, the more frequently the countries have collaborated. It is clear here that the red cluster is the prominent one among the clusters; it has four countries in total. Among the countries in the red cluster, “China” is the leading one, with a total link strength of 18. China has collaborated with all the other countries in Figure 7, which clearly indicates the productivity and significance of China. However, China has collaborated the most times (7) with the USA. The next influential country in the red cluster is the “USA”, with a total link strength of 9. However, the USA has collaborated with only three countries (China, South Korea, Canada). South Korea and Canada have collaborated with only China and the USA. In the green cluster, there are only three countries. In this cluster, all the countries are from Europe (Germany, Spain, England). These countries are collaborating with each other along with China, which implies that European researchers in this particular area generally collaborate with other European researchers. Finally, Japan is the only country in the blue cluster. Though Japan is an Asian country, it is not in the same cluster as China. However, Japan is collaborating with only China, with a link strength of 1.

#### 4.3.3. Co-Occurrence and Timeline View Analysis

Keywords of a research paper are very important tools for understanding the research topic of an article. The keywords are said to co-occur when they are present in a single



article. In this subsection, this work will present a co-occurrence analysis of the keywords to map the research trends and highlight the research hotspots in the field of image-based crack detection algorithms. From the total 129 publications, we have obtained 519 keywords altogether using VOSviewer. Among the 519 keywords, 30 keywords satisfied the threshold that we set as the least number of co-occurrences of a keyword (value 5). Figure 8 presents the network visualization of the publications' keyword co-occurrences.



**Figure 8.** Network visualization of co-occurrences of the keywords.

The keywords are presented by the nodes or circles in Figure 8. The size of a node reveals the weight or the number of occurrences of a keyword. The bigger nodes represent the most weighted or frequently occurring keywords. On the other hand, if any circle or node is small, then it means that the keyword has not occurred so many times in the publications. According to the terminology, it can be noticed from Figure 8 that “crack detection” is the keyword with the highest number of occurrences. A few other keywords with a higher number of occurrences include “deep learning” (25), “damage detection” (21), and “system” (17). The connection line between the nodes is also important information. If there is a line between two nodes, then it implies that these keywords appeared together. The thickness of these lines reveals the link strength; in other words, it indicates the number of co-occurrences between the keywords. The thicker the line is, the more co-occurrences

the keywords have. From Figure 8, it can be noticed that the keyword “crack detection” has the highest link strength (121) among the keywords. The node “crack detection” has a higher link strength or a thicker line with “deep learning” (11), “image processing” (9), “concrete” (9), and “damage detection” (9). The relationship of “deep learning”, “image processing”, “concrete”, and “damage detection” with “crack detection” implies the close integration between the keywords. It is a clear indication that during those crucial 10 years, deep learning-based image processing was highly utilized for detecting cracks and damage on reinforced concrete structures.

Another important thing in Figure 8 to notice is the distance among the nodes. The distance among the nodes represents the semantic similarity of the keywords. The keywords which have stronger similarity are situated within a shorter distance. In contrast, a longer distance denotes a lower similarity between the nodes. VOSviewer divides the keywords of a dataset into several clusters or sets of keywords based on their similarity. From the same figure, it can be observed that the keywords are divided into a total of three clusters denoted by three different colors (red, blue, and green). Table 10 summarizes the clusters.

**Table 10.** Summary of the resulting clusters related to keywords analysis.

Cluster Color	Observed Keywords	No. of Keywords
Red	damage detection, algorithm, identification, model, system, inspection, deep, convolutional neural network, neural-network, recognition, CNN, images	12
Green	deep learning, crack, neural-networks, pavement crack detection, 3d asphalt surfaces, semantic segmentation, structural health monitoring, computer vision, image processing	9
Blue	crack detection, concrete, edge detection, vision, bridge inspection, segmentation, edge-detection, road crack detection	9

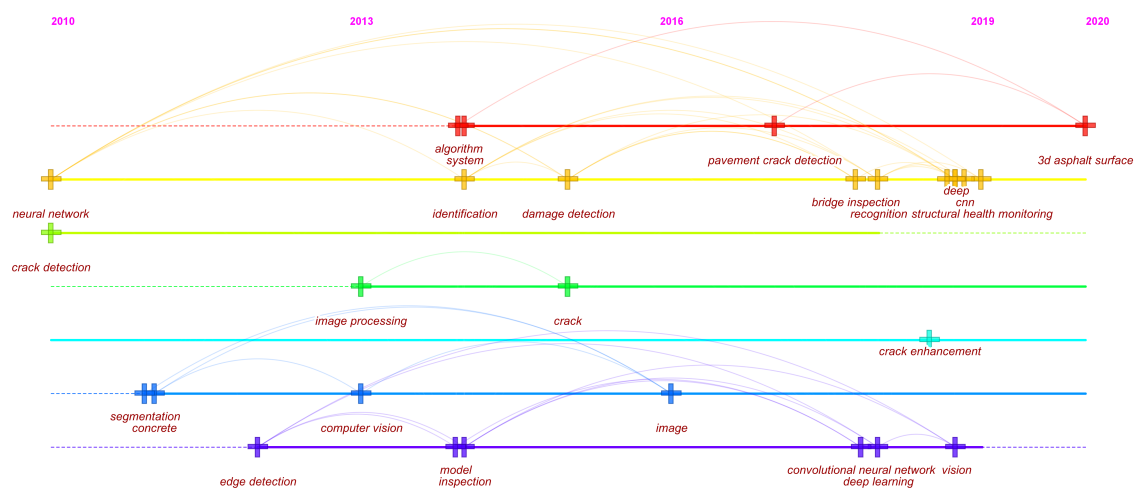
From Figure 8, it can be noticed that the red cluster is the prominent one among the clusters containing 12 keywords. The most frequent keyword in the red cluster is “damage detection” (21), which has a total of 25 links; this means that it co-occurred with 25 different keywords in the articles. The other notable keywords in this cluster include “system” (17), “algorithm” (16), “inspection” (14), and “model” (12), which highlight the technical and mathematical aspects in the case of damage or crack detection. The green cluster’s core is on “deep learning” (25), with a close linkage with other keywords such as “image processing” (18) and “computer vision” (9), highlighting the importance of deep learning-based IPTs for crack detection. The blue cluster connects “crack detection” (45), which is the most frequent keyword in the publications, with “concrete” (12), “segmentation” (11), and “convolutional neural network” (11), highlighting the crack detection and segmentation on reinforced concrete structures.

After analyzing the clusters, we have extracted the top 10 keywords in the publications. Table 11 summarizes the top 10 keywords with their frequencies, links, and link strength. This table is ordered by the frequency of the keywords. From Table 11, it can be observed that among the keywords, “crack detection”, “damage detection”, “system”, “inspection”, and “identification” are connected with all other keywords on the list, so each of them has nine links, which clearly indicates that these terms are closely integrated and indivisibly connected, as well as that they are the core keywords in this research area. One more noticeable thing from this table is that the keywords which have higher frequencies may not have higher link strength in all cases. For example, “image processing” is at number 4 on the list based on the frequency (18) and has 7 links, but its link strength is only 19, which indicates that it has not co-occurred so many times with other keywords.

**Table 11.** Summary of the top 10 keywords.

Keyword	Frequency	Links	Link Strength
crack detection	45	9	61
deep learning	25	9	44
damage detection	21	9	46
image processing	18	7	19
system	17	9	37
algorithm	16	7	27
inspection	14	9	39
model	12	7	33
identification	12	9	27
concrete	12	7	21

For this work, we have also made a timeline view of the keywords (which met the threshold) using the CiteSpace software to present the development trend of the important topics during 2010–2020 (Figure 9). From Figure 9, it can be noticed that there are a total of four stages in terms of time. In the first stage, from 2010 to 2013, the prominent keywords were “neural network”, “crack detection”, “image processing”, and “computer vision”. The research on crack detection in this stage was dependent on Neural Networks and image processing. In the later stage, from 2013 to 2016, the research on crack detection began to increase and especially focused on the mathematical models and technical aspects. For this reason, the prominent keywords of this period were “algorithm”, “system”, “model”, and “inspection”. The research on crack detection had a revolution during the third stage (2016–2019). The researchers started utilizing deep learning- and convolutional neural network-based techniques to detect cracks in reinforced concrete structures. The notable keywords of this period were “deep learning”, “convolutional neural network”, “pavement crack detection”, and “bridge inspection”. Finally, in the last stage from 2019 to 2020, the number of keywords is too low. In this stage, the specific focus was on “structural health monitoring” and “3d asphalt surfaces”. As per our observation, this happened due to the emergence of state-of-the-art technologies already in the third stage (2016–2019). These technologies were also employed in 2019–2020, and no newer technology was developed for detecting cracks during that time period. Table 12 lists the keywords of crack detection-related publications that occurred during the four different periods.

**Figure 9.** The timeline view of the keywords.

**Table 12.** The keywords of crack detection-related publications occurred during four different periods.

Periods	Keywords
2010–2013	neural network, crack detection, segmentation, concrete, edge detection, computer vision, image processing
2013–2016	algorithm, system, identification, damage detection, crack, image, model, inspection
2016–2019	pavement crack detection, bridge inspection, recognition, deep, cnn, crack enhancement, convolutional neural network, deep learning, vision
2019–2020	structural health monitoring, 3d asphalt surfaces

## 5. Critical Analysis

After analyzing the previous survey papers and performing bibliometric analysis based on keywords, we have found out that in recent years, DL methods are more viable and have received much more attention from researchers. As a consequence, we decided to present a critical analysis of the papers based on DL techniques (especially) for elucidating and acquiring knowledge on DL methods used for crack detection. Therefore, after omitting the articles based on traditional techniques by using the methodology described in Section 3, this work ended up with 65 papers (Figure 2). We have grouped these 65 papers based on the type of computer vision technique used in them, i.e., classification, object detection, and segmentation. Then, we analyzed the 65 papers based on their problem statements, methodologies, and results. After summarizing the papers, we have brought forward some questions as follows:

- Q1. Which DL method is used in an article?
- Q2. Which backbone is used by the DL method?
- Q3. Which DL library is used by an article?
- Q4. Which datasets are used by an article?
- Q5. Which concrete surface is taken into consideration by an article?
- Q6. Which loss function is used in an article?
- Q7. Which optimizer is used in an article?
- Q8. Which annotation tool is used by an article?
- Q9. What performances are achieved by an article?

The answers to these are summarized in Tables 13–15 for the papers of different categories.

### 5.1. Classification

In [49], Tran et al. presented a two-step sequential Mask region-based Convolutional Neural Network (Mask-RCNN) model to classify pavement crack type and severity level of the cracks. The authors trained, validated, and tested their model using 32,563 images which were collected by a CMOS vision sensor mounted on a road screening vehicle. After completing the training process, the model was able to classify three types of cracks (i.e., longitudinal, transverse, fatigue) as well as the severity level of the cracks (i.e., low, medium, high). Tran et al. mentioned that the model was 92.10% accurate and showed 96.32% and 94.67% average precision and recall, respectively. The authors compared their methods with a few other classification techniques and showed that their model was more accurate than others, and they also claimed that they performed crack classification problems with the highest (nine) number of classes. Moreover, the authors measured the widths of the cracks; though the predicted width was slightly different from the original width that they considered, this error, however, is acceptable.

In [50], Wang et al. proposed a new framework for detecting cracks by fine-tuning the AlexNet architecture. The authors considered the class imbalance problem and the presence of disturbance on non-crack images and solved the issues by developing an active learning method. The proposed framework used a sliding window approach to filter out



the images and divided one image into four training images, which increased the number of training images and facilitated the classification task. Wang et al. trained their model and obtained 97.55% accuracy. In addition, the authors compared their model with ChaNet and showed that their method outperformed ChaNet in terms of all evaluation metrics.

In [51], Zhang et al. proposed a hybrid method based on IoT (Internet of Things) technology and a CNN model for classifying cracks as well as monitoring the structural health condition of concrete bridges in real time. In this work, Zhang et al. first preprocessed the crack images by converting the images into grayscale, increasing the contrast using a piecewise linear function, and denoising the images using wavelet transformation. Then, the authors developed a CNN model to classify different types of cracks (i.e., small, large, serious cracks) for measuring the severity level of the cracks. Zhang et al. trained their model with 300 images and obtained an accuracy of more than 90%.

In [52], Dung et al. detected cracks on the joints of bridges using three deep learning methods. Firstly, the authors developed an SCNN model from scratch to classify crack images. Secondly, the authors utilized a pre-trained VGG-16 model and finally fine-tuned the top layers of the VGG-16 model for detecting cracks. The authors trained the model by using the images collected from a previous fatigue crack inspection at Tokyo University and demonstrated that the third method outperformed the previous two methods in terms of accuracy (97%), while the other two methods produced 90% and 94% accuracy, respectively. They also mentioned that data augmentation helped to increase the accuracy at a rate of 5%, 2%, and 5% of the models, respectively.

In [35], Cha et al. presented a CNN-based approach to classify concrete images as “crack” or “non-crack”. The authors designed their model with 4 convolution layers, 2 pooling layers, and 1 softmax layer and trained the model with 332 images collected from a building of the University of Manitoba, Canada. After training the model, Cha et al. tested their model with 55 images and the model was able to classify the images successfully with an accuracy of 98.22%. They compared their model with the Canny edge detector and Sobel operator and demonstrated the advantage of the CNN model over the traditional models.

In [53], Nehdi et al. presented a classifier based on CNN and Otsu image processing techniques for classifying the presence of cracks and the position of cracks in concrete structures. The authors trained their model with 20,000 images and achieved 96.17% accuracy. Nehdi et al. developed their classifier in such a way that it can classify three things (i.e., crack or not, the position of cracks, a combination of two) simultaneously. The authors also quantify the length, width, and angle of the cracks successfully.

In [37], Dorafshan et al. compared the performance of a few conventional edge detectors (i.e., Roberts, Prewitt, Sobel, etc.) methods with Deep Convolutional Neural Networks (DCNNs) for classifying cracks in concrete images. The authors considered AlexNet as the DCNN model and trained it in three different modes (i.e., Fine-tuned, Transfer learning, and classifier). Dorafshan et al. mentioned that the AlexNet models achieved 97–98% classification accuracy, whereas the traditional methods managed to achieve 53–79% accuracy. They also showed that AlexNet with a transfer learning scheme obtained the highest accuracy.

In [54], Gopalkrishnan et al. deployed a DCNN framework based on a truncated VGG-16 model for classifying asphalt and Portland cement concrete images as “crack” or “non-crack”. The authors utilized a subset of images from the dataset of FHWA and LPTT programs. Then, they extracted features using the pre-trained VGG-16 model and classified crack images using different classifiers (i.e., NN classifier, Random Forest, SVM, Logistic Regression). Gopalkrishnan et al. demonstrated that the NN classifier along with the VGG-16 model obtained the highest (90%) classification accuracy.

In [55], Yang et al. presented a transfer learning method based on VGG-16 to classify cracks on civil infrastructures. They trained their model on three different datasets (i.e., CCIC, BCD, SDNET) and obtained an accuracy of 99.83%, 99.72%, and 97.07%, respectively. Yang et al. presented three modes of transfer learning (i.e., sample transfer

learning, model transfer learning, and parameter transfer learning) and demonstrated that the parameter transfer learning mode is the best among all three models.

In [56], Li et al. presented a deep learning model by modifying the AlexNet architecture and integrating the Exhaustive Search Technique into the CNN model for classifying cracks on concrete structures. The authors prepared a dataset containing 60,000 images and after training the model, they achieved 99.07% accuracy. In addition, Lie et al. integrated their trained model into a mobile phone application so that people can detect cracks easily.

**Table 13.** Summary of Deep Learning techniques for crack classification.

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Performance (%)
[49]	Mask RCNN	ResNet-101	Keras, Tensorflow	Own collection	Asphalt pavement	-	Adam	Accuracy = 92.10%, Precision = 96.32%, Recall = 94.67%
[50]	CNN	AlexNet	MATLAB	Own collection	Concrete structure	Cross entropy	-	Accuracy = 97.22%, Precision = 90.53%, Recall = 83.37%, F1-score = 84.63%
[51]	CNN	-	MATLAB 2016A	Own collection	Bridge deck	-	Back propagation	Accuracy = 90%
[52]	CNN	VGG16	-	Metropolitan expressway Co. Ltd.	Bridge deck	Binary cross-entropy	Stochastic Gradient Descent (SGD)	Accuracy = 97%
[35]	CNN	MatConvNet	-	Own collection from University of Manitoba	Concrete structures	Softmax loss function	SGD	Accuracy = 98%
[53]	CNN	-	-	CCIC	Concrete structures	Binary cross-entropy	Natural Gradient Descent (NGD)	Accuracy = 96.17%
[37]	DCNN	AlexNet	MATLAB 2018A	Simulated panels from SMASH Lab	Bridge deck	-	-	Accuracy = 97%
[54]	CNN	VGG16	Keras	FHWA, LTPP	Pavement	-	Adam	Accuracy = 90%, Cohen's Kappa score = 74.2%
[55]	DCNN	VGG16	Keras	CCIC, SDNET, BCD	Concrete structures	-	-	Accuracy = 99.15%, 92.59%, 98.97%
[56]	CNN	AlexNet	Caffe	Own collection	Concrete structures	Softmax loss function	SGD	Accuracy = 99.06%

'-' denotes the paper did not provide the particular information.

## 5.2. Detection

In [57], Deng et al. utilized a modified form of the You Only Look Once (YOLO) version 2 algorithm where the base extractor was the VGG-16 architecture for detecting cracks using bounding boxes on concrete surfaces having a complex background for the first time. The authors took two classes (i.e., crack, handwriting scripts) into account and gathered images to train their model for distinguishing cracks from handwritten scripts on concrete surfaces. After training the model, the authors evaluated the model's performance and robustness by using three images containing both cracks and handwritten scripts. Deng et al. showed that their model successfully detects objects like cracks and handwritten scripts with confidence scores of each class. The authors also compared their model with the Faster-RCNN model and observed that their model outperformed the Faster-RCNN model in terms of accuracy and inference speed.

In [58], Li et al. designed a convolutional neural network called Skip-Squeeze-and-Excitation Networks (SSEnets) by embedding Skip-Squeeze-Excitation (SSE) module and



the Atrous Spatial Pyramid Pooling (ASPP) module with CNN layers for detecting cracks on concrete bridges. The authors utilized the SSE module for reducing the vanishing gradient problem as well as the computational complexity of the deeper network. They used an ASPP module containing a depthwise separable convolutional layer for extracting features of a crack image in a multi-scale rate which facilitated the model to detect the cracks with different widths. Li et al. trained their model with the Xu et al. dataset [59] and got 97.77% accuracy for detecting cracks. After detecting crack successfully, the authors compared their model with Xu's model and ResNets model and they showed that their model performed better in terms of accuracy, F1 score, and running time.

In [60], Chen et al. presented a deep learning framework consisted of a CNN model and Naive-Bayes decision-making algorithm for detecting cracks with bounding boxes at underwater nuclear power plant surfaces. The authors maintained the spatiotemporal coherence using the CNN and discarded false-positive samples by the Naive-Bayes fusion scheme. The authors demonstrated that their model achieved a 98.3% hit rate while the false positive rate was only 0.1%.

In [61], Park et al. utilized tiny YOLO for detecting cracks in real-time on concrete structures. In this work, the authors also quantify the crack images using structured light projected from laser beams. To avoid the installment error of the sensors, Park et al. used the Jig module and successfully measured the lengths and the widths of the cracks. They claimed that their model successfully detected cracks with bounding boxes where the model achieved 94% accuracy and 98% precision.

In [62], Majdifard et al. presented a hybrid method based on YOLO v2 and U-Net for segmenting cracks and detecting crack severity on road images. In a previous work, Majdifard et al. built a dataset and trained their model using YOLO V2. However, in this later work, they integrated U-net and detected different types of cracks (i.e., alligator crack, block crack, longitudinal crack) accurately by eliminating the presence of shadow problem, presence of cars on the roads, and various other problems. The authors also compared their method's condition index with PASER (Pavement Surface Evaluation and Rating) ratings.

In [63], Deng et al. proposed a faster region-based convolutional neural network (Faster-RCNN) for detecting cracks on the concrete bridges with contaminated backgrounds (i.e., presence of handwritten scripts). The Faster-RCNN model consisted of RPN which was utilized to generate bounding boxes and the Fast-RCN model for detecting cracks and handwritten scripts. The authors extracted the crack features by using Zeiler-Fergus Network (ZF-Net) as the base CNN model for which 5 shareable layers were shared by both RPN and Fast-RCN. The authors trained their model with 160 images and demonstrated that their model can successfully detect tiny cracks and handwritten scripts. Deng et al. also compared their model with YOLO V2 and claimed that their model is superior.

In [64], Li et al. proposed a crack detection framework named Crack Deep Network (CrackDN) for detecting sealed and unsealed cracks on road images. The authors developed the CrackDN model based on a FRCNN architecture where they used ZF-Net as the feature extraction module. Li et al. integrated a sensitivity detection network along with ZF-Net and added them to a Region Proposal Refinement Network (RPRN) for detecting the cracks. The authors collected images by mounting smartphones and cameras and trained their model using the images of different complex backgrounds (i.e., variation of illuminations, shadings, and markings on-road). Li et al. evaluated their model in terms of accuracy, precision, and recall and also compared the performance with faster-RCNN and SSD300. They demonstrated that their model performed better than the compared models with respect to all evaluated metrics.

In [65], Ma et al. presented an FCN model based on ResNet-101 to detect and localize cracks on pavement images. The proposed model utilized the base model (ResNet-101) for feature extraction, RPN for predicting the cracks as well as generating the bounding boxes, and a position-sensitive ROI (Region of Interest) pooling for predicting the output map. The authors trained their model by using both CCIC and CIDB datasets and got 91.4% and 86.4% accuracy respectively. Ma et al. examined the influence of data augmentation

and online hard example mining techniques on their model and claimed that using the techniques increased their model's accuracy.

In [66], Co et al. presented a model based on Alex-Net for detecting cracks on concrete surfaces. They collected images from concrete surfaces and categorized them into a total of five classes including cracks, plants, intact surfaces, two crack-like patterns. They trained the model using transfer learning and fine-tuned the Alex-Net. After classifying the images, Cho et al. utilized a probability map in the third stage and detected the cracks with bounding boxes. Their model produced accuracy, precision, and recall of 98%, 86.73% and 88.68% respectively. They utilized a drone to capture real images and performed an on-site experiment. The authors demonstrated that their model successfully detected cracks except for only one thin crack.

In [67], Chang et al. compared the performance of eight different deep learning models for detecting different types of cracks on road images. The authors considered 4 models based on SSD (i.e., SSD MobileNet-v1, SSD MobileNet-V2, SSD inception V2, SSDLite-MobileNet-V2) and 4 models based on faster R-CNN (i.e., Faster RCNN inception V2, Faster R-CNN ResNet-50, Faster-RCNN ResNet-101, Faster RCNN Inception ResNet-V2) for comparison. The authors utilized a dataset containing 15,435 images to train and evaluate the models. Cheng et al. demonstrated that the Faster-RCNN models performed better in terms of mAP (mean Average Precision) while SSD models are faster in terms of inference time than the Faster RCNN models.

In [68], Li et al. presented a DL model based on coarse-to-fine region localization method for detecting cracks on concrete tunnels. They collected images from different tunnels and annotated as well as processed them. After that, the authors designed a Faster RCNN model with 5 regular CNN layers, an RPN layer for localizing cracks, and an ROI pooling layer for classifying cracks. Finally, Li et al. deployed an edge detection-based method based on a median filter for generating fine detection of the results. They trained their model with their own dataset and their model detected cracks successfully with 93.6% mAP.

**Table 14.** Summary of Deep Learning techniques for crack detection.

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Annotation Tool	Performance (%)
[57]	YOLO v2	VGG16	MATLAB 2019A	Own collection	Concrete bridges	SDGM	SGD	Labeler app in MATLAB	map = 77%
[58]	SSEnets	VGG16	Pytorch	Bridge crack dataset of Xu et al. [59]	Bridge deck	-	SGD	-	Accuracy = 97.77%, Precision = 95.45%, Recall = 97.67%
[60]	NB-CNN	Customized CNN	-	Own collection	Nuclear power plant	-	-	Manually	Hit rate = 98.3%, AUC = 79.2%
[61]	YOLO V3-tiny	Customized CNN	-	Own collection	Concrete structures	-	-	YOLO V3-tiny	Accuracy = 94%, Precision = 98%
[62]	YOLO v2 + U-Net	-	-	Own collection	Pavement	-	GEP	Python based software	F1-score = 84%, Precision = 93%, Recall = 77%
[63]	Faster R-CNN	ZF-net	MATLAB 2018B	Own collection	Bridge	Momentum loss function	SGD	Labeler app in MATLAB	mAP = 79%, F1 = 67%
[64]	CrackDN	Customized CNN	Tensorflow	Own collection	Pavement	Multitask loss function	-	-	Accuracy = 85%
[65]	FCN	ResNet-101	Caffe	CCIC,CIDB	Pavement	Binary cross-entropy	SGD	Manually	Accuracy = 91.4%, 86.4%
[66]	CNN	AlexNet	MATLAB	Own collection	Concrete structures	-	-	-	Average precision = 86.73%, Average recall = 88.68%
[67]	Faster R-CNN, SSD	-	-	BCD	Road	-	-	-	mAP = 27.66%, 19.45%
[68]	Faster R-CNN	ZF-Net	Caffe	Own collection	Tunnel	Regression loss	SGD	Manually	mAP = 93.6%

‘-’ denotes the paper did not provide the particular information.

### 5.3. Segmentation

In [69], Li et al. presented a convolutional encoder–decoder network (CedNet) for detecting the cracks at the pixel level by using the DenseNet-121 architecture as the encoder part of the proposed CedNet. In this work, the authors built a dataset for crack detection including 1800 images and trained the model by utilizing their own dataset. After successfully detecting the cracks with 98.90% accuracy, the authors performed perspective transformation to correctly construct the distorted predicted images. They also measured the width of the cracks and determined the orientation of the cracks by employing the Euclidean distance transformation and least square principle, respectively. Li et al. compared their model's performance with Mask-RCNN as well as FCN and showed that their model was able to detect cracks more accurately than those two models even if the crack was thin.

In [70], Huyan et al. proposed an encoder–decoder-based architecture named CrackU-Net by improving the “U”-shaped model U-Net for detecting cracks on pavement images. the authors deployed a 3D data collection system for building the dataset, which consisted of 3000 images. In this work, Huyan et al. took the problem of false positive crack detection into consideration and successfully improved it using their model. The proposed model produced 99.01% accuracy and it outperformed some well-known traditional methods (e.g., Sobel, Roberts, LG) as well as FCN and U-Net for the pixel-level segmentation of pavement crack images.

In [71], Chen et al. exploited the rotation-invariant property of the cracks for the first time and as a consequence, they integrated active rotating filters (ARFs) with an FCN model named DeepCrack and presented a new model called ARF-crack for detecting cracks at the pixel level. The authors assessed their model on four different benchmark datasets including DeepCrack, CFD, Crack500, and GAPS384. The authors presented visually that their model was able to segment cracks accurately for all the datasets and they claimed through some numerical results (e.g., average precision, recall) that their model outperformed the DeepCrack, FPHBN, and IRA-Crack models. They also mentioned that the proposed model needs a fewer number of parameters and less time to be trained.

In [72], Pan et al. developed a deep learning model named spatial-channel hierarchical network (SCHNet) by employing the VGG-19 model as the baseline for segmenting cracks in reinforced concrete structures. The authors integrated a self-attention mechanism with their proposed model by running three different modules (feature pyramid attention module, spatial attention module, and channel attention module) to establish a relationship between pixels and improve the reliability of crack segmentation. Pan et al. trained their model with the SDNET2018 dataset and selected Mean IOU as the evaluation metrics of their task. They mentioned that usage of each attention module increased the model's IoU (Intersection over Union) gradually and it finally ended up at 85.31%. The authors compared their model with a few other state-of-the-art methods and ensured that their model was the superior one. They also tested their model under various conditions (i.e., holes, shadow on the surfaces, rough surface) and each and every time, their model successfully segmented the cracks.

In [73], Kalfarisi et al. presented two deep learning methods; one is FRCNN-FED, which is a combination of faster region-based convolutional neural network (FRCNN) and structured random forest edge detection (SRFED) methods, and the other is Mask-RCNN. The authors attempted to detect cracks using bounding boxes and segment the cracks simultaneously. Kalfarisi et al. trained their model with some images which were collected during some real-life structure inspection and also evaluated their technique's performance by detecting the cracks from the images of roads, bridges, buildings, and so on. They claimed that their model was able to detect cracks as well as measure the length and width of cracks successfully.

In [74], Lee et al. presented a semi-supervised learning method for detecting cracks in concrete structures. To reduce the cost of acquiring a vast quantity of data for supervised learning, Lee et al. developed an adversarial network to produce labeled confidence maps from unlabelled images. After that, the authors applied a multiscale segmentation learning

network instead of encoding decoder architecture to segment crack images efficiently. The authors trained their method using METU and USU datasets and achieved 98.176% accuracy. To show the robustness of the method, Lee et al. compared their technique with a few other encoder–decoder-based models, and their method outperformed all of the compared methods in terms of all evaluated metrics.

In [75], Fan et al. modified the U-Net architecture by embedding a Hierarchical feature learning (HF) module and a multi-dilation module (MDM) and proposed a novel framework named U-Hierarchical Dilated network (U-HDN) for detecting cracks on asphalt pavements at the pixel level. The authors employed MDM with different dilation rates for extracting crack features of different context sizes and HF for predicting the feature maps on different scales and fused them to obtain an accurate segmented image of pavement cracks. Li et al. trained their model on both CFD and AgileRN datasets and their model showed better precision and recall values than all the compared methods.

In [76], Gil et al. developed a novel method named ConnCrack by combining a conditional Wasserstein generative adversarial network (cWGAN) and connectivity maps for crack detection on pavement images. In this work, the authors also published a dataset named EdmCrack600 containing 600 images and trained their model with both EdmCrack600 and CFD datasets. The authors evaluated their model in terms of precision, recall, and F1-score. They compared their model's performance with a few conventional methods (i.e., Canny, CrackTree, CrackForest) and deep learning methods (ResNet 152-FCN, VGG19-FCN, Cracknet-V). Gil et al. demonstrated that their model outperformed other methods by means of all parameters; however, they noticed that the model performed better using the CFD dataset than the EdmCrack600 dataset.

In [77], Alipour et al. presented a fully convolutional neural network named Crackpix based on the VGG-16 architecture in order to perform semantic segmentation of cracks on concrete structures. The authors employed five FCN architectures (i.e., FCN32s, FCN16s, FCN8s, FCN4s, FCN2s) and trained their model using images collected from several bridges, roads, and building surfaces. The method was able to segment cracks successfully with 92.17% validation accuracy, and the authors claimed that it was the first FCN model which could segment images of arbitrary sizes.

In [78], Ji et al. utilized DeepLabV3+ for segmenting cracks on pavement images. The authors also deployed a crack quantification algorithm named the fast parallel training (FPT) algorithm for calculating the length, width, area, and ratio of the cracks. Ji et al. trained their model using a dataset of 300 images and the model successfully segmented several types of cracks (i.e., single crack, multiple crack, intersecting crack, alligator crack). They evaluated their model by means of the MIoU metric, which was calculated as 0.7331. Ji et al. compared their model with few other state-of-the-art deep learning models (i.e., FCN, DeepCrack, Encoder–Decoder). The authors demonstrated that their model could predict unseen crack images better than all the compared methods.

In [79], Wei et al. designed an algorithm based on GAN (Generative Adversarial Network) and neural style transfer for detecting cracks on road images. The authors produced trained images from only one sample crack image using the GAN simulator. Then, Wei et al. utilized a segmentation algorithm named Seg which produced an F1-score of 0.82 and successfully predicted the cracks at the pixel level.

In [80], Lau et al. presented a U-Net model in which the encoder is a ResNet-34 architecture for segmenting cracks on pavement images. The authors trained their model on both CFD and crack500 datasets. After completing the training session, the authors demonstrated that their model produced F1-scores of 96% and 73% for the CFD and crack500 datasets, respectively, as well as predicted the pixels which would contain cracks on pavement images successfully. Lau et al. compared their method with a few other U-Net-based models and the FCN model and showed that their method performed better than all the compared methods according to precision, recall, and F1-score. Then, the authors performed several ablation techniques (i.e., training the model using frozen layers and not using the frozen layers, using the SCSE module and not using the SCSE module) to check

the increase in the performance and robustness of the model. The authors demonstrated with numerical values that the usage of the ablation studies helped the model perform better.

In [81], Song et al. presented a deep learning model based on ResNet in which a multiscale dilated attention (MDA) module and feature fusion upsampling (FFU) modules are embedded to detect cracks at the pixel level on pavement images. The authors utilized the MDA module for extracting high-level features and the FFU for restoring the crack spatial resolution. Song et al. trained their model using the dataset named “CrackDataset” and produced higher precision, recall, and F1-score than a few other state-of-the-art deep learning methods, such as SegNet, U-Net, PSPNet, DL-V3+, and DFN. After detecting the cracks at the pixel level, Song et al. classified the types of cracks (i.e., Transversal, Longitudinal, Block, Alligator) and severity level of cracks (i.e., normal, medium, high) by identifying the branches and measuring the height as well as weight of the cracks. The authors demonstrated that their model obtained over 95% accuracy in terms of classifying the cracks.

In [36], Dong et al. proposed an encoder–decoder-based FCN network for semantic segmentation of cracks on concrete surfaces. The authors selected the encoder architecture of the FCN model by conducting a classification task on a crack dataset using different deep learning models (i.e., VGG-16, InceptionV3, ResNet). VGG-16 outperformed the two other models in terms of classification accuracy and was deployed as the encoder of the FCN model. Dong et al. trained the FCN model using 600 annotated images and the model successfully detected crack and non-crack pixels on concrete images with an average precision of 90.9%.

In [82], Bang et al. developed a fully convolutional encoder–decoder-based network where the ResNet-152 architecture was used as the encoder for detecting cracks on black box images at the pixel level. The authors installed some black box cameras on the vehicles and collected images extracted from videos captured by the cameras to train their model. The authors demonstrated that their model successfully segmented the crack images with a recall and precision of 71.98% and 77.68%, respectively. The authors also compared their model with other pre-trained networks such as VGG-16, ResNet-50, ResNet-101, and SegNet and showed that their model outperformed all the compared methods according to all the compared evaluation metrics.

In [83], Yao et al. presented a novel concept to reduce the computational complexity of the encoder–decoder-based architecture for detecting concrete cracks at the pixel level. They proposed a switching module named SWT consisting of a binary classification header that would classify crack and non-crack images and would pass only the positive samples to the decoder module while directly outputting the negative map, without passing the samples to the decoder module. Yao et al. integrated their switching concept on U-Net and the DeepCrack model by utilizing the datasets CrackTree 206 and as well as AIMCrack. The authors demonstrated that their method did not diminish the performance, and also reduced the computation time and computation complexities by both quantitative and qualitative analysis. At the end, they showed that U-Net and the DeepCrack model ran about 30.7% and 62.9% faster with SWT than without SWT.

In [84], Cai et al. developed an FCN model named pavement and bridge crack segmentation network (PCSN-512) by modifying the SegNet architecture for performing semantic segmentation on the crack images of pavement and bridge decks. The authors built a dataset of 5000 images and trained their model using the “Adadelta” crack images with perplexing backgrounds and also by comparing the method with a few other state-of-the-art networks (i.e., FCN, MRCNN, PCSN). The authors demonstrated that the proposed PCSN-512 segmented the images successfully with 93% accuracy and outperformed the compared models in terms of inference time, precision, and recall.

In [85], Liu et al. utilized U-Net for the first time to detect cracks on concrete images. The authors collected a total of 84 images from Huazong University, China, and trained their model with the Adam optimizer. Liu et al. evaluated their model by means of three



metrics (i.e., precision, recall, F1-score) and also compared their model with Cha's CNN [35] and an FCN model. The authors claimed and demonstrated by quantitative analysis that their model is better than all the compared methods (that they selected).

In [86], Qu et al. performed both classification and semantic segmentation on pavement crack images. The authors fine-tuned the LeNet-5 model for the classification task and modified the VGG-16 model by reducing some convolution layers, adding a  $1 \times 1$ -Conv layer after an Eltwise layer, using the horizontal expansion method for detecting cracks at the pixel level. Qu et al. also built two datasets named CCD1500 and CCD861 in this work. They trained their model with CCD861, CFD, DeepCrack, and Crack200 datasets and demonstrated that their model performed better than all the compared methods (i.e., VGG-16, U-Net, Percolation) for each dataset.

In [87], Fan et al. deployed an ensemble learning technique on a CNN to detect cracks in pavement images. The authors used only convolution layers and fully connected layers without any pooling layers in each individual CNN model, as the pooling layer loses important pixel information. The authors averaged the output of each CNN model and presented the predicted pavement images. Fan et al. trained their model on both CFD and AgileRN datasets; they conducted an experiment with the number of CNN models to be ensembled and finally selected three CNN models, as they obtained the highest results from three ensembled CNN models for both datasets. They also compared their model with a few other state-of-the-art methods and claimed that their model outperformed all the compared methods in terms of precision, recall, and F1-score.

In [88], Feng et al. presented a novel method based on the U-Net architecture for detecting cracks on road images. The authors added residual identity blocks on the U-Net and passed the extracted information of different layers to the final layer by adding the weighted values of the pixel so that no original information could be lost. The authors trained their model on the CFD dataset and demonstrated that it achieved precision, recall, F1-score, and dice coefficient of 94.29%, 99.36%, 96.76%, and 86.95%, respectively.

In [89], Shen et al. developed a deep learning framework named CrackSegNet for detecting cracks on concrete tunnels. The authors designed their model based on U-Net and by adding dilated convolution layers on the encoder stage and integrating a Spatial Pyramid pooling Module (SPP) at the end of the encoder stage. They trained their model on images collected from Zhejiang province, China. The authors experimented with their framework by using different forms (i.e., dilated convolution, skip connection, SPP) and demonstrated that their model performed better by using dilated convolution layers in terms of all evaluation metrics (IoU, precision, recall, F1-score).

In [90], Alipour et al. developed a deep learning framework based on the Res-18 architecture to detect cracks on multiple types of infrastructure. In this work, the authors attempted to develop a model which would be able to achieve good accuracy on any kind of surface (i.e., concrete surface, asphalt surface). Alipour et al. presented three schemes (i.e., joint training, sequential training, ensemble training) to endow their model with adaptivity. The authors trained their model with two different datasets and showed that the joint training method obtained the highest accuracy (97.8%). They also demonstrated that their model outperformed two material-specific models (i.e., Cha et al. [35] and Eisenbach et al. [91]) in terms of accuracy.

In [92], Wu et al. presented a deep learning model named cascade mask region convolutional neural network (cascade mask RCNN) for segmenting and detecting cracks simultaneously on stay cables of bridges. The authors used an inspection robot to collect images from stay cables and trained their model with the images. Wu et al. showed that their model achieved an IoU index of 74.3 on infrastructure and was able to detect cracks successfully. The authors compared their model with U-Net, PSPnet, FCN8s, Linknet, and Enet and demonstrated that their model outperformed the compared methods in terms of IoU, recall, and F1-score, while a few other models were better than the proposed model with respect to accuracy and precision.



In [93], Zheng et al. compared the performance of several state-of-the-art deep learning models (i.e., RFCN, FCN, RCNN) for detecting cracks on building images. The authors analyzed the working principle of each model and presented the quantitative results of the models in the case of predicting crack images. Zheng et al. considered accuracy, precision, recall, and IoU as the evaluation metrics and demonstrated that RFCN is superior among the models.

In [94], Manjurul et al. employed an FCN model by using the VGG-16 architecture as an encoder to predict cracks on concrete structures at the pixel level. The authors trained their model using the dataset of the Middle East Technical University and examined the performance of the model in terms of precision (91.3%), recall (94.1%), and F1-score (92.7%). They demonstrated that their model can predict the cracks successfully and compared their model with SVM and CNN. Manjurul et al. showed that their model outperformed the compared models in terms of all evaluated metrics.

In [95], Hoon et al. presented a deep learning method based on U-Net to detect cracks on underground tunnel images. For improving the performance of the framework, Hoon et al. added a ternary classifier to the U-Net for reducing the number of false positives. The authors collected images from several tunnels in Korea (i.e., Masung, Habuncheon, Sangock) and trained their model using the images to detect cracks. Hoon et al. compared their model with U-Net, Att-Unet, and DeeplabV3+ and demonstrated that their model obtained the highest precision (88%, 75%), recall (47%, 45%), and F1-score (61%, 56%) for the images of Sangock and Habuncheon tunnels.

In [96], Li et al. presented a two-stage deep learning model for detecting cracks on concrete bridges. In the first state, the authors used smaller receptive fields ( $3 \times 3$ ) and smaller sizes of images ( $18 \times 18$ ) to produce the confidence map. In the second stage, the utilized model was the same but the input size was the output ( $64 \times 64$ ) of the first stage and the receptive field was also bigger ( $5 \times 5$ ). After producing the confidence map, they fused it with the previous one and finally obtained the predicted result. The authors used convolution layers and three densely connected layers in the DL model for extracting features. They collected 65 images from different bridges and trained the model using the images. Li et al. compared their model with STRUM and the Canny edge detector and demonstrated that their model outperformed the compared methods in terms of accuracy (99.55%) and precision (78.49%).

In [97], Liu et al. presented a deep learning framework named NB-FCN consisting of a VGG-16 architecture and a naive Bayes decision technique. The FCN model extracted essential features to recognize and segment crack images. In addition, the authors used a naive Bayes probability fusion scheme to again classify the crack images for reducing the false detection rate. The authors utilized a device named Bridge Substructure Detection (BSD-10) to collect images from different bridges and trained their model using the SGD algorithm. The special characteristic of this model is that it can detect cracks successfully with different kinds of complexities on the surface (handwriting, water stains, peel-off). Liu et al. compared their model with the CrackTree algorithm, Random Structured Forest algorithm, CNN and demonstrated that their model is superior in terms of accuracy and inference time.

In [98], Pan et al. detected cracks on the U-rib-to-deck welded joint area of bridges by proposing a deep learning algorithm based on VGG-Net. The authors also tested the performance of different models including ResNet, Deeplab, and PSPnet and demonstrated that their model works better in terms of precision and recall.

In [99], Wang et al. presented a DL named CrackNet for detecting cracks on 3D asphalt surfaces. The authors developed the CrackNet model with one input layer, two convolution layers, and two fully connected layers. However, the authors did not use any pooling layers; rather, they compared each pixel with its neighboring pixels to achieve pixel-level accuracy. Wang et al. trained their model with 1800 images and tested their model with 200 images. They observed that their model achieved 90.13%, 87.63%, and 88.86% for precision, recall, and F1-score, respectively. They also compared their model's performance with

Pixel-SVM and 3D shadow modeling and showed that their model performed better than the other two methods.

In [100], Wang et al. proposed a recurrent neural network (RNN)-based model named CrackNet-R for segmenting cracks on pavement images. The authors employed a novel recurrent unit called gated recurrent multilayer perceptron (GRMLP) instead of LSTM and GRU for obtaining deeper abstraction, as it conducted multilayer transformation at each gating unit. The authors trained their model by using the images extracted from the PaveVision3D system. After a successful training session, the model achieved 93.06% segmentation accuracy. Wang et al. compared their model with CrackNet, CrackNet-LSTM, and CrackNet-GRU and showed that their model outperformed the compared methods in terms of accuracy, precision, and recall, and it was also four times faster.

In [101], Li et al. presented an FCN model using VGG19 as the encoder architecture for detecting cracks on concrete structures. The authors collected 800 crack images from different roads and building walls for training their model. The authors mentioned that their model could segment cracks with 97.96% accuracy, 81.73% precision, and 78.97% recall, though their model was less accurate than the CrackNet model. After successfully detecting the cracks, Li et al. predicted the crack skeleton and measured crack height and width with a minimum error rate.

In [102], Zhang et al. presented a context-aware-based segmentation network for detecting cracks in concrete structures. First, the authors utilized the sliding window approach for localizing image patches, and then the authors employed SegNet to classify crack pixels from the image patches. Finally, Zhang et al. proposed and deployed a context-aware overlapping patch fusion (CAOPF) scheme for integrating the output of every patch to generate a final output map. The authors tested their model on three different datasets and achieved an F1-score of 82.34%, 82.52%, and 79.37%, respectively.

In [103], Xiang et al. presented an encoder–decoder architecture based on FCN integrated with a pyramid pooling module as well as an attention mechanism module for detecting cracks on pavement images. The authors used a pyramid pooling module for extracting global context information and an attention mechanism module for improving the representation ability of the encoder–decoder architecture. Furthermore, the authors employed dilated convolution layers for reducing the information loss due to pooling layers. Xiang et al. trained their model on three different datasets (i.e., Crack500, CrackTree200, CFD) and compared their model with CrackIT, CrackForest, FPHBN, and SegNet models. They demonstrated their model's superiority by visualizing predicted images and in terms of MPA and MIoU.

In [104], Zhang et al. presented a new model named CrackNet-V by modifying the original CrackNet architecture for detecting pavement cracks at the pixel level. CrackNet-V consists of three units (i.e., preprocessing layer, convolutional layer, output unit). Zhang et al. did not use any pooling layers, like the original CrackNet model, and developed a novel activation function named leaky rectified tanh function in their work. They trained their model using the images of the PaveVision3D system and obtained 84.3% precision, 90.12% recall, and 87.12% F1-score, which is better than the original CrackNet architecture.

In [105], Wu et al. proposed a sample and structure-guided network based on U-Net for segmenting cracks on road images. The authors introduced the structure-guided method to solve the problem of illumination variation and shadow in the case of detecting cracks. Wu et al. trained their model on CrackForest, ALE, CrackTree200, and CrackPV datasets. After completing the training session, they demonstrated that their model can successfully detect cracks with 90.11% recall and 43.30% precision.

In [106], Choi et al. presented a model named Semantic Segmentation Network (SDDNet) for detecting cracks on concrete structures. The SDDNet consists of several standard convolution layers, separable convolution layers, a modified ASPP module, and a decoder module. Choi et al. generated a dataset named the Crack200 dataset in their work and trained their model using the dataset. Choi et al. demonstrated that their model can successfully detect cracks even with complex backgrounds with F1 of 81.9% and mIoU of 84.6%. They also showed that their model is 46 times faster and 88 times smaller than the compared Deepcrack model.

In [107], Li et al. presented a semi-supervised learning framework based on an adversarial learning technique for detecting cracks. In this work, Li et al. intend to reduce the labor of manual annotation; as a consequence, they employed an adversarial network for generating a supervisory signal from unlabeled images. Then the authors deployed a discriminator network based on the DensNet architecture for predicting an output feature map of segmented cracks. Li et al. utilized both CFD and AigleRN datasets to train their model and achieved about 95.91% segmentation accuracy. The authors compared their model with FCN and Hybrid Crack Detector and demonstrated their model's superiority in terms of precision, recall, and F1-score.

In [108], Song et al. designed a model named CrackSeg consisting of a multiscale dilated convolution module, upsampling module, and some convolution as well as pooling blocks for detecting cracks at the pixel level in the presence of complex backgrounds. The authors built a new dataset with a total of 8196 images in their work and trained the model. However, they tested their model with CFD and AgileRN datasets along with their own dataset and achieved mIoU, F1-score, recall, and precision of 73.53%, 97.92%, 97.85%, and 98.00%, respectively. They compared their model with other state-of-the-art models (i.e., CrackForest, SegNet, U-Net, Deeplabv3+, PSPNet, DeepCrack) and claimed with quantitative analysis that their model outperformed the compared methods in terms of all evaluated metrics.

In [109], Zu et al. developed a weakly supervised model based on autoencoders for detecting cracks on asphalt concrete bridge decks. The authors differentiated the data using the autoencoder and then extracted imported features by deploying a K-means clustering algorithm. After that, the authors used a CNN model with encoder–decoder and skip connection for segmenting the cracks. Zu et al. utilized a dataset of 46,632 images and achieved 98% accuracy after training their model with the dataset.

In [110], Yang et al. proposed a novel method named Feature pyramid and Hierarchical Boosting Network (FPHBN) for detecting cracks on pavement images. The authors designed the model with bottom-up convolutional layers, which are basically the first five layers of the VGG architecture, a feature pyramid pooling module for extracting context information of different levels, deconvolutional layers, and a hierarchical boosting module for reweighting the samples. Yang et al. trained their model with five different datasets (i.e., Crack500, GAPs384, CrackTree200, CFD, AgileRN) and introduced a new evaluation metric named AIOU. They also compared their model with HED, RCF, FCN, and CrackForest models and demonstrated that their model outperformed all of the models in terms of AIOU, ODS, and OIS for all of the datasets.

In [111], Ye et al. proposed an FCN model names Ci-Net for detecting cracks in concrete structures. In the feature extraction part, the authors used six convolutional layers and two pooling layers. On the other hand, Ye et al. utilized six deconvolutional layers and two upsampling layers in the decoder module for information restoration and generating predicted images. The authors trained their model with the images of the CrackForest and TITS2016 datasets by employing the SGD algorithm. They demonstrated that their model achieved 84% precision, 82% recall, and 72.7% IoU. The authors also showed the model's superiority over the Canny edge detector and Sobel operator by visualizing the predicted images.

In [112], Zhang et al. presented an improved U-Net model named CrackNet for detecting concrete cracks at the pixel level. They proposed a total of four CrackUnet models (CrackNet7, CrackNet11, CrackNet15, CrackNet19) based on the number of convolutional layers. In this work, the authors utilized the CrackForest dataset and achieved 98.72% precision, 92.84% recall, and 95.44% F1-score. They compared the CrackUnet models and demonstrated that CrackUnet 19 performed the best, even performing better than the FCN model.

From the critical analysis and Tables 13–15, it can be seen that the researchers used a wide variety of datasets. Though most of them used their own collected private datasets, there are still a few public benchmark datasets. Table 16 presents a list of the datasets along with their access link, so that new researchers can easily find databases to start their research work in this field.

**Table 15.** Summary of Deep Learning techniques for crack segmentation.

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Annotation Tool	Performance (%)
[69]	CedNet	DenseNet-121	Caffe	Own collection	Building	-	-	Manually	Accuracy = 98.90%, Precision = 93.58%, Recall = 93.18%, F1-score = 87.23%, IoU = 98.82%
[70]	CrackU-Net	U-Net	Tensorflow	Own collection	Road	-	Adam	-	Accuracy = 99.01%, Precision = 98.56%, Recall = 97.98%, F1-score = 98.42%
[71]	ARF-crack	DeepCrack	-	DeepCrack, CFD, Crack500, GAPS384	Pavement	-	-	-	Average precision = 76.45%, 76.9%, 48.9%
[72]	SCHNet	VGG19	Tensorflow	SDNET2018	Bridge deck	Cross-entropy	-	LabelMe	mIoU = 85.31%
[73]	FRCNN, Mask RCNN	Inception ResNet-V2	Tensorflow	Own collection	Bridge	-	-	LabelImg	Average precision = 66%, 78%
[74]	Multiscale Adversarial NN	-	Pytorch	METU	concrete structures	Customized Loss function	Adam	LEAR	Accuracy = 98.176%, MIoU = 88.936%, F1 = 88.789%
[75]	U-HDN	U-Net	Pytorch	CFD, AgileRN	Road	Customized loss function	-	-	Precision = 94.5%, 92.1%, Recall = 93.6%, 93.1%
[76]	ConnCrack	VGG16	-	CFD	pavement	CwGAN loss	-	-	Precision = 96.79%, Recall = 87.75%, F1-score = 91.96%
[77]	CrackPix	VGG16	Pytorch	Own collection	concrete structures	-	-	Image Labeler tool of MATLAB	Precision = 91.24%, F1-score = 91.70%
[78]	DeepLabV3+	-	Tensorflow	Own collection	Pavement	Regression Loss	-	LabelMe	mIoU = 83.42%
[79]	DSS framework	GAN	Pytorch	Roadcrack	Road	-	Adam	-	F1-score = 82%
[80]	U-Net	ResNet-34	Pytorch, Fastai	CFD, Crack500	Pavement	Dice coefficient loss	Adamw	-	F1-score = 96%, 73%
[81]	Customized model	ResNet-40	Tensorflow	CrackDataset	Pavement	-	-	-	Precision = 98.74%, Recall = 98.05%, F1-score = 98.39%
[36]	FCN	VGG16	Keras	CCIC	Concrete structures	Binary cross-entropy	RMSprop	LIBLABEL	AP = 90%

Table 15. Cont.

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Annotation Tool	Performance (%)
[82]	FCN	ResNet-152	Tensorflow	Own collection	Roads	Cross-entropy	SGD	LEAR	Recall = 71.98%, Precision = 77.68%
[83]	UNet, DeepCrack	VGG13, VGG16	Tensorflow	CrackTree200	pavement	-	Adam	Manually	Recall = 83.1%, 80.0%, Precision = 77.5%, 76.2%
[84]	PCSN-512	VGG16	Keras, Tensorflow	Own collection	Bridge, pavement	Categorical cross-entropy	Adadelta	Manually	mAP83%,
[85]	U-Net	-	-	Own collection	Concrete structures	Focal loss function	Adam	Manually	Precision = 96%, Recall = 81%, F1-score = 88%
[86]	VGG16	-	Caffe	CCD1500	Concrete structures	Cross-entropy	-	-	Precision = 88.9%, Recall = 81%, F1-score = 88%
[87]	Ensemble CNN	-	Tensorflow	CFD, AgileRN	Pavement	Cross-entropy	-	-	Precision = 95.52%, 93.02% Recall = 95.2%, 91.6%
[88]	Pyramid Residual Network	-	-	Own collection	Road	-	Adam	-	Precision = 90.64%, Recall = 94.92%, F1-score = 92.73%
[89]	CrackSegNet	VGG16	Keras	Own collection	Tunnel	Binary cross-entropy	BP	Manually with photoshop	PA = 98.88%, Precision = 66.49%, F1-score = 63.09%
[90]	ResNet-18	-	Pytorch	CCIC, GAPS	Concrete and asphalt structures	Customized loss function	SGD	-	Accuracy = 97.95%, 94.3%
[92]	Cascaded Mask RCNN	ResNet	Pytorch	Own collection	stay cables	Multitask loss function	-	Manually	IoU = 74.3% Accuracy = 99.6%, Precision = 82.1%, Recall = 88.32%
[93]	RFCN	-	Tensorflow	Own collection	Roads, Bridge	Customized loss function	-	LabelMe	pA = 80.44%, mIoU = 80.15%
[94]	FCN	VGG16	-	METU	Concrete structures	-	-	-	precision = 91.3%, Recall = 94.1%, F1-score = 92.7%
[95]	U-Net+Ternary classifier	-	Pytorch	Own collection	Tunnel	Binary cross-entropy	Adam	Manually	Recall = 92%, Precision = 47%, F1-score = 61%

Table 15. Cont.

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Annotation Tool	Performance (%)
[96]	Customized CNN	-	Pytorch	Own collection	Bridge	-	-	-	F1-score = 84%, Accuracy = 99.55%, Precision = 78.49%
[97]	NB-FCN	VGG19	Tensorflow	Own collection	Bridge	Customized Loss function	SGD	LabelMe	Accuracy = 97.96%, Precision = 81.73%, Recall = 78.97%
[98]	FCN	VGGNet	MXNet	Own collection	Bridge	2D cross-entropy	Adam	-	AP = 96.7%
[99]	CrackNet	-	C++	PaveVision 3D	Asphalt surface	Cross-entropy	SGD	-	Precision = 90.13%, Recall = 87.63%, F1-score = 88.86%
[100]	CrackNet-R	RNN	-	PaveVision3D	Asphalt surface	-	-	-	Precision = 88.89%, Recall = 95%, F1-score = 91.84%
[101]	FCN	VGG19	Tensorflow	Own collection	Concrete structures	Cross-entropy	Adam	Manually	Accuracy = 97.96%, Precision = 81.73%, Recall = 78.8 = 97%
[102]	SegNet	VGG16	MATLAB R 2018A	CFD, TRIMMD	Concrete structures	-	-	-	Precision = 82%, 79%, Recall = 82.83%, 85.38%
[103]	FCN	-	Tensorflow	Crack500	Pavement	Softmax loss function	-	-	map = 77%
[104]	CrackNet-V	VGG	-	PaveVision3D	Asphalt pavement	Cross-entropy	SGD	manually	Precision = 84.31%, Recall = 90.12%, F1-score = 87.12%
[105]	U-Net	-	Pytorch	CrackTree200, ALE, CrackForest	Road	Focal loss	-		Recall = 90.11%, 93.99%, 76.23%
[106]	SDDNet	Customized CNN	-	Own collection	Concrete structures	mIoU loss	Adam	Affinity photo	IoU = 84.6%
[107]	DenseNet	-	Pytorch	CFD, AgileRN	Pavement	Cross-entropy	SGD	GAN technology	Accuracy = 95.91%
[108]	CracSeg	-	Tensorflow	CrackDataset	Pavement	Cross-entropy	Adam	-	Precision = 98%, Recall = 97.85%, F1-score = 97.92%



**Table 15.** *Cont.*

Ref	Method	Backbone	Framework	Dataset	Surface	Loss Function	Optimizer	Annotation Tool	Performance (%)
[109]	TDB-Net	FCN	Tensorflow	Own collection	Bridge deck	Binary cross-entropy	SGD	Manually	Accuracy = 98%
[110]	FPHBN	VGG	Caffe	Crack500	Pavement	-	-	Manually	Average Intersection Over Union (AIU) = 48%
[111]	Ci-Net	LeNet-5	-	CrackForest, TITS	Crack structures	-	Adam	-	Precision = 84%, Recall = 82%, IoU = 72.7%
[112]	CrackUNet	U-Net	Keras	CrackForest	Concrete structures	GDL	SGD	LabelMe	Precision = 92.84%, Recall = 92.84%, F1-score = 95.44%

‘-’ denotes the paper did not provide the particular information.

**Table 16.** List of the public benchmark datasets along with access links, all accessed on 28 April 2023.

Dataset	Links
CCIC/METU	<a href="https://data.mendeley.com/datasets/5y9wdsg2zt/2">https://data.mendeley.com/datasets/5y9wdsg2zt/2</a>
BCD	<a href="https://github.com/tjdxhy/crack-detection">https://github.com/tjdxhy/crack-detection</a>
SDTNET2018	<a href="https://digitalcommons.usu.edu/all_datasets/48/">https://digitalcommons.usu.edu/all_datasets/48/</a>
RDD2022	<a href="https://github.com/sekilab/RoadDamageDetector/">https://github.com/sekilab/RoadDamageDetector/</a>
CFD	<a href="https://github.com/cuilimeng/CrackForest-dataset">https://github.com/cuilimeng/CrackForest-dataset</a>
Crack500	<a href="https://bit.ly/3HbEC6d">https://bit.ly/3HbEC6d</a>
GAPS384	<a href="https://bit.ly/3HbEC6d">https://bit.ly/3HbEC6d</a>
CrackTree200	<a href="https://bit.ly/3HbEC6d">https://bit.ly/3HbEC6d</a>
PaveVision 3D	<a href="http://www.pvision3d.com/">http://www.pvision3d.com/</a>
TRIMMD	<a href="https://trid.trb.org/view/1371755">https://trid.trb.org/view/1371755</a>
TITS	<a href="https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html">https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html</a>
AigleRN	<a href="https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html">https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html</a>
Crackdataset	<a href="https://downloads.hindawi.com/journals/jat/2020/6412562.f1.zip">https://downloads.hindawi.com/journals/jat/2020/6412562.f1.zip</a>
ALE	<a href="https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html">https://www.irit.fr/~Sylvie.Chambon/Crack_Detection_Database.html</a>
DeepCrack	<a href="https://github.com/yhlleo/DeepCrack/tree/master/dataset">https://github.com/yhlleo/DeepCrack/tree/master/dataset</a>

## 6. Findings and Future Research Scope

### 6.1. Findings of the Study

In this work, we have presented a bibliometric analysis as well as a critical analysis of a few selected papers related to image-based crack detection methods. During the bibliometric analysis, our target was to determine the research trends, influential authors, journals, publications, countries, important research terms, and collaboration patterns. We list our findings from the bibliometric analysis below.

- The publication rate in the earlier years (2010–2013) was too low; less than five papers were published each year. After 2013, the number of published articles per year begins to accelerate and fluctuates in the range of six to nine during the years 2014–2019. However, the number of published articles increases dramatically in 2019 (29 papers were published). In 2020, the publication rate also follows an upward trajectory (42 papers were published).
- Ying Chen, Zhong Qu, Weigang Zou, Wei Li, Qingquan Li, Young-zin Cha, Choi Wooram, Oral Buyukozturk, Qingquan Li, and Mao Qin ZOU are the influential authors in this research field.
- Computer-aided Civil Infrastructure and Engineering, Sensors, Journal of Computing in Civil Engineering, Automation in Construction, and Construction and Building Materials are the most cited journals.
- Refs. [35,38,39,43,44] are among the most influential publications of this research field.
- The highly influential countries are China, the USA, Germany, and Japan.
- The important research terms are crack detection, deep learning, damage detection, image processing, system algorithm, inspection, model, identification, and concrete.

In the critical analysis section of our work, we have classified the papers based on their utilized techniques and described the ins and outs of the papers. We present a list of our findings from the critical analysis below.

- Deep learning techniques for detecting cracks are classified into three categories including classification, detection, and segmentation.
- Among the techniques, crack segmentation is widely adopted by researchers.
- CNN, Faster-RCNN, FCN, and U-net are the most used DL methods for performing crack classification, detection, and segmentation tasks, respectively.
- VGG-16 is the most utilized backbone among the DL methods.
- Most of the works performed their DL tasks on Tensorflow and PyTorch frameworks.
- LabelMe, LEAR, LIBLABEL, and LabelImg are the most widely adopted annotation tools.
- SGD and the Adam optimizer are utilized for optimizing the DL model by most of the researchers.
- Fine-tuning the deep learning architectures [66], using a transfer learning scheme [37], modifying deep learning architectures by adding convolutional layers [86], adding residual identity blocks [88], and removing pooling layers [87,99] can increase the accuracy for detecting cracks.
- Modifying the deep learning models by integrating various modules, including the MDM module [75], SCSE module [80], ASPP module [58], and the attention mechanism [103] can also increase the performance of the DL methods for detecting cracks.
- Utilizing of modules (i.e., SSE module [58], SWT module [83]) can also reduce the computational complexities and reduce the inference time of the DL model.

### 6.2. Future Research Direction

The previous section listed the findings of our study. Moreover, along with the findings, we have also determined a set of research scopes and directions for future researchers by analyzing the extracted DL-based papers.

- It is our understanding that the segmentation of concrete cracks using DL techniques is going to be an engrossing research topic in this field. Researchers can focus on developing and modifying benchmark DL methods for segmenting concrete cracks with better accuracy. The design and integration of attention mechanisms, the ASPP module, SSE module, SCSE module, and other modules can be a promising research topic for researchers, as several research papers showed that usage of the modules can increase accuracy.
- Very few research works [58,83] took reducing computation complexity as well as inference time into consideration. This can be a prominent research direction in order to develop lightweight and fast DL models and deploy them in low-cost devices, as real-time crack monitoring is important.
- Refs. [62–64,72] highlighted the presence of noise, such as shadow problems, shadows, contaminated backgrounds, road markings, rough surfaces, and variations in illumination, as challenging scenarios for detecting the cracks and provided solutions. However, more research should be carried out to develop robust models to tackle these issues. As a result, this can be pointed out as a huge research scope for new researchers.
- Another important perspective is to take class imbalance problems into consideration, as in [50]. As only a few pixels in an image contain crack information, DL models are very likely to face the class imbalance problem, which may hamper the classification accuracy. As a result, it also should be a research concern for future researchers.
- Though a few research works [49,53,81,101] already focused on this, there is still plenty to be researched in developing algorithms to extract the geometric information of cracks from the segmented images. As a consequence, the researchers will be able to monitor the length, width, area, and severity level of the cracks.
- Collecting data for research is always a laborious task for researchers. New researchers in this field can reduce their efforts by putting their focus on collecting data using drones and vehicles, as in [66,82]. It could be more effective if the researchers follow the research direction of [5] and develop a robotic vehicle for both collecting data and detecting cracks in real time.
- As the DL model is data-hungry method and it needs plenty of labeled images to be trained, researchers need to put a huge amount of effort into collecting images and labelling them. For solving these issues, ref. [79] comes up with an interesting solution, producing train images using a GAN simulator from only one sample image. Refs. [74,107] showed methods for labelling the images automatically by developing semi-supervised techniques using adversarial networks. New researchers can devote their efforts in this direction, as it could create a revolution in the research of DL for crack detection by providing plenty of labeled data within a shorter time and with less labor.

## 7. Discussions and Conclusions

In this article, we have presented a literature review of the existing papers on IPT-based crack detection techniques. IPTs have proved themselves to be essential parts of crack detection research. This review article has provided both scientometric and critical analysis of the prevailing papers (within the first crucial decade in this specific area of research). Bibliometric review offers several advantages over traditional systematic reviews in research evaluation and analysis. It uses objective and quantitative measures, such as citation counts and H-index, to gauge research impact, processes large quantities of data from multiple sources, facilitates the timely identification of emerging trends, and presents complex data in visual formats. Additionally, it fosters transparency and reproducibility in research evaluation and analysis. Thus, bibliometric review is a potent tool for gauging research impact and affords researchers valuable insights into emerging trends, research gaps, and potential collaborations in their field. This work performs bibliometric analysis to determine the influential authors, publications, geographical locations, modern research

trends, and possible future research directions in this field. This was carried out so that researchers can be familiarized with the pioneers in this field, can follow the prominent publications to gather knowledge, can be aware of the sources in which to publish their articles to receive more attention, and can be aware of possible future research directions in this field to pursue further research in this field. Based on the scientometric analysis conducted in this work, it is found that [35] is the most influential publications, “Computer-aided Civil Infrastructure and Engineering” is the most popular among the journals, and Ying Chen, Zhong Qu, and Weigang Zou are among the pioneer researchers in this field. It can be seen that among various technologies, DL-based techniques have contributed to a booming spike in the prosperity curve of crack detection applications. Furthermore, we have found that DL is considered the most modern technology today from the keyword timeline analysis.

Furthermore, this work has presented a thorough survey on a few scrutinized DL-based papers and also abridged many essential insights of the papers. Moreover, some captivating directive research questions have been yielded as an annex to the primary findings from the reviewed articles. This article articulates the answers to the questions related to the robustness and viability of various papers or DL techniques in this research area. In addition to this, this research work enlists a few benchmark datasets extracted from the DL-based paper along with their links so that new researchers can easily find necessary data to start their research in this field. It is our understanding that the segmentation of concrete cracks using DL techniques is going to be an engrossing research topic in this field. By means of feasible outcomes and practical application, segmentation can spearhead DL practice in concrete crack detection. It would be a rational move for the researchers to channel their research work toward crack segmentation utilizing DL techniques. The researchers should focus on developing modified DL architectures, integrating various modules, and introducing loss functions to increase the pixel segmentation accuracy. In addition to this, researchers should focus on reducing the computational complexity in order to implement the DL models on low-cost devices for real-time monitoring. Furthermore, it would be beneficial if they used segmented pictures to extract the geometric features of the cracks. We hope that researchers from both academia and industry will receive enough critical information and knowledge on DL-based crack detection techniques from this work that they will be able to contribute to this domain by incorporating this information into their research works.

**Author Contributions:** Conceptualization, M.A.-M.K., S.-H.K., A.-S.K.P. and A.-A.N.; Formal analysis, M.A.-M.K., S.-H.K., A.-S.K.P. and A.-A.N.; Funding acquisition, S.-H.K.; Investigation, M.A.-M.K. and S.-H.K.; Project administration, M.A.-M.K., S.-H.K., A.-S.K.P. and A.-A.N.; Resources, M.A.-M.K., S.-H.K., A.-S.K.P. and A.-A.N.; Software, M.A.-M.K. and S.-H.K.; Supervision, S.-H.K.; Validation, S.-H.K., A.-S.K.P. and A.-A.N.; Visualization, M.A.-M.K., S.-H.K. and A.-A.N.; Writing—original draft, M.A.-M.K.; Writing—review and editing, S.-H.K., A.-S.K.P. and A.-A.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Korea Institute of Marine Science and Technology Promotion (KIMST) grant funded by the Ministry of Oceans and Fisheries for the project titled ‘Development of smart maintenance monitoring techniques to prepare for disaster and deterioration of port infrastructures’.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The works in the paper were performed at the department of ICT integrated Ocean Smart Cities Engineering at Dong-A University, Busan, South Korea, when Md. Al-Masrur Khan was a master’s degree student at Dong-A University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Peng, T.; Kavva, T.S.; Jang, Y.-M.; Kim, B.-W. Concrete Crack Detection using Relative Standard Deviation for Image Thresholding. *Int. J. Eng. Res. Technol.* **2020**, *13*, 2720. [\[CrossRef\]](#)
- Ho, S.K.; White, R.M.; Lucas, J. A vision system for automated crack detection in welds. *Meas. Sci. Technol.* **1990**, *1*, 287–294. [\[CrossRef\]](#)
- Moon, H.; Jung, H.K.; Lee, C.W.; Park, G. Camera image processing for automated crack detection of pressed panel products. *Act. Passiv. Smart Struct. Integr. Syst.* **2017**, *10164*, 1016409.
- Runnemalm, A.; Broberg, P. Surface crack detection using infrared thermography and ultraviolet excitation. In Proceedings of the 2014 International Conference on Quantitative InfraRed Thermography, Bordeaux, France, 7–11 July 2014.
- La, H.M.; Gucunski, N.; Kee, S.-H.; Nguyen, L.V. Data analysis and visualization for the bridge deck inspection and evaluation robotic system. *Vis. Eng.* **2015**, *3*, 6. [\[CrossRef\]](#)
- Su, T.-C. Assessment of Cracking Widths in a Concrete Wall Based on TIR Radiance of Cracking. *Sensors* **2020**, *20*, 4980. [\[CrossRef\]](#)
- Nigam, R.; Singh, S.K. Crack detection in a beam using wavelet transform and photographic measurements. *Structures* **2020**, *25*, 436–447. [\[CrossRef\]](#)
- Gehri, N.; Mata-Falcón, J.; Kaufmann, W. Automated crack detection and measurement based on digital image correlation. *Constr. Build. Mater.* **2020**, *256*, 119383. [\[CrossRef\]](#)
- Yamaguchi, T.; Nakamura, S.; Hashimoto, S. An efficient crack detection method using percolation-based image processing. In Proceedings of the 2008 3rd IEEE Conference on Industrial Electronics and Applications, Singapore, 3–5 June 2008.
- Talab, A.M.; Huang, Z.; Xi, F.; HaiMing, L. Detection crack in image using Otsu method and multiple filtering in image processing techniques. *Optik* **2016**, *127*, 1030–1033. [\[CrossRef\]](#)
- Yun, H.-B.; Mokhtari, S.; Wu, L. Crack Recognition and Segmentation Using Morphological Image-Processing Techniques for Flexible Pavements. *Transp. Res. Rec. J. Transp. Res. Board* **2015**, *2523*, 115–124. [\[CrossRef\]](#)
- Chen, Z.Q.; Hutchinson, T.C. Image-Based Framework for Concrete Surface Crack Monitoring and Quantification. *Adv. Civ. Eng.* **2010**, *2010*, 215295. [\[CrossRef\]](#)
- Tong, X.; Guo, J.; Ling, Y.; Yin, Z. A new image-based method for concrete bridge bottom crack detection. In Proceedings of the 2011 International Conference on Image Analysis and Signal Processing, Wuhan, China, 21–23 October 2011.
- Mathavan, S.; Vaheesan, K.; Kumar, A.; Chandrakumar, C.; Kamal, K.; Rahman, M.; Stonecliffe-Jones, M. Detection of pavement cracks using tiled fuzzy Hough transform. *J. Electron. Imaging* **2017**, *26*, 1. [\[CrossRef\]](#)
- Sari, Y.; Prakoso, P.B.; Baskara, A.R. Road Crack Detection using Support Vector Machine (SVM) and OTSU Algorithm. In Proceedings of the 2019 6th International Conference on Electric Vehicular Technology (ICEVT), Bali, Indonesia, 18–21 November 2019.
- Shi, Y.; Cui, L.; Qi, Z.; Meng, F.; Chen, Z. Automatic Road Crack Detection Using Random Structured Forests. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3434–3445. [\[CrossRef\]](#)
- Yusof, N.A.; Osman, M.K.; Hussain, Z.; Noor, M.H.; Ibrahim, A.; Tahir, N.M.; Abidin, N.Z. Automated Asphalt Pavement Crack Detection and Classification using Deep Convolution Neural Network. In Proceedings of the 2019 9th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 29 November–1 December 2019.
- Zhang, Q.; Barri, K.; Babanajad, S.K.; Alavi, A.H. Real-Time Detection of Cracks on Concrete Bridge Decks Using Deep Learning in the Frequency Domain. *Engineering* **2020**, *7*, 1786–1796. [\[CrossRef\]](#)
- Vijayan, S.; Geethalakshmi, D.S.N. A Survey on Crack Detection Using Image Proces Techniques and Deep Learning Algorithms. *Int. J. Pure Appl. Math.* **2018**, *118*, 215–220.
- McCann, D.M.; Forde, M.C. Review of NDT methods in the assessment of concrete and masonry structures. *NDT E Int.* **2001**, *34*, 71–84. [\[CrossRef\]](#)
- Jahanshahi, M.R.; Kelly, J.S.; Masri, S.F.; Sukhatme, G.S. A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. *Struct. Infrastruct. Eng.* **2009**, *5*, 455–486. [\[CrossRef\]](#)
- Yao, Y.; Tung, S.-T.E.; Glisic, B. Crack detection and characterization techniques-An overview. *Struct. Control Health Monit.* **2014**, *21*, 1387–1413. [\[CrossRef\]](#)
- Zakeri, H.; Nejad, F.M.; Fahimifar, A. Image Based Techniques for Crack Detection, Classification and Quantification in Asphalt Pavement: A Review. *Arch. Comput. Methods Eng.* **2016**, *24*, 935–977. [\[CrossRef\]](#)
- Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [\[CrossRef\]](#)
- Milovanović, B.; Pečur, I.B. Review of Active IR Thermography for Detection and Characterization of Defects in Reinforced Concrete. *J. Imaging* **2016**, *2*, 11. [\[CrossRef\]](#)
- Gopalakrishnan, K. Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review. *Data* **2018**, *3*, 28. [\[CrossRef\]](#)
- Sharma, R.; Potnis, D.A.; Chourasia, V. Review of Image Based Concrete Crack Detection. In Proceedings of the 2021 International Conference on Advances in Technology, Management & Education (ICATME), Bhopal, India, 8–9 January 2021.
- Hsieh, Y.-A.; Tsai, Y.J. Machine Learning for Crack Detection: Review and Model Performance Comparison. *J. Comput. Civ. Eng.* **2020**, *34*, 04020038. [\[CrossRef\]](#)
- Koch, C.; Georgieva, K.; Kasireddy, V.; Akinci, B.; Fieguth, P. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Adv. Eng. Inform.* **2015**, *29*, 196–210. [\[CrossRef\]](#)



30. Hussain, A.; Akhtar, S. Review of Non-Destructive Tests for Evaluation of Historic Masonry and Concrete Structures. *Arab. J. Sci. Eng.* **2017**, *42*, 925–940. [\[CrossRef\]](#)
31. Popovics, J.S.; Rose, J.L. A survey of developments in ultrasonic NDE of concrete. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **1997**, *30*, 258. [\[CrossRef\]](#)
32. Bhat, S.; Naik, S.; Gaonkar, M.; Sawant, P.; Aswale, S.; Shetgaonkar, P. A Survey On Road Crack Detection Techniques. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020.
33. van Eck, N.J.; Waltman, L. *VOSviewer Manual*; University Leiden: Leiden, The Netherlands, 2020.
34. Chen, C. Visualizing and Exploring Scientific Literature with CiteSpace. In Proceedings of the 2018 Conference on Human Information Interaction & Retrieval, New Brunswick, NJ, USA, 11–15 March 2018.
35. Cha, Y.-J.; Choi, W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 361–378. [\[CrossRef\]](#)
36. Dung, C.V.; Anh, L.D. Autonomous concrete crack detection using deep fully convolutional neural network. *Autom. Constr.* **2019**, *99*, 52–58. [\[CrossRef\]](#)
37. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [\[CrossRef\]](#)
38. Zou, Q.; Cao, Y.; Li, Q.; Mao, Q.; Wang, S. CrackTree: Automatic crack detection from pavement images. *Pattern Recognit. Lett.* **2012**, *33*, 227–238. [\[CrossRef\]](#)
39. Yeum, C.M.; Dyke, S.J. Vision-Based Automated Crack Detection for Bridge Inspection. *Comput.-Aided Civ. Infrastruct. Eng.* **2015**, *30*, 759–770. [\[CrossRef\]](#)
40. Prasanna, P.; Dana, K.J.; Gucunski, N.; Basily, B.B.; La, H.M.; Lim, R.S.; Parvardeh, H. Automated Crack Detection on Concrete Bridges. *IEEE Trans. Autom. Sci. Eng.* **2016**, *13*, 591–599. [\[CrossRef\]](#)
41. Li, Q.; Zou, Q.; Zhang, D.; Mao, Q. FoSA: F\* Seed-growing Approach for crack-line detection from pavement images. *Image Vis. Comput.* **2011**, *29*, 861–872. [\[CrossRef\]](#)
42. Torok, M.M.; Golparvar-Fard, M.; Kochersberger, K.B. Image-Based Automated 3D Crack Detection for Post-disaster Building Assessment. *J. Comput. Civ. Eng.* **2014**, *28*. [\[CrossRef\]](#)
43. Yamaguchi, T.; Hashimoto, S. Fast crack detection method for large-size concrete surface images using percolation-based image processing. *Mach. Vis. Appl.* **2009**, *21*, 797–809. [\[CrossRef\]](#)
44. Oliveira, H.; Correia, P.L. Automatic Road Crack Detection and Characterization. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 155–168. [\[CrossRef\]](#)
45. Nishikawa, T.; Yoshida, J.; Sugiyama, T.; Fujino, Y. Concrete Crack Detection by Multiple Sequential Image Filtering. *Comput.-Aided Civ. Infrastruct. Eng.* **2011**, *27*, 29–47. [\[CrossRef\]](#)
46. Gavilán, M.; Balcones, D.; Marcos, O.; Llorca, D.F.; Sotelo, M.A.; Parra, I.; Ocaña, M.; Aliseda, P.; Yarza, P.; Amírola, A. Adaptive Road Crack Detection System by Pavement Classification. *Sensors* **2011**, *11*, 9628–9657. [\[CrossRef\]](#)
47. Zalama, E.; Gómez-García-Bermejo, J.; Medina, R.; Llamas, J. Road Crack Detection Using Visual Features Extracted by Gabor Filters. *Comput.-Aided Civ. Infrastruct. Eng.* **2013**, *29*, 342–358. [\[CrossRef\]](#)
48. Fujita, Y.; Hamamoto, Y. A robust automatic crack detection method from noisy concrete surfaces. *Mach. Vis. Appl.* **2010**, *22*, 245–254. [\[CrossRef\]](#)
49. Tran, T.S.; Tran, V.P.; Lee, H.J.; Flores, J.M.; Le, V.P. A two-step sequential automated crack detection and severity classification process for asphalt pavements. *Int. J. Pavement Eng.* **2020**, *23*, 2019–2033. [\[CrossRef\]](#)
50. Wang, Z.; Xu, G.; Ding, Y.; Wu, B.; Lu, G. A vision-based active learning convolutional neural network model for concrete surface crack detection. *Adv. Struct. Eng.* **2020**, *23*, 2952–2964. [\[CrossRef\]](#)
51. Zhang, L.; Zhou, G.; Han, Y.; Lin, H.; Wu, Y. Application of Internet of Things Technology and Convolutional Neural Network Model in Bridge Crack Detection. *IEEE Access* **2018**, *6*, 39442–39451. [\[CrossRef\]](#)
52. Dung, C.V.; Sekiya, H.; Hirano, S.; Okatani, T.; Miki, C. A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. *Autom. Constr.* **2019**, *102*, 217–229. [\[CrossRef\]](#)
53. Flah, M.; Suleiman, A.R.; Nehdi, M.L. Classification and quantification of cracks in concrete structures using deep learning image-based techniques. *Cem. Concr. Compos.* **2020**, *114*, 103781. [\[CrossRef\]](#)
54. Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* **2017**, *157*, 322–330. [\[CrossRef\]](#)
55. Yang, Q.; Shi, W.; Chen, J.; Lin, W. Deep convolution neural network-based transfer learning method for civil infrastructure crack detection. *Autom. Constr.* **2020**, *116*, 103199. [\[CrossRef\]](#)
56. Li, S.; Zhao, X. Image-Based Concrete Crack Detection Using Convolutional Neural Network and Exhaustive Search Technique. *Adv. Civ. Eng.* **2019**, *2019*, 6520620. [\[CrossRef\]](#)
57. Deng, J.; Lu, Y.; Lee, V.C.-S. Imaging-based crack detection on concrete surfaces using You Only Look Once network. *Struct. Health Monit.* **2020**, *20*, 147592172093848. [\[CrossRef\]](#)
58. Li, H.; Xu, H.; Tian, X.; Wang, Y.; Cai, H.; Cui, K.; Chen, X. Bridge Crack Detection Based on SSENets. *Appl. Sci.* **2020**, *10*, 4230. [\[CrossRef\]](#)

59. Xu, H.; Su, X.; Wang, Y.; Cai, H.; Cui, K.; Chen, X. Automatic Bridge Crack Detection Using a Convolutional Neural Network. *Appl. Sci.* **2019**, *9*, 2867. [\[CrossRef\]](#)
60. Chen, F.-C.; Jahanshahi, M.R. NB-CNN: Deep Learning-Based Crack Detection Using Convolutional Neural Network and Naïve Bayes Data Fusion. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4392–4400. [\[CrossRef\]](#)
61. Park, S.E.; Eem, S.-H.; Jeon, H. Concrete crack detection and quantification using deep learning and structured light. *Constr. Build. Mater.* **2020**, *252*, 119096. [\[CrossRef\]](#)
62. Majidifard, H.; Adu-Gyamfi, Y.; Buttlar, W.G. Deep machine learning approach to develop a new asphalt pavement condition index. *Constr. Build. Mater.* **2020**, *247*, 118513. [\[CrossRef\]](#)
63. Deng, J.; Lu, Y.; Lee, V.C.S. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *35*, 373–388. [\[CrossRef\]](#)
64. Huyan, J.; Li, W.; Tighe, S.; Zhai, J.; Xu, Z.; Chen, Y. Detection of sealed and unsealed cracks with complex backgrounds using deep convolutional neural network. *Autom. Constr.* **2019**, *107*, 102946. [\[CrossRef\]](#)
65. Ma, D.; Fang, H.; Xue, B.; Wang, F.; Msekh, M.A.; Chan, C.L. Intelligent Detection Model Based on a Fully Convolutional Neural Network for Pavement Cracks. *Comput. Model. Eng. Sci.* **2020**, *123*, 1267–1291. [\[CrossRef\]](#)
66. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [\[CrossRef\]](#)
67. Cao, M.-T.; Tran, Q.-V.; Nguyen, N.-M.; Chang, K.-T. Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources. *Adv. Eng. Inform.* **2020**, *46*, 101182. [\[CrossRef\]](#)
68. Li, C.; Xu, P.; Niu, L.; Chen, Y.; Sheng, L.; Liu, M. Tunnel crack detection using coarse-to-fine region localization and edge detection. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2019**, *9*, e1308. [\[CrossRef\]](#)
69. Li, S.; Zhao, X. Automatic Crack Detection and Measurement of Concrete Structure Using Convolutional Encoder-Decoder Network. *IEEE Access* **2020**, *8*, 134602–134618. [\[CrossRef\]](#)
70. Huyan, J.; Li, W.; Tighe, S.; Xu, Z.; Zhai, J. CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection. *Struct. Control. Health Monit.* **2020**, *27*. [\[CrossRef\]](#)
71. Chen, F.-C.; Jahanshahi, M.R. ARF-Crack: Rotation invariant deep fully convolutional network for pixel-level crack detection. *Mach. Vis. Appl.* **2020**, *31*, 47. [\[CrossRef\]](#)
72. Pan, Y.; Zhang, G.; Zhang, L. A spatial-channel hierarchical deep learning network for pixel-level automated crack detection. *Autom. Constr.* **2020**, *119*, 103357. [\[CrossRef\]](#)
73. Kalfarisi, R.; Wu, Z.Y.; Soh, K. Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization. *J. Comput. Civ. Eng.* **2020**, *34*, 04020010. [\[CrossRef\]](#)
74. Shim, S.; Kim, J.; Cho, G.-C.; Lee, S.-W. Multiscale and Adversarial Learning-Based Semi-Supervised Semantic Segmentation Approach for Crack Detection in Concrete Structures. *IEEE Access* **2020**, *8*, 170939–170950. [\[CrossRef\]](#)
75. Fan, Z.; Li, C.; Chen, Y.; Wei, J.; Loprencipe, G.; Chen, X.; Mascio, P.D. Automatic Crack Detection on Road Pavements Using Encoder-Decoder Architecture. *Materials* **2020**, *13*, 2960. [\[CrossRef\]](#) [\[PubMed\]](#)
76. Mei, Q.; Gül, M. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Constr. Build. Mater.* **2020**, *256*, 119397. [\[CrossRef\]](#)
77. Alipour, M.; Harris, D.K.; Miller, G.R. Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks. *J. Comput. Civ. Eng.* **2019**, *33*, 04019040. [\[CrossRef\]](#)
78. Ji, A.; Xue, X.; Wang, Y.; Luo, X.; Xue, W. An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement. *Autom. Constr.* **2020**, *114*, 103176. [\[CrossRef\]](#)
79. Wei, T.; Cao, D.; Zheng, C.; Yang, Q. A simulation-based few samples learning method for surface defect segmentation. *Neurocomputing* **2020**, *412*, 461–476. [\[CrossRef\]](#)
80. Lau, S.L.; Chong, E.K.; Yang, X.; Wang, X. Automated Pavement Crack Segmentation Using U-Net-Based Convolutional Neural Network. *IEEE Access* **2020**, *8*, 114892–114899. [\[CrossRef\]](#)
81. Song, W.; Jia, G.; Jia, D.; Zhu, H. Automatic Pavement Crack Detection and Classification Using Multiscale Feature Attention Network. *IEEE Access* **2019**, *7*, 171001–171012. [\[CrossRef\]](#)
82. Bang, S.; Park, S.; Kim, H.; Kim, H. Encoder–decoder network for pixel-level road crack detection in black-box images. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 713–727. [\[CrossRef\]](#)
83. Chen, H.; Lin, H.; Yao, M. Improving the Efficiency of Encoder-Decoder Architecture for Pixel-Level Crack Detection. *IEEE Access* **2019**, *7*, 186657–186670. [\[CrossRef\]](#)
84. Chen, T.; Cai, Z.; Zhao, X.; Chen, C.; Liang, X.; Zou, T.; Wang, P. Pavement crack detection and recognition using the architecture of segNet. *J. Ind. Inf. Integr.* **2020**, *18*, 100144. [\[CrossRef\]](#)
85. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Autom. Constr.* **2019**, *104*, 129–139. [\[CrossRef\]](#)
86. Qu, Z.; Mei, J.; Liu, L.; Zhou, D.-Y. Crack Detection of Concrete Pavement With Cross-Entropy Loss Function and Improved VGG16 Network Model. *IEEE Access* **2020**, *8*, 54564–54573. [\[CrossRef\]](#)
87. Fan, Z.; Li, C.; Chen, Y.; Mascio, P.D.; Chen, X.; Zhu, G.; Loprencipe, G. Ensemble of Deep Convolutional Neural Networks for Automatic Pavement Crack Detection and Measurement. *Coatings* **2020**, *10*, 152. [\[CrossRef\]](#)

88. Feng, H.; Xu, G.; Guo, Y. Multi-scale classification network for road crack detection. *IET Intell. Transp. Syst.* **2018**, *13*, 398–405. [\[CrossRef\]](#)
89. Ren, Y.; Huang, J.; Hong, Z.; Lu, W.; Yin, J.; Zou, L.; Shen, X. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Constr. Build. Mater.* **2020**, *234*, 117367. [\[CrossRef\]](#)
90. Alipour, M.; Harris, D.K. Increasing the robustness of material-specific deep learning models for crack detection across different materials. *Eng. Struct.* **2020**, *206*, 110157. [\[CrossRef\]](#)
91. Eisenbach, M.; Stricker, R.; Seichter, D.; Amende, K.; Debes, K.; Sesselmann, M.; Ebersbach, D.; Stoeckert, U.; Gross, H.-M. How to get pavement distress detection ready for deep learning? A systematic approach. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017.
92. Hou, S.; Dong, B.; Wang, H.; Wu, G. Inspection of surface defects on stay cables using a robot and transfer learning. *Autom. Constr.* **2020**, *119*, 103382. [\[CrossRef\]](#)
93. Zheng, M.; Lei, Z.; Zhang, K. Intelligent detection of building cracks based on deep learning. *Image Vis. Comput.* **2020**, *103*, 103987. [\[CrossRef\]](#)
94. Islam, M.M.; Kim, J.-M. Vision-Based Autonomous Crack Detection of Concrete Structures Using a Fully Convolutional Encoder–Decoder Network. *Sensors* **2019**, *19*, 4251. [\[CrossRef\]](#)
95. Han, J.H.; Moon, Y.S.; Lee, C.H.; Kim, I.S. Crack Detection Method for Tunnel Lining Surfaces using Ternary Classifier. *KSII Trans. Internet Inf. Syst.* **2020**, *14*.
96. Li, Y.; Zhao, W.; Zhang, X.; Zhou, Q. A Two-Stage Crack Detection Method for Concrete Bridges Using Convolutional Neural Networks. *IEICE Trans. Inf. Syst.* **2018**, *E101.D*, 3249–3252. [\[CrossRef\]](#)
97. Li, G.; Liu, Q.; Zhao, S.; Qiao, W.; Ren, X. Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system. *Meas. Sci. Technol.* **2020**, *31*, 075403. [\[CrossRef\]](#)
98. Wang, D.; Dong, Y.; Pan, Y.; Ma, R. Machine Vision-Based Monitoring Methodology for the Fatigue Cracks in U-Rib-to-Deck Weld Seams. *IEEE Access* **2020**, *8*, 94204–94219. [\[CrossRef\]](#)
99. Zhang, A.; Wang, K.C.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces Using a Deep-Learning Network. *Comput.-Aided Civ. Infrastruct. Eng.* **2017**, *32*, 805–819. [\[CrossRef\]](#)
100. Zhang, A.; Wang, K.C.; Fei, Y.; Liu, Y.; Chen, C.; Yang, G.; Li, J.Q.; Yang, E.; Qiu, S. Automated Pixel-Level Pavement Crack Detection on 3D Asphalt Surfaces with a Recurrent Neural Network. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *34*, 213–229. [\[CrossRef\]](#)
101. Yang, X.; Li, H.; Yu, Y.; Luo, X.; Huang, T.; Yang, X. Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 1090–1109. [\[CrossRef\]](#)
102. Zhang, X.; Rajan, D.; Story, B. Concrete crack detection using context-aware deep semantic segmentation network. *Comput.-Aided Civ. Infrastruct. Eng.* **2019**, *34*, 951–971. [\[CrossRef\]](#)
103. Xiang, X.; Zhang, Y.; Saddik, A.E. Pavement crack detection network based on pyramid structure and attention mechanism. *IET Image Process.* **2020**, *14*, 1580–1586. [\[CrossRef\]](#)
104. Fei, Y.; Wang, K.C.; Zhang, A.; Chen, C.; Li, J.Q.; Liu, Y.; Yang, G.; Li, B. Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 273–284. [\[CrossRef\]](#)
105. Wu, S.; Fang, J.; Zheng, X.; Li, X. Sample and Structure-Guided Network for Road Crack Detection. *IEEE Access* **2019**, *7*, 130032–130043. [\[CrossRef\]](#)
106. Choi, W.; Cha, Y.-J. SDDNet: Real-Time Crack Segmentation. *IEEE Trans. Ind. Electron.* **2020**, *67*, 8016–8025. [\[CrossRef\]](#)
107. Li, G.; Wan, J.; He, S.; Liu, Q.; Ma, B. Semi-Supervised Semantic Segmentation Using Adversarial Learning for Pavement Crack Detection. *IEEE Access* **2020**, *8*, 51446–51459. [\[CrossRef\]](#)
108. Song, W.; Jia, G.; Zhu, H.; Jia, D.; Gao, L. Automated Pavement Crack Damage Detection Using Deep Multiscale Convolutional Features. *J. Adv. Transp.* **2020**, *2020*, 1–11. [\[CrossRef\]](#)
109. Zhu, J.; Song, J. Weakly supervised network based intelligent identification of cracks in asphalt concrete bridge deck. *Alex. Eng. J.* **2020**, *59*, 1307–1317. [\[CrossRef\]](#)
110. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1525–1535. [\[CrossRef\]](#)
111. Ye, X.-W.; Jin, T.; Chen, P.-Y. Structural crack detection using deep learning-based fully convolutional networks. *Adv. Struct. Eng.* **2019**, *22*, 3412–3419. [\[CrossRef\]](#)
112. Zhang, L.; Shen, J.; Zhu, B. A research on an improved UNET-based concrete crack detection algorithm. *Struct. Health Monit.* **2020**, *20*, 1864–1879. [\[CrossRef\]](#)

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.