Construction and Building Materials

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

--Manuscript Draft--

Manuscript Number:	CONBUILDMAT-D-22-04772R2
Article Type:	Review Article
Keywords:	defect datasets; infrastructure defect inspection; classfication; segmentation and detection; learning-based approaches
Corresponding Author:	Guidong YANG The Chinese University of Hong Kong Hong Kong, Hong Kong CHINA
First Author:	Guidong YANG
Order of Authors:	Guidong YANG
	Kangcheng Liu
	Jihan Zhang
	Benyun Zhao
	Zuoquan Zhao
	Xi Chen
	Ben M. Chen
Abstract:	Deep learning breakthrough stimulates new research trends in civil infrastructure inspection, whereas the lack of quality-guaranteed, human-annotated, free-of-charge, and publicly available defect datasets with sufficient amounts of data hinders the progress of deep learning in defect inspection. To boost research in deep learning-based visual defect inspection, this paper first reviews and summarizes 40 publicly available defect datasets, covering common defects in various types of buildings and infrastructures. The taxonomy of the datasets is proposed based on specific deep learning objectives (classification, segmentation, and detection). Clarifications are also made for each dataset regarding its corresponding data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, algorithms adopted for validation, annotation levels, context levels, and publication license for future utilization. Consequently, the summarized defect datasets offer around 13.38 M labeled images, cover more than 5 defect types, 5 infrastructure types, 5 material types, and 3 levels of image context. Given that the crack is a common interest in civil engineering, this paper further combines existing datasets with self-labeled crack images to establish a benchmark dataset providing more than 15,000 and 11,000 labeled images for crack classification and segmentation, respectively. Based on the established crack dataset, experiments are conducted for classification, segmentation, and the subsequent non-maximum suppression-based detection tasks. The proposed multi-branch self-attention module and multi-stage-fused attentional pyramid network have been successfully adapted into the state-of-the-art (SOTA) classification network- Swin Transformer and segmentation networks including DeepLab V3+ , DenseNet , and Full Resolution ResNet . The resulting classification network achieves 88.0% accuracy, and the adapted segmentation models reach 77.8, 77.6, 76.9% mIoU (mean Intersection over Union), respectively. Moreover, a comprehensive comparison between 11 SOTA classification algorithms and 12 SOTA segmentation algorithms has been conducted. The algorithms proposed in this work are shown to achieve satisfactory performance with an acceptable efficiency on modern graphic processing units. Detailed suggestions are provided for constructing high-quality datasets and inspection algorithms. Finally, this paper remarks on the quantity, diversity, difficulty, and scalability of the reviewed defect datasets, feasibility on robotic platforms, superiority of proposed algorithms, and criticality of algorithm comparison results, formulating a solid baseline for future defect inspection research.

Suggested Reviewers:	Bo Xia Queensland University of Technology paul.xia@qut.edu.au
	Yupeng Wu University of Nottingham Yupeng.Wu@nottingham.ac.uk
	Yong Xia The Hong Kong Polytechnic University y.xia@polyu.edu.hk
	Changwen Chen The Hong Kong Polytechnic University changwen.chen@polyu.edu.hk
	Kay Chen TAN The Hong Kong Polytechnic University kaychen.tan@polyu.edu.hk

Dear Editor:

We submit our manuscript entitled "*Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection*" to **Construction and Building Materials**. We have read and abided by the statement of ethical standards for manuscripts submitted to **Construction and Building Materials**.

This paper presents the first comprehensive review of the existing publicly available datasets for deep learning-based visual defect inspection. Besides, this paper conducts the first systematic comparison of the state-of-the-art (SOTA) algorithms for defect classification, segmentation, and detection, with crack as the typical research interest to conduct the case study. Moreover, this study proposes deep learning-based network architectures based on the adaptations to SOTA algorithms for crack classification, segmentation, and subsequent detection based on non-maximum suppression. Furthermore, suggestions are proposed for the construction of high-quality defect datasets and the development of defect inspection algorithms. The results from this study can provide a solid baseline for future research in defect inspection.

The work described has not been submitted elsewhere for publication, and all the authors listed have approved the manuscript that is enclosed.

Yours sincerely,

Guidong Yang, Kangcheng Liu*, Jihan Zhang, Benyun Zhao, Zuoquan Zhao, Xi Chen* and Ben M. Chen

Unmanned Systems Research Group, Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China

Highlights

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

Guidong Yang,Kangcheng Liu,Jihan Zhang,Benyun Zhao,Zuoquan Zhao,Xi Chen,Ben M. Chen

- Review of the datasets for deep learning-based visual defect inspection
- Comparison of the algorithms for defect classification, segmentation and detection
- Proposed deep learning-based network architectures for defect inspection
- Suggestions on developing defect datasets and defect inspection algorithms

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

Guidong Yang[1], Kangcheng Liu[*,1], Jihan Zhang, Benyun Zhao, Zuoquan Zhao, Xi Chen[*] and Ben M. Chen

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China

ARTICLE INFO

Keywords:
defect datasets
infrastructure defect inspection
classfication
segmentation and detection
learning-based approaches

ABSTRACT

Deep learning breakthrough stimulates new research trends in civil infrastructure inspection, whereas the lack of quality-guaranteed, human-annotated, free-of-charge, and publicly available defect datasets with sufficient amounts of data hinders the progress of deep learning in defect inspection. To boost research in deep learning-based visual defect inspection, this paper first reviews and summarizes 40 publicly available defect datasets, covering common defects in various types of buildings and infrastructures. The taxonomy of the datasets is proposed based on specific deep learning objectives (classification, segmentation, and detection). Clarifications are also made for each dataset regarding its corresponding data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, algorithms adopted for validation, annotation levels, context levels, and publication license for future utilization. Consequently, the summarized defect datasets offer around $13.38M$ labeled images, cover more than 5 defect types, 5 infrastructure types, 5 material types, and 3 levels of image context. Given that the crack is a common interest in civil engineering, this paper further combines existing datasets with self-labeled crack images to establish a benchmark dataset providing more than 15,000 and 11,000 labeled images for crack classification and segmentation, respectively. Based on the established crack dataset, experiments are conducted for classification, segmentation, and the subsequent non-maximum suppression-based detection tasks. The proposed *multi-branch self-attention module* and *multi-stage-fused attentional pyramid network* have been successfully adapted into the state-of-the-art (SOTA) classification network-*Swin Transformer* and segmentation networks including *DeepLab V3+*, *DenseNet*, and *Full Resolution ResNet*. The resulting classification network achieves 88.0% accuracy, and the adapted segmentation models reach 77.8%, 77.6%, 76.9% mIoU (mean Intersection over Union), respectively. Moreover, a comprehensive comparison between 11 SOTA classification algorithms and 12 SOTA segmentation algorithms has been conducted. The algorithms proposed in this work are shown to achieve satisfactory performance with an acceptable efficiency on modern graphic processing units. Detailed suggestions are provided for constructing high-quality datasets and inspection algorithms. Finally, this paper remarks on the quantity, diversity, difficulty, and scalability of the reviewed defect datasets, feasibility on robotic platforms, superiority of proposed algorithms, and criticality of algorithm comparison results, formulating a solid baseline for future defect inspection research.

1. Introduction

Civil infrastructures such as pavements, bridges, buildings, tunnels, and dams suffer from performance degradation caused by structure deterioration, external loads, weather impact, poor workmanship, poor design, and natural disasters [1, 2, 3]. Periodical defect inspection is a necessary and pivotal measure to ensure the energy efficiency and the functional safety of civil structures. The subsequent rehabilitation measures can be then carried out according to the inspection results. Periodical defect inspection is often conducted through Non-Destructive Testing (NDT), which can avoid the physical damage caused by the traditional sample collection process [4]. NDT techniques include infrared thermography (IRT) [5], photogrammetry [6], laser scanning [7], impact echos (IE) [8], and ground-penetrating radars (GPR) [9]. Currently, periodical manual defect inspection is predominant in infrastructure maintenance, where inspectors make use of NDT devices to evaluate structural health [10]. However, inspectors may be exposed to complex site environment with potential health hazards and safety risks [11]. Furthermore, such subjective inspection can be error-prone [12], labor-intensive [13], and time-consuming [14], not conducive to the subsequent rehabilitation [15]. For example, traditional methods make wrong predictions easily under not well-controlled illuminations, and cost more manpower to accomplish the inspection task. Manual inspections often span several weeks to months, resulting in outdated evaluation at the time of rehabilitation.

Due to the aforementioned limitations, more and more researchers tend to incorporate machine learning and deep learning algorithms into automatic defect inspection solutions. Especially in recent years, deep learning has become the main stream solution due to its unprecedented breakthrough. Deep learning-based solutions are evolving to automate the defect inspection efficiently [16, 17, 18, 19].

*Corresponding author
kcliu@mae.cuhk.edu.hk (Kangcheng Liu); xichen002@cuhk.edu.hk (Xi Chen)
ORCID(s): 0000-0003-2168-9057 (Xi Chen)
[1]Both authors contributed equally to this work

Nevertheless, compared to the successful application of deep learning in natural language processing, facial recognition [20], image processing [21, 22], and 3D vision for autonomous aerial and ground vehicles [23, 24, 25], research in deep learning-based defect inspection is still restricted. The most critical reason is the lack of quality-guaranteed, human-annotated, free-of-charge, and publicly available defect datasets, which are beneficial to training highly accurate neural networks for defect inspection based on supervised learning [26]. Although there exist reviews focusing on NDT devices for building inspection [4] and segmentation algorithms for pavement crack detection [27], they neither provide a comprehensive review of the datasets spanning different infrastructure and defect types nor a systematic comparison of deep learning algorithms for visual inspection. Thus, it is essential and meaningful to make a comprehensive review and systematic comparison of existing **publicly available** datasets and algorithms to boost deep learning-based defect inspection. To the authors' best knowledge, this paper is the first comprehensive review of **publicly available** civil infrastructure inspection datasets, and the first that provides a systematic review and comparison of **publicly available** state-of-the-art (SOTA) algorithms for surface defect inspection.

Motivated by the aforementioned difficulties, this paper intends to promote research in deep learning-based defect inspection by conducting a comprehensive review on existing publicly available defect datasets with a systematic comparison between SOTA algorithms for the task of classification, segmentation, and detection on a constructed crack dataset. The major contributions of this paper are as follows:

- A comprehensive review of the existing publicly available datasets for deep learning-based visual defect inspection.
- A systematic comparison of the SOTA algorithms for defect classification, segmentation and detection, with crack as the typical research interest for a case study.
- Proposed deep learning-based network architectures based on the adaptations to SOTA algorithms for crack classification, segmentation and subsequent detection with non-maximum suppression.
- Suggestions on developing high-quality defect datasets and defect inspection algorithms.

The remainder of this paper is organized as follows. Section 2 is the literature review methodology. Section 3 shows the review results of the datasets and corresponding methods. The self-established crack dataset, results of the comparison between our methods and SOTAs for crack classification, segmentation, and detection are presented and discussed in Section 4. Based on the review and comparison, Section 5 points out existing barriers to building a high-quality and large-scale defect dataset and offers corresponding suggestions. Also, the systematical suggestions on methodology to conduct highly-effective defect recognition are provided. Conclusions and future work are presented in Section 6 to form a comprehensive baseline for studies on civil infrastructure defect inspection.

2. Literature review methodology

A comprehensive review of the literature related to **publicly available** datasets for deep learning-based visual defect inspection was conducted using Google Scholar. Based on keywords searching, a considerable amount of literature most relevant to the research interest was acquired. The literature was filtrated according to the following procedures: (1) Title, abstract, and conclusion screening; (2) Dataset public availability checking; (3) Full-text screening to extract critical features of the datasets. Specifically, the following features of the defect dataset were selected and summarized, they are: *data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, annotation levels, context levels, publication license, algorithms adopted for validation, algorithm training strategies*, and *data augmentation methods*. These critical features are of the utmost concern when developing deep learning-based solutions for defect inspection. The main focus of the review is on visual inspection datasets, i.e., datasets with optical images supplemented with IRT images. Datasets with data from other NDT devices (see e.g., IE and GPR) are beyond the scope of this review.

3. Review results on datasets and corresponding methods

Based on the above literature review on publicly available defect datasets with optical images supplemented with IRT images. Altogether 40 defect datasets are summarized, illustrated, and demonstrated. Figure 1 shows the taxonomy of summarized defect datasets based on different aspects. For each dataset, its corresponding data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, algorithms for validation, annotation levels, image context levels, and publication license are clarified. In this paper, the taxonomy of these datasets is elaborated as per specific deep learning objectives (annotation levels). The datasets are grouped into classification-oriented, detection-oriented, and segmentation-oriented, with patch-level, bounding-box-level, and pixel-level annotation respectively.

As demonstrated in Figure 1, the summarized defect datasets cover various types of infrastructure such as pavements, bridges, buildings, tunnels, and dams with different materials such as concrete, asphalt, steel, masonry, and wood. These datasets cover the most common defect types: crack, spalling, delamination, corrosion, and efflorescence. As to data types, most datasets utilize optical images (in terms of grey-scale and color images), with IRT images [26, 28], IE signals [29, 30], and GPR signals [30] as alternatives. Optical images are typically used to detect surface defects of the structure, while IRT images, IE signals,

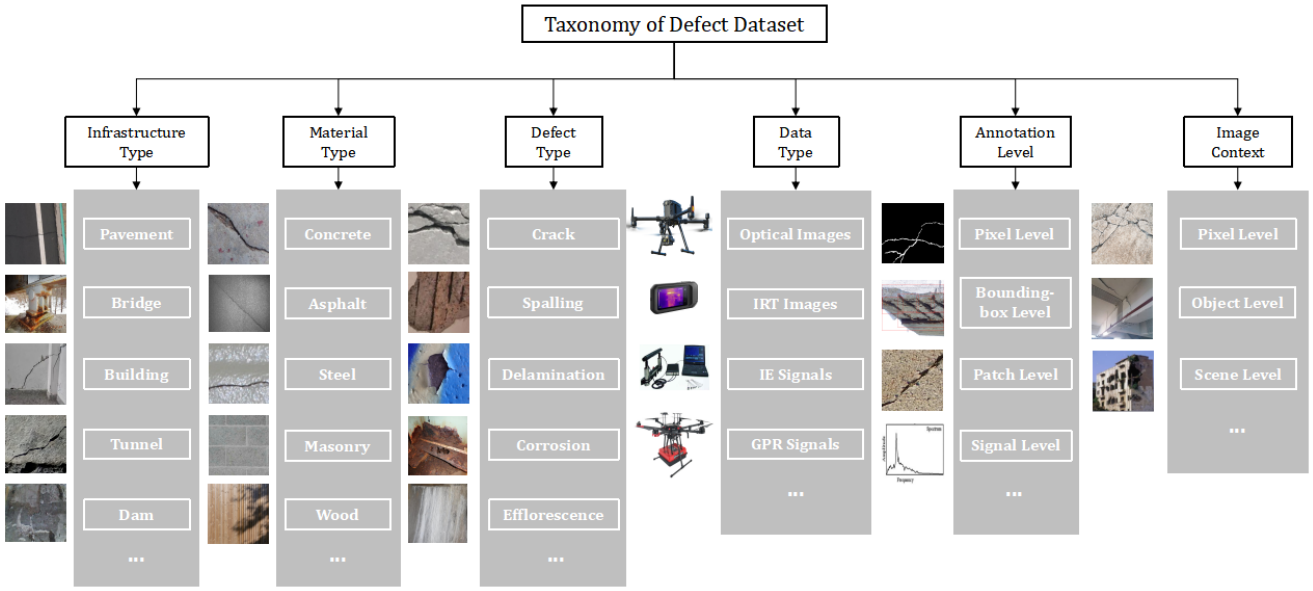

Figure 1: Taxonomy of defect datasets.

Table 1
A summary of publicly available **classification-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data Source / Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
GAPs-v1 [13]	2017	6.3 M	64 × 64	Cameras on ground vehicle	a. Crack b. Pothole c. Inlaid patch d. Applied patch e. Open joint	Pavement	Asphalt	Patch Level	Pixel Level	Private License, for Academic Use Only
GAPs-v2 [14]	2019	6.7 M	64 × 64 to 256 × 256	Cameras on ground vehicle	Same as GAPs-v1	Pavement	Asphalt	Patch Level	Pixel Level	Private License, for Academic Use Only
CBID [32]	2017	1028	229 × 229	Not clarified	a. Crack b. Water seepage c. Spalling, etc.	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Xu [33]	2019	6069	224 × 224	Camera on UAV	Crack	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Philipp [34]	2019	3607	Multiple	Hand-held camera	a. Crack, Efflorescence b. General defects (e.g. graffiti, moss, etc) c. Scaling, spalling d. Exposed reinforcement, Rust staining	Bridge	Concrete and Steel	Patch Level	Pixel Level	CC BY 4.0 License
KrakN [12]	2020	16114	224 × 224	Hand-held cameras	Thin crack(< 0.2mm)	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
DCTCD [35]	2021	250	512 × 512	Cameras on UAV	Thin crack (>= 0.1mm)	Bridge	Concrete	Patch Level	Pixel Level	GNU General Public License v3.0
CCIC [36]	2018	40000	227 × 227	Hand-held camera	Crack	Building	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
$\phi - Net$ [3]	2020	36413	448 × 448	Crawled from internet Hand-held cameras	Spalling, etc.	Building	Concrete, Steel, Masonry, and Wood	Patch Level	Pixel Level Object & Scene Level	CC BY-NC-SA 4.0 License
CSSC [10]	2017	45024 44263	130 × 130 100 × 100	Crawled from internet	Crack and Spalling	Bridge and Building	Concrete	Patch Level	Pixel Level	Not Clarified
SDNET2018 [37]	2018	56092	256 × 256	Hand-held camera	Crack	Bridge, Building and Pavement	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Qurishee [26]	2020	2088	4032 × 3024 and 5312 × 2988	Not clarified	Crack	Not clarified	Concrete	Patch Level	Pixel Level	CC BY 4.0 License

and GPR signals can reveal subsurface defects. Besides, these datasets vary in the level of image context information, i.e., the pixel level, object level, and scene level. The data contained in different datasets are collected via hand-held sensors, robotic platforms, or UAV platforms. In particular, compared to the hand-held cameras and wall-climbing robots, the UAV platform combined with visual-inertial odometry offers a feasible solution for defect data collection and localization in the GPS-denied environment, e.g., defect inspection under the bridge [10, 31]. Subsection 3.1-3.3 illustrate classification, segmentation, and detection-oriented datasets with optical images supplemented with IRT images respectively. Within each subsection, the defect datasets are further grouped based on the type of targeted civil infrastructure. Subsection 3.4 describes the status and trend of data collection and labeling procedures.

3.1. Classification-oriented datasets

In this subsection, each dataset described is labeled either at the image level or at the patch level (if multiple image patches are cropped from the raw image) to conduct multi-class classification between different defect categories or binary classification of a particular defect between defect and non-defect categories. It should be noted that the classification datasets can also be used to detect defects based on the sliding window technique, e.g., Histogram of Oriented Gradients (HOG) Detector [38], Deformable Part-based Model (DPM) [39], and Overfeat detector [40]. In general, the sliding window technique is to slide a window to go through all possible locations and scales in the image and further classify each image patch bounded by the window to check whether the image patch contains the target object (the defect in our case) or not [41]. In this manner, the detection problem can be converted to a classification problem, and the defect in the original image can be detected. Table 1 shows the summary of publicly available classification-oriented defect datasets. Each dataset's corresponding data volume, data resolution, data source, defects categories covered, infrastructure types focused, material types targeted, annotation level, and image context level are clarified. These datasets are firstly sorted according to the corresponding infrastructure type and then sorted in chronological order. Table 2

shows the corresponding algorithms used for validating the datasets. For each dataset, its network structures and training strategies are listed. Figure 2 shows exemplary images for each classification-oriented dataset.

3.1.1. Pavements

The German Asphalt Pavement Distress (GAPs) datasets have three versions, i.e. GAPs-v1 [13], GAPs-v2 [14], and GAPs-10m [15]. GAPs-v1 [13] dataset is the first standardized, quality-controlled, patch-level annotated, free of charge, and publicly available dataset with a decent size enough to train neural networks for asphalt pavement distress classification. The data collection procedure strictly follows the regulations developed by the German Road and Transportation Research Association (FGSV). The images are downward-facing road images collected by a surface camera system composed of two photogrammetrically calibrated cameras. The GAPs-v1 dataset contains 1,969 grey-scale images (8-bit) comprising 1,418 images for training, 51 images for validation, and 500 images for testing. Each resulting image has a resolution of 1920×1080, with a pixel resolution of 1.2 mm $\times$ 1.2 mm. Each high-resolution image is annotated to impose 64×64 bounding boxes enclosing pavement distress (defined by FGSV), which covers cracks, potholes, inlaid patches, applied patches, and open joints. Each image is further sliced into multiple 64×64 image patches. Thus, the dataset has 4.9 M patches for training, 200 k patches for validation, and 1.2 M patches for testing. Cracks are the dominant distress class in the GAPs-v1 dataset. Various crack types are included: single or multiple cracking, longitudinal or transversal cracking, alligator cracking, and sealed cracks. The GAPs-v1 dataset is dedicated to the binary classification of pavement distress. All of the aforementioned damage classes are labeled as 'Distress', while intact road patches are labeled as 'Normal'.

GAPs-v2 [14] is an improvement on the GAPs-v1 dataset, it provides more data, refined annotations, and more context compared to GAPs-v1. Five hundred additional images with a size of 1920×1080 are collected following the regulations developed by FGSV. Altogether 2,468 grey-scale images (8 bit) are further divided into a training set (1,417 images), a validation set (51 images), a validation-test set (500 images), and a test set (500 images). Based on these images, 692,377 and 6,035,404 image patches are extracted for road distress and intact road, respectively, to form the entire dataset. The respective proportion of intact roads, cracks, applied patches, inlaid patches, potholes, and open joints in the full dataset are 89.71%, 7.28%, 1.72%, 0.75%, 0.30%, and 0.24%. GAPs-v2 also refines annotations by providing a smaller bounding box for non-damage space and solving conflicting annotations. Moreover, GAPs-v2 offers multiple patch sizes (64×64 to 256×256) with more image context since different image patch sizes will influence the trade-off between damage detection quality and inference speed of the neural network [14]. In addition to the above refinements, GAPs-v2 contains a CIFAR-like [42] or MNIST-like [43] subset consisting of 50,000 patches for training and 10,000 patches for validation, validation-test, and test. The subset's proportion of intact road, cracks, applied patches, inlaid patches, potholes, and open joints are 60%, 20%, 10%, 5%, 3% and 2% respectively. The publicly available GAPs-v2 dataset is still dedicated to the binary classification of pavement distress, i.e., 'Distress' or 'Normal'. GAPs-10m [15] dataset provides pixel-level annotation for pavement distress segmentation. This dataset will be illustrated in the Subsection 3.2 of this survey.

3.1.2. Bridges

Cambridge Bridge Inspection Dataset (CBID) [32] is a dataset for evaluating the classification performance of different bridge defects. The dataset contains 1,028 image patches with a resolution of 229×229. The dataset is further partitioned into two subsets containing bridge patches with (337 patches) and without (691 patches) defects. However, the dataset doesn't explicitly illustrate the data collection procedure and defects classes contained in the dataset.

Xu *et al.* [33] build up a dataset for binary classification of concrete bridge crack. The original dataset [49] contains 2068 crack images collected by a UAV equipped with a camera that has a resolution of 1024×1024. To improve the classification robustness of the network, crack images with bridge shadings, strong light, and water stains are wittingly included in the dataset. Each image in the original dataset is further cropped into multiple 512×512 image patches. After filtering blurred patches, a new dataset containing 6,069 patches is obtained. The acquired dataset comprises 4,058 crack images and 2,011 background images. The number of patches for the training and validation sets is 4,856 and 1,213, respectively. Afterward, Xu *et al.* [33] further crop all the patches into smaller 256×256 patches and flip the patches from the training set in order to meet the input requirement of the network.

Philipp *et al.* [34] provides the first patch-level-annotated dataset for multi-classification of concrete bridge defects covering cracks, efflorescence, scaling, spalling, and general defects (e.g., graffiti and moss). To consider possible defect combinations required by inspection guidelines, they also provide two other datasets for the binary classification of the exposed reinforcement and rust staining (corrosion). The total number of image patches in the multi-classification dataset and two binary-classification datasets are 3,607. The detailed distribution of the data volume for each defect type in the corresponding dataset is clarified in [34]. The patches do not have a consistent resolution since they are acquired from 38,408 images by slicing and labeling the defect area, with 21,284 images collected in the on-site experiment and 17,124 images provided by authorities. The image collection procedure adopts a 42-Mp camera and takes the shooting range, on-surface resolution (0.1 mm), camera focus, lighting condition, and surface angle between the subject surface and the camera optical axis into account for high-quality images.

KrakN [12] dataset is dedicated to thin crack detection. For the training set, over 900 pictures with a size of

Table 2
A summary of network architectures and training strategies adopted by the corresponding **classification-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
GAPs-v1 [13]	2017	a. ASINVOS Net [13] b. ASINVOS-mod Net [13]	✗	✓	✗
GAPs-v2 [14]	2019	a. ASINVOS Net [13] b. ResNet-10, -18, -34, -50 [44]	✓	✓	Adversarial training, Rotation, Translation
Xu [33]	2019	a. Xu's Net (with ASPP Module) [33] b. ResNet-18, -34, -50 [44] c. VGG-16, -19 [45]	✗	✓	Cropping, Flipping
Philipp [34]	2019	- Inception V3 [46]	✓	✗	✗
KrakN [12]	2020	- KrakN Net [12]	✓	✗	Cropping
CCIC [36]	2018	a. AlexNet [47] b. VGG-16, -19 [45] c. GoogLeNet [48] d. ResNet-50, -101, -152 [44]	✓	✗	Cropping
$\phi - Net$ [3]	2020	a. VGG-16, -19 [45] b. ResNet-50 [44]	✓	✗	Cropping
CSSC [10]	2017	- VGG-16 [45]	✓	✗	Cropping, Picking, Rotation, Sampling
SDNET2018 [37]	2018	- AlexNet [47]	✓	✓	Cropping

4248×2850 are collected from a cracked bridge pillar in good lighting conditions and at a close-up shooting distance ($20 - 30$ cm). Image cropping and labeling are conducted within 4 hours by using a self-developed semi-automatic tool. Only cracks and background surfaces are labeled as two classes. Afterward, $8,057$ image patches are acquired for cracks and background classes, respectively. Over $3,000$ images are collected from multiple scenarios and cameras for the validation set.

Drone Captured Tiny Crack Dataset (DCTCD) [35] consists of 250 images with complex textures (scrawls, surface corrosion, and efflorescences). DCTCD concentrates on bridge thin crack detection. All of the crack images are collected by a drone under bridge beams and pier inner walls. With controlled camera shooting distance, angles, and lighting conditions, the range of image pixel resolution is $0.1 - 0.2$ mm, which is beneficial to thin crack detection. Image color jitter, ISO noises, defocus blur, and motion blur are involved to imitate real application scenarios. The whole dataset is further split into five subsets according to different edge complexity factors defined in [35].

3.1.3. Buildings

Concrete Crack Images for Classification (CCIC) [36] dataset provides $40,000$ image patches with a size of 224×224, cropped from 500 high-resolution (4032×3024) images. The original images are collected from walls and floors of multiple concrete buildings, with various concrete surface finishes (plastering, exposed, and paint). During the image collection procedure, the camera directly faces the subject surface, and the data collection is finished in a single day to ensure consistent image illumination conditions.

Pacific Earthquake Engineering Research (PEER) Hub ImageNet (ϕ-Net) [3] provides $36,413$ image patches with building defects, which are collected and cropped from $100,000$ images collected from the field experiment and the Internet. Each image is labeled with 8 attributes related to local and global building information. Afterward, eight subsets are extracted for the classification of each attribute respectively.

3.1.4. Aggregated

Concrete Structure Spalling and Crack (CSSC) [10] dataset is the first released dataset for concrete spalling and crack detection. The initial dataset consists of $1,232$ images totally, with 278 spalling images and 954 crack images. All of the images are collected from the Internet through keyword searching. These images cover several types of infrastructure (e.g., bridges and buildings). Thus, the CSSC dataset is an aggregated dataset. The dataset also provides two subsets containing image patches with sizes of 100×100 and 130×130 for each defect class. Each patch in the subsets is annotated either as a 'True' or 'False' label, where the 'True' label stands for the patch with defects and the 'False' label represents the patch without defects or the patch with defects but does not meet the pixel threshold defined in [10]. For concrete spalling, the number of patches in the two subsets are $19,123$ ($7,376$ for 'True', $11,747$ for 'False') and $19,924$ ($8,574$ for 'True', $11,350$ for 'False') respectively. For concrete crack, the amounts of patches in the two subsets are $25,140$ ($13,448$ for 'True', $11,652$ for 'False'), $25,100$ ($13,422$ for 'True', $11,678$ for 'False') respectively. In addition to the patch-level annotated subsets, the CSSC dataset also annotates the initial images at pixel level according to the suggestions from the experts in civil engineering.

Structural Defects NET (SDNET2018) [37] is a patch-level annotated dataset for concrete crack classification. Altogether 230 images with sizes of 4068×3456 are acquired through a 16-MP camera. These images cover reinforced concrete building walls (72 images), bridge decks (54 images), and unreinforced concrete pavements (104 images). The working distance between the camera (without zoom) and the subject is 500 mm during the image acquisition. Each image is partitioned into multiple 256×256 image patches. Each image and patch cover a rough area of

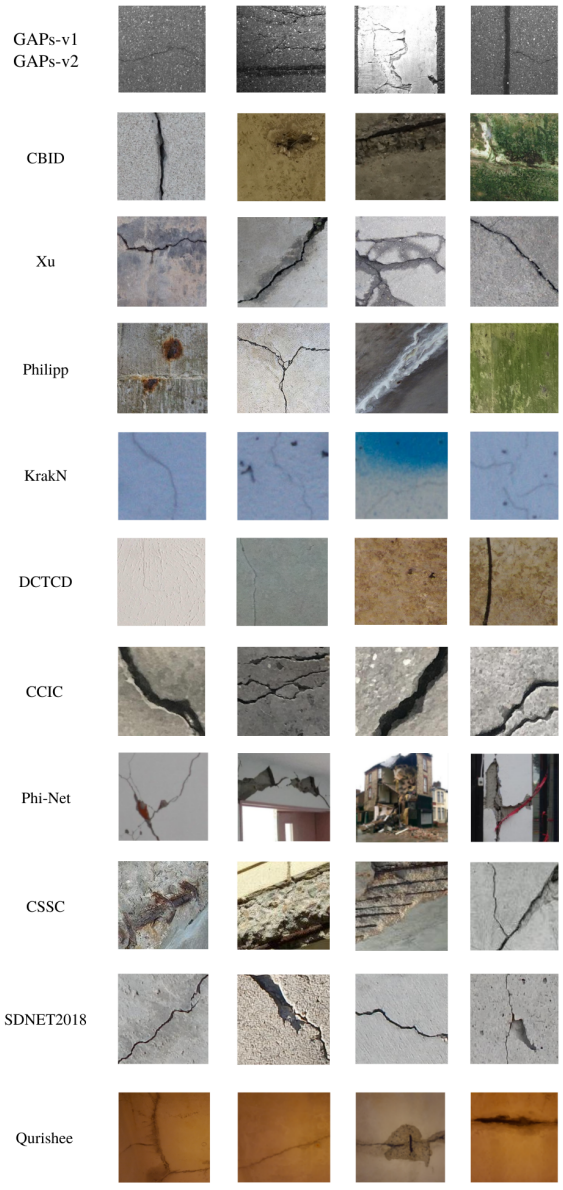

Figure 2: Exemplary images from datasets for infrastructure defects classification. From top row to bottom row, they are image patches from GAPs-v1 [13] & GAPs-v2 [14], CBID [32], Xu [33], Philipp [34], KrakN [12], DCTCD [35], CCIC [36], ϕ-Net [3], CSSC [10], SDNET2018 [37], and Qurishee [26] respectively.

1000 mm × 850 mm and 60 mm × 60 mm, respectively. Besides, each patch is annotated as 'Cracked' or 'Uncracked'. After partition and annotation, the respective amounts of patches for bridge decks, building walls, and pavements are 13,620 (2,025 for 'Cracked', 11,595 for 'Uncracked'), 18,138 (3,851 for 'Cracked', 14,287 for 'Uncracked'), and 24,334 (2,608 for 'Cracked', 21,276 for 'Uncracked'). On the whole, the dataset contains 8,484 patches with crack and 47,608 patches without crack. In addition to the sufficient data volume, the dataset also provides the range of crack width (0.06 mm-25 mm) covered, which will benefit the deep neural network to identify crack with size variety. Besides, the dataset intentionally incorporates images with various obstructions, including shadows, stains, edges, rough surface finishes, inclusions, voids, joints, and surface scaling, improving the robustness and generalization ability of the deep neural network real applications.

Qurishee *et al.* [26] proposed a dataset about concrete cracks. This dataset has 1,499 crack images and 589 non-crack images. The data are with two resolutions, 4032 × 3024 and 5312 × 2988. The resolutions of the data are relatively high and can show details that are not easily observed. However, equipment used to collect the data, algorithms for validating the dataset, and infrastructure type targeted are not specified.

3.2. Segmentation-oriented datasets

In this subsection, each dataset illustrated is annotated at the pixel level to conduct defects segmentation. Compared to the bounding-box-level annotation, pixel-level annotation can localize the defects more accurately and clearly. Table 3 shows the summary of publicly available segmentation-oriented defect datasets. These datasets are firstly sorted by corresponding infrastructure type, and then sorted in chronological order. Table 4 shows the corresponding algorithms and training strategies adopted for validating the datasets. Fig 3 show exemplary images for datasets targeting at different types of infrastructure.

3.2.1. Pavements

There are pixel-level-annotated datasets that are firstly used by traditional machine learning algorithms. They are Sylvie [50], CrackTree [51], Amhaz [52], and CrackForest Dataset (CFD) [53]. These datasets are dedicated to the crack segmentation of asphalt pavement. Due to their positive influence on the subsequent deep learning-based methods, corresponding attributes are also summarized and listed in Table 3. Since this paper focuses on datasets for deep learning, readers can find a more detailed description of the aforementioned datasets in the corresponding papers [50, 51, 52, 53].

For deep learning-based crack segmentation of asphalt pavement, Yang *et al.* propose the Crack500 dataset [54, 55] which has pixel-wise annotation and comprises 500 crack images with a resolution of 2000 × 1500. Each image is further cropped into 16 non-overlapping image patches, whereas patches in which the number of crack pixels is smaller than a certain threshold are discarded. Furthermore, based on GAPs-v1 [13] dataset, Yang *et al.* provide the GAPs384 dataset [55], in which 384 pavement images (1920 × 1080) containing only crack distress are selected and annotated at pixel level.

EdmCrack600 [56, 57, 58] dataset offers 600 backward-facing images with pixel-level annotation for pavement crack segmentation. All images are with a resolution of 1920 × 1080 and extracted from videos recorded by a sports camera mounted on the rear of a moving vehicle. The images vary in weather conditions, environmental effects, blurring effects, and noise.

Table 3
A summary of publicly available **segmentation-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data source/Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
Sylvie [50]	2011	42	Multiple	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
CrackTree [51]	2012	206	800 × 600	Not clarified	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
Amhaz [52]	2016	68	Multiple	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
CFD [53]	2016	118	480 × 320	Hand-held camera	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
Crack500 [54, 55]	2019	3368	640 × 360	Hand-held camera	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
GAPs384 [55]	2019	384	1920 × 1080	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Private License, for Academic Use Only
EdmCrack600 [56, 57, 58]	2020	600	1920 × 1080	Camera on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Scene Level	CC BY-NC-ND 4.0 License
GAPs-10m [15]	2021	20	5030 × 11505	Cameras on ground vehicle	23 distresses	Pavement	Asphalt	Pixel Level	Scene Level	Private License, for Academic Use Only
Highway-crack [59]	2021	5275	512 × 512	Cameras on UAV	Crack	Pavement	Asphalt	Pixel Level	Scene Level	Not Clarified
CCIC-600 [36]	2019	600	227 × 227	Not clarified	Crack	Bridge	Concrete	Pixel Level	Pixel Level	CC BY 4.0 License
BCL [60]	2021	11000	256 × 256	Hand-held cameras	Crack	Bridge	Concrete, Masonry, and Steel	Pixel Level	Pixel Level	CC0 1.0 License
CCSSS [61]	2021	440	512 × 512	Not clarified	Corrosion	Bridge	Steel	Pixel Level	Pixel & Object Level	CC0 1.0 License
LCW [62]	2021	440	512 × 512	Not clarified	Corrosion	Bridge	Steel	Pixel Level	Scene Level	CC0 1.0 License
DeepCrack [63]	2019	537	544 × 384	Not clarified	Crack	Building	Concrete, Asphalt	Pixel Level	Pixel Level	Private License, for Academic Use Only
Bai-2020 [64]	2020	853	256 × 256	Not clarified	Crack	Building	Concrete	Pixel Level	Pixel Level Object & Scene Level	GNU General Public License v3.0
Masonry [65]	2021	11491	224 × 224	Crawled from internet Hand-held cameras	Crack	Building	Masonry	Pixel Level	Pixel & Object Level	GNU General Public License v3.0
Ren [66]	2020	919	512 × 512	Hand-held camera	Crack	Tunnel	Concrete	Pixel Level	Pixel Level	MIT License
Sandra (IRT) [28]	2020	517	320 × 240	Hand-held thermal cameras	Crack, Spalling, Patches, Delamination	Dam	Concrete	Pixel Level	Pixel Level	Not Clarified
UAV75 [67]	2019	75	512 × 512	Camera on UAV	Crack	Not Clarified	Not Clarified	Pixel Level	Pixel Level	GNU General Public License v3.0
CSD [68]	2020	11298	448 × 448	Crawled from internet	Crack	Multiple	Multiple	Pixel Level	Pixel Level	Not Clarified
Bai-2021 [69]	2021	2229	Multiple	Crawled from internet	Crack, Spalling	Building, Bridge	Concrete	Pixel Level	Pixel Level Object & Scene Level	MIT License
CCCD [70]	2021	10995	448 × 448	Crawled from internet	Crack	Multiple	Multiple	Pixel Level	Pixel Level	CC0 1.0 License

Table 4
A summary of network architectures and training strategies adopted by the corresponding **segmentation-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
Sylvie [50]	2011	- Morph [50] (Morphological Analysis) - GaMM [50] (Multiscale Analysis and Local Crack Modelling)	Not Applicable	Not Applicable	✗
CrackTree [51]	2012	- CrackTree [51] (Minimum Spanning Trees)	Not Applicable	Not Applicable	✗
Amhaz [52]	2016	- Minimal Path Selection [52]	Not Applicable	Not Applicable	✗
CFD [53]	2016	- CrackForest [53] (Random Structured Forests)	Not Applicable	Not Applicable	Re-defined crack tokens
Crack500 [54, 55] GAPs384 [55]	2019 2019	- Feature Pyramid and Hierarchical Boosting Network [55] (FPHBN)	✗	✓	Cropping
EdmCrack600 [56, 57, 58]	2020	- ConnCrack [57] (cWGAN-based training)	✗	✓	Flipping, Cropping
GAPs-10m [15]	2021	a. U-Net [71], U-Net [71] (Xception [72]) b. An Encoder (Resnet-18, -50 [44])-Decoder (PSPNet [73]) Network	✗	✓	Flipping, Patch rotation, Patch scaling Modifying brightness, contrast, noise
Highway-crack [59]	2021	- U-Net [71] (Lighter Encoder and Attention Module)	✗	✓	Flipping, Rotation
BCL [60]	2021	a. U-Net [71] (Pruned Version) b. FCN [74] (VGG [45]) c. DeepLab V3 [75]	✗	✓	Cropping
CCSSS [61]	2021	- DeepLab V3+ [76]	✗	✓	Resizing
LCW [62]	2021	- DeepLab V3+ [76]	✗	✓	Resizing
DeepCrack [63]	2019	- DeepCrack [63]	✗	✓	Rotation, Cropping, Flipping
Bai-2020 [64]	2020	a. ResNet-152 [44] b. U-Net [71]	✓	✗	Resizing
Masonry [65]	2021	a. VGG-16 [45], ResNet-34, -50 [44] b. DenseNet-121, -169 [77], Inception V3 [46] c. MobileNet [78], MobileNet V2 [79] d. DeepLab V3+ [76], FCN [74] (VGG-16 [45]) e. U-Net [71] (with various backbones) f. FPN [80] (with various backbones)	✓	✗	Cropping
Ren [66]	2020	- CrackSegNet [66]	✓	✗	Rotation, Translation, Scaling, Shearing
Sandra (IRT) [28]	2021	a. VGG-16 [45] b. ResNet-18 [44] c. ResNet-50 [44] d. MobileNet V2 [79] e. Xception [72]	✗	✓	Cropping, Resizing Reflection, Translation
UAV75 [67]	2019	- CrackNausNet [67]	✓	✗	Resizing, Cropping, Rotation, Flipping
CSD [68]	2020	a. U-Net [71] (VGG-16 [45]) b. U-Net [71] (ResNet-101 [44])	✓	✗	Resizing
Bai-2021 [69]	2021	a. Mask R-CNN [81] (Cascade) b. Mask R-CNN [81] (APANet [82, 83]) c. Mask R-CNN [81] (HRNet [84])	✗	✓	Flipping, Rotation, Cropping
CCCD [70]	2021	- DeepLab V3+ [76]	✗	✓	Resizing

Based on the GAPs-v1 [13], and GAPs-v2 [14] dataset, Ronny *et al.* propose GAPs-10m [15] dataset annotated at the pixel level. The original dataset comprises 394 high-resolution downward-facing pavement surface images collected by following government regulations. All images are taken at different road sections to cover pavement distresses and object classes. 23 pavement distresses and object classes are defined by experts. The original dataset is then partitioned into a training set, a validation set, and a test set. As a subset of the validation set, the publicly available GAPs-10m dataset consists of 20 images (complying with German federal regulations) with a consistent resolution of 5030×11505. It is named after GAPs-10m since a single image covers 10 m in the image height direction. The dataset offers certain challenges, such as image artifacts caused by harsh sunlight and image stitching and the difficulty of distinguishing certain distress from the intact pavement surface.

Hong *et al.* [59] propose two datasets for highway crack segmentation. The first dataset is annotated at the pixel level based on the public dataset Aerial Crack Dataset [85], which is only annotated at the bounding-box level. After relabeling and cropping, the resulting dataset contains $4,118$ images with a resolution of 512×512. To validate the generalization ability of their proposed model, they constructed a second dataset comprising $1,157$ highway crack images collected by a UAV. These images are taken after a 6.4-level earthquake in China and annotated at the pixel level, with an image resolution of 5 cm and a UAV flight height of 200 m.

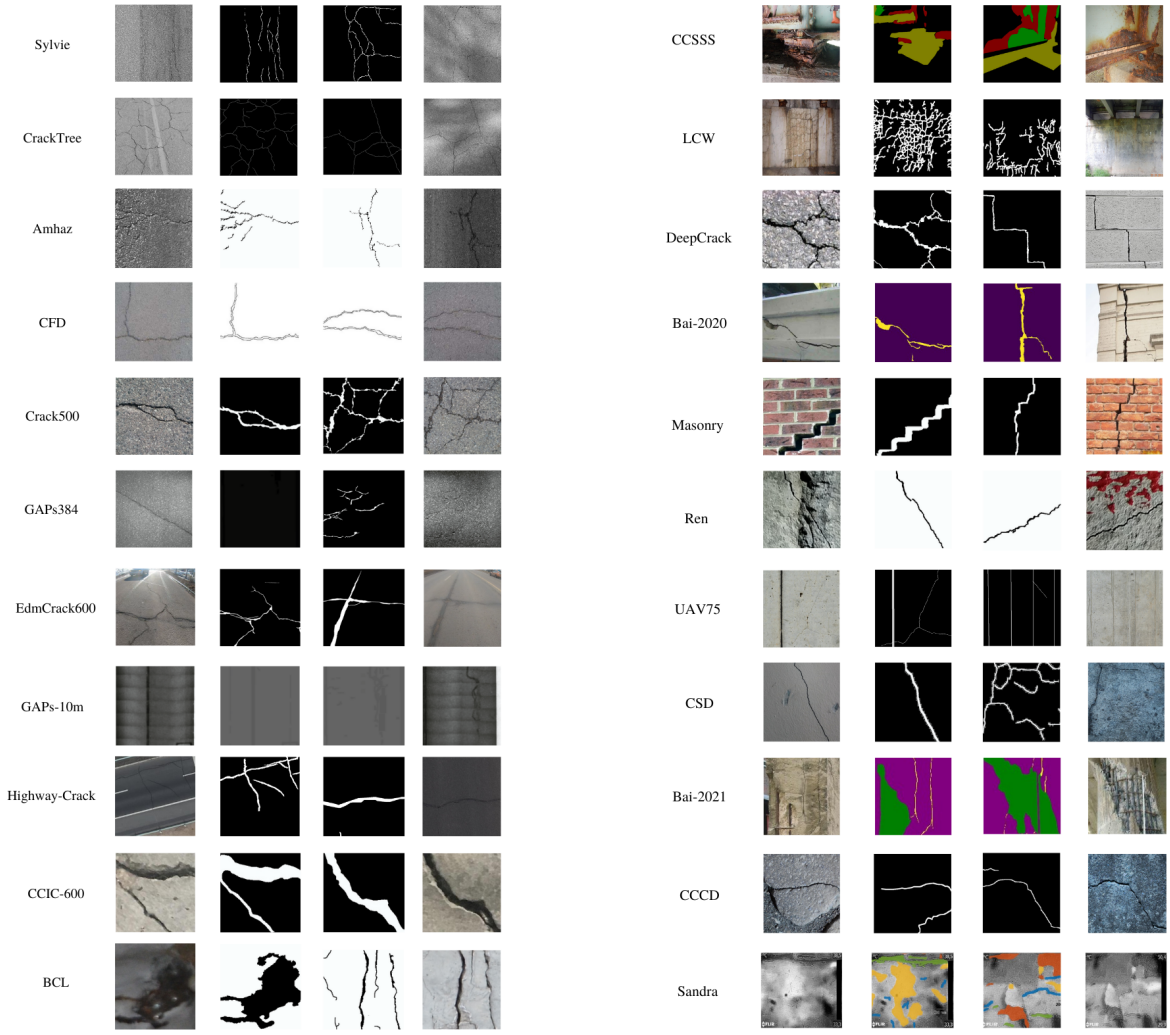

Figure 3: Exemplary images from datasets for pavement defects segmentation. From top row to bottom row, they are image patches and corresponding segmentation labels from Sylvie [50], CrackTree [51], Amhaz [52], CFD [53], Crack500 [54, 55], GAPs384 [55], EdmCrack600 [56, 57, 58], GAPs-10m [15], Highway-Crack [59], CCIC-600 [36], BCL [60], CCSSS [61], LCW [62], DeepCrack [63], Bai-2020 [64], Masonry [65], Ren [66], UAV75 [67], CSD, Bai-2021 [69], CCCD [70], Sandra [28] respectively. Each row represents two pairs of image patches with corresponding segmentation labels.

3.2.2. Bridges

CCIC-600[1] is an extension of the CCIC [36] dataset, aiming at concrete crack segmentation. Six hundred image patches are selected from the CCIC dataset and annotated at the pixel level. Bridge Crack Library (BCL) [60] dataset consists of 11,000 image patches with a size of 256×256 for bridge crack segmentation. This dataset covers three types of bridge materials, i.e., concrete, masonry, and steel. All the crack image patches are cropped from 1,180 raw crack images, with 1,000 nonsteel crack images and 180 steel crack images. Nonsteel crack images are collected by bridge inspection engineers through field inspection on 50 in-service bridges in China, with crack width within millimeters. Steel crack images are provided by the first International Project Competition for Structural Health Monitoring (IPC-SHM) [86]. The dataset can be divided into three subsets, with 5,769 nonsteel cracks, 2,036 steel cracks, and 3,195 crack-like motifs (cropped from 25,000 non-crack images for steel structures), respectively. A large proportion of crack-like motifs (such as shadows, stains, and water spots) are introduced intentionally to resolve the class imbalance between nonsteel crack and steel crack and improve model robustness.

Corrosion Condition State Semantic Segmentation (CCSSS) [61] dataset is devoted to the segmentation of bridge condition state. The dataset comprises 440 high-resolution images with a resolution of 512 × 512. The images are acquired from the bridge inspection reports and finely annotated at the pixel level by following government guidelines. This dataset is the first dataset to grade the corrosion state of bridges. The corrosion state is semantically annotated in four-level, i.e., good, fair, poor, and severe. The Labeled

[1] CCIC-600

Cracks in the Wild (LCW) [62] dataset is dedicated in scene-level bridge crack segmentation. The dataset consists of 3,817 images collected from bridge inspection reports. For the training purpose, all the images are resized to 512×512. All of the original images and resized images are publicly available.

3.2.3. Buildings, tunnels, and dams

DeepCrack [63] dataset is composed of 537 images for building surface crack segmentation. All the images are with a resolution of 544×384. This dataset covers multiple surface textures (bare, dirty, and rough), structure materials (concrete and asphalt), and crack scales (1 pixel to 180 pixels), which makes it be a challenging dataset. Bai-2020 [64] is a dataset for building crack localization. In addition to the images with pixel-level and object-level context information, the dataset contains some images at the structural (scene) level. The dataset contains 853 images with a resolution of 256×256.

The Masonry [65] dataset pays attention to cracks on the masonry walls of the buildings. The dataset contains 469 raw images either acquired from the Internet or captured by field experiments from several buildings in the Netherlands. Each image is divided into multiple image patches. The dataset includes images with varying scales, resolutions, crack appearances, and types of noisy backgrounds for more robust segmentation.

Ren *et al.* [66] provides a crack segmentation dataset focusing on tunnel environment. The raw images with a size of 4032×3016 are captured in a tunnel from China. Each raw image is further cropped into multiple image patches. Data augmentation techniques such as rotation, translation, scaling, and shearing are adopted to increase data volume.

Infrared images can be used to reveal subsurface defects. Sandra [28] proposes a segmentation dataset of white-hot infrared images containing four defect labels: crack, spalling, patches, and delamination. All images contain delamination and cracks, although some images do not contain spalling and patches. There are totally 517 images collected by FLIR and the resolution of data is 320×240. All annotations of these infrared images are labelled based on the corresponding optical images and engineering knowledge.

3.2.4. Aggregated

UAV75 [67] is a crack segmentation dataset emphasizing the images collected by the UAV. Compared with images captured by hand-held digital cameras and smartphones, the images acquired by the UAV may suffer from low resolution, low crack intensity, and re-occurring planking patterns. The authors notice that planking patterns may result in false-positive results. The planking class is added to the label space to distinguish planking patterns from cracks.

Bai-2021 [69] is an extension of the dataset [64]. They all focus on extreme events such as major earthquakes. Compared with the former version, Bai-2021 includes more images (2,229 additional images) with various resolutions from 147×288 to 4600×3070, more scenes including buildings and bridges, and more structural failures including cracks and spalling. Data augmentation is used to increase data volume.

Crack Segmentation Dataset (CSD)[2] is an aggregate dataset that merges 300 self-collected images (labeled at the pixel level) with several other crack segmentation datasets [10, 51, 52, 53, 54, 55, 63]. The dataset contains 11,298 images with a consistent resolution of 448×448. These images taking several cases into account, they are images containing pure crack, pseudo crack, crack with noise, crack with moss, and crack in large context. There is a high degree of similarity between CSD and Concrete Crack Conglomerate Dataset (CCCD) [70], which is also a conglomeration of several other crack segmentation datasets [10, 51, 52, 53, 54, 55, 63].

3.3. Detection-oriented datasets

The classification-oriented datasets (see Section 3.1) with images annotated at the image level can be used to conduct defect detection based on the sliding window techniques. Besides, there also exists detection-oriented defect datasets with images annotated at the bounding-box level to conduct multi-defect detection. Road Damage Dataset 2018 (RDD-2018) [87], RDD-2019, RDD-2020, and Pavement-Image-Dataset (PID) [92] are for pavement damage detection, while COncrete DEfect BRidge IMage (CODEBRIM) dataset [93] focuses on the multi-defect detection of concrete bridges. Compared to the defect segmentation, bounding-box-level annotation is benifical to the real-time defect detection and deployment. Table 5 shows the summary of publicly available detection-oriented defect datasets. These datasets are firstly sorted by corresponding infrastructure type, and then sorted in chronological order. Table 6 shows the corresponding algorithms and training strategies adopted for validating the datasets.

3.3.1. Pavements

Maeda *et al.* publish Road Damage Dataset 2018 (RDD-2018) [87], which is the first dataset for large-scale road damage detection. The dataset comprises 9,053 frontal-facing road images which contains 15,435 damage instances in Japan. All images with a uniform resolution of 600×600 are collected by a smartphone installed on the dashboard of the vehicle. These images have diverse background in terms of weather and surface conditions, which resembles the real-world scenarios. This dataset covers 8 damage classes such as cracks and corrosions, which is defined by government guidelines. A more detailed illustration of damage type and distribution can be found in [87]. For each damage in the image, the damage class and corresponding bounding box location are labeled. The number of images in the training set and validation set is 7,240 and 1,813 respectively. RDD-2018 was used as the benchmark dataset in Road Damage Detection and Classification Challenge (RDDCC) [104].

[2] CSD

Table 5
A summary of publicly available **detection-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data source/Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
RDD-2018 [87]	2018	9053	600 × 600	Camera on ground vehicle	Cracks and corrosions (8 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
RDD-2019 [88]	2019	13135	600 × 600	Camera on ground vehicle	Cracks and corrosions (9 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
RDD-2020 [89, 90, 91]	2020	26336	600 × 600 720 × 720	Cameras on ground vehicle	Cracks and potholes (4 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
PID [92]	2020	7237	640 × 640	Crawled from internet	Cracks (9 damage classes)	Pavement	Not clarified	Bounding-box Level	Scene Level	Not Clarified
Qurishee (IRT) [26]	2020	108 (IRT) 2620	up to 1024 × 768 up to 838 × 809	Not clarified Hand-held phone and UAV	Cracks Cracks (18 damage classes)	Pavement	Asphalt	Bounding-box Level	Pixel Level	CC BY 4.0 License
CODEBRIM [93]	2019	1590	up to 6000 × 4000	Hand-held cameras Cameras on UAV	Cracks and corrosions (5 damage classes)	Bridge	Concrete	Bounding-box Level	Pixel level Object & Scene Level	Private License, for Academic Use Only
GC10-DET [94]	2020	3570	up to 2048 × 1000	Hand-held cameras	Cracks and corrosions (10 damage classes)	Industrial plant	Steel	Bounding-box Level	Pixel Level	CC BY 4.0 License

Table 6
A summary of network architectures and training strategies adopted by the corresponding **detection-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
RDD-2018 [87]	2018	a. SSD [95] (Inception V2 [96]) b. SSD [95] (MobileNet [78])	✗	✓	Flipping
RDD-2019 [88]	2019	a. SSD [95] (ResNet-50 [44]) b. SSD [95] (MobileNet [78])	✗	✓	PG-GAN [97], Poisson blending [98]
RDD-2020 [89, 90, 91]	2020	- SSD [95] (MobileNet [78])	✓	✗	✗
PID [92]	2020	a. YOLO V2 [99] b. Fast R-CNN [100]	✓	✗	✗
Qurishee (IRT) [26]	2020	- Faster R-CNN [101]	Not clarified	Not clarified	Not clarified
CODEBRIM [93]	2019	a. MetaQNN [102] b. Efficient Neural Architecture Search [93]	✗	✓	Cropping
GC10-DET [94]	2020	a. SSD [95] (VGG-16 [45]) b. Faster R-CNN [101] (ResNet-50 [44]) c. YOLO V2 (DarkNet-19) [99] d. YOLO V3 (DarkNet-53) [103] e. SSD [95] (VGG-16 [45])	✓	✗	Patches, Scaling

RDD-2019 [88] dataset is an extension and refinement of the RDD-2018 dataset. Compared to RDD-2018, RDD-2019 increases the number of annotated frontal-facing road images from 9,053 to 13,135, resulting in 30,989 road damage instances. All the newly added images with a resolution of 600 × 600 are still collected in Japan through a vehicle-mounted smartphone. Besides, all the images contained in RDD-2018 are reviewed, quality-controlled, and reannotated by road managers. A new class called 'utility hole' is added into the RDD-2019 dataset to discriminate the damage class 'pothole' from it. To expand the size of dataset, the authors apply progressive growing Generative Adversarial Network (PG-GAN) to generate synthetic images with 'pothole' damage class, more results can be found in [88]. However, the RDD-2019 dataset only includes real images.

RDD-2020 [89] is an extension of RDD-2019 [88] by incorporating additional road images taken in the Czech and India, which makes this dataset more heterogeneous and conducive to network robustness. It offers 26,336 frontal-facing road images collected by a vehicle-mounted smartphone in Japan, Czech, and India. These images contain more than 31,000 road damage instances with a wide variety of light and weather conditions. The whole dataset is partitioned into a training set and two test sets, their respective amounts of images contained are 21,041, 2,631, and 2,664. Images for Japan and Czech have a consistent resolution of 600×600, while for India, the image resolution is 720 × 720. This dataset is dedicated to road damage detection, unlike RDD-2018 and RDD-2019, it only covers 4 damage classes, i.e. potholes, alligator cracks, longitudinal cracks, and transverse cracks. Some extra damage classes are included in images collected in Japan for data consistency, more details can be found in [90]. For each damage in the image from the training set, the damage class and its corresponding bounding box coordinates are labeled. RDD-2020 dataset has also been used as the benchmark dataset by Global Road Damage Detection Challenge (GRDDC) [105], performance of state-of-the-art solutions can be found in [91].

Pavement Image Dataset (PID) [92] collects 7,237 images of 22 different pavement sections in the USA from Google street view. The images, with a 640 × 640 resolution, come from two types of camera views, including a wide view and a top-down view. Images from the wide view are used to detect pavement distresses, and top-down view images are employed to calculate the crack density for automated pavement rating in the future. The pavement distresses in this dataset consist of 9 crack types, including reflective, transverse, block, longitudinal, alligator, sealed transverse, sealed longitudinal, and lane longitudinal cracking, along with potholes. The numbers of images used in training and testing are 5,789 and 1,448, respectively, and the tool used to annotate images is python-based Openlabling software.

Qurishee *et al.* [26] propose a pavement crack detection dataset with 336 test images and 2,284 training images. All the images are collected by a hand-held mobile phone camera and a drone's camera. There is a total of 11 categories of flexible pavement cracks and 7 classes of rigid pavement cracks. These images are labelled by the open-source tool LabelImg with more than 50 hours of manual labour. In addition, they also propose a very small but high-resolution infrared dataset with 24 test images and 84 training images.

Table 7
A summary of manual and semi-automatic labeling methods of classification, segmentation and detection (first sorted by the payment situation, then sorted by the degree of automation of the labeling)

Name	Annotation level	Other types of input	Format of the exported dataset	Automatic labeling	Local Deployment	Web-based Deployment	Free-of-charge
Ybat [107]	Bounding-box Level	✗	a. Pascal VOC format b. YOLO format c. COCO format	✗	✗	✓	✓
LabelImg [108]	Bounding-box Level	✗	a. Pascal VOC format b. YOLO format c. CreateML format	✗	✓(Win, Linux, macOS)	✗	✓
LabelMe [109]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. COCO format	✗	✓(Win, Linux, macOS)	✗	✓
VIA [110]	Bounding-box Level	✓(audio, video)	a. COCO format b. VIA format c. CSV format	✗	✗	✓	✓
VoTT [111]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. TFrecord format c. VoTT format d. CSV format	✗	✓(Win, Linux, macOS)	✓	✓
PixelAnnotationTool [112]	Pixel Level	✗	- Only mask images (PNG files)	✓	✓(Win, Linux, macOS)	✓	✓
CVAT [113]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. COCO format d. TFrecord format e. CVAT format (and other 13 formats)	✓	✓(Win, Linux, macOS)	✓	✓
RectLabel [114]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. CreateML format d. CSV format	✓	✓(macOS)	✓	✗
Labelbox [115]	Bounding-box Level Pixel Level	✓(audio, video, text)	- Only JSON files containing labels	✓	✗	✓	✗
V7 Darwin [116]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. CVAT format d. Darwin format	✓	✗	✓	✗

3.3.2. Bridge

COncrete DEfect BRidge IMage (CODEBRIM) dataset [93] focuses on the defects of concrete bridges. The images with defects are captured from 30 bridges by UAV and can be divided into five classes: crack, spallation, exposed reinforcement bar, efflorescence, and corrosion. In order to detect minor defects from different scales, cameras with high resolution (up to 6000×4000) and large focal lengths are adopted to collect images. One highlight of this dataset is that the images are labeled with multi-class, and the defects in the same image can be overlapped. There are only $1,590$ high-resolution images in this dataset, but the total number of labeling box are $7,806$, $5,354$ of which are overlapping defect and $2,506$ of which are non-overlapping.

3.3.3. Industrial plant

GC10-DET [94] dataset pays attention to the surface defect in a real industrial plant. The images with a resolution of 2048×1000 are captured by a set of linear array CCD cameras with a direct current light source to avoid the presence of stripes produced by an alternating current. The pixel size of the camera is $7.04\ \mu m \times 7.04\ \mu m$. Compared with the NEU-DET dataset [106], GC10-DET has more data and a greater variety of defect types: punching, weld line, crescent gap, water spots, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding. With real scenes, high-precision collection tools, and high-resolution data, the AI models can be greatly enhanced and highly robust after training on this dataset.

3.4. Data collection and labeling

The data collection and labeling procedure can be summarized as follows: they are both labor-intensive and costly procedures. The first step in data collection is to survey the target site in advance to make the collection plan and select the collection equipment. Weather, light, and equipment all affect the quality of the dataset. After collecting the original data, it is necessary to clean it and eliminate similar and ambiguous data artificially. The next step is data labeling. Although some mature methods have been proposed for the labeling of classification and pixel-level segmentation [117] tasks, and some commercial software has been deployed on the website for user-friendly labeling, the efficient labeling strategy for large-scale unlabeled datasets is still in its infancy. This subsection illustrates and summarizes several labeling tools and their properties, including the annotation level, input data type, export format, labeling automation level, deployment configuration, and public accessibility in Table 7. The most commonly used open-source annotation tool is also compared with another semi-automatic open-source annotation tool to highlight the efficiency of automatic labeling.

With the popularity of deep learning-based image processing, many open-source annotation tools have emerged, including Ybat [107], a web-based annotation tool specially designed for the YOLO [99, 103, 118] series algorithm. And the classic annotation tool LabelImg [108], the most widely used open-source annotation tool LabelMe [109], VGG Image Annotator (VIA [110]) developed by VGG [45] network team that can efficiently annotate faces, and VoTT [111], a web-based annotation tool developed by Microsoft team. In addition to the above common open-source manual annotation tools, many semi-automatic annotation methods are also free-of-charge. PixelAnnotationTool [112] is a semi-automatic annotation tool for semantic and instance segmentation annotation tasks. CVAT [113] is a powerful and community-established semi-automatic annotation tool that

supports exporting 18 different data formats. During the annotating process, the target needs to be clicked by several key points, and then CVAT will automatically annotate the target. Although the functions of open-source annotation tools can meet our daily needs, they are still inferior to commercial annotation tools. RectLabel [114] is an annotation tool aiming at macOS users. RectLabel is unique in its ability to split images into uneven pieces, which the user can adjust to speed up automatic annotation. Labelbox [115] and V7 Darwin [116] are commercial annotation tools that can be used by simply logging into their web pages. Both can invite teammates to join in for annotation, orchestrate complex workflows, visualize annotation results and processes, and optionally train a specific network to improve the accuracy of automated annotation. In addition to the open-source and commercial tools, some researchers attempt to utilize machine learning algorithms for automatic labeling to get preliminary labeling results which can be then manually refined for accurate labeling in a much shorter time [119, 120, 121].

To compare the efficiency of the purely manual and the semi-automatic open-source tools, LabelMe (manual) is evaluated against CVAT (semi-automatic). The efficiency of CVAT is twice that of LabelMe, especially for some ordinary objects, such as vehicles and pedestrians. The automatic labeling algorithm identifies the object in two seconds with adjustable selection box details. The automatic labeling algorithm is prone to errors for defect datasets, such as cracks and spalling. However, in our experiment on the self-collected data, long cracks are divided into many small parts for labeling and the boundaries of labeled polygons are modified in a centralized way, saving about one-third of the time for manually labeling an RGB image full of cracks with a resolution of 6000×4000.

4. Comparison of SOTA algorithms for crack inspection

Before illustrating our algorithm comparison in detail, it is noteworthy that there exist recently published valuable works performing crack classification and segmentation tasks in different scenarios comparatively. Hallee *et al.* [123] pay their attention to masonry crack detection, where they systematically compare the domain adaption performance between the convolutional neural network (CNN) and traditional machine learning methods based on handcrafted features, including Support Vector Machine (SVM), Random Forest (RF), Gaussian Process (GP), Multi-Layer Perceptron (MLP), Naive Bayes (NB), and Quadratic Discriminant Analysis (QDA). The critical conclusion [123] is that successful domain adaption is possible in both the CNN and simple classifiers if trained on a wide range of masonry shapes, colors, and lighting conditions, complying with our conclusion in Subsection 5.1.1, Subsection 5.2.3, and Subsection 6.1.

Loverdos *et al.* [124] are dedicated to automating brick and crack segmentation of masonry walls. Regarding brick segmentation, extensive comparison experiments are conducted among networks, including U-Net, DeepLab V3+, LinkNet, and Feature Pyramid Network (FPN), all with various configurations. As to crack segmentation, SOTA architectures (with multiple backbones, training strategies, and loss functions) including DeepCrack, DeepLab V3+, Fully Convolutional Network (FCN), U-Net, and FPN are systematically compared to identify the best model configuration. The results are impressive when the brick segmentation and crack detection outputs are coupled. The essential remarks [124] are that deep learning methods allow for improving model performance by increasing the dataset used for training and validation, and the model performance can continually be enhanced by acquiring additional samples of the classified elements and desired features. These valuable remarks show the necessity and importance of our summarized datasets. Rezaie *et al.* [125] focuses on the crack segmentation of the stone masonry walls. They systematically compare a threshold method based on Digital Image Correlation (DIC) results and a deep learning-based method named TernausNet. The remarks are on the superiority of the deep learning method and its potential benefits for DIC methods and predictive models for damage level evaluation.

Based on the previous literature review, it can be found that crack is the dominant defect category in common structures [1, 13]. Its recognition is significant for a variety of applications such as the fault analysis and safe operations of public infrastructures such as bridge [126], building [36], and the electrical power grid [127]. Therefore, this research further develops a self-established crack classification and semantic segmentation dataset, based on which SOTA inspection algorithms are compared. We have developed an adapted Swin Transformer [122] from previous cutting-edge algorithms for crack classification as shown in Figure 5, and proposed a multi-layer fused attentional pyramid network for crack semantic segmentation as shown in Figure 6, respectively. Extensive experimental results show that the proposed approaches achieve comparable performance and efficiency to current SOTA approaches. Moreover, comprehensive comparisons between existing SOTA algorithms for crack classification and segmentation are conducted to provide a comprehensive baseline for future research in infrastructure defect inspection.

4.1. Our self-established crack classification and semantic segmentation dataset

A large-scale dataset for both crack classification and segmentation tasks is first established. Data for the classification task contain more than 15,000 images with image-level labels (crack or non-crack), while those for semantic segmentation contain more than 11,000 images with detailed pixel-level labeling. For crack semantic segmentation, 42% of images are derived from the internet and the remaining 58% is collected in our on-site inspection. And the corresponding percentage for classification is 36% and 64%, respectively. The preliminary version of our dataset has been released online at the following link to benefit the research

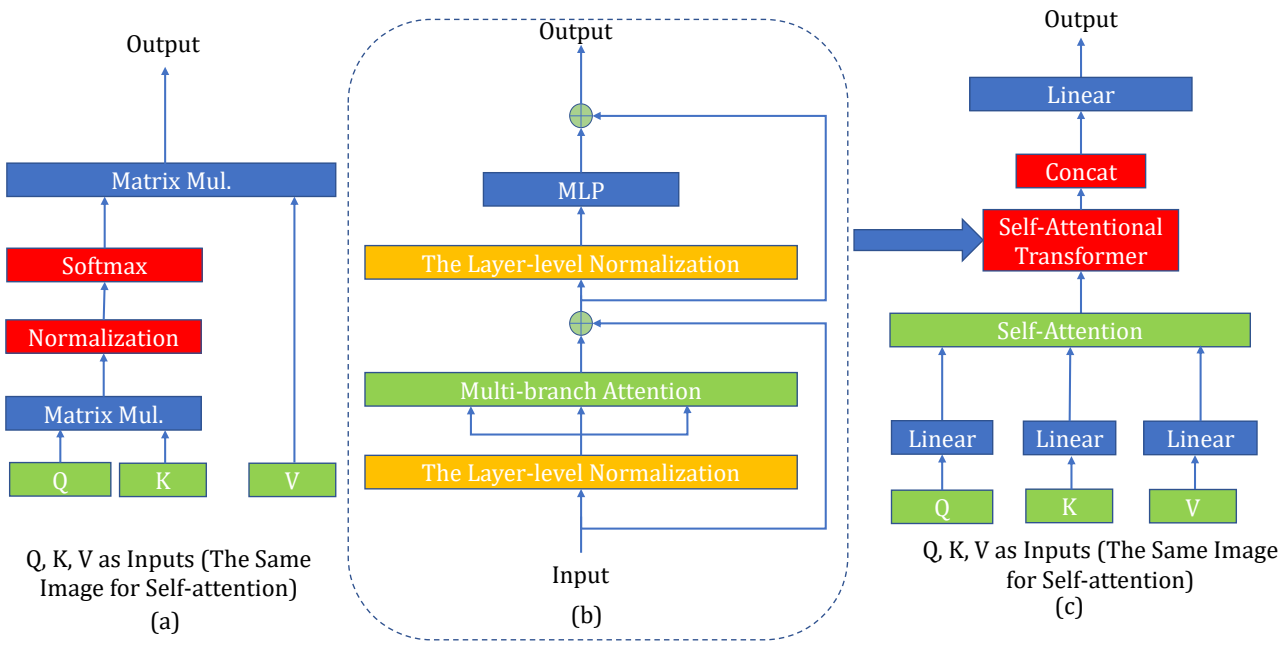

Figure 4: The detailed structure of the attentional module to be integrated into the Swin Transformer [122]. In subfigure (a), we have illustrated the network component of the original attention-based network. In subfigure (b), we have shown the multi-branch self-attention module we proposed to integrate into the current Swin Transformer [122] to boost the performance. Summarizing the whole network in the dash-line rectangle as a new module and integrating it into subfigure (c) as a self-attention transformer, the performance of the original transformer can be improved according to our experiments.

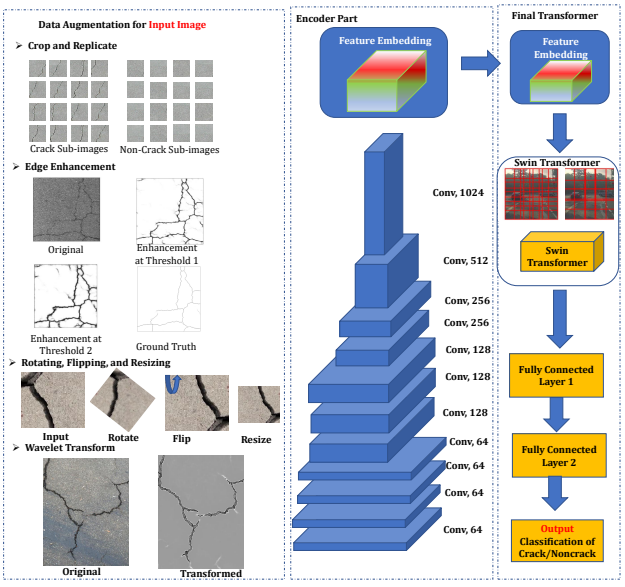

Figure 5: The detailed structure of the classification network adopted by us adapted from the Swin Transformer [122]. The network can achieve SOTA performance with the fine-tuning.

community for defect recognition[3]. Typical results are also shown on the website attached. Currently, this dataset can be used to perform crack recognition on the pavements effectively, and will be enriched further for building and tunnel inspections. In this way, it can be used for general UAV-based infrastructure inspections.

[3]Our Established Datasets Preliminary Version

4.2. Comparison of SOTA algorithms for crack classification

This subsection focuses on the task of crack classification, predicting whether a specific image contains a crack or not. Based on our self-established dataset, the existing SOTAs methods are compared, and the evaluation metric of the classification efficiency and effectiveness is detailed. Extensive experiments show that both the adapted Swin Transformer shown in Fig. 5 and the traditional convolutional network ResNeSt [128] show the best performance for crack classification. However, the ResNeSt [128] shows greater performance in the inference speed, and is more favorable in real industrial applications.

4.2.1. The definition of evaluation metric of crack classification

For the crack classification problem, the **accuracy** can be simply defined as follows:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (1)$$

For the binary classification, accuracy can be further defined in terms of positives and negatives as follows:

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

Where TP, TN, FP, FN stand for True Positives, True Negatives, False Positives, and False Negatives, respectively. Accuracy is the most direct metric for the one-hot prediction task of image-level crack classification, and also the

fairest. Then, we use the accuracy to make a comprehensive comparison between the SOTA algorithms for the task of crack classification. Also, the validation inference time on the tested images with the resolution of 1500×960 is used to evaluate the efficiency of diverse SOTA methods.

It is noteworthy that the accuracy metric can be misleading if there exists a class imbalance in the dataset. In our training dataset for crack classification, crack and non-crack images are balanced with the proportion of 41.3% and 58.7%, respectively. In the image classification task of more than 2,000 images, the indeterminacy caused by the class imbalance can be neglected if the training set is not extremely imbalanced. Various techniques [129] can be adopted encountering class imbalance, such as under-sampling, oversampling, merging similar classes, and data augmentation.

4.2.2. Algorithms illustration

The network architectures are depicted and detailed in Figure 4 and Figure 5 for our crack classification-based crack detection applications. For the network settings, in this work, we only utilize the Swin Transformer [122]-based approach as an example. The transformers [122, 130] are the most up-to-date transformer-based network architectures for the general vision task of image classification. The transformer [130]-based network architecture has recently surpassed the CNN network architectures with its massive network architecture consisting of fully-connected network layers with a huge number of parameters. We firstly illustrate the basic ideas of our proposed multi-branch attentional transformers. As shown in Figure 4, in subfigure (a), we have illustrated the network component of the original attention-based network. In subfigure (b), we have shown the proposed multi-branch self-attention module to leverage the semantic correlation among various transformed feature representations for image patches. Finally, it can be integrated into the current Swin Transformer [122] to boost the performance. Also, the multi-branch self-attention module in subfigure (b) has the advantages of a larger receptive field and multi-branch concatenated feature representations, which are both significant to better modeling the contextual information within the image. After summarizing the whole network in the dash-line rectangle (subfigure (b)) as a new module and integrating it into subfigure (c) as a self-attentional transformer, the performance of the original transformer can be improved according to our experiments. The sliding window-based approach is utilized for the final crack classification-based crack detection.

It should be noted that the original Swin Transformer [122] can not handle the high-resolution testing image in training, which will result in out-of-memory for ordinary GPU devices. Therefore, we have cropped the original crack images into 60×60 sub-images for the ease of training with transformer [122]. As shown in Figure 5, our framework consists of data augmentation, the encoder part for feature embedding, and the final transformer. To deploy the computation-intensive transformer [122]-based models for crack classification, images should firstly be split/cropped into sub-patches to make it memory-efficient and computationally tractable for the original GPU such as NVIDIA GTX 1080 with 8 GB memory. The data augmentation is also of great significance to the final performance, for the fact that it can create more training samples for the better instance discrimination at the feature level. In this work, we have proposed to use the following four kinds of data augmentation. The crop and replicate, the edge enhancement, the rotating, flipping, resizing, and finally, the wavelet transform. The encoder of the network converts the input image to a feature embedding. Finally, the sub-images are fed into the Swin Transformer for the crack classification task.

We have also utilized the current popular architectures such as the ResNeXt-101 [131], the ResNeSt-101 [128] for doing the crack classifications. Also, the crack classification task takes longer in validation time as shown in Table 8 because we directly tested on the images with a large resolution of 1500×960. We have also tested and utilized the up-to-date transformer-based network architectures. Table 8 shows the related results. Note that our utilized Swin Transformer is also based on the widely adopted attentional feature correlation mining networks [132], which is the fundamental component of all transformer-based networks.

4.2.3. Detailed partitions of our dataset for crack classification

For the task of crack classification, the training set consists of 10,000 images. Moreover, the validation and test sets are composed of 4,500 and 500 images, respectively. For the large memory consumption of the transformer and the fairness of comparisons, we have utilized 120×100 sub-image for the training, and we have used 500 images with a resolution of 1500×960 for testing.

4.2.4. Experimental settings

For the task of **crack classification**, we train all compared networks in a unified setting. We train networks for 500 epochs on a single NVIDIA 2080Ti GPU with a batch size of 32 during training and 16 during testing. The initial learning rate is $5{\times}10^{-3}$ and decays by five times every 100 epochs. We select 500 epochs because training for 500 epochs is enough for the convergence of networks. Finally, we select the network weights that have the best performance on the validation set to do testing on the test set. We implement it in *Tensorflow* and optimize it with Adam optimizer [133]. Training the models to convergence takes approximately 9.5 hours for our self-established dataset with various crack patterns for the ResNeXt-101 [131] for example. All the models are trained from scratch for the task of crack classification. Furthermore, all our results are obtained from the results three times on average. Therefore, we have guaranteed fairness and robustness in all of our comparisons. In the future, we will also explore the possibility of large-scale pre-training and transfer learning-based approaches to achieve the relatively large-scale crack classification of more than a million images. However, although the performance

Table 8
The comparison of **crack classification** results between SOTA algorithms for the tested images with the resolution of 1500 × 960.

Network Architecture	Accuracy/%	Validation Time /ms
AlexNet [47, 134]	81.8	698.6
VGG-16 [45]	86.4	678.5
VGG-19 [45]	87.1	689.6
GoogLeNet [48]	83.6	875.5
ResNet-101 [44]	87.2	617.5
ResNeXt-101 [131]	87.9	1213.5
ResNeSt-101 [128]	88.2	1063.8
Swin Transformer-Base [130] [122]	87.7	2382.3
Swin Transformer-MB [130] [122]	88.0	2587.5
ShuffleNet [135]	85.7	1567.7
ShuffleNet V2 [136]	86.3	1645.8

of the large-scale pre-training is very prominent, the efficient network architectures and the efficient training strategies must be explored to put the large-scale pre-training into practice. Otherwise, it will remain a complex problem for academic research without the availability of high computational power.

4.2.5. The optimization loss function

For the optimization loss function, for simplicity and to guarantee the fairness of comparisons, we have adopted the unified cross-entropy optimization loss. The cross-entropy loss was used for the network training of all networks, including the SOTA networks and our proposed ones, which can be formulated as follows:

$$L = -\sum_{x^{In}} [y(x^{In}) \cdot log(p(x^{In})) + (1 - y(x^{In})) \cdot log(1 - p(x^{In}))] \quad (3)$$

where $p(x^{In})$ represents the predicted possibility of whether an input image x^{In} is a crack image, and y is the label of the input image. For crack image, $y = 1$. For non-crack image, $y = 0$. The loss can be utilized for the end-to-end training of the network framework. And finally, we present our experimental results.

4.2.6. Experiment results of crack classification

We conduct experiments to test the performance of various crack classification networks. The networks we tested have covered a broad range, which consists of current SOTA network architectures, including the classical AlexNet [134], and the newly proposed vision-transformers [122, 130]. As shown in the Table 8, we have also tested with other SOTA network backbones for crack classification, such as the vision transformer (ViT) [122, 130] which has the best performance among various methods in recent vision benchmarks. The results demonstrate that the recent approach, such as the ResNeSt [128] also has comparable performance with ViT, and has a much faster inference speed compared to the ViT. It can be seen that when using our multi-branch attentional layer in the Swin Transformer [122] (denoted as Swin Transformer-MB in Table 8), the performance can be boosted a little with a merely marginal increase on the computational cost (0.2 s validation time increase for the inference per image of 1500 × 960). It can be demonstrated that although the vision transformer-based methods can achieve remarkable performance, the computational and memory costs should be considered in the deployment stage. For the robotics applications with real-time requirements, the faster methods such as the ResNet-101 [130] or ResNeSt-101 [128] are more preferred for efficiency considerations. Also, although the Swin Transformer [122] based methods have comparable or slightly better performance under various circumstances compared with the typical convolutional network [45], and residual network [128] based methods, it requires a large inference time, which is unacceptable in real-time applications. Therefore, taking the efficiency and accuracy of both into consideration, the ResNeSt [128] is the best choice for the crack classification task.

4.3. Comparison of SOTA algorithms for crack segmentation

This subsection takes crack semantic segmentation as a case study of defect recognition in modern infrastructures. As mentioned in Subsection 4.1, the images are labeled in pixel levels for segmentation and summarized into our self-established dataset, based on which performances of various network architectures are compared in detail. Utilizing our designed network architecture (Fig 6) combined with existing SOTA network backbones such as ResNet [44], ResNeXt [131], and VGG [137], the performance will be enhanced when the domain gap between the source and target test data is not large. Table 9 presents the comparison results of recent crack segmentation methods such as the DeepCrack [63].

4.3.1. The definition of evaluation metrics of crack semantic segmentation

We have utilized various metrics for a fair evaluation of the performance of different methods, as shown in Table 9. The inference time is the testing time for an image of resolution in 600 × 480 for the task of crack segmentation. We define the average precision, mean Intersection over Union (mIoU), precision, recall, best F-measure on our test set for a fixed threshold (DS), and the total F-measure on our test set for the threshold on each image (IS) in the same settings as the [138]. These evaluation metrics are commonly recognized and adopted evaluation metrics for comparisons in defect identification. Also, the validation inference time on the test images is used to evaluate the efficiency of diverse SOTA methods. Our experimental results are conducted three times to obtain an average value for fair comparisons.

4.3.2. Algorithms illustration

We have developed algorithms for crack segmentation and the subsequent detection based on non-maximum suppression (NMS). To keep the paper brief, we will merely take our proposed Attentional Pyramid Scene Parsing-based network architecture integrated with full resolution ResNet [139] as an example case for algorithms illustration. As

shown in Fig 6, the network adopts the typical encoder-decoder-based basic structure for semantic segmentation. Unlike the transformer-based model for the task of crack classification, the input image can be directly fed into the encoder of the network structure based on these convolutional neural network-based segmentation models. As shown by the red module linking the intermediate feature output encoder and the decoder of the network in Figure 6, we have also incorporated the attentional transformer for the self-correlated feature extraction of the image at the pixel level. Utilizing this kind of design, the correlated features in the embedding space will be effectively enhanced, and the distinct features will be well separated. The attentional transformer shown in red in Figure 6 is used to enhance the feature correlation mining capacity of the network. All the decoder features are ultimately concatenated to give the final predictions. This kind of network design can make the model focus more on the critical zones of the images and pay less attention to the insignificant ones. Also, the attention-based transformer can be beneficial in increasing and enlarging the spatial contextual information and fusing them with the low-level feature representations in the encoder. We adapted it based on the SOTA attention-based [132] transformer and integrated it into our network structure. Through this kind of network architecture, the low-level feature cues, such as the edges and corners, and the high-level semantic cues, such as the crack patterns, can be fully utilized and learned based on the training data. Moreover, we use selective search-based methods to do the detection. The selective search-based methods [140] use the traditional sliding window-based approaches for object detection. Furthermore, we utilized the efficient nearest neighbor query methods to find the next sliding window for detection and efficiently do the final object detection. Finally, as shown in the Algorithm 1, we have summarized the proposed detailed procedures for NMS-based object detection. Denote B as the list of the initially obtained detection boxes. S contains the corresponding detection scores. And N_t is the NMS threshold. The set D is utilized to store the final box. As shown in Fig. 7, we can utilize NMS to obtain the most typical object detection bounding boxes obtained from selective search-based methods in the original RGB images.

4.3.3. Detailed partitions of our dataset for crack semantic segmentation

For the tasks of **crack semantic segmentation**, we have partitioned the original dataset into the training set, the validation set, and the test set. The dataset consists of more than 11,000 images with a resolution of 600 × 480. We have utilized 6,000 images for training, 3,000 images for validation, and the remaining 1,650 images for testing.

4.3.4. Experimental settings

We adopt the same setting as crack classification except that the initial learning rate is $1{\times}10^{-4}$. Training the model to convergence takes approximately 17.5 hours for our self-established dataset with various crack patterns. All the models are trained from scratch for the task of semantic segmentation.

Algorithm 1: The non-maximum suppression based algorithm for object detection (Simplified Version)

Input: The **input** initial detection boxes B, the related corresponding detection scores S, the related NMS threshold N_t
Output: The **output** final detection boxes D and the corresponding detection score S.

$D \leftarrow \emptyset$
while $B \neq empty$ **do**
  Select the maximum value in the set of S, and give this value to m. $m \leftarrow argmax(S)$
  $M \leftarrow b_m$
  $D \leftarrow D \cup M$
  $B \leftarrow B - M$
  for b_i *in* B **do**
    if $iou(M, b_i) > N_t$ **then**
      $B \leftarrow B - b_i$; $S \leftarrow S - s_i$;
  return D, S

4.3.5. The optimization loss functions

In addition to the network architectures illustrated above, we further illustrate the optimization functions used for the network training. In real situations, the crack is usually thin, which means most of the pixels in the captured images are non-crack. Different from the traditional cross-entropy loss, we have proposed our class-balanced loss function to tackle the problem of extreme class imbalance in the task of crack semantic segmentation. Also, we have also proposed the multi-stage fused loss, which can operate well with our proposed multi-stage fused pyramid network to boost the network performance. The optimization loss function is detailed as follows. We calculate the total number of crack and non-crack pixels in the training images are p and q respectively. The class frequencies of crack and non-crack are $\frac{p}{p+q}$ and $\frac{q}{p+q}$, while the median for the 2 classes is 0.5. Then the median divided by the class frequency gives the weight of two classes. In our case, the weights of the loss function for the crack pixels and non-crack pixels are $\alpha_1 = \frac{p+q}{2p}$ and $\alpha_2 = \frac{p+q}{2q}$ respectively. Then for each side-output layer, the improved loss function for the h-th side outputs $L^h_{side}(W)$ can be formulated as:

$$L^h_{side}(W) = -\alpha_2 \sum_{j \in S^-} log(1 - P(W)) \\ -\alpha_1 \sum_{j \in S^+} log(P(W)) \tag{4}$$

where $h = 1, 2 \cdots, H$ respectively are the convolutional stages of the network. The H denotes the total stages. S^+ and S^- are the total number of crack pixels and non-crack pixels respectively for an input image. And P denotes the predicted possibility of each pixel to be a crack one. The W denotes the

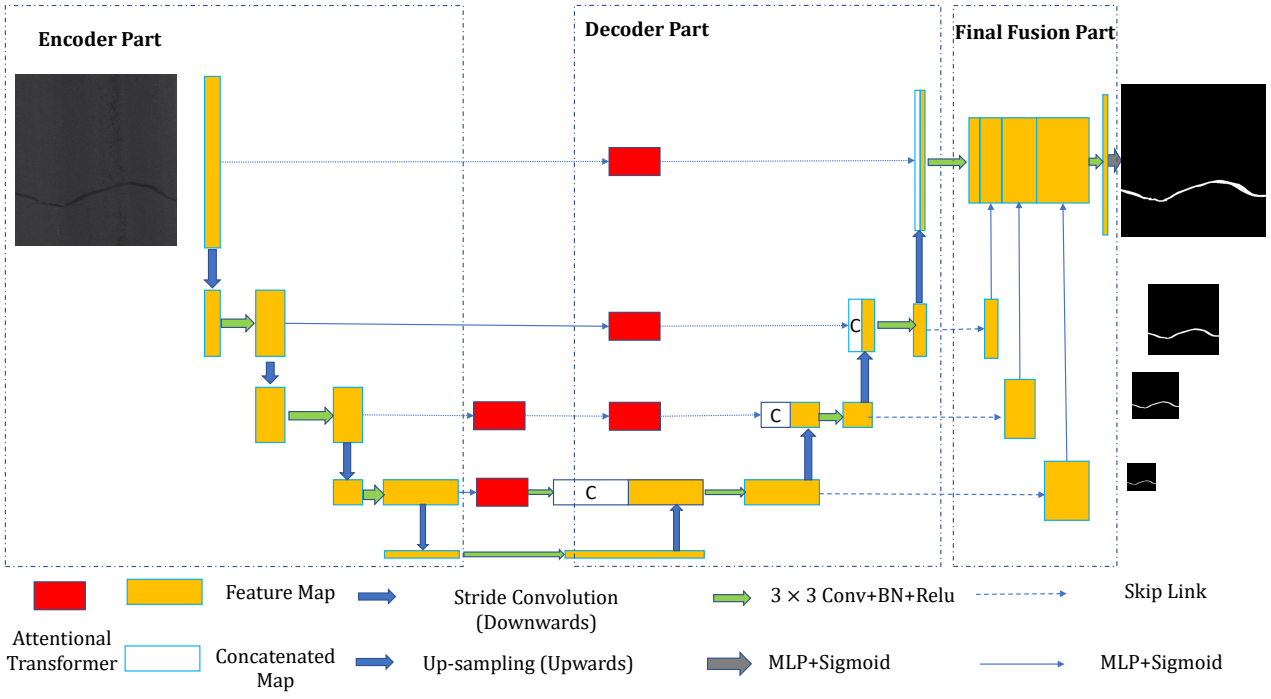

Figure 6: The proposed Multi-Stage-Fused Attentional Pyramid Network structure. We have proposed an encoder-decoder network architecture with skip-connections enhanced by the attentional transformer module. The attentional transformer module is added to better enhance the final segmentation performance.

Table 9
The comparison of semantic segmentation results between our proposed and various current SOTA methods

Methods	Inference Times/ (ms)	Thres (0-1)	Average Precision	mIoU	Precision	Recall	DS	IS
Original Hierarchical Neural Network [141]	165	0.49	82.3	75.9	74.6	76.5	75.6	77.5
SegNet [142]	215	0.52	80.2	75.6	73.3	74.8	74.1	74.7
FCN-8s [143]	176	0.55	81.1	76.9	74.2	75.5	74.8	75.8
U-Net [143]	168	0.53	82.1	77.1	73.7	74.9	74.3	75.3
DeepLab V2 [144]	192	0.55	83.2	78.7	76.9	75.9	76.4	75.6
DeepLab V3 [144]	226	0.50	83.6	79.3	74.9	74.9	74.9	75.7
PSPNet V1 [145]	257	0.49	83.4	79.8	75.5	75.6	75.6	76.3
ASPP-Net [146]	266	0.51	85.2	78.9	75.4	75.7	75.6	76.2
DeepCrack [63]	708	0.50	78.6	76.9	71.2	72.3	71.7	72.3
CrackNet based DeepLab V3+ [141] [144] (Our)	252	0.45	**86.3**	**77.8**	**75.3**	**75.6**	**75.5**	**75.8**
CrackNet based DenseNet [141] [147] (Our)	502	0.51	**86.6**	**77.6**	**76.1**	**75.1**	**75.6**	**76.3**
CrackNet based Full Res-ResNet [141] [139] (Our)	324	0.56	**87.3**	**76.9**	**76.6**	**75.5**	**76.1**	**76.6**

weights of the whole proposed transformer-based network shown in Fig. 5. Next the improved loss function $L_{fuse}(W)$ for the fused output can be also written as:

$$L_{fuse}(W) = -\alpha_2 \sum_{j \in S_-} log(1 - P(W)) \\ -\alpha_1 \sum_{j \in S_+} log(P(W)) \tag{5}$$

And then the total optimization loss function $L_{total}(W)$ is written as:

$$L_{total}(W) = \sum_{j=1}^{J} (\sum_{h=1}^{H} L_{side}^{h}(W) + L_{fuse}(W)) \tag{6}$$

The multi-stage fused optimization loss functions has the advantages of considering both the low-level feature in the early stages of the network such as edges and corners, and the high-level semantic information in the deeper stages of the network. Thus, the multi-stage hierarchical information can be extracted and fused in an adapted manner and this kind of information is further formulated into the network optimizations to boost the final segmentation performance in an explicit way.

4.3.6. Experiment results of crack semantic segmentation

The results of crack segmentation have been shown in Figure 7. We have utilized the characteristics of our proposed network shown in Figure 6 to construct multi-stage deep hierarchical feature representations for each tested network. The feature pyramid network has been demonstrated to

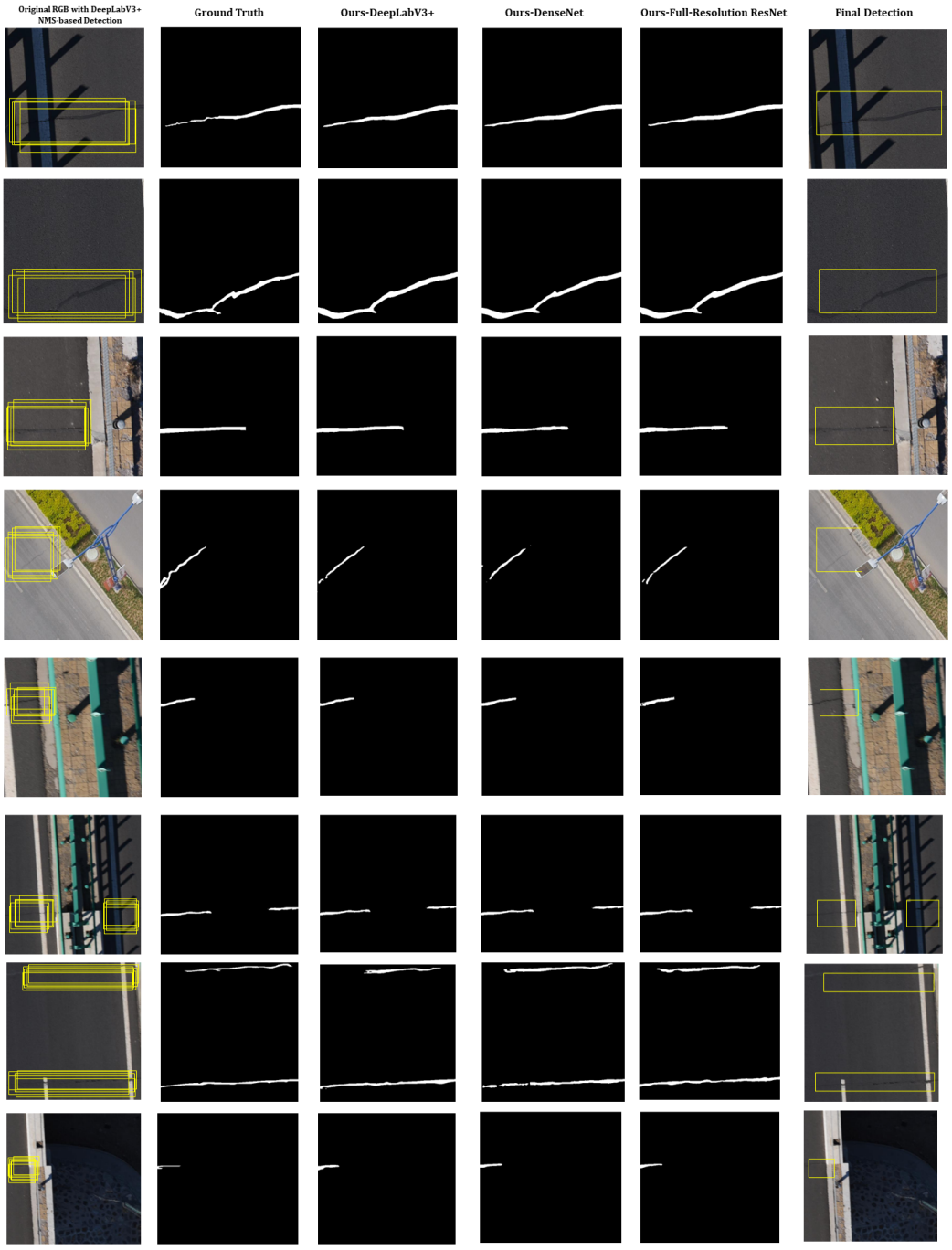

Figure 7: The pixel-level semantic segmentation results of our adapted CrackNet shown in Fig. 6 integrated with SOTA deep convolutional networks for semantic segmentation after conducting real-site pavement inspection. The black color denotes the background, while the white color denotes the segmented results of various cracks. Our approach can realize accurate segmentation of the cracks under different complex backgrounds, with large shadows, road greenings, and roadside bricks.

be very effective in the hierarchical and multi-layer fused feature extraction [148, 149]. As mentioned, the proposed network shown in Figure 6 adopts the encoder-decoder-based architecture. It is highly effective in feature extraction because we fuse the representations from diverse network stages to obtain a compound representation integrating low-level geometry cues (including edges and corners) and high-level semantics (including the category and semantic feature representations). Also, we have successfully integrated our proposed network with SOTA dilated convolutions network Deeplab V3+ [76] as shown in the third column of Figure 7. In the deeper stages of the network, we choose to use the dilated convolutions instead of the original classical convolutions because the dilated convolutions provide a much larger receptive field. And the spatial resolutions can be well maintained. Also, the dilated convolutions generate the feature

maps of various scales compared with the input data, which further improves the scale robustness of the proposed network. Moreover, we have utilized more elaborately designed network architectures such as the DenseNet [147] and SENet [150]. The densely connected residual learning is conducted to obtain a better feature representation. The full-resolution residual networks (FRRNs) [139] utilize the two streams to fuse the multi-scale global contextual information with the pixel-level local information. The first stream carries information at full resolution to achieve accurate segmentation of the boundaries of various shapes. The second stream utilizes a series of max-pooling operations to obtain the high-level feature for recognition. The FRRNs [139] couples these two streams and finally provide a hierarchically fused segmentation map. In our work, we have adapted the original FRRNs [139] for a better multi-stage fusion of hierarchical features based on our design shown in Figure 6. For the crack segmentation, a certain segmentation threshold (denoted as Thres on Table 9) require to be chosen for obtaining the final binary segmentation map. We have chosen the threshold for various tested segmentation methods based on their original implementation in their original paper.

We have successfully integrated our proposed attentional pyramid network architecture shown in Fig. 6 with current SOTA deep network models for semantic segmentation, such as the DeepLab V3+ [144], DenseNet [147], and FRRNs [139]. As shown in Fig. 7, it can be demonstrated that integrated with our proposed method [141], the DeepLab V3+ [144], DenseNet [147], and FRRNs [139] all show superior performance when encountered with the diverse background if the network training parameters are fine-tuned. We have shown the typical detection results under diverse complicated circumstances. Our approach can realize very accurate segmentation of the cracks under different complex backgrounds, with large shadows, road greenings, and roadside bricks. It can be seen that the shadows can be properly handled. Although shadow pixels do not have sharp contrasts with the background pixels of road surface, the network will not mistakenly recognize the high-contrast shadows as cracks. From the last column, we have also shown the object detection results with the non-maximum suppression (NMS)-based post-processing approach. The details of the NMS algorithm are shown in Algorithm 1. It is demonstrated that it can suppress the redundant bounding boxes obtained from the semantic segmentation, and select the most typical detection bounding box result based on selective search for object detection [140]. Also, it can be seen that the selective search methods can find candidates with excellent efficiency and robustness. The object detection follows the semantic segmentation results to do more accurate crack object detection, which demonstrates the effectiveness and robustness of our proposed method [141]. The final results of crack semantic segmentation performance are also summarized in Table 9. It can be demonstrated that our proposed method can be successfully integrated with SOTA methods and shows consistently better performance compared with other ones. The performance increment can be ascribed to the effective network design of attentional transformer module and effective multi-stage fusion strategies. We have integrated our proposed network architecture in Figure 6 with three typical segmentation backbone networks. As shown in Table 9, it can be seen that the three proposed networks all achieve SOTAs semantic segmentation performance in terms of mIoU. Also, the provided comprehensive comparisons between existing SOTA algorithms for crack classification and segmentation can provide a solid baseline for future research in industrial infrastructural defects inspection.

5. Suggestions on datasets and methodology

5.1. The suggestions on constructing a defect dataset

5.1.1. Classification-oriented dataset

Classification task is the basic building block to the detection and segmentation task. To build up a high-quality dataset for defect classification, the defect categories should be firstly defined according to the government inspection guidelines. Then, the data collection procedure should be conducted and recorded in a controlled environment by strictly following inspection guidelines. The data collection system should be developed or chosen for specific application scenarios. The accuracy and robustness of the object classification algorithm face several challenges posed by object viewpoint variation, intraclass variation (e.g., the same type of crack but with a different background or color intensity), the difficulty of identifying fine-grained categories (e.g., various types of the crack), background clutter, illumination changes, deformation, and occlusion. The dataset can wittingly incorporate images with the challenges mentioned above to improve the accuracy and robustness. Besides, it should be noticed that there exist conflicts between the labeling results of different annotators. The effect of annotation conflict can be alleviated by introducing a self-checking mechanism during the labeling process or utilizing label smoothing techniques during the network training process [15]. Moreover, data augmentation (e.g., crop and flip) can be adopted to increase the data volume.

5.1.2. Segmentation- and detection-oriented dataset

The dataset should be recorded in a standardized way. The corresponding infrastructure type, material type, defect type, data type, sensor specifications, data collection procedure, and geometric properties of the defects should be recorded. The dataset should have sufficient data and defect diversity to train a superior defect detector. Traditional data augmentation and GAN-based data augmentation (e.g., Defect-GAN [151]) can be used to increase the data volume. The context level of the dataset also matters. The pixel-level context is conducive to the network training process, while the object-level context is beneficial to localizing the defects, relating the defects to the structure, and further evaluating the hazard level of defects. The scene-level context can increase the generalization ability of the trained model in real

applications. It is promising to build up a multi-modal defect dataset (e.g., SDNET2021 [30]). RGB images are conducive to detecting surface defects, while IRT images, IE signals, and GPR signals reveal subsurface defects. It should be noted that there are conflicts between different annotation results (even when annotated by experts) [15], which will influence the training result.

5.2. The suggestions on defect visual inspection methodologies

5.2.1. Developing advanced methods and algorithms

In real industrial applications, the specific infrastructure to be inspected can not be easily accessed. Although intelligent industrial robots such as UAVs or UGVs with sensing capacity have been developed, complicated autonomous localization, navigation, and planning algorithms should be developed to collect high-quality data on the target infrastructure to be inspected. In most cases, merely limited high-quality data for the inspected target can be collected, and the labeling process is time-consuming and cumbersome. Therefore, to train and deploy an effective crack recognition model for modern industrial applications, firstly, the efficient labeling strategy should be further explored to achieve highly efficient labeling, which we have discussed in detail in Table 7. Secondly, the domain gap should be considered in establishing the dataset. Domain adaptation is a great method that can expand the applications of the crack recognition model across different domains. From our experience, effective domain adaptation in crack detection and segmentation can be achieved if we take the intrinsic information into consideration and formulate them into the optimization of the deep network model. The intrinsic information in the images includes depth and edge information. For crack recognition, the edges reveal the most likely pixels that belong to cracks. Moreover, the drastic change in depth information can also indicate the change in the 3D geometric structures. Therefore, they all play an essential role in finding the intrinsic feature representations of cracks and can be well utilized to improve the generalization capacity of the learning-based deep neural network models.

5.2.2. Using more advanced 3D sensors

Intrinsically, the defects such as crack and spalling are structural damages. And the geometric patterns of them can be captured very easily by 3D sensors. Therefore, advanced sensors, such as the 3D industrial cameras, the advanced high-precision industrial LiDAR sensors, and the industrial laser scans should be incorporated to better enhance the 3D geometrical information, which is just complementary to the 2D visual information. Subsequently, to better enhance the performance in defects recognition, the fusion networks or mechanisms should be further developed to boost the performance by utilizing the complementary characteristics of multiple sensors.

5.2.3. Constructing high-quality database to boost the performance of SOTA Methods

The algorithm design and the database are complementary to each other. The highly effective algorithms and high-quality datasets can both boost the final defects recognition performance in a mutually beneficial way. According to our experiments, various SOTA networks have nearly equal performance in the task of crack semantic segmentation. From our experience, the issue that matters most in achieving highly accurate industrial defects recognition lies in two aspects regarding the constructed dataset. The first is the amount of the training data, and the second is the quality of the data. To be more specific, firstly, the amount of the training data should be sufficient enough to support various types of defects, such as the most typical infrastructural damages with varying patterns including crack and spalling. Also, the domain gap between the training set and the on-site captured test images of the infrastructures to be inspected should be as small as possible. Secondly, the quality of the training data should also be guaranteed, which means the geometric patterns of various defects are largely covered in the established dataset. When faced with real industrial applications, the quantity and the quality of the dataset should be evaluated carefully to guarantee robustness in inspections. When evaluating the performance, a high-quality large-scale dataset will also be beneficial to the robust and fair comparisons between diverse learning-based defects identification approaches.

5.2.4. Algorithms illustration and recommendation for weakly-supervised defect recognition without sufficient labeling for industrial applications

In real industrial applications, according to our experiments, it can be seen that the fully supervised defects classification, detection, and segmentation approaches have an upper bound in recognition accuracy, even with a fully labeled training set and no domain gap between the training and test set. Moreover, their actual performance depends more on the effectiveness of the learned model from limited labeled data. The detection accuracy may also experience a considerable drop when there is a large domain gap between the source labeled datasets and target unlabeled defects to be inspected. Therefore, this subsection discusses several promising weakly supervised algorithm approaches to alleviate the data hunger problem in defect recognition.

For **semantic segmentation**, many weakly or semi-supervised approaches have been proposed to reduce the demand for large-amount of annotated datasets, such as weakly supervised image segmentation methods with image-level labels [152]. Attention mechanism with a transformer-based network design can be used to extract the semantic affinity between various contextual objects, using the affinity from attention (AFA) module to refine and improve the quality of the pseudo labels. For the semi-supervised semantic segmentation, U2PL [153] has been proposed to make better use of the unreliable samples in the unlabeled data. Because a large amount of unlabeled data contains a great deal of

meaningful information in both low-level geometry and high-level semantics, the U2PL can make full use of the unlabeled data of low reliability as negative samples to boost the performance of the semantic segmentation models.

For **object detection**, the remarkable work Dense-Teacher [154] with a newly defined teacher-student model can be adopted to improve the performance of the single-stage object detector. The threshold-based object detection models also rely on NMS and therefore depend on accurate semantic segmentation results as shown in our experiments. However, choosing an inappropriate threshold will result in noisy pseudo labels. The teacher model gives a dense model of the whole feature map and proposes the quality focal loss to supervise the output of the student model. Using the mean-teacher scheme, the DTG-SSOD [155] can provide dense supervision for the teacher model with iterative NMS clustering and rank match strategies. Therefore, more abundant features and information on the unlabeled data are utilized.

In addition, a single model of multi-task learning can be utilized to handle the **object classification, semantic segmentation, and object detection** simultaneously for real applications. It has great potential to enhance real-time performance and construct memory-efficient learned models with real-time performance for multiple tasks. In 2D multi-task learning, classical works have formulated it as a multi-objective optimization problem and jointly optimized every target with network training [156]. The cracks and other structural defects can be regarded as 3D geometric changes, with point clouds captured by LiDAR sensors or RGB-D cameras. For multi-task learning based on 3D point clouds in a weakly supervised setting, the approach proposed in [157] can tackle the 3D scene understanding problem with limited labels, and can be integrated seamlessly with different neural network backbones to achieve 3D scene perception with multiple down-streams tasks. There is still considerable room for improving 2D/3D multi-task learning for defect recognition given their data characteristics and advantages.

6. Conclusions and outlook

This paper summarizes 40 publicly available defect datasets for deep learning-based classification, segmentation, and detection tasks. The architectures are suggested for the task of classification and semantic segmentation, while multiple deep learning-based models are trained, validated, and tested, and the performances are compared in a very detailed way. Critical remarks on the review and comparison results as well as future research directions are summarized as below.

6.1. Remarks on review and comparison results

Based on the comprehensive review and systematic comparison in this paper, major findings with deep learning-based defect inspection are presented as follows:

(1) The **quantity** of the summarized defect datasets: The volume of summarized publicly available datasets reaches around 13.38 M, with approximately 13.25 M, 0.061 M, 0.064 M for classification, segmentation, and detection respectively. The quantity of the dataset shrinks dramatically when the inspection task transfers from classification to high-level segmentation and detection, because segmentation and detection tasks require further annotation exhausting resources of the researchers. Considering the significant impact of the labor-intensive labeling process on productivity, SOTA labeling tools are summarized and compared in Subsection 3.4. The labeling process with a semi-automatic labeling tool is found to save about 33% of the time compared with a manual labeling tool. Furthermore, to alleviate data scarcity, the inspection research community is enriched with our self-established defect dataset, which contains more than 15,000 and 11,000 images for defect classification and semantic segmentation, respectively.

(2) The **diversity** of the summarized defect datasets: The dataset diversity lies in the defect type, infrastructure type, material type, and image context. The reviewed datasets cover more than 5 most common and vital defect types including crack, spalling, delamination, corrosion, and efflorescence, as well as more than 5 civil infrastructures including the pavement, bridge, building, tunnel, and dam. 5 material types including concrete, asphalt, steel, masonry, and wood are targeted with 3 image context levels (i.e., pixel, object and scene). The diversity in material types and image context level is essential since deep learning-based defect inspection algorithms depend highly on the diverse content and context features to generalize effectively.

(3) The **difficulty** for constructing a high-quality defect dataset: As to defect classification, 7 challenges get in the way of developing accurate and robust classification algorithms, including viewpoint variation, intra-class variation, difficulty of identifying fine-grained categories, background clutter, illumination changes, defect deformation, and occlusion. The established dataset needs to wittingly contain images with the aforementioned features to adapt the deep learning-based algorithms with higher accuracy and robustness. Regarding defect segmentation and detection, an additional main difficulty lies in annotating the defect image accurately and efficiently. Some attempts to utilize machine learning algorithms for automatic labeling is identified to get preliminary labeling results which can be then manually refined for accurate labeling in a much shorter time [119, 120, 121].

(4) The **feasibility** of the data collection platforms for defect inspection: Among the summarized 40 visual defect datasets, 10 of them are collected via cameras installed on ground vehicles, 6 of them are acquired by cameras on UAV platforms, 13 of them are obtained via hand-held cameras, and the rest are crawled from the internet. The ground vehicle is preferred as a data collection platform for pavement inspection due to its stability, accessibility, and long-duration ability. The UAV platform

is preferred as a feasible and cost-effective solution to conduct defect inspection of bridges and high-rise buildings. It is noteworthy that a valuable dataset collected by the UAV, named "Highway-crack dataset [59]", contains highway crack images taken just after a 6.4-level earthquake in China, revealing UAV's rapid response capability. Hand-held cameras are the most common data collection tools but their field of view (FOV) is limited, accompanied by image occlusion and perspective distortion, resulting in the incorrect recognition of defects and their geometric properties.

(5) The **scalability** for establishing a large-quantity defect dataset: Data augmentation methods, composed of basic image manipulations (e.g., kernel filters, geometric transformations, random erasing, and color space transformations) and deep learning approaches (based on adversarial learning, neural style transfer, and GAN) [158], are identified to efficiently expand the data volume of the defect dataset [88, 151].

(6) The **superiority** of our proposed algorithms: We have proposed the multi-branch self-attention module and multi-stage-fused attentional pyramid network architecture. As to crack classification, the multi-branch self-attention module is successfully integrated into the Swin Transformer [122] to get the adapted Swin Transformer-MB network, which achieves 88.0% accuracy slightly better than the original Swin Transformer with 87.7% accuracy and ranks in the second place out of 11 SOTA classification networks. For crack semantic segmentation, the multi-stage-fused attentional pyramid network architecture is successfully combined with SOTA segmentation networks such as DeepLab V3+ [144], DenseNet [147], and FRRNs [139]. The resulting models achieve satisfactory performances among 12 SOTA segmentation networks, with 77.8%, 77.6%, 76.9% mIoU respectively and an acceptable efficiency on the modern graphic processing unit.

(7) The **criticality** of algorithm comparison results: We have systematically compared 11 SOTA classification networks in terms of the accuracy and efficiency and 12 SOTA segmentation networks in terms of the widely-accepted accuracy metrics and efficiency. Based on the comparison results, suggestions are provided regarding the model deployment on robotic platforms and the development of semi-supervised algorithms for defect inspection. A good starting point is set up for the follow-up researchers and practitioners.

6.2. Outlook for automatic defect inspection

Following concluding marks, potential research topics are proposed as below for defect inspection:

(1) **Establish a multi-modal benchmark dataset**: A large-scale multi-modal dataset containing data collected from multiple sensors, such as optical cameras, IRT cameras, depth cameras, IE, GPR, ultrasonic sensors, and industrial LiDAR, will be conducive to defect localization and quantifying. One such dataset called "SDNET-2021 [30]" is identified for detecting the subsurface defects of the bridge decks and benchmarking advanced deep learning models. Advanced data fusion methods will be required to tackle the defect inspection more accurately with the established multi-modal dataset.

(2) **Standardize the summarized visual defect dataset**: The research community lacks a widely-accepted large-scale benchmark dataset for advancing and fairly comparing deep learning algorithms for visual defect inspection. Despite the systematically summarized 40 publicly available defect datasets, enormous efforts are still needed to standardize all the datasets into a unified benchmark defect dataset.

(3) **Develop datasets and algorithms for evaluating defect hazard level and predicting structure deterioration**: The core objective of the defect inspection is to quantify the hazard level of the defect. Except for the CCSSS [61] dataset, no publicly available visual dataset is found to evaluate defect hazard levels. Besides, the prediction of structure deterioration needs more research attention for the estimation of optimal rehabilitation measures [159, 160, 161].

(4) **Develop autonomous robotic platforms**: Most robotic platforms for defect data collection still rely on manual or remote control, requiring at least one operator to be exposed to uncomfortable and dangerous environments. An autonomous data collection platform [162, 163] can not only enhance the operation safety, but also accelerate the inspection process with improved objectivity and accuracy, providing a better reference for follow-up maintenance decisions and rehabilitation measures.

(5) **Develop automated defect inspection pipelines**: The community still lacks an integral defect inspection pipeline, which can automatically register the corresponding information (e.g., localization, quantity, hazard level, and tendency) of the defect to civil infrastructure management systems. Some attempts have been made in [164] and [165], where the 2D defect inspection results are mapped to reconstructed 3D model from LiDAR data and further registered to the building information modeling (BIM) system [165] or geographic information system (GIS) [166]. The digital twin (DT) system can also benefit the data storage and analysis process [167].

CRediT authorship contribution statement

Guidong Yang: Conceptualization, Investigation, Formal analysis, Writing - Original Draft. **Kangcheng Liu:** Conceptualization, Investigation, Methodology, Formal analysis, Writing - Original Draft. **Jihan Zhang:** Investigation, Formal analysis, Writing - Original Draft. **Benyun Zhao:** Investigation, Formal analysis, Writing - Original Draft. **Zuoquan Zhao:** Investigation, Formal analysis, Writing -

Original Draft. **Xi Chen:** Conceptualization, Resources, Supervision, Writing - Review & Editing, Project administration. **Ben M. Chen:** Conceptualization, Funding acquisition, Resources, Supervision, Writing - Review & Editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reproted in the paper.

Acknowledgements

This project is supported in part by the Research Grants Council of Hong Kong SAR (Grant No: 14209020 and Grant No: 14206821) and in part by the Hong Kong Centre for Logistics Robotics (HKCLR).

References

[1] Wai-Kiong Chong and Sui-Pheng Low. Assessment of defects at construction and occupancy stages. *Journal of Performance of Constructed facilities*, 19(4):283–289, 2005.

[2] Wai-Kiong Chong and Sui-Pheng Low. Latent building defects: causes and design strategies to prevent them. *Journal of performance of constructed facilities*, 20(3):213–221, 2006.

[3] Yuqing Gao and Khalid M Mosalam. PEER Hub ImageNet: A large-scale multiattribute benchmark data set of structural images. *Journal of Structural Engineering*, 146(10):04020198, 2020.

[4] Yasser El Masri and Tarek Rakha. A scoping review of non-destructive testing (NDT) techniques in building performance diagnostic inspections. *Construction and Building Materials*, 265:120542, 2020.

[5] Mahesh Yumnam, Hina Gupta, Debdutta Ghosh, and Jayaprakash Jaganathan. Inspection of concrete structures externally reinforced with FRP composites using active infrared thermography: A review. *Construction and Building Materials*, 310:125265, 2021.

[6] Ali Akbar Shirzadi Javid, Parviz Ghoddousi, Gholamreza Ghodrati Amiri, and Khalil Donyadideh. A new photogrammetry method to study the relationship between thixotropy and bond strength of multi-layers casting of self-consolidating concrete. *Construction and Building Materials*, 204:530–540, 2019.

[7] Junzhi Zhang, Jin Huang, Chuanqing Fu, Le Huang, and Hailong Ye. Characterization of steel reinforcement corrosion in concrete using 3D laser scanning techniques. *Construction and Building Materials*, 270:121402, 2021.

[8] Wei Jiang, Youjun Xie, Jianxian Wu, and Guangcheng Long. Influence of age on the detection of defects at the bonding interface in the CRTS III slab ballastless track structure via the impact-echo method. *Construction and Building Materials*, 265:120787, 2020.

[9] Mezgeen Rasol, Jorge C. Pais, Vega Pérez-Gracia, Mercedes Solla, Francisco M. Fernandes, Simona Fontul, David Ayala-Cabrera, Franziska Schmidt, and Hossein Assadollahi. GPR monitoring for road transport infrastructure: A systematic review and machine learning insights. *Construction and Building Materials*, 324:126686, 2022.

[10] Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep concrete inspection using unmanned aerial vehicle towards CSSC database. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 24–28, 2017.

[11] Devdatt P Purohit, NA Siddiqui, Abhishek Nandan, and Bikarama P Yadav. Hazard identification and risk assessment in construction industry. *International Journal of Applied Engineering Research*, 13(10):7639–7667, 2018.

[12] Mateusz Żarski, Bartosz Wójcik, and Jarosław Adam Miszczak. KrakN: Transfer learning framework for thin crack detection in infrastructure maintenance. *arXiv preprint arXiv:2004.12337*, 2020.

[13] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst-Michael Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *2017 international joint conference on neural networks (IJCNN)*, pages 2039–2047. IEEE, 2017.

[14] Ronny Stricker, Markus Eisenbach, Maximilian Sesselmann, Klaus Debes, and Horst-Michael Gross. Improving visual road condition assessment by extensive experiments on the extended gaps dataset. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[15] Ronny Stricker, Dustin Aganian, Maximilian Sesselmann, Daniel Seichter, Marius Engelhardt, Roland Spielhofer, Matthias Hahn, Astrid Hautz, Klaus Debes, and Horst-Michael Gross. Road surface segmentation-pixel-perfect distress and object detection for road assessment. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1789–1796. IEEE, 2021.

[16] Sattar Dorafshan, Robert J. Thomas, and Marc Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186:1031–1045, 2018.

[17] Ruoxu Ren, Terence Hung, and Kay Chen Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, 2018.

[18] Jun Kang Chow, Kuan fu Liu, Pin Siang Tan, Zhaoyu Su, Jimmy Wu, Zhaofeng Li, and Yu-Hsing Wang. Automated defect inspection of concrete structures. *Automation in Construction*, 132:103959, 2021.

[19] Qiuchen Zhu and Quang Ha. A bidirectional self-rectifying network with bayesian modelling for vision-based crack detection. *IEEE Transactions on Industrial Informatics*, 2022.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[21] Yuzhi Zhao, Lai-Man Po, Tingyu Lin, Xuehui Wang, Kangcheng Liu, Yujia Zhang, Wing-Yin Yu, Pengfei Xian, and Jingjing Xiong. Legacy photo editing with learned noise prior. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2103–2112, 2021.

[22] Kangcheng Liu, Yuzhi Zhao, Zhi Gao, and Ben M Chen. WeakLabel3D-Net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[23] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Conv: Large-scale lidar point clouds understanding leveraging feature correlation mining and geometric-aware modeling. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12896–12902. IEEE, 2021.

[24] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Net: A fast and accurate framework for large-scale lidar point cloud understanding. *IEEE Transactions on Cybernetics*, 2022.

[25] Kexin Guo, Zhirong Qiu, Cunxiao Miao, Abdul Hanif Zaini, Chun-Lin Chen, Wei Meng, and Lihua Xie. Ultra-wideband-based localization for quadcopter navigation. *Unmanned Systems*, 04(01):23–34, 2016.

[26] Murad Al Qurishee, Weidong Wu, Babatunde Atolagbe, Joseph Owino, Ignatius Fomunung, and Mbakisya Onyango. Creating a dataset to boost civil engineering deep learning research and application. *Engineering*, 12(3):151–165, 2020.

[27] Narges Kheradmandi and Vida Mehranfar. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, 321:126162, 2022.

[28] Sandra Pozzer, Ehsan Rezazadeh Azar, Francisco Dalla Rosa, and Zacarias Martin Chamberlain Pravia. Semantic segmentation of

defects in infrared thermographic images of highly damaged concrete structures. *Journal of Performance of Constructed Facilities*, 35(1):04020131, 2021.

[29] Sattar Dorafshan and Hoda Azari. Deep learning models for bridge deck evaluation using impact echo. *Construction and Building Materials*, 263:120109, 2020.

[30] Eberichi Ichi and Sattar Dorafshan. SDNET2021: Annotated NDE dataset for Structural Defects. 2021.

[31] Kexin Guo, Zhirong Qiu, Cunxiao Miao, Abdul Hanif Zaini, Chun-Lin Chen, Wei Meng, and Lihua Xie. Ultra-wideband-based localization for quadcopter navigation. *Unmanned Systems*, 04(01):23–34, 2016.

[32] P. Huethwohl. Cambridge bridge inspection dataset. *Online*, 2017.

[33] Hongyan Xu, Xiu Su, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. Automatic bridge crack detection using a convolutional neural network. *Applied Sciences*, 9(14):2867, 2019.

[34] Philipp Hüthwohl, Ruodan Lu, and Ioannis Brilakis. Multi-classifier for reinforced concrete bridge defects. *Automation in Construction*, 105:102824, 2019.

[35] Mingpeng Li. Concrete-crack-detection dataset. 2020.

[36] C. F. Özgenel and Arzu Gönenç Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 693–700, July 2018.

[37] Sattar Dorafshan, Robert J Thomas, and Marc Maguire. Sdnet2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in brief*, 21:1664–1668, 2018.

[38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[39] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[40] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[41] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[42] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[49] Yun Wang, Ju Zhang, Jing Liu, Yin Zhang, Zhi Chen, Chun Li, Kai He, and Rui Yan. Research on crack detection algorithm of the concrete bridge based on image processing. *Procedia Computer Science*, 154:610–616, 01 2019.

[50] Sylvie Chambon and Jean-Marc Moliard. Automatic road pavement assessment with image processing: review and comparison. *International Journal of Geophysics*, 2011, 2011.

[51] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[52] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016.

[53] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[54] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

[55] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.

[56] Qipei Mei, Mustafa Gül, and Md Riasat Azim. Densely connected deep neural network considering connectivity of pixels for automatic crack detection. *Automation in Construction*, 110:103018, 2020.

[57] Qipei Mei and Mustafa Gül. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Construction and Building Materials*, 256:119397, 2020.

[58] Qipei Mei, Mustafa Gül, and Nima Shirzad-Ghaleroudkhani. Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles. *Journal of Civil Structural Health Monitoring*, 10:653–665, 2020.

[59] Zhonghua Hong, Fan Yang, Haiyan Pan, Ruyan Zhou, Yun Zhang, Yanling Han, Jing Wang, Shuhu Yang, Peng Chen, Xiaohua Tong, et al. Highway crack segmentation from unmanned aerial vehicle images using deep learning. *IEEE Geoscience and Remote Sensing Letters*, 2021.

[60] Xiao-Wei Ye, T Jin, ZX Li, SY Ma, Y Ding, and YH Ou. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *Journal of Structural Engineering*, 147(11):04721008, 2021.

[61] Eric Bianchi and Matthew Hebdon. Corrosion Condition State Semantic Segmentation Dataset. 12 2021.

[62] Eric Bianchi and Matthew Hebdon. Labeled Cracks in the Wild (LCW) Dataset. 10 2021.

[63] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[64] Y Bai, Bing Zha, Halil Sezen, and Alper Yilmaz. Deep cascaded neural networks for automatic detection of structural damage and cracks from images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:411–417, 2020.

[65] Dimitris Dais, Ihsan Engin Bal, Eleni Smyrou, and Vasilis Sarhosis. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, 125:103606, 2021.

[66] Yupeng Ren, Jisheng Huang, Zhiyou Hong, Wei Lu, Jun Yin, Lejun Zou, and Xiaohua Shen. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234:117367, 2020.

[67] Christian Benz, Paul Debus, Huy Khanh Ha, and Volker Rodehorst. Crack segmentation on UAS-based imagery using transfer learning. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6, 2019.

[68] khanhha. Crack segmentation. https://github.com/khanhha/crack_segmentation, 2020.

[69] Yongsheng Bai, Halil Sezen, and Alper Yilmaz. Detecting cracks and spalling automatically in extreme events by end-to-end deep learning frameworks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:161–168, 2021.

[70] Eric Bianchi and Matthew Hebdon. Concrete Crack Conglomerate Dataset. 10 2021.

[71] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[72] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[73] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[75] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[76] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[77] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[78] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[79] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[80] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[81] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[82] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[83] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6688–6697, 2019.

[84] Yongsheng Bai, Halil Sezen, and Alper Yilmaz. End-to-end deep learning methods for automated damage detection in extreme events at various scales. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6640–6647. IEEE, 2021.

[85] Bo Wang. Aerialcrackdataset: Towards object detection with dataset. https://github.com/arasharchor/AerialCrackDetection_Keras, 2017.

[86] Hui Li and Billie F. Spencer Jr. the first International Project Competition for Structural Health Monitoring. *Journal of Structural Engineering*, 2020.

[87] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, and Hiroshi Omata. Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12):1127–1141, 2018.

[88] Hiroya Maeda, Takehiro Kashiyama, Yoshihide Sekimoto, Toshikazu Seto, and Hiroshi Omata. Generative adversarial network for road damage detection. *Computer-Aided Civil and Infrastructure Engineering*, 36(1):47–60, 2021.

[89] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto. RDD2020: An annotated image dataset for automatic road damage detection using deep learning. *Data in brief*, 36:107133, 2021.

[90] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Alexander Mraz, Takehiro Kashiyama, and Yoshihide Sekimoto. Transfer learning-based road damage detection for multiple countries. *CoRR*, abs/2008.13101, 2020.

[91] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto. Global road damage detection: State-of-the-art solutions. *CoRR*, abs/2011.08740, 2020.

[92] Hamed Majidifard, Peng Jin, Yaw Adu-Gyamfi, and William G. Buttlar. Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. *Transportation Research Record: Journal of the Transportation Research Board*, 2674(2):328–339, Feb 2020.

[93] Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11196–11205, 2019.

[94] Xiaoming Lv, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6), 2020.

[95] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[96] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[97] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[98] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, page 313–318, New York, NY, USA, 2003. Association for Computing Machinery.

[99] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[100] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[101] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[102] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016.

[103] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[104] Donald Kossmann and Bing Liu. Road damage detection and classification challenges. https://cci.drexel.edu/bigdata/bigdata2018/BigDataCupChallenges.html, 2018.

[105] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto. Global road damage detection: State-of-the-art solutions. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5533–5539. IEEE, 2020.

[106] Kechen Song and Yunhui Yan. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285:858–864, 2013.

[107] Ybat. Ybat. https://github.com/drainingsun/ybat.

[108] LabelImg. Labelimg. https://github.com/tzutalin/labelImg.

[109] LabelMe. Labelme. https://github.com/wkentaro/labelme.

[110] VIA. Via. https://www.robots.ox.ac.uk/~vgg/software/via.

[111] VoTT. Vott. https://github.com/microsoft/VoTT#build-and-run-from-source.

[112] PixelAnnotationTool. Pixelannotationtool. https://github.com/abreheret/PixelAnnotationTool.

[113] CVAT. Cvat. https://github.com/openvinotoolkit/cvat.

[114] RectLabel. Rectlabel. https://rectlabel.com/.

[115] Labelbox. Labelbox. https://labelbox.com/product/platform/annotate.

[116] V7 Darwin. V7 darwin. https://www.v7labs.com/.

[117] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[118] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[119] Yizheng Chen, Jia Liang, Xingyu Gu, Qipeng Zhang, Hanyu Deng, and Shuwei Li. An improved minimal path selection approach with new strategies for pavement crack segmentation. *Measurement*, 184:109877, 2021.

[120] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[121] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3060–3069, January 2021.

[122] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *International Conference on Computer Vision (ICCV)*, 2021.

[123] Mitchell J Hallee, Rebecca K Napolitano, Wesley F Reinhart, and Branko Glisic. Crack detection in images of masonry using cnns. *Sensors*, 21(14):4929, 2021.

[124] Dimitrios Loverdos and Vasilis Sarhosis. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*, 140:104389, 2022.

[125] Amir Rezaie, Radhakrishna Achanta, Michele Godio, and Katrin Beyer. Comparison of crack segmentation using digital image correlation measurements and deep learning. *Construction and Building Materials*, 261:120474, 2020.

[126] Eric Bianchi. *COCO-Bridge: Common Objects in Context Dataset and Benchmark for Structural Detail Detection of Bridges*. PhD thesis, Virginia Tech, 2019.

[127] Kangcheng Liu, Yanbin Qu, Hak-Man Kim, and Huihui Song. Avoiding frequency second dip in power unreserved control during wind power rotational speed recovery. *IEEE transactions on power systems*, 33(3):3097–3106, 2017.

[128] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[129] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[130] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[131] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[133] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[134] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[135] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6848–6856, 2018.

[136] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[137] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[138] Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3):1498–1512, 2018.

[139] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017.

[140] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[141] Kangcheng Liu, Xiaodong Han, and Ben M Chen. Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 381–387. IEEE, 2019.

[142] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[143] Weiwei Sun and Ruisheng Wang. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm. *IEEE Geoscience and Remote Sensing Letters*, 15(3):474–478, 2018.

[144] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[145] Zhenyu Zhang, Shouwei Gao, and Zheng Huang. An automatic glioma segmentation system using a multilevel attention pyramid scene parsing network. *Current Medical Imaging*, 17(6):751–761, 2021.

[146] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable

convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[147] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[148] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[149] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Net: Fast large-scale lidar point cloudsunderstanding network leveraging correlatedfeature mining and geometric-aware modelling. *arXiv preprint arXiv:2012.09439*, 2020.

[150] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[151] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2524–2534, 2021.

[152] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16846–16855, 2022.

[153] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4248–4257, 2022.

[154] Hongyu Zhou, Zheng Ge, Songtao Liu, Weixin Mao, Zeming Li, Haiyan Yu, and Jian Sun. Dense teacher: Dense pseudo-labels for semi-supervised object detection. *arXiv preprint arXiv:2207.02541*, 2022.

[155] Gang Li, Xiang Li, Yujie Wang, Yichao Wu, Ding Liang, and Shanshan Zhang. DTG-SSOD: Dense teacher guidance for semi-supervised object detection. *arXiv preprint arXiv:2207.05536*, 2022.

[156] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[157] Kangcheng Liu, Yuzhi Zhao, Zhi Gao, and Ben M Chen. WeakLabel3D-Net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5108–5115. IEEE, 2022.

[158] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[159] Monica Santamaria Ariza, Ivan Zambon, Hélder S. Sousa, Jose Antonio Campos e Matos, and Alfred Strauss. Comparison of forecasting models to predict concrete bridge decks performance. *Structural Concrete*, 21(4):1240–1253, 2020.

[160] Shouyan Jiang, Linxin Zhao, and Chengbin Du. Structural deformation prediction model based on extreme learning machine algorithm and particle swarm optimization. *Structural Health Monitoring*, page 14759217211072237, 2022.

[161] Mohammed Alsharqawi, Tarek Zayed, and Saleh Abu Dabous. Integrated condition rating and forecasting method for bridge decks using visual inspection and ground penetrating radar. *Automation in Construction*, 89:135–145, 2018.

[162] Elisabeth Menendez, Juan G Victores, Roberto Montero, Santiago Martínez, and Carlos Balaguer. Tunnel structural inspection and assessment using an autonomous robotic system. *Automation in Construction*, 87:117–126, 2018.

[163] Jacob J Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting. *Journal of Computing in Civil Engineering*, 35(2):04020064, 2021.

[164] Jacob J Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting. *Journal of Computing in Civil Engineering*, 35(2):04020064, 2021.

[165] Jun Kang Chow, Kuan-fu Liu, Pin Siang Tan, Zhaoyu Su, Jimmy Wu, Zhaofeng Li, and Yu-Hsing Wang. Automated defect inspection of concrete structures. *Automation in Construction*, 132:103959, 2021.

[166] Kaiwen Chen, Georg Reichard, Abiola Akanmu, and Xin Xu. Geo-registering UAV-captured close-range images to GIS-based spatial model for building façade inspections. *Automation in Construction*, 122:103503, 2021.

[167] Jihan Zhang, Ruoyu Wang, Guidong Yang, Kangcheng Liu, Chuanxiang Gao, Yu Zhai, Xi Chen, and Ben M. Chen. Sim-in-real: Digital twin based uav inspection process,. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 784–801. IEEE, 2022.

Highlights

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

Guidong Yang,Kangcheng Liu,Jihan Zhang,Benyun Zhao,Zuoquan Zhao,Xi Chen,Ben M. Chen

- Review of the datasets for deep learning-based visual defect inspection
- Comparison of the algorithms for defect classification, segmentation and detection
- Proposed deep learning-based network architectures for defect inspection
- Suggestions on developing defect datasets and defect inspection algorithms

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

Guidong Yang[1], Kangcheng Liu[*,1], Jihan Zhang, Benyun Zhao, Zuoquan Zhao, Xi Chen[*] and Ben M. Chen

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China

ARTICLE INFO

Keywords:
defect datasets
infrastructure defect inspection
classfication
segmentation and detection
learning-based approaches

ABSTRACT

Deep learning breakthrough stimulates new research trends in civil infrastructure inspection, whereas the lack of quality-guaranteed, human-annotated, free-of-charge, and publicly available defect datasets with sufficient amounts of data hinders the progress of deep learning in defect inspection. To boost research in deep learning-based visual defect inspection, this paper first reviews and summarizes 40 publicly available defect datasets, covering common defects in various types of buildings and infrastructures. The taxonomy of the datasets is proposed based on specific deep learning objectives (classification, segmentation, and detection). Clarifications are also made for each dataset regarding its corresponding data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, algorithms adopted for validation, annotation levels, context levels, and publication license for future utilization. Consequently, the summarized defect datasets offer around $13.38M$ labeled images, cover more than 5 defect types, 5 infrastructure types, 5 material types, and 3 levels of image context. Given that the crack is a common interest in civil engineering, this paper further combines existing datasets with self-labeled crack images to establish a benchmark dataset providing more than 15,000 and 11,000 labeled images for crack classification and segmentation, respectively. Based on the established crack dataset, experiments are conducted for classification, segmentation, and the subsequent non-maximum suppression-based detection tasks. The proposed *multi-branch self-attention module* and *multi-stage-fused attentional pyramid network* have been successfully adapted into the state-of-the-art (SOTA) classification network-*Swin Transformer* and segmentation networks including *DeepLab V3+*, *DenseNet*, and *Full Resolution ResNet*. The resulting classification network achieves 88.0% accuracy, and the adapted segmentation models reach 77.8%, 77.6%, 76.9% mIoU (mean Intersection over Union), respectively. Moreover, a comprehensive comparison between 11 SOTA classification algorithms and 12 SOTA segmentation algorithms has been conducted. The algorithms proposed in this work are shown to achieve satisfactory performance with an acceptable efficiency on modern graphic processing units. Detailed suggestions are provided for constructing high-quality datasets and inspection algorithms. Finally, this paper remarks on the quantity, diversity, difficulty, and scalability of the reviewed defect datasets, feasibility on robotic platforms, superiority of proposed algorithms, and criticality of algorithm comparison results, formulating a solid baseline for future defect inspection research.

1. Introduction

Civil infrastructures such as pavements, bridges, buildings, tunnels, and dams suffer from performance degradation caused by structure deterioration, external loads, weather impact, poor workmanship, poor design, and natural disasters [1, 2, 3]. Periodical defect inspection is a necessary and pivotal measure to ensure the energy efficiency and the functional safety of civil structures. The subsequent rehabilitation measures can be then carried out according to the inspection results. Periodical defect inspection is often conducted through Non-Destructive Testing (NDT), which can avoid the physical damage caused by the traditional sample collection process [4]. NDT techniques include infrared thermography (IRT) [5], photogrammetry [6], laser scanning [7], impact echos (IE) [8], and ground-penetrating radars (GPR) [9]. Currently, periodical manual defect inspection is predominant in infrastructure maintenance, where inspectors make use of NDT devices to evaluate structural health [10]. However, inspectors may be exposed to complex site environment with potential health hazards and safety risks [11]. Furthermore, such subjective inspection can be error-prone [12], labor-intensive [13], and time-consuming [14], not conducive to the subsequent rehabilitation [15]. For example, traditional methods make wrong predictions easily under not well-controlled illuminations, and cost more manpower to accomplish the inspection task. Manual inspections often span several weeks to months, resulting in outdated evaluation at the time of rehabilitation.

Due to the aforementioned limitations, more and more researchers tend to incorporate machine learning and deep learning algorithms into automatic defect inspection solutions. Especially in recent years, deep learning has become the main stream solution due to its unprecedented breakthrough. Deep learning-based solutions are evolving to automate the defect inspection efficiently [16, 17, 18, 19].

*Corresponding author
kcliu@mae.cuhk.edu.hk (Kangcheng Liu); xichen002@cuhk.edu.hk (Xi Chen)
ORCID(s): 0000-0003-2168-9057 (Xi Chen)
[1]Both authors contributed equally to this work

Nevertheless, compared to the successful application of deep learning in natural language processing, facial recognition [20], image processing [21, 22], and 3D vision for autonomous aerial and ground vehicles [23, 24, 25], research in deep learning-based defect inspection is still restricted. The most critical reason is the lack of quality-guaranteed, human-annotated, free-of-charge, and publicly available defect datasets, which are beneficial to training highly accurate neural networks for defect inspection based on supervised learning [26]. Although there exist reviews focusing on NDT devices for building inspection [4] and segmentation algorithms for pavement crack detection [27], they neither provide a comprehensive review of the datasets spanning different infrastructure and defect types nor a systematic comparison of deep learning algorithms for visual inspection. Thus, it is essential and meaningful to make a comprehensive review and systematic comparison of existing **publicly available** datasets and algorithms to boost deep learning-based defect inspection. To the authors' best knowledge, this paper is the first comprehensive review of **publicly available** civil infrastructure inspection datasets, and the first that provides a systematic review and comparison of **publicly available** state-of-the-art (SOTA) algorithms for surface defect inspection.

Motivated by the aforementioned difficulties, this paper intends to promote research in deep learning-based defect inspection by conducting a comprehensive review on existing publicly available defect datasets with a systematic comparison between SOTA algorithms for the task of classification, segmentation, and detection on a constructed crack dataset. The major contributions of this paper are as follows:

- A comprehensive review of the existing publicly available datasets for deep learning-based visual defect inspection.
- A systematic comparison of the SOTA algorithms for defect classification, segmentation and detection, with crack as the typical research interest for a case study.
- Proposed deep learning-based network architectures based on the adaptations to SOTA algorithms for crack classification, segmentation and subsequent detection with non-maximum suppression.
- Suggestions on developing high-quality defect datasets and defect inspection algorithms.

The remainder of this paper is organized as follows. Section 2 is the literature review methodology. Section 3 shows the review results of the datasets and corresponding methods. The self-established crack dataset, results of the comparison between our methods and SOTAs for crack classification, segmentation, and detection are presented and discussed in Section 4. Based on the review and comparison, Section 5 points out existing barriers to building a high-quality and large-scale defect dataset and offers corresponding suggestions. Also, the systematical suggestions on methodology to conduct highly-effective defect recognition are provided. Conclusions and future work are presented in Section 6 to form a comprehensive baseline for studies on civil infrastructure defect inspection.

2. Literature review methodology

A comprehensive review of the literature related to **publicly available** datasets for deep learning-based visual defect inspection was conducted using Google Scholar. Based on keywords searching, a considerable amount of literature most relevant to the research interest was acquired. The literature was filtrated according to the following procedures: (1) Title, abstract, and conclusion screening; (2) Dataset public availability checking; (3) Full-text screening to extract critical features of the datasets. Specifically, the following features of the defect dataset were selected and summarized, they are: *data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, annotation levels, context levels, publication license, algorithms adopted for validation, algorithm training strategies*, and *data augmentation methods*. These critical features are of the utmost concern when developing deep learning-based solutions for defect inspection. The main focus of the review is on visual inspection datasets, i.e., datasets with optical images supplemented with IRT images. Datasets with data from other NDT devices (see e.g., IE and GPR) are beyond the scope of this review.

3. Review results on datasets and corresponding methods

Based on the above literature review on publicly available defect datasets with optical images supplemented with IRT images. Altogether 40 defect datasets are summarized, illustrated, and demonstrated. Figure 1 shows the taxonomy of summarized defect datasets based on different aspects. For each dataset, its corresponding data volume, data resolution, data source, defect categories covered, infrastructure types focused, material types targeted, algorithms for validation, annotation levels, image context levels, and publication license are clarified. In this paper, the taxonomy of these datasets is elaborated as per specific deep learning objectives (annotation levels). The datasets are grouped into classification-oriented, detection-oriented, and segmentation-oriented, with patch-level, bounding-box-level, and pixel-level annotation respectively.

As demonstrated in Figure 1, the summarized defect datasets cover various types of infrastructure such as pavements, bridges, buildings, tunnels, and dams with different materials such as concrete, asphalt, steel, masonry, and wood. These datasets cover the most common defect types: crack, spalling, delamination, corrosion, and efflorescence. As to data types, most datasets utilize optical images (in terms of grey-scale and color images), with IRT images [26, 28], IE signals [29, 30], and GPR signals [30] as alternatives. Optical images are typically used to detect surface defects of the structure, while IRT images, IE signals,

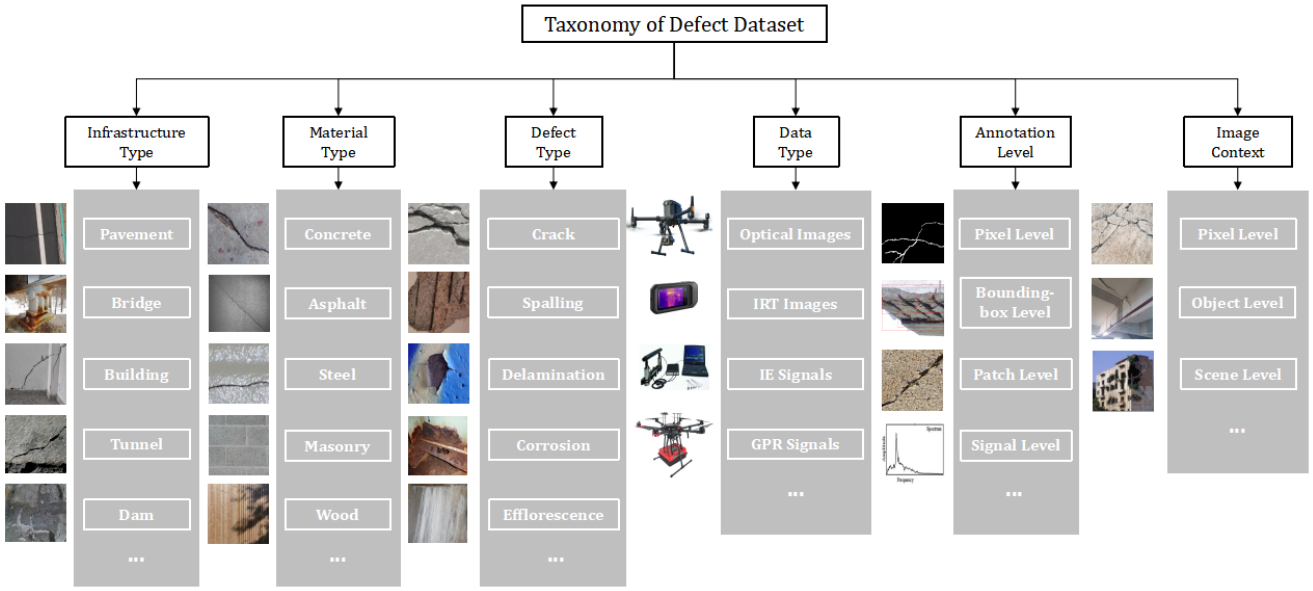

Figure 1: Taxonomy of defect datasets.

Table 1
A summary of publicly available **classification-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data Source / Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
GAPs-v1 [13]	2017	6.3 M	64 × 64	Cameras on ground vehicle	a. Crack b. Pothole c. Inlaid patch d. Applied patch e. Open joint	Pavement	Asphalt	Patch Level	Pixel Level	Private License, for Academic Use Only
GAPs-v2 [14]	2019	6.7 M	64 × 64 to 256 × 256	Cameras on ground vehicle	Same as GAPs-v1	Pavement	Asphalt	Patch Level	Pixel Level	Private License, for Academic Use Only
CBID [32]	2017	1028	229 × 229	Not clarified	a. Crack b. Water seepage c. Spalling, etc.	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Xu [33]	2019	6069	224 × 224	Camera on UAV	Crack	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Philipp [34]	2019	3607	Multiple	Hand-held camera	a. Crack, Efflorescence b. General defects (e.g. graffiti, moss, etc) c. Scaling, spalling d. Exposed reinforcement, Rust staining	Bridge	Concrete and Steel	Patch Level	Pixel Level	CC BY 4.0 License
KrakN [12]	2020	16114	224 × 224	Hand-held cameras	Thin crack(< 0.2mm)	Bridge	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
DCTCD [35]	2021	250	512 × 512	Cameras on UAV	Thin crack (>= 0.1mm)	Bridge	Concrete	Patch Level	Pixel Level	GNU General Public License v3.0
CCIC [36]	2018	40000	227 × 227	Hand-held camera	Crack	Building	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
$\phi - Net$ [3]	2020	36413	448 × 448	Crawled from internet Hand-held cameras	Spalling, etc.	Building	Concrete, Steel, Masonry, and Wood	Patch Level	Pixel Level Object & Scene Level	CC BY-NC-SA 4.0 License
CSSC [10]	2017	45024 44263	130 × 130 100 × 100	Crawled from internet	Crack and Spalling	Bridge and Building	Concrete	Patch Level	Pixel Level	Not Clarified
SDNET2018 [37]	2018	56092	256 × 256	Hand-held camera	Crack	Bridge, Building and Pavement	Concrete	Patch Level	Pixel Level	CC BY 4.0 License
Qurishee [26]	2020	2088	4032 × 3024 and 5312 × 2988	Not clarified	Crack	Not clarified	Concrete	Patch Level	Pixel Level	CC BY 4.0 License

and GPR signals can reveal subsurface defects. Besides, these datasets vary in the level of image context information, i.e., the pixel level, object level, and scene level. The data contained in different datasets are collected via hand-held sensors, robotic platforms, or UAV platforms. In particular, compared to the hand-held cameras and wall-climbing robots, the UAV platform combined with visual-inertial odometry offers a feasible solution for defect data collection and localization in the GPS-denied environment, e.g., defect inspection under the bridge [10, 31]. Subsection 3.1-3.3 illustrate classification, segmentation, and detection-oriented datasets with optical images supplemented with IRT images respectively. Within each subsection, the defect datasets are further grouped based on the type of targeted civil infrastructure. Subsection 3.4 describes the status and trend of data collection and labeling procedures.

3.1. Classification-oriented datasets

In this subsection, each dataset described is labeled either at the image level or at the patch level (if multiple image patches are cropped from the raw image) to conduct multi-class classification between different defect categories or binary classification of a particular defect between defect and non-defect categories. It should be noted that the classification datasets can also be used to detect defects based on the sliding window technique, e.g., Histogram of Oriented Gradients (HOG) Detector [38], Deformable Part-based Model (DPM) [39], and Overfeat detector [40]. In general, the sliding window technique is to slide a window to go through all possible locations and scales in the image and further classify each image patch bounded by the window to check whether the image patch contains the target object (the defect in our case) or not [41]. In this manner, the detection problem can be converted to a classification problem, and the defect in the original image can be detected. Table 1 shows the summary of publicly available classification-oriented defect datasets. Each dataset's corresponding data volume, data resolution, data source, defects categories covered, infrastructure types focused, material types targeted, annotation level, and image context level are clarified. These datasets are firstly sorted according to the corresponding infrastructure type and then sorted in chronological order. Table 2

shows the corresponding algorithms used for validating the datasets. For each dataset, its network structures and training strategies are listed. Figure 2 shows exemplary images for each classification-oriented dataset.

3.1.1. Pavements

The German Asphalt Pavement Distress (GAPs) datasets have three versions, i.e. GAPs-v1 [13], GAPs-v2 [14], and GAPs-10m [15]. GAPs-v1 [13] dataset is the first standardized, quality-controlled, patch-level annotated, free of charge, and publicly available dataset with a decent size enough to train neural networks for asphalt pavement distress classification. The data collection procedure strictly follows the regulations developed by the German Road and Transportation Research Association (FGSV). The images are downward-facing road images collected by a surface camera system composed of two photogrammetrically calibrated cameras. The GAPs-v1 dataset contains 1,969 grey-scale images (8-bit) comprising 1,418 images for training, 51 images for validation, and 500 images for testing. Each resulting image has a resolution of 1920×1080, with a pixel resolution of 1.2 mm $\times$ 1.2 mm. Each high-resolution image is annotated to impose 64×64 bounding boxes enclosing pavement distress (defined by FGSV), which covers cracks, potholes, inlaid patches, applied patches, and open joints. Each image is further sliced into multiple 64×64 image patches. Thus, the dataset has 4.9 M patches for training, 200 k patches for validation, and 1.2 M patches for testing. Cracks are the dominant distress class in the GAPs-v1 dataset. Various crack types are included: single or multiple cracking, longitudinal or transversal cracking, alligator cracking, and sealed cracks. The GAPs-v1 dataset is dedicated to the binary classification of pavement distress. All of the aforementioned damage classes are labeled as 'Distress', while intact road patches are labeled as 'Normal'.

GAPs-v2 [14] is an improvement on the GAPs-v1 dataset, it provides more data, refined annotations, and more context compared to GAPs-v1. Five hundred additional images with a size of 1920×1080 are collected following the regulations developed by FGSV. Altogether 2,468 grey-scale images (8 bit) are further divided into a training set (1,417 images), a validation set (51 images), a validation-test set (500 images), and a test set (500 images). Based on these images, 692,377 and 6,035,404 image patches are extracted for road distress and intact road, respectively, to form the entire dataset. The respective proportion of intact roads, cracks, applied patches, inlaid patches, potholes, and open joints in the full dataset are 89.71%, 7.28%, 1.72%, 0.75%, 0.30%, and 0.24%. GAPs-v2 also refines annotations by providing a smaller bounding box for non-damage space and solving conflicting annotations. Moreover, GAPs-v2 offers multiple patch sizes (64×64 to 256×256) with more image context since different image patch sizes will influence the trade-off between damage detection quality and inference speed of the neural network [14]. In addition to the above refinements, GAPs-v2 contains a CIFAR-like [42] or MNIST-like [43] subset consisting of 50,000 patches for training and 10,000 patches for validation, validation-test, and test. The subset's proportion of intact road, cracks, applied patches, inlaid patches, potholes, and open joints are 60%, 20%, 10%, 5%, 3% and 2% respectively. The publicly available GAPs-v2 dataset is still dedicated to the binary classification of pavement distress, i.e., 'Distress' or 'Normal'. GAPs-10m [15] dataset provides pixel-level annotation for pavement distress segmentation. This dataset will be illustrated in the Subsection 3.2 of this survey.

3.1.2. Bridges

Cambridge Bridge Inspection Dataset (CBID) [32] is a dataset for evaluating the classification performance of different bridge defects. The dataset contains 1,028 image patches with a resolution of 229×229. The dataset is further partitioned into two subsets containing bridge patches with (337 patches) and without (691 patches) defects. However, the dataset doesn't explicitly illustrate the data collection procedure and defects classes contained in the dataset.

Xu *et al.* [33] build up a dataset for binary classification of concrete bridge crack. The original dataset [49] contains 2068 crack images collected by a UAV equipped with a camera that has a resolution of 1024×1024. To improve the classification robustness of the network, crack images with bridge shadings, strong light, and water stains are wittingly included in the dataset. Each image in the original dataset is further cropped into multiple 512×512 image patches. After filtering blurred patches, a new dataset containing 6,069 patches is obtained. The acquired dataset comprises 4,058 crack images and 2,011 background images. The number of patches for the training and validation sets is 4,856 and 1,213, respectively. Afterward, Xu *et al.* [33] further crop all the patches into smaller 256×256 patches and flip the patches from the training set in order to meet the input requirement of the network.

Philipp *et al.* [34] provides the first patch-level-annotated dataset for multi-classification of concrete bridge defects covering cracks, efflorescence, scaling, spalling, and general defects (e.g., graffiti and moss). To consider possible defect combinations required by inspection guidelines, they also provide two other datasets for the binary classification of the exposed reinforcement and rust staining (corrosion). The total number of image patches in the multi-classification dataset and two binary-classification datasets are 3,607. The detailed distribution of the data volume for each defect type in the corresponding dataset is clarified in [34]. The patches do not have a consistent resolution since they are acquired from 38,408 images by slicing and labeling the defect area, with 21,284 images collected in the on-site experiment and 17,124 images provided by authorities. The image collection procedure adopts a 42-Mp camera and takes the shooting range, on-surface resolution (0.1 mm), camera focus, lighting condition, and surface angle between the subject surface and the camera optical axis into account for high-quality images.

KrakN [12] dataset is dedicated to thin crack detection. For the training set, over 900 pictures with a size of

Table 2
A summary of network architectures and training strategies adopted by the corresponding **classification-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
GAPs-v1 [13]	2017	a. ASINVOS Net [13] b. ASINVOS-mod Net [13]	✗	✓	✗
GAPs-v2 [14]	2019	a. ASINVOS Net [13] b. ResNet-10, -18, -34, -50 [44]	✓	✓	Adversarial training, Rotation, Translation
Xu [33]	2019	a. Xu's Net (with ASPP Module) [33] b. ResNet-18, -34, -50 [44] c. VGG-16, -19 [45]	✗	✓	Cropping, Flipping
Philipp [34]	2019	- Inception V3 [46]	✓	✗	✗
KrakN [12]	2020	- KrakN Net [12]	✓	✗	Cropping
CCIC [36]	2018	a. AlexNet [47] b. VGG-16, -19 [45] c. GoogLeNet [48] d. ResNet-50, -101, -152 [44]	✓	✗	Cropping
$\phi - Net$ [3]	2020	a. VGG-16, -19 [45] b. ResNet-50 [44]	✓	✗	Cropping
CSSC [10]	2017	- VGG-16 [45]	✓	✗	Cropping, Picking, Rotation, Sampling
SDNET2018 [37]	2018	- AlexNet [47]	✓	✓	Cropping

4248×2850 are collected from a cracked bridge pillar in good lighting conditions and at a close-up shooting distance ($20 - 30$ cm). Image cropping and labeling are conducted within 4 hours by using a self-developed semi-automatic tool. Only cracks and background surfaces are labeled as two classes. Afterward, $8,057$ image patches are acquired for cracks and background classes, respectively. Over $3,000$ images are collected from multiple scenarios and cameras for the validation set.

Drone Captured Tiny Crack Dataset (DCTCD) [35] consists of 250 images with complex textures (scrawls, surface corrosion, and efflorescences). DCTCD concentrates on bridge thin crack detection. All of the crack images are collected by a drone under bridge beams and pier inner walls. With controlled camera shooting distance, angles, and lighting conditions, the range of image pixel resolution is $0.1 - 0.2$ mm, which is beneficial to thin crack detection. Image color jitter, ISO noises, defocus blur, and motion blur are involved to imitate real application scenarios. The whole dataset is further split into five subsets according to different edge complexity factors defined in [35].

3.1.3. Buildings

Concrete Crack Images for Classification (CCIC) [36] dataset provides $40,000$ image patches with a size of 224×224, cropped from 500 high-resolution (4032×3024) images. The original images are collected from walls and floors of multiple concrete buildings, with various concrete surface finishes (plastering, exposed, and paint). During the image collection procedure, the camera directly faces the subject surface, and the data collection is finished in a single day to ensure consistent image illumination conditions.

Pacific Earthquake Engineering Research (PEER) Hub ImageNet (ϕ-Net) [3] provides $36,413$ image patches with building defects, which are collected and cropped from $100,000$ images collected from the field experiment and the Internet. Each image is labeled with 8 attributes related to local and global building information. Afterward, eight subsets are extracted for the classification of each attribute respectively.

3.1.4. Aggregated

Concrete Structure Spalling and Crack (CSSC) [10] dataset is the first released dataset for concrete spalling and crack detection. The initial dataset consists of $1,232$ images totally, with 278 spalling images and 954 crack images. All of the images are collected from the Internet through keyword searching. These images cover several types of infrastructure (e.g., bridges and buildings). Thus, the CSSC dataset is an aggregated dataset. The dataset also provides two subsets containing image patches with sizes of 100×100 and 130×130 for each defect class. Each patch in the subsets is annotated either as a 'True' or 'False' label, where the 'True' label stands for the patch with defects and the 'False' label represents the patch without defects or the patch with defects but does not meet the pixel threshold defined in [10]. For concrete spalling, the number of patches in the two subsets are $19,123$ ($7,376$ for 'True', $11,747$ for 'False') and $19,924$ ($8,574$ for 'True', $11,350$ for 'False') respectively. For concrete crack, the amounts of patches in the two subsets are $25,140$ ($13,448$ for 'True', $11,652$ for 'False'), $25,100$ ($13,422$ for 'True', $11,678$ for 'False') respectively. In addition to the patch-level annotated subsets, the CSSC dataset also annotates the initial images at pixel level according to the suggestions from the experts in civil engineering.

Structural Defects NET (SDNET2018) [37] is a patch-level annotated dataset for concrete crack classification. Altogether 230 images with sizes of 4068×3456 are acquired through a 16-MP camera. These images cover reinforced concrete building walls (72 images), bridge decks (54 images), and unreinforced concrete pavements (104 images). The working distance between the camera (without zoom) and the subject is 500 mm during the image acquisition. Each image is partitioned into multiple 256×256 image patches. Each image and patch cover a rough area of

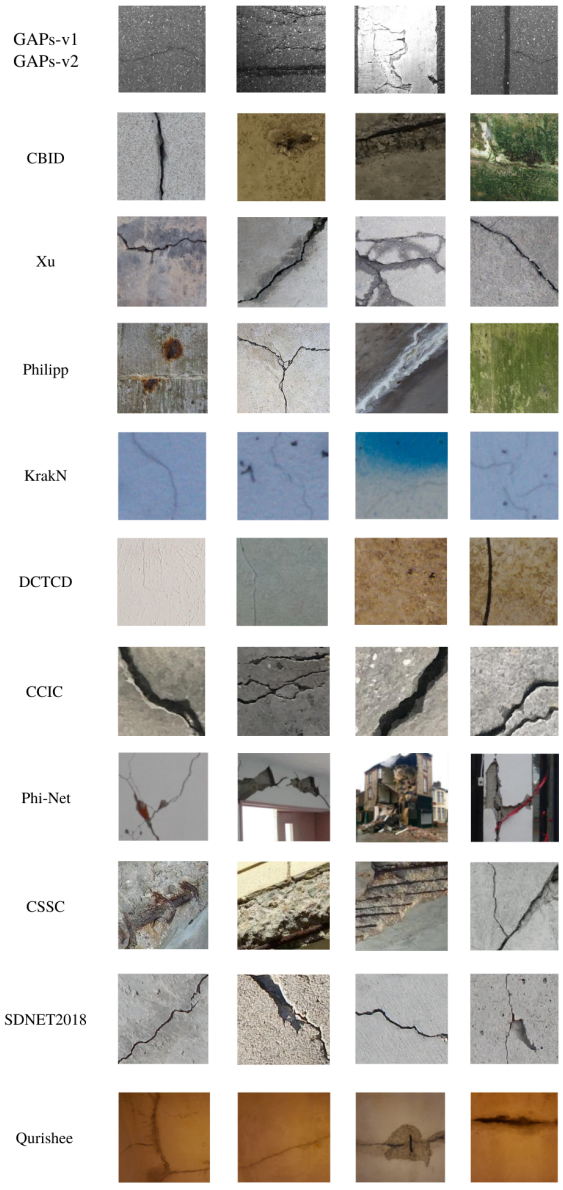

Figure 2: Exemplary images from datasets for infrastructure defects classification. From top row to bottom row, they are image patches from GAPs-v1 [13] & GAPs-v2 [14], CBID [32], Xu [33], Philipp [34], KrakN [12], DCTCD [35], CCIC [36], ϕ-Net [3], CSSC [10], SDNET2018 [37], and Qurishee [26] respectively.

1000 mm × 850 mm and 60 mm × 60 mm, respectively. Besides, each patch is annotated as 'Cracked' or 'Uncracked'. After partition and annotation, the respective amounts of patches for bridge decks, building walls, and pavements are 13,620 (2,025 for 'Cracked', 11,595 for 'Uncracked'), 18,138 (3,851 for 'Cracked', 14,287 for 'Uncracked'), and 24,334 (2,608 for 'Cracked', 21,276 for 'Uncracked'). On the whole, the dataset contains 8,484 patches with crack and 47,608 patches without crack. In addition to the sufficient data volume, the dataset also provides the range of crack width (0.06 mm-25 mm) covered, which will benefit the deep neural network to identify crack with size variety. Besides, the dataset intentionally incorporates images with various obstructions, including shadows, stains, edges, rough surface finishes, inclusions, voids, joints, and surface scaling, improving the robustness and generalization ability of the deep neural network real applications.

Qurishee *et al.* [26] proposed a dataset about concrete cracks. This dataset has 1,499 crack images and 589 non-crack images. The data are with two resolutions, 4032 × 3024 and 5312 × 2988. The resolutions of the data are relatively high and can show details that are not easily observed. However, equipment used to collect the data, algorithms for validating the dataset, and infrastructure type targeted are not specified.

3.2. Segmentation-oriented datasets

In this subsection, each dataset illustrated is annotated at the pixel level to conduct defects segmentation. Compared to the bounding-box-level annotation, pixel-level annotation can localize the defects more accurately and clearly. Table 3 shows the summary of publicly available segmentation-oriented defect datasets. These datasets are firstly sorted by corresponding infrastructure type, and then sorted in chronological order. Table 4 shows the corresponding algorithms and training strategies adopted for validating the datasets. Fig 3 show exemplary images for datasets targeting at different types of infrastructure.

3.2.1. Pavements

There are pixel-level-annotated datasets that are firstly used by traditional machine learning algorithms. They are Sylvie [50], CrackTree [51], Amhaz [52], and CrackForest Dataset (CFD) [53]. These datasets are dedicated to the crack segmentation of asphalt pavement. Due to their positive influence on the subsequent deep learning-based methods, corresponding attributes are also summarized and listed in Table 3. Since this paper focuses on datasets for deep learning, readers can find a more detailed description of the aforementioned datasets in the corresponding papers [50, 51, 52, 53].

For deep learning-based crack segmentation of asphalt pavement, Yang *et al.* propose the Crack500 dataset [54, 55] which has pixel-wise annotation and comprises 500 crack images with a resolution of 2000 × 1500. Each image is further cropped into 16 non-overlapping image patches, whereas patches in which the number of crack pixels is smaller than a certain threshold are discarded. Furthermore, based on GAPs-v1 [13] dataset, Yang *et al.* provide the GAPs384 dataset [55], in which 384 pavement images (1920 × 1080) containing only crack distress are selected and annotated at pixel level.

EdmCrack600 [56, 57, 58] dataset offers 600 backward-facing images with pixel-level annotation for pavement crack segmentation. All images are with a resolution of 1920 × 1080 and extracted from videos recorded by a sports camera mounted on the rear of a moving vehicle. The images vary in weather conditions, environmental effects, blurring effects, and noise.

Table 3
A summary of publicly available **segmentation-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data source/Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
Sylvie [50]	2011	42	Multiple	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
CrackTree [51]	2012	206	800 × 600	Not clarified	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
Amhaz [52]	2016	68	Multiple	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
CFD [53]	2016	118	480 × 320	Hand-held camera	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
Crack500 [54, 55]	2019	3368	640 × 360	Hand-held camera	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Not Clarified
GAPs384 [55]	2019	384	1920 × 1080	Cameras on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Pixel Level	Private License, for Academic Use Only
EdmCrack600 [56, 57, 58]	2020	600	1920 × 1080	Camera on ground vehicle	Crack	Pavement	Asphalt	Pixel Level	Scene Level	CC BY-NC-ND 4.0 License
GAPs-10m [15]	2021	20	5030 × 11505	Cameras on ground vehicle	23 distresses	Pavement	Asphalt	Pixel Level	Scene Level	Private License, for Academic Use Only
Highway-crack [59]	2021	5275	512 × 512	Cameras on UAV	Crack	Pavement	Asphalt	Pixel Level	Scene Level	Not Clarified
CCIC-600 [36]	2019	600	227 × 227	Not clarified	Crack	Bridge	Concrete	Pixel Level	Pixel Level	CC BY 4.0 License
BCL [60]	2021	11000	256 × 256	Hand-held cameras	Crack	Bridge	Concrete, Masonry, and Steel	Pixel Level	Pixel Level	CC0 1.0 License
CCSSS [61]	2021	440	512 × 512	Not clarified	Corrosion	Bridge	Steel	Pixel Level	Pixel & Object Level	CC0 1.0 License
LCW [62]	2021	440	512 × 512	Not clarified	Corrosion	Bridge	Steel	Pixel Level	Scene Level	CC0 1.0 License
DeepCrack [63]	2019	537	544 × 384	Not clarified	Crack	Building	Concrete, Asphalt	Pixel Level	Pixel Level	Private License, for Academic Use Only
Bai-2020 [64]	2020	853	256 × 256	Not clarified	Crack	Building	Concrete	Pixel Level	Pixel Level Object & Scene Level	GNU General Public License v3.0
Masonry [65]	2021	11491	224 × 224	Crawled from internet Hand-held cameras	Crack	Building	Masonry	Pixel Level	Pixel & Object Level	GNU General Public License v3.0
Ren [66]	2020	919	512 × 512	Hand-held camera	Crack	Tunnel	Concrete	Pixel Level	Pixel Level	MIT License
Sandra (IRT) [28]	2020	517	320 × 240	Hand-held thermal cameras	Crack, Spalling, Patches, Delamination	Dam	Concrete	Pixel Level	Pixel Level	Not Clarified
UAV75 [67]	2019	75	512 × 512	Camera on UAV	Crack	Not Clarified	Not Clarified	Pixel Level	Pixel Level	GNU General Public License v3.0
CSD [68]	2020	11298	448 × 448	Crawled from internet	Crack	Multiple	Multiple	Pixel Level	Pixel Level	Not Clarified
Bai-2021 [69]	2021	2229	Multiple	Crawled from internet	Crack, Spalling	Building, Bridge	Concrete	Pixel Level	Pixel Level Object & Scene Level	MIT License
CCCD [70]	2021	10995	448 × 448	Crawled from internet	Crack	Multiple	Multiple	Pixel Level	Pixel Level	CC0 1.0 License

Table 4
A summary of network architectures and training strategies adopted by the corresponding **segmentation-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
Sylvie [50]	2011	- Morph [50] (Morphological Analysis) - GaMM [50] (Multiscale Analysis and Local Crack Modelling)	Not Applicable	Not Applicable	✗
CrackTree [51]	2012	- CrackTree [51] (Minimum Spanning Trees)	Not Applicable	Not Applicable	✗
Amhaz [52]	2016	- Minimal Path Selection [52]	Not Applicable	Not Applicable	✗
CFD [53]	2016	- CrackForest [53] (Random Structured Forests)	Not Applicable	Not Applicable	Re-defined crack tokens
Crack500 [54, 55] GAPs384 [55]	2019 2019	- Feature Pyramid and Hierarchical Boosting Network [55] (FPHBN)	✗	✓	Cropping
EdmCrack600 [56, 57, 58]	2020	- ConnCrack [57] (cWGAN-based training)	✗	✓	Flipping, Cropping
GAPs-10m [15]	2021	a. U-Net [71], U-Net [71] (Xception [72]) b. An Encoder (Resnet-18, -50 [44])-Decoder (PSPNet [73]) Network	✗	✓	Flipping, Patch rotation, Patch scaling Modifying brightness, contrast, noise
Highway-crack [59]	2021	- U-Net [71] (Lighter Encoder and Attention Module)	✗	✓	Flipping, Rotation
BCL [60]	2021	a. U-Net [71] (Pruned Version) b. FCN [74] (VGG [45]) c. DeepLab V3 [75]	✗	✓	Cropping
CCSSS [61]	2021	- DeepLab V3+ [76]	✗	✓	Resizing
LCW [62]	2021	- DeepLab V3+ [76]	✗	✓	Resizing
DeepCrack [63]	2019	- DeepCrack [63]	✗	✓	Rotation, Cropping, Flipping
Bai-2020 [64]	2020	a. ResNet-152 [44] b. U-Net [71]	✓	✗	Resizing
Masonry [65]	2021	a. VGG-16 [45], ResNet-34, -50 [44] b. DenseNet-121, -169 [77], Inception V3 [46] c. MobileNet [78], MobileNet V2 [79] d. DeepLab V3+ [76], FCN [74] (VGG-16 [45]) e. U-Net [71] (with various backbones) f. FPN [80] (with various backbones)	✓	✗	Cropping
Ren [66]	2020	- CrackSegNet [66]	✓	✗	Rotation, Translation, Scaling, Shearing
Sandra (IRT) [28]	2021	a. VGG-16 [45] b. ResNet-18 [44] c. ResNet-50 [44] d. MobileNet V2 [79] e. Xception [72]	✗	✓	Cropping, Resizing Reflection, Translation
UAV75 [67]	2019	- CrackNausNet [67]	✓	✗	Resizing, Cropping, Rotation, Flipping
CSD [68]	2020	a. U-Net [71] (VGG-16 [45]) b. U-Net [71] (ResNet-101 [44])	✓	✗	Resizing
Bai-2021 [69]	2021	a. Mask R-CNN [81] (Cascade) b. Mask R-CNN [81] (APANet [82, 83]) c. Mask R-CNN [81] (HRNet [84])	✗	✓	Flipping, Rotation, Cropping
CCCD [70]	2021	- DeepLab V3+ [76]	✗	✓	Resizing

Based on the GAPs-v1 [13], and GAPs-v2 [14] dataset, Ronny *et al.* propose GAPs-10m [15] dataset annotated at the pixel level. The original dataset comprises 394 high-resolution downward-facing pavement surface images collected by following government regulations. All images are taken at different road sections to cover pavement distresses and object classes. 23 pavement distresses and object classes are defined by experts. The original dataset is then partitioned into a training set, a validation set, and a test set. As a subset of the validation set, the publicly available GAPs-10m dataset consists of 20 images (complying with German federal regulations) with a consistent resolution of 5030×11505. It is named after GAPs-10m since a single image covers 10 m in the image height direction. The dataset offers certain challenges, such as image artifacts caused by harsh sunlight and image stitching and the difficulty of distinguishing certain distress from the intact pavement surface.

Hong *et al.* [59] propose two datasets for highway crack segmentation. The first dataset is annotated at the pixel level based on the public dataset Aerial Crack Dataset [85], which is only annotated at the bounding-box level. After relabeling and cropping, the resulting dataset contains $4,118$ images with a resolution of 512×512. To validate the generalization ability of their proposed model, they constructed a second dataset comprising $1,157$ highway crack images collected by a UAV. These images are taken after a 6.4-level earthquake in China and annotated at the pixel level, with an image resolution of 5 cm and a UAV flight height of 200 m.

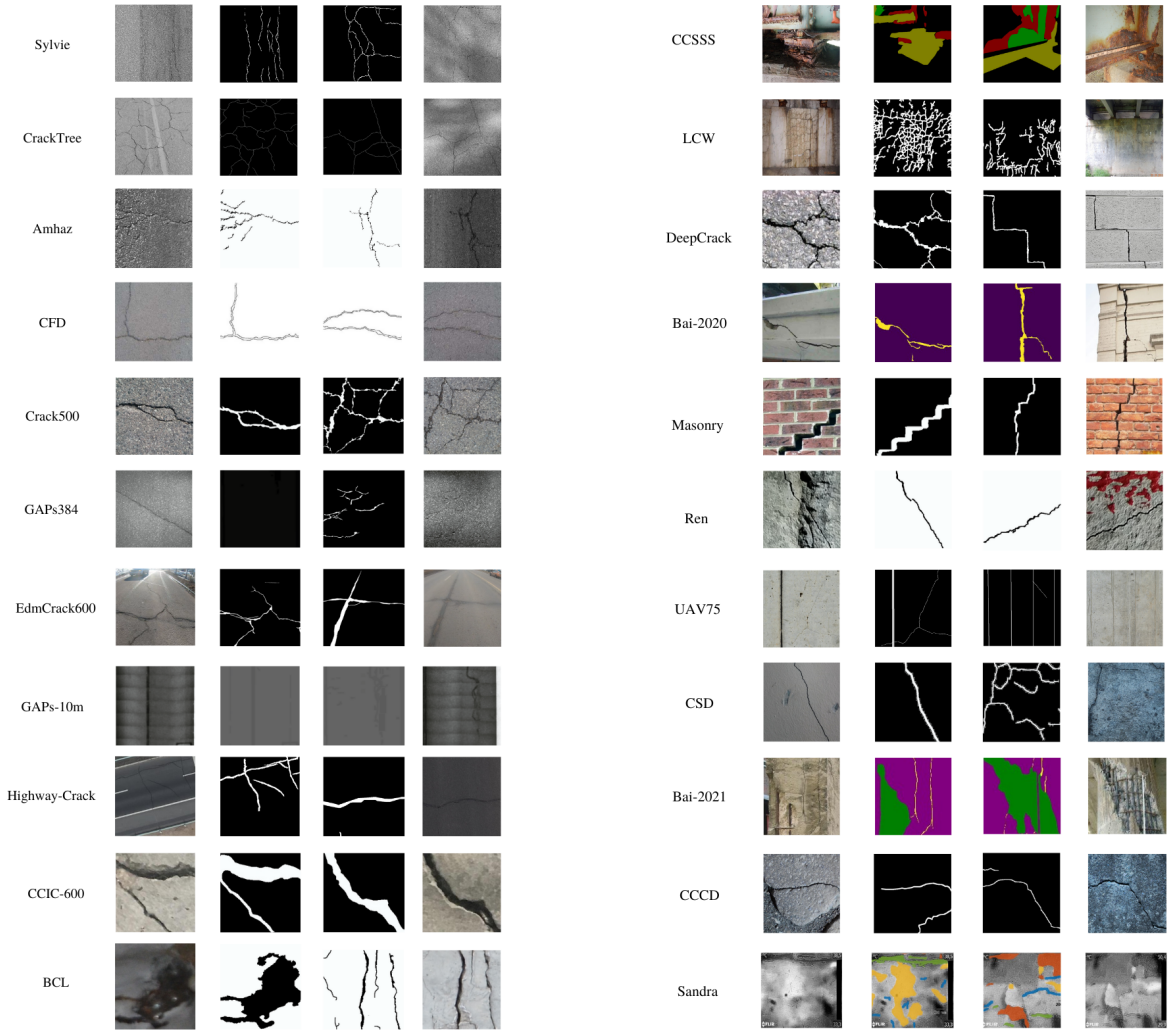

Figure 3: Exemplary images from datasets for pavement defects segmentation. From top row to bottom row, they are image patches and corresponding segmentation labels from Sylvie [50], CrackTree [51], Amhaz [52], CFD [53], Crack500 [54, 55], GAPs384 [55], EdmCrack600 [56, 57, 58], GAPs-10m [15], Highway-Crack [59], CCIC-600 [36], BCL [60], CCSSS [61], LCW [62], DeepCrack [63], Bai-2020 [64], Masonry [65], Ren [66], UAV75 [67], CSD, Bai-2021 [69], CCCD [70], Sandra [28] respectively. Each row represents two pairs of image patches with corresponding segmentation labels.

3.2.2. Bridges

CCIC-600[1] is an extension of the CCIC [36] dataset, aiming at concrete crack segmentation. Six hundred image patches are selected from the CCIC dataset and annotated at the pixel level. Bridge Crack Library (BCL) [60] dataset consists of 11,000 image patches with a size of 256×256 for bridge crack segmentation. This dataset covers three types of bridge materials, i.e., concrete, masonry, and steel. All the crack image patches are cropped from 1,180 raw crack images, with 1,000 nonsteel crack images and 180 steel crack images. Nonsteel crack images are collected by bridge inspection engineers through field inspection on 50 in-service bridges in China, with crack width within millimeters. Steel crack images are provided by the first International Project Competition for Structural Health Monitoring (IPC-SHM) [86]. The dataset can be divided into three subsets, with 5,769 nonsteel cracks, 2,036 steel cracks, and 3,195 crack-like motifs (cropped from 25,000 non-crack images for steel structures), respectively. A large proportion of crack-like motifs (such as shadows, stains, and water spots) are introduced intentionally to resolve the class imbalance between nonsteel crack and steel crack and improve model robustness.

Corrosion Condition State Semantic Segmentation (CCSSS) [61] dataset is devoted to the segmentation of bridge condition state. The dataset comprises 440 high-resolution images with a resolution of 512 × 512. The images are acquired from the bridge inspection reports and finely annotated at the pixel level by following government guidelines. This dataset is the first dataset to grade the corrosion state of bridges. The corrosion state is semantically annotated in four-level, i.e., good, fair, poor, and severe. The Labeled

[1] CCIC-600

Cracks in the Wild (LCW) [62] dataset is dedicated in scene-level bridge crack segmentation. The dataset consists of 3,817 images collected from bridge inspection reports. For the training purpose, all the images are resized to 512×512. All of the original images and resized images are publicly available.

3.2.3. Buildings, tunnels, and dams

DeepCrack [63] dataset is composed of 537 images for building surface crack segmentation. All the images are with a resolution of 544×384. This dataset covers multiple surface textures (bare, dirty, and rough), structure materials (concrete and asphalt), and crack scales (1 pixel to 180 pixels), which makes it be a challenging dataset. Bai-2020 [64] is a dataset for building crack localization. In addition to the images with pixel-level and object-level context information, the dataset contains some images at the structural (scene) level. The dataset contains 853 images with a resolution of 256×256.

The Masonry [65] dataset pays attention to cracks on the masonry walls of the buildings. The dataset contains 469 raw images either acquired from the Internet or captured by field experiments from several buildings in the Netherlands. Each image is divided into multiple image patches. The dataset includes images with varying scales, resolutions, crack appearances, and types of noisy backgrounds for more robust segmentation.

Ren *et al.* [66] provides a crack segmentation dataset focusing on tunnel environment. The raw images with a size of 4032×3016 are captured in a tunnel from China. Each raw image is further cropped into multiple image patches. Data augmentation techniques such as rotation, translation, scaling, and shearing are adopted to increase data volume.

Infrared images can be used to reveal subsurface defects. Sandra [28] proposes a segmentation dataset of white-hot infrared images containing four defect labels: crack, spalling, patches, and delamination. All images contain delamination and cracks, although some images do not contain spalling and patches. There are totally 517 images collected by FLIR and the resolution of data is 320×240. All annotations of these infrared images are labelled based on the corresponding optical images and engineering knowledge.

3.2.4. Aggregated

UAV75 [67] is a crack segmentation dataset emphasizing the images collected by the UAV. Compared with images captured by hand-held digital cameras and smartphones, the images acquired by the UAV may suffer from low resolution, low crack intensity, and re-occurring planking patterns. The authors notice that planking patterns may result in false-positive results. The planking class is added to the label space to distinguish planking patterns from cracks.

Bai-2021 [69] is an extension of the dataset [64]. They all focus on extreme events such as major earthquakes. Compared with the former version, Bai-2021 includes more images (2,229 additional images) with various resolutions from 147×288 to 4600×3070, more scenes including buildings and bridges, and more structural failures including cracks and spalling. Data augmentation is used to increase data volume.

Crack Segmentation Dataset (CSD)[2] is an aggregate dataset that merges 300 self-collected images (labeled at the pixel level) with several other crack segmentation datasets [10, 51, 52, 53, 54, 55, 63]. The dataset contains 11,298 images with a consistent resolution of 448×448. These images taking several cases into account, they are images containing pure crack, pseudo crack, crack with noise, crack with moss, and crack in large context. There is a high degree of similarity between CSD and Concrete Crack Conglomerate Dataset (CCCD) [70], which is also a conglomeration of several other crack segmentation datasets [10, 51, 52, 53, 54, 55, 63].

3.3. Detection-oriented datasets

The classification-oriented datasets (see Section 3.1) with images annotated at the image level can be used to conduct defect detection based on the sliding window techniques. Besides, there also exists detection-oriented defect datasets with images annotated at the bounding-box level to conduct multi-defect detection. Road Damage Dataset 2018 (RDD-2018) [87], RDD-2019, RDD-2020, and Pavement-Image-Dataset (PID) [92] are for pavement damage detection, while COncrete DEfect BRidge IMage (CODEBRIM) dataset [93] focuses on the multi-defect detection of concrete bridges. Compared to the defect segmentation, bounding-box-level annotation is benifical to the real-time defect detection and deployment. Table 5 shows the summary of publicly available detection-oriented defect datasets. These datasets are firstly sorted by corresponding infrastructure type, and then sorted in chronological order. Table 6 shows the corresponding algorithms and training strategies adopted for validating the datasets.

3.3.1. Pavements

Maeda *et al.* publish Road Damage Dataset 2018 (RDD-2018) [87], which is the first dataset for large-scale road damage detection. The dataset comprises 9,053 frontal-facing road images which contains 15,435 damage instances in Japan. All images with a uniform resolution of 600×600 are collected by a smartphone installed on the dashboard of the vehicle. These images have diverse background in terms of weather and surface conditions, which resembles the real-world scenarios. This dataset covers 8 damage classes such as cracks and corrosions, which is defined by government guidelines. A more detailed illustration of damage type and distribution can be found in [87]. For each damage in the image, the damage class and corresponding bounding box location are labeled. The number of images in the training set and validation set is 7,240 and 1,813 respectively. RDD-2018 was used as the benchmark dataset in Road Damage Detection and Classification Challenge (RDDCC) [104].

[2] CSD

Table 5
A summary of publicly available **detection-oriented** defect datasets (first sorted by infrastructure type, then sorted in chronological order)

Dataset	Year	Num.of Image Patches	Resolution	Data source/Platform	Defect Type	Structure Type	Material Type	Annotation Level	Image Context	License
RDD-2018 [87]	2018	9053	600 × 600	Camera on ground vehicle	Cracks and corrosions (8 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
RDD-2019 [88]	2019	13135	600 × 600	Camera on ground vehicle	Cracks and corrosions (9 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
RDD-2020 [89, 90, 91]	2020	26336	600 × 600 720 × 720	Cameras on ground vehicle	Cracks and potholes (4 damage classes)	Pavement	Asphalt	Bounding-box Level	Scene Level	CC BY-SA 4.0 License
PID [92]	2020	7237	640 × 640	Crawled from internet	Cracks (9 damage classes)	Pavement	Not clarified	Bounding-box Level	Scene Level	Not Clarified
Qurishee (IRT) [26]	2020	108 (IRT) 2620	up to 1024 × 768 up to 838 × 809	Not clarified Hand-held phone and UAV	Cracks Cracks (18 damage classes)	Pavement	Asphalt	Bounding-box Level	Pixel Level	CC BY 4.0 License
CODEBRIM [93]	2019	1590	up to 6000 × 4000	Hand-held cameras Cameras on UAV	Cracks and corrosions (5 damage classes)	Bridge	Concrete	Bounding-box Level	Pixel level Object & Scene Level	Private License, for Academic Use Only
GC10-DET [94]	2020	3570	up to 2048 × 1000	Hand-held cameras	Cracks and corrosions (10 damage classes)	Industrial plant	Steel	Bounding-box Level	Pixel Level	CC BY 4.0 License

Table 6
A summary of network architectures and training strategies adopted by the corresponding **detection-oriented** datasets

Dataset	Year	Network Structure	Transfer Learning	Trained from Scratch	Data Augmentation
RDD-2018 [87]	2018	a. SSD [95] (Inception V2 [96]) b. SSD [95] (MobileNet [78])	✗	✓	Flipping
RDD-2019 [88]	2019	a. SSD [95] (ResNet-50 [44]) b. SSD [95] (MobileNet [78])	✗	✓	PG-GAN [97], Poisson blending [98]
RDD-2020 [89, 90, 91]	2020	- SSD [95] (MobileNet [78])	✓	✗	✗
PID [92]	2020	a. YOLO V2 [99] b. Fast R-CNN [100]	✓	✗	✗
Qurishee (IRT) [26]	2020	- Faster R-CNN [101]	Not clarified	Not clarified	Not clarified
CODEBRIM [93]	2019	a. MetaQNN [102] b. Efficient Neural Architecture Search [93]	✗	✓	Cropping
GC10-DET [94]	2020	a. SSD [95] (VGG-16 [45]) b. Faster R-CNN [101] (ResNet-50 [44]) c. YOLO V2 (DarkNet-19) [99] d. YOLO V3 (DarkNet-53) [103] e. SSD [95] (VGG-16 [45])	✓	✗	Patches, Scaling

RDD-2019 [88] dataset is an extension and refinement of the RDD-2018 dataset. Compared to RDD-2018, RDD-2019 increases the number of annotated frontal-facing road images from 9,053 to 13,135, resulting in 30,989 road damage instances. All the newly added images with a resolution of 600 × 600 are still collected in Japan through a vehicle-mounted smartphone. Besides, all the images contained in RDD-2018 are reviewed, quality-controlled, and reannotated by road managers. A new class called 'utility hole' is added into the RDD-2019 dataset to discriminate the damage class 'pothole' from it. To expand the size of dataset, the authors apply progressive growing Generative Adversarial Network (PG-GAN) to generate synthetic images with 'pothole' damage class, more results can be found in [88]. However, the RDD-2019 dataset only includes real images.

RDD-2020 [89] is an extension of RDD-2019 [88] by incorporating additional road images taken in the Czech and India, which makes this dataset more heterogeneous and conducive to network robustness. It offers 26,336 frontal-facing road images collected by a vehicle-mounted smartphone in Japan, Czech, and India. These images contain more than 31,000 road damage instances with a wide variety of light and weather conditions. The whole dataset is partitioned into a training set and two test sets, their respective amounts of images contained are 21,041, 2,631, and 2,664. Images for Japan and Czech have a consistent resolution of 600×600, while for India, the image resolution is 720 × 720. This dataset is dedicated to road damage detection, unlike RDD-2018 and RDD-2019, it only covers 4 damage classes, i.e. potholes, alligator cracks, longitudinal cracks, and transverse cracks. Some extra damage classes are included in images collected in Japan for data consistency, more details can be found in [90]. For each damage in the image from the training set, the damage class and its corresponding bounding box coordinates are labeled. RDD-2020 dataset has also been used as the benchmark dataset by Global Road Damage Detection Challenge (GRDDC) [105], performance of state-of-the-art solutions can be found in [91].

Pavement Image Dataset (PID) [92] collects 7,237 images of 22 different pavement sections in the USA from Google street view. The images, with a 640 × 640 resolution, come from two types of camera views, including a wide view and a top-down view. Images from the wide view are used to detect pavement distresses, and top-down view images are employed to calculate the crack density for automated pavement rating in the future. The pavement distresses in this dataset consist of 9 crack types, including reflective, transverse, block, longitudinal, alligator, sealed transverse, sealed longitudinal, and lane longitudinal cracking, along with potholes. The numbers of images used in training and testing are 5,789 and 1,448, respectively, and the tool used to annotate images is python-based Openlabling software.

Qurishee *et al.* [26] propose a pavement crack detection dataset with 336 test images and 2,284 training images. All the images are collected by a hand-held mobile phone camera and a drone's camera. There is a total of 11 categories of flexible pavement cracks and 7 classes of rigid pavement cracks. These images are labelled by the open-source tool LabelImg with more than 50 hours of manual labour. In addition, they also propose a very small but high-resolution infrared dataset with 24 test images and 84 training images.

Table 7
A summary of manual and semi-automatic labeling methods of classification, segmentation and detection (first sorted by the payment situation, then sorted by the degree of automation of the labeling)

Name	Annotation level	Other types of input	Format of the exported dataset	Automatic labeling	Local Deployment	Web-based Deployment	Free-of-charge
Ybat [107]	Bounding-box Level	✗	a. Pascal VOC format b. YOLO format c. COCO format	✗	✗	✓	✓
LabelImg [108]	Bounding-box Level	✗	a. Pascal VOC format b. YOLO format c. CreateML format	✗	✓(Win, Linux, macOS)	✗	✓
LabelMe [109]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. COCO format	✗	✓(Win, Linux, macOS)	✗	✓
VIA [110]	Bounding-box Level	✓(audio, video)	a. COCO format b. VIA format c. CSV format	✗	✗	✓	✓
VoTT [111]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. TFrecord format c. VoTT format d. CSV format	✗	✓(Win, Linux, macOS)	✓	✓
PixelAnnotationTool [112]	Pixel Level	✗	- Only mask images (PNG files)	✓	✓(Win, Linux, macOS)	✓	✓
CVAT [113]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. COCO format d. TFrecord format e. CVAT format (and other 13 formats)	✓	✓(Win, Linux, macOS)	✓	✓
RectLabel [114]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. CreateML format d. CSV format	✓	✓(macOS)	✓	✗
Labelbox [115]	Bounding-box Level Pixel Level	✓(audio, video, text)	- Only JSON files containing labels	✓	✗	✓	✗
V7 Darwin [116]	Bounding-box Level Pixel Level	✓(video)	a. Pascal VOC format b. YOLO format c. CVAT format d. Darwin format	✓	✗	✓	✗

3.3.2. Bridge

COncrete DEfect BRidge IMage (CODEBRIM) dataset [93] focuses on the defects of concrete bridges. The images with defects are captured from 30 bridges by UAV and can be divided into five classes: crack, spallation, exposed reinforcement bar, efflorescence, and corrosion. In order to detect minor defects from different scales, cameras with high resolution (up to 6000×4000) and large focal lengths are adopted to collect images. One highlight of this dataset is that the images are labeled with multi-class, and the defects in the same image can be overlapped. There are only $1,590$ high-resolution images in this dataset, but the total number of labeling box are $7,806$, $5,354$ of which are overlapping defect and $2,506$ of which are non-overlapping.

3.3.3. Industrial plant

GC10-DET [94] dataset pays attention to the surface defect in a real industrial plant. The images with a resolution of 2048×1000 are captured by a set of linear array CCD cameras with a direct current light source to avoid the presence of stripes produced by an alternating current. The pixel size of the camera is $7.04\ \mu m \times 7.04\ \mu m$. Compared with the NEU-DET dataset [106], GC10-DET has more data and a greater variety of defect types: punching, weld line, crescent gap, water spots, oil spot, silk spot, inclusion, rolled pit, crease, and waist folding. With real scenes, high-precision collection tools, and high-resolution data, the AI models can be greatly enhanced and highly robust after training on this dataset.

3.4. Data collection and labeling

The data collection and labeling procedure can be summarized as follows: they are both labor-intensive and costly procedures. The first step in data collection is to survey the target site in advance to make the collection plan and select the collection equipment. Weather, light, and equipment all affect the quality of the dataset. After collecting the original data, it is necessary to clean it and eliminate similar and ambiguous data artificially. The next step is data labeling. Although some mature methods have been proposed for the labeling of classification and pixel-level segmentation [117] tasks, and some commercial software has been deployed on the website for user-friendly labeling, the efficient labeling strategy for large-scale unlabeled datasets is still in its infancy. This subsection illustrates and summarizes several labeling tools and their properties, including the annotation level, input data type, export format, labeling automation level, deployment configuration, and public accessibility in Table 7. The most commonly used open-source annotation tool is also compared with another semi-automatic open-source annotation tool to highlight the efficiency of automatic labeling.

With the popularity of deep learning-based image processing, many open-source annotation tools have emerged, including Ybat [107], a web-based annotation tool specially designed for the YOLO [99, 103, 118] series algorithm. And the classic annotation tool LabelImg [108], the most widely used open-source annotation tool LabelMe [109], VGG Image Annotator (VIA [110]) developed by VGG [45] network team that can efficiently annotate faces, and VoTT [111], a web-based annotation tool developed by Microsoft team. In addition to the above common open-source manual annotation tools, many semi-automatic annotation methods are also free-of-charge. PixelAnnotationTool [112] is a semi-automatic annotation tool for semantic and instance segmentation annotation tasks. CVAT [113] is a powerful and community-established semi-automatic annotation tool that

supports exporting 18 different data formats. During the annotating process, the target needs to be clicked by several key points, and then CVAT will automatically annotate the target. Although the functions of open-source annotation tools can meet our daily needs, they are still inferior to commercial annotation tools. RectLabel [114] is an annotation tool aiming at macOS users. RectLabel is unique in its ability to split images into uneven pieces, which the user can adjust to speed up automatic annotation. Labelbox [115] and V7 Darwin [116] are commercial annotation tools that can be used by simply logging into their web pages. Both can invite teammates to join in for annotation, orchestrate complex workflows, visualize annotation results and processes, and optionally train a specific network to improve the accuracy of automated annotation. In addition to the open-source and commercial tools, some researchers attempt to utilize machine learning algorithms for automatic labeling to get preliminary labeling results which can be then manually refined for accurate labeling in a much shorter time [119, 120, 121].

To compare the efficiency of the purely manual and the semi-automatic open-source tools, LabelMe (manual) is evaluated against CVAT (semi-automatic). The efficiency of CVAT is twice that of LabelMe, especially for some ordinary objects, such as vehicles and pedestrians. The automatic labeling algorithm identifies the object in two seconds with adjustable selection box details. The automatic labeling algorithm is prone to errors for defect datasets, such as cracks and spalling. However, in our experiment on the self-collected data, long cracks are divided into many small parts for labeling and the boundaries of labeled polygons are modified in a centralized way, saving about one-third of the time for manually labeling an RGB image full of cracks with a resolution of 6000×4000.

4. Comparison of SOTA algorithms for crack inspection

Before illustrating our algorithm comparison in detail, it is noteworthy that there exist recently published valuable works performing crack classification and segmentation tasks in different scenarios comparatively. Hallee *et al.* [123] pay their attention to masonry crack detection, where they systematically compare the domain adaption performance between the convolutional neural network (CNN) and traditional machine learning methods based on handcrafted features, including Support Vector Machine (SVM), Random Forest (RF), Gaussian Process (GP), Multi-Layer Perceptron (MLP), Naive Bayes (NB), and Quadratic Discriminant Analysis (QDA). The critical conclusion [123] is that successful domain adaption is possible in both the CNN and simple classifiers if trained on a wide range of masonry shapes, colors, and lighting conditions, complying with our conclusion in Subsection 5.1.1, Subsection 5.2.3, and Subsection 6.1.

Loverdos *et al.* [124] are dedicated to automating brick and crack segmentation of masonry walls. Regarding brick segmentation, extensive comparison experiments are conducted among networks, including U-Net, DeepLab V3+, LinkNet, and Feature Pyramid Network (FPN), all with various configurations. As to crack segmentation, SOTA architectures (with multiple backbones, training strategies, and loss functions) including DeepCrack, DeepLab V3+, Fully Convolutional Network (FCN), U-Net, and FPN are systematically compared to identify the best model configuration. The results are impressive when the brick segmentation and crack detection outputs are coupled. The essential remarks [124] are that deep learning methods allow for improving model performance by increasing the dataset used for training and validation, and the model performance can continually be enhanced by acquiring additional samples of the classified elements and desired features. These valuable remarks show the necessity and importance of our summarized datasets. Rezaie *et al.* [125] focuses on the crack segmentation of the stone masonry walls. They systematically compare a threshold method based on Digital Image Correlation (DIC) results and a deep learning-based method named TernausNet. The remarks are on the superiority of the deep learning method and its potential benefits for DIC methods and predictive models for damage level evaluation.

Based on the previous literature review, it can be found that crack is the dominant defect category in common structures [1, 13]. Its recognition is significant for a variety of applications such as the fault analysis and safe operations of public infrastructures such as bridge [126], building [36], and the electrical power grid [127]. Therefore, this research further develops a self-established crack classification and semantic segmentation dataset, based on which SOTA inspection algorithms are compared. We have developed an adapted Swin Transformer [122] from previous cutting-edge algorithms for crack classification as shown in Figure 5, and proposed a multi-layer fused attentional pyramid network for crack semantic segmentation as shown in Figure 6, respectively. Extensive experimental results show that the proposed approaches achieve comparable performance and efficiency to current SOTA approaches. Moreover, comprehensive comparisons between existing SOTA algorithms for crack classification and segmentation are conducted to provide a comprehensive baseline for future research in infrastructure defect inspection.

4.1. Our self-established crack classification and semantic segmentation dataset

A large-scale dataset for both crack classification and segmentation tasks is first established. Data for the classification task contain more than 15,000 images with image-level labels (crack or non-crack), while those for semantic segmentation contain more than 11,000 images with detailed pixel-level labeling. For crack semantic segmentation, 42% of images are derived from the internet and the remaining 58% is collected in our on-site inspection. And the corresponding percentage for classification is 36% and 64%, respectively. The preliminary version of our dataset has been released online at the following link to benefit the research

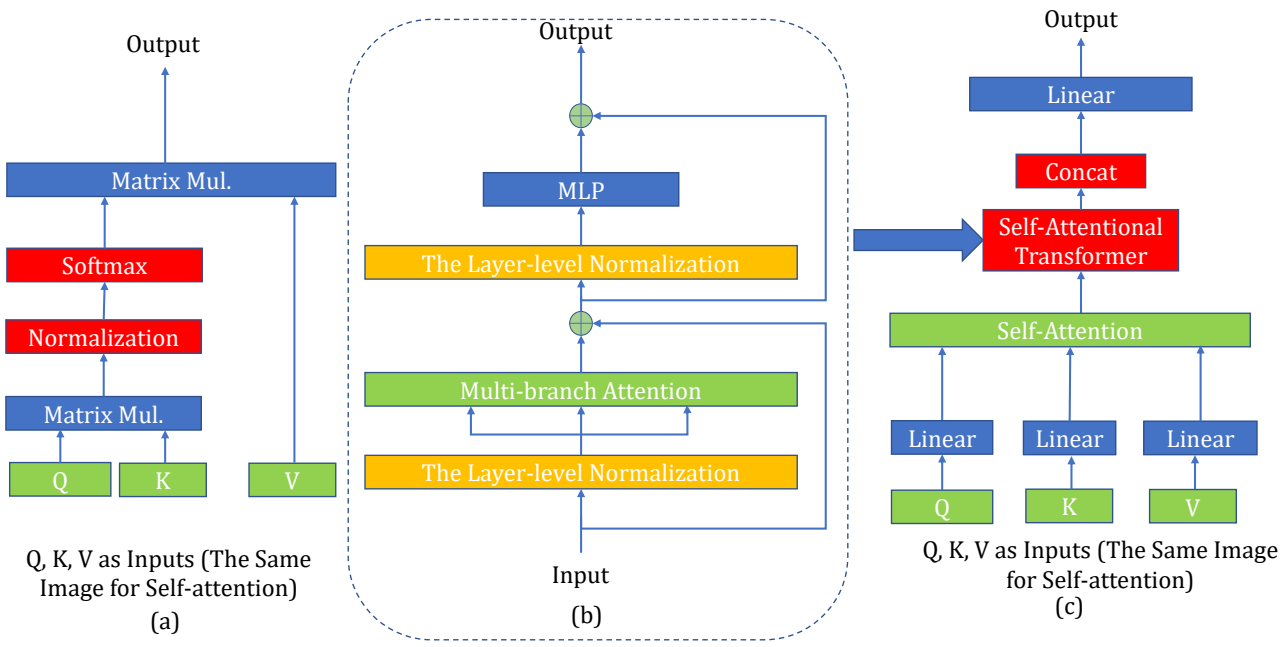

Figure 4: The detailed structure of the attentional module to be integrated into the Swin Transformer [122]. In subfigure (a), we have illustrated the network component of the original attention-based network. In subfigure (b), we have shown the multi-branch self-attention module we proposed to integrate into the current Swin Transformer [122] to boost the performance. Summarizing the whole network in the dash-line rectangle as a new module and integrating it into subfigure (c) as a self-attention transformer, the performance of the original transformer can be improved according to our experiments.

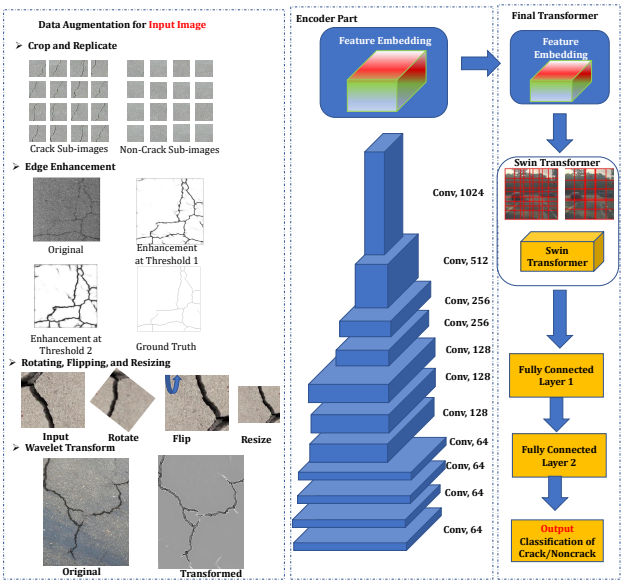

Figure 5: The detailed structure of the classification network adopted by us adapted from the Swin Transformer [122]. The network can achieve SOTA performance with the fine-tuning.

community for defect recognition[3]. Typical results are also shown on the website attached. Currently, this dataset can be used to perform crack recognition on the pavements effectively, and will be enriched further for building and tunnel inspections. In this way, it can be used for general UAV-based infrastructure inspections.

[3]Our Established Datasets Preliminary Version

4.2. Comparison of SOTA algorithms for crack classification

This subsection focuses on the task of crack classification, predicting whether a specific image contains a crack or not. Based on our self-established dataset, the existing SOTAs methods are compared, and the evaluation metric of the classification efficiency and effectiveness is detailed. Extensive experiments show that both the adapted Swin Transformer shown in Fig. 5 and the traditional convolutional network ResNeSt [128] show the best performance for crack classification. However, the ResNeSt [128] shows greater performance in the inference speed, and is more favorable in real industrial applications.

4.2.1. The definition of evaluation metric of crack classification

For the crack classification problem, the **accuracy** can be simply defined as follows:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (1)$$

For the binary classification, accuracy can be further defined in terms of positives and negatives as follows:

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

Where TP, TN, FP, FN stand for True Positives, True Negatives, False Positives, and False Negatives, respectively. Accuracy is the most direct metric for the one-hot prediction task of image-level crack classification, and also the

fairest. Then, we use the accuracy to make a comprehensive comparison between the SOTA algorithms for the task of crack classification. Also, the validation inference time on the tested images with the resolution of 1500×960 is used to evaluate the efficiency of diverse SOTA methods.

It is noteworthy that the accuracy metric can be misleading if there exists a class imbalance in the dataset. In our training dataset for crack classification, crack and non-crack images are balanced with the proportion of 41.3% and 58.7%, respectively. In the image classification task of more than $2,000$ images, the indeterminacy caused by the class imbalance can be neglected if the training set is not extremely imbalanced. Various techniques [129] can be adopted encountering class imbalance, such as under-sampling, oversampling, merging similar classes, and data augmentation.

4.2.2. Algorithms illustration

The network architectures are depicted and detailed in Figure 4 and Figure 5 for our crack classification-based crack detection applications. For the network settings, in this work, we only utilize the Swin Transformer [122]-based approach as an example. The transformers [122, 130] are the most up-to-date transformer-based network architectures for the general vision task of image classification. The transformer [130]-based network architecture has recently surpassed the CNN network architectures with its massive network architecture consisting of fully-connected network layers with a huge number of parameters. We firstly illustrate the basic ideas of our proposed multi-branch attentional transformers. As shown in Figure 4, in subfigure (a), we have illustrated the network component of the original attention-based network. In subfigure (b), we have shown the proposed multi-branch self-attention module to leverage the semantic correlation among various transformed feature representations for image patches. Finally, it can be integrated into the current Swin Transformer [122] to boost the performance. Also, the multi-branch self-attention module in subfigure (b) has the advantages of a larger receptive field and multi-branch concatenated feature representations, which are both significant to better modeling the contextual information within the image. After summarizing the whole network in the dash-line rectangle (subfigure (b)) as a new module and integrating it into subfigure (c) as a self-attentional transformer, the performance of the original transformer can be improved according to our experiments. The sliding window-based approach is utilized for the final crack classification-based crack detection.

It should be noted that the original Swin Transformer [122] can not handle the high-resolution testing image in training, which will result in out-of-memory for ordinary GPU devices. Therefore, we have cropped the original crack images into 60×60 sub-images for the ease of training with transformer [122]. As shown in Figure 5, our framework consists of data augmentation, the encoder part for feature embedding, and the final transformer. To deploy the computation-intensive transformer [122]-based models for crack classification, images should firstly be split/cropped into sub-patches to make it memory-efficient and computationally tractable for the original GPU such as NVIDIA GTX 1080 with 8 GB memory. The data augmentation is also of great significance to the final performance, for the fact that it can create more training samples for the better instance discrimination at the feature level. In this work, we have proposed to use the following four kinds of data augmentation. The crop and replicate, the edge enhancement, the rotating, flipping, resizing, and finally, the wavelet transform. The encoder of the network converts the input image to a feature embedding. Finally, the sub-images are fed into the Swin Transformer for the crack classification task.

We have also utilized the current popular architectures such as the ResNeXt-101 [131], the ResNeSt-101 [128] for doing the crack classifications. Also, the crack classification task takes longer in validation time as shown in Table 8 because we directly tested on the images with a large resolution of 1500×960. We have also tested and utilized the up-to-date transformer-based network architectures. Table 8 shows the related results. Note that our utilized Swin Transformer is also based on the widely adopted attentional feature correlation mining networks [132], which is the fundamental component of all transformer-based networks.

4.2.3. Detailed partitions of our dataset for crack classification

For the task of crack classification, the training set consists of 10,000 images. Moreover, the validation and test sets are composed of 4,500 and 500 images, respectively. For the large memory consumption of the transformer and the fairness of comparisons, we have utilized 120×100 sub-image for the training, and we have used 500 images with a resolution of 1500×960 for testing.

4.2.4. Experimental settings

For the task of **crack classification**, we train all compared networks in a unified setting. We train networks for 500 epochs on a single NVIDIA 2080Ti GPU with a batch size of 32 during training and 16 during testing. The initial learning rate is $5{\times}10^{-3}$ and decays by five times every 100 epochs. We select 500 epochs because training for 500 epochs is enough for the convergence of networks. Finally, we select the network weights that have the best performance on the validation set to do testing on the test set. We implement it in *Tensorflow* and optimize it with Adam optimizer [133]. Training the models to convergence takes approximately 9.5 hours for our self-established dataset with various crack patterns for the ResNeXt-101 [131] for example. All the models are trained from scratch for the task of crack classification. Furthermore, all our results are obtained from the results three times on average. Therefore, we have guaranteed fairness and robustness in all of our comparisons. In the future, we will also explore the possibility of large-scale pre-training and transfer learning-based approaches to achieve the relatively large-scale crack classification of more than a million images. However, although the performance

Table 8
The comparison of **crack classification** results between SOTA algorithms for the tested images with the resolution of 1500 × 960.

Network Architecture	Accuracy/%	Validation Time /ms
AlexNet [47, 134]	81.8	698.6
VGG-16 [45]	86.4	678.5
VGG-19 [45]	87.1	689.6
GoogLeNet [48]	83.6	875.5
ResNet-101 [44]	87.2	617.5
ResNeXt-101 [131]	87.9	1213.5
ResNeSt-101 [128]	88.2	1063.8
Swin Transformer-Base [130] [122]	87.7	2382.3
Swin Transformer-MB [130] [122]	88.0	2587.5
ShuffleNet [135]	85.7	1567.7
ShuffleNet V2 [136]	86.3	1645.8

of the large-scale pre-training is very prominent, the efficient network architectures and the efficient training strategies must be explored to put the large-scale pre-training into practice. Otherwise, it will remain a complex problem for academic research without the availability of high computational power.

4.2.5. The optimization loss function

For the optimization loss function, for simplicity and to guarantee the fairness of comparisons, we have adopted the unified cross-entropy optimization loss. The cross-entropy loss was used for the network training of all networks, including the SOTA networks and our proposed ones, which can be formulated as follows:

$$L = -\sum_{x^{In}} [y(x^{In}) \cdot log(p(x^{In})) + (1 - y(x^{In})) \cdot log(1 - p(x^{In}))] \quad (3)$$

where $p(x^{In})$ represents the predicted possibility of whether an input image x^{In} is a crack image, and y is the label of the input image. For crack image, $y = 1$. For non-crack image, $y = 0$. The loss can be utilized for the end-to-end training of the network framework. And finally, we present our experimental results.

4.2.6. Experiment results of crack classification

We conduct experiments to test the performance of various crack classification networks. The networks we tested have covered a broad range, which consists of current SOTA network architectures, including the classical AlexNet [134], and the newly proposed vision-transformers [122, 130]. As shown in the Table 8, we have also tested with other SOTA network backbones for crack classification, such as the vision transformer (ViT) [122, 130] which has the best performance among various methods in recent vision benchmarks. The results demonstrate that the recent approach, such as the ResNeSt [128] also has comparable performance with ViT, and has a much faster inference speed compared to the ViT. It can be seen that when using our multi-branch attentional layer in the Swin Transformer [122] (denoted as Swin Transformer-MB in Table 8), the performance can be boosted a little with a merely marginal increase on the computational cost (0.2 s validation time increase for the inference per image of 1500 × 960). It can be demonstrated that although the vision transformer-based methods can achieve remarkable performance, the computational and memory costs should be considered in the deployment stage. For the robotics applications with real-time requirements, the faster methods such as the ResNet-101 [130] or ResNeSt-101 [128] are more preferred for efficiency considerations. Also, although the Swin Transformer [122] based methods have comparable or slightly better performance under various circumstances compared with the typical convolutional network [45], and residual network [128] based methods, it requires a large inference time, which is unacceptable in real-time applications. Therefore, taking the efficiency and accuracy of both into consideration, the ResNeSt [128] is the best choice for the crack classification task.

4.3. Comparison of SOTA algorithms for crack segmentation

This subsection takes crack semantic segmentation as a case study of defect recognition in modern infrastructures. As mentioned in Subsection 4.1, the images are labeled in pixel levels for segmentation and summarized into our self-established dataset, based on which performances of various network architectures are compared in detail. Utilizing our designed network architecture (Fig 6) combined with existing SOTA network backbones such as ResNet [44], ResNeXt [131], and VGG [137], the performance will be enhanced when the domain gap between the source and target test data is not large. Table 9 presents the comparison results of recent crack segmentation methods such as the DeepCrack [63].

4.3.1. The definition of evaluation metrics of crack semantic segmentation

We have utilized various metrics for a fair evaluation of the performance of different methods, as shown in Table 9. The inference time is the testing time for an image of resolution in 600 × 480 for the task of crack segmentation. We define the average precision, mean Intersection over Union (mIoU), precision, recall, best F-measure on our test set for a fixed threshold (DS), and the total F-measure on our test set for the threshold on each image (IS) in the same settings as the [138]. These evaluation metrics are commonly recognized and adopted evaluation metrics for comparisons in defect identification. Also, the validation inference time on the test images is used to evaluate the efficiency of diverse SOTA methods. Our experimental results are conducted three times to obtain an average value for fair comparisons.

4.3.2. Algorithms illustration

We have developed algorithms for crack segmentation and the subsequent detection based on non-maximum suppression (NMS). To keep the paper brief, we will merely take our proposed Attentional Pyramid Scene Parsing-based network architecture integrated with full resolution ResNet [139] as an example case for algorithms illustration. As

shown in Fig 6, the network adopts the typical encoder-decoder-based basic structure for semantic segmentation. Unlike the transformer-based model for the task of crack classification, the input image can be directly fed into the encoder of the network structure based on these convolutional neural network-based segmentation models. As shown by the red module linking the intermediate feature output encoder and the decoder of the network in Figure 6, we have also incorporated the attentional transformer for the self-correlated feature extraction of the image at the pixel level. Utilizing this kind of design, the correlated features in the embedding space will be effectively enhanced, and the distinct features will be well separated. The attentional transformer shown in red in Figure 6 is used to enhance the feature correlation mining capacity of the network. All the decoder features are ultimately concatenated to give the final predictions. This kind of network design can make the model focus more on the critical zones of the images and pay less attention to the insignificant ones. Also, the attention-based transformer can be beneficial in increasing and enlarging the spatial contextual information and fusing them with the low-level feature representations in the encoder. We adapted it based on the SOTA attention-based [132] transformer and integrated it into our network structure. Through this kind of network architecture, the low-level feature cues, such as the edges and corners, and the high-level semantic cues, such as the crack patterns, can be fully utilized and learned based on the training data. Moreover, we use selective search-based methods to do the detection. The selective search-based methods [140] use the traditional sliding window-based approaches for object detection. Furthermore, we utilized the efficient nearest neighbor query methods to find the next sliding window for detection and efficiently do the final object detection. Finally, as shown in the Algorithm 1, we have summarized the proposed detailed procedures for NMS-based object detection. Denote B as the list of the initially obtained detection boxes. S contains the corresponding detection scores. And N_t is the NMS threshold. The set D is utilized to store the final box. As shown in Fig. 7, we can utilize NMS to obtain the most typical object detection bounding boxes obtained from selective search-based methods in the original RGB images.

4.3.3. Detailed partitions of our dataset for crack semantic segmentation

For the tasks of **crack semantic segmentation**, we have partitioned the original dataset into the training set, the validation set, and the test set. The dataset consists of more than 11,000 images with a resolution of 600 × 480. We have utilized 6,000 images for training, 3,000 images for validation, and the remaining 1,650 images for testing.

4.3.4. Experimental settings

We adopt the same setting as crack classification except that the initial learning rate is $1{\times}10^{-4}$. Training the model to convergence takes approximately 17.5 hours for our self-established dataset with various crack patterns. All the models are trained from scratch for the task of semantic segmentation.

Algorithm 1: The non-maximum suppression based algorithm for object detection (Simplified Version)

Input: The **input** initial detection boxes B, the related corresponding detection scores S, the related NMS threshold N_t
Output: The **output** final detection boxes D and the corresponding detection score S.

$D \leftarrow \emptyset$
while $B \neq empty$ **do**
  Select the maximum value in the set of S, and give this value to m. $m \leftarrow argmax(S)$
  $M \leftarrow b_m$
  $D \leftarrow D \cup M$
  $B \leftarrow B - M$
  for b_i *in* B **do**
    if $iou(M, b_i) > N_t$ **then**
      $B \leftarrow B - b_i$; $S \leftarrow S - s_i$;
  return D, S

4.3.5. The optimization loss functions

In addition to the network architectures illustrated above, we further illustrate the optimization functions used for the network training. In real situations, the crack is usually thin, which means most of the pixels in the captured images are non-crack. Different from the traditional cross-entropy loss, we have proposed our class-balanced loss function to tackle the problem of extreme class imbalance in the task of crack semantic segmentation. Also, we have also proposed the multi-stage fused loss, which can operate well with our proposed multi-stage fused pyramid network to boost the network performance. The optimization loss function is detailed as follows. We calculate the total number of crack and non-crack pixels in the training images are p and q respectively. The class frequencies of crack and non-crack are $\frac{p}{p+q}$ and $\frac{q}{p+q}$, while the median for the 2 classes is 0.5. Then the median divided by the class frequency gives the weight of two classes. In our case, the weights of the loss function for the crack pixels and non-crack pixels are $\alpha_1 = \frac{p+q}{2p}$ and $\alpha_2 = \frac{p+q}{2q}$ respectively. Then for each side-output layer, the improved loss function for the h-th side outputs $L^h_{side}(W)$ can be formulated as:

$$L^h_{side}(W) = -\alpha_2 \sum_{j \in S^-} log(1 - P(W)) \\ -\alpha_1 \sum_{j \in S^+} log(P(W)) \tag{4}$$

where $h = 1, 2 \cdots, H$ respectively are the convolutional stages of the network. The H denotes the total stages. S^+ and S^- are the total number of crack pixels and non-crack pixels respectively for an input image. And P denotes the predicted possibility of each pixel to be a crack one. The W denotes the

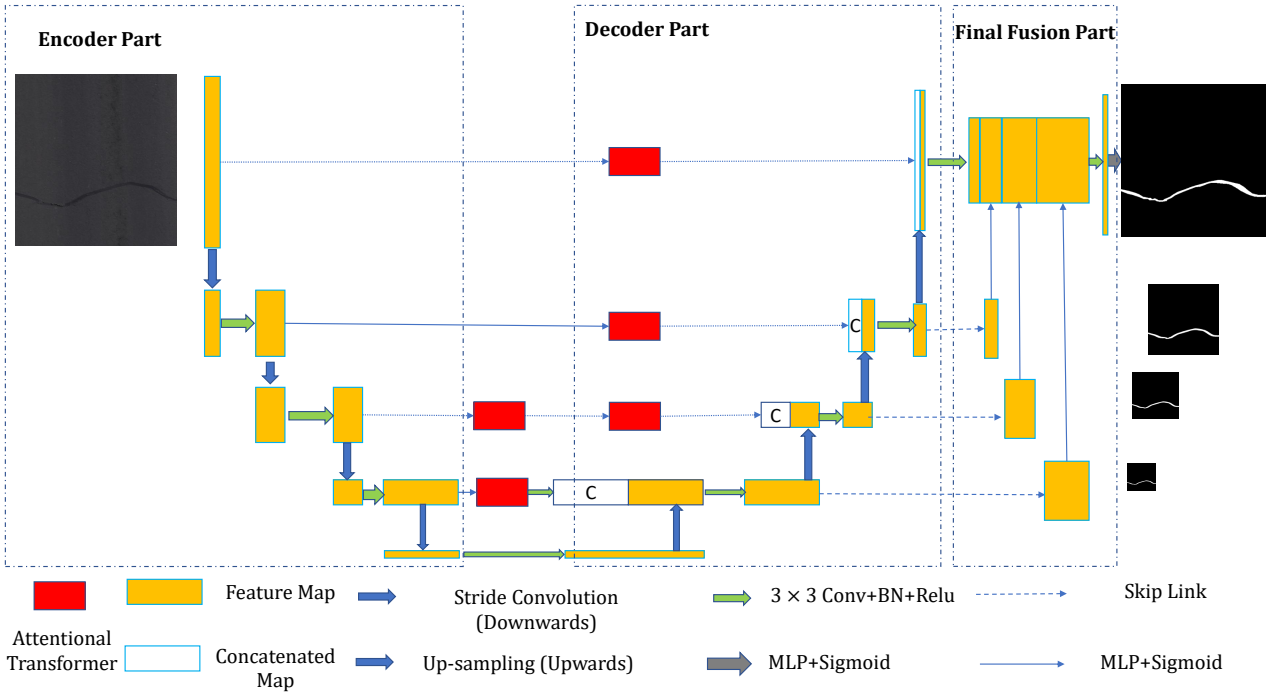

Figure 6: The proposed Multi-Stage-Fused Attentional Pyramid Network structure. We have proposed an encoder-decoder network architecture with skip-connections enhanced by the attentional transformer module. The attentional transformer module is added to better enhance the final segmentation performance.

Table 9
The comparison of semantic segmentation results between our proposed and various current SOTA methods

Methods	Inference Times/ (ms)	Thres (0-1)	Average Precision	mIoU	Precision	Recall	DS	IS
Original Hierarchical Neural Network [141]	165	0.49	82.3	75.9	74.6	76.5	75.6	77.5
SegNet [142]	215	0.52	80.2	75.6	73.3	74.8	74.1	74.7
FCN-8s [143]	176	0.55	81.1	76.9	74.2	75.5	74.8	75.8
U-Net [143]	168	0.53	82.1	77.1	73.7	74.9	74.3	75.3
DeepLab V2 [144]	192	0.55	83.2	78.7	76.9	75.9	76.4	75.6
DeepLab V3 [144]	226	0.50	83.6	79.3	74.9	74.9	74.9	75.7
PSPNet V1 [145]	257	0.49	83.4	79.8	75.5	75.6	75.6	76.3
ASPP-Net [146]	266	0.51	85.2	78.9	75.4	75.7	75.6	76.2
DeepCrack [63]	708	0.50	78.6	76.9	71.2	72.3	71.7	72.3
CrackNet based DeepLab V3+ [141] [144] (Our)	252	0.45	**86.3**	**77.8**	**75.3**	**75.6**	**75.5**	**75.8**
CrackNet based DenseNet [141] [147] (Our)	502	0.51	**86.6**	**77.6**	**76.1**	**75.1**	**75.6**	**76.3**
CrackNet based Full Res-ResNet [141] [139] (Our)	324	0.56	**87.3**	**76.9**	**76.6**	**75.5**	**76.1**	**76.6**

weights of the whole proposed transformer-based network shown in Fig. 5. Next the improved loss function $L_{fuse}(W)$ for the fused output can be also written as:

$$L_{fuse}(W) = -\alpha_2 \sum_{j \in S_-} log(1 - P(W)) \\ -\alpha_1 \sum_{j \in S_+} log(P(W)) \tag{5}$$

And then the total optimization loss function $L_{total}(W)$ is written as:

$$L_{total}(W) = \sum_{j=1}^{J} (\sum_{h=1}^{H} L_{side}^{h}(W) + L_{fuse}(W)) \tag{6}$$

The multi-stage fused optimization loss functions has the advantages of considering both the low-level feature in the early stages of the network such as edges and corners, and the high-level semantic information in the deeper stages of the network. Thus, the multi-stage hierarchical information can be extracted and fused in an adapted manner and this kind of information is further formulated into the network optimizations to boost the final segmentation performance in an explicit way.

4.3.6. Experiment results of crack semantic segmentation

The results of crack segmentation have been shown in Figure 7. We have utilized the characteristics of our proposed network shown in Figure 6 to construct multi-stage deep hierarchical feature representations for each tested network. The feature pyramid network has been demonstrated to

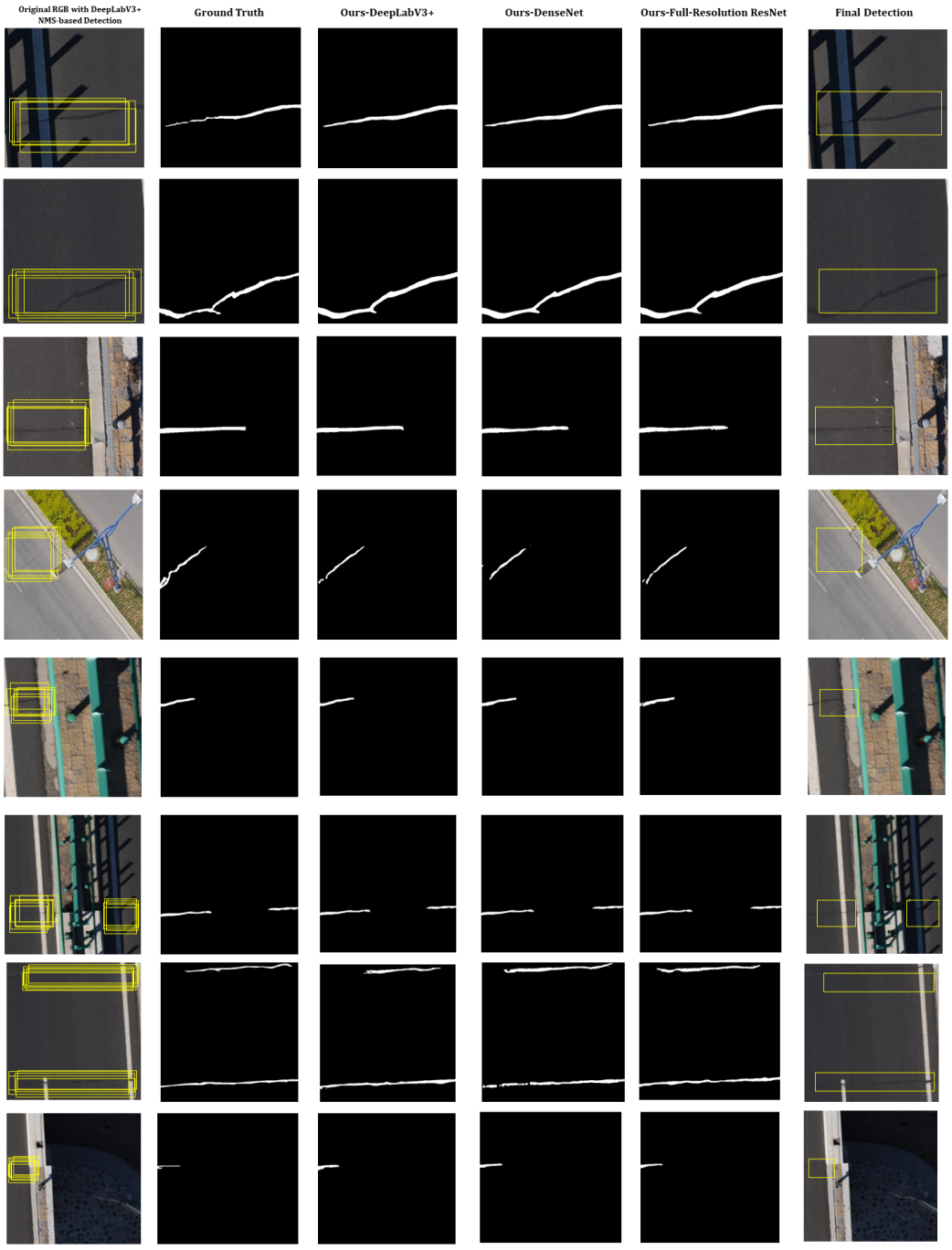

Figure 7: The pixel-level semantic segmentation results of our adapted CrackNet shown in Fig. 6 integrated with SOTA deep convolutional networks for semantic segmentation after conducting real-site pavement inspection. The black color denotes the background, while the white color denotes the segmented results of various cracks. Our approach can realize accurate segmentation of the cracks under different complex backgrounds, with large shadows, road greenings, and roadside bricks.

be very effective in the hierarchical and multi-layer fused feature extraction [148, 149]. As mentioned, the proposed network shown in Figure 6 adopts the encoder-decoder-based architecture. It is highly effective in feature extraction because we fuse the representations from diverse network stages to obtain a compound representation integrating low-level geometry cues (including edges and corners) and high-level semantics (including the category and semantic feature representations). Also, we have successfully integrated our proposed network with SOTA dilated convolutions network Deeplab V3+ [76] as shown in the third column of Figure 7. In the deeper stages of the network, we choose to use the dilated convolutions instead of the original classical convolutions because the dilated convolutions provide a much larger receptive field. And the spatial resolutions can be well maintained. Also, the dilated convolutions generate the feature

maps of various scales compared with the input data, which further improves the scale robustness of the proposed network. Moreover, we have utilized more elaborately designed network architectures such as the DenseNet [147] and SENet [150]. The densely connected residual learning is conducted to obtain a better feature representation. The full-resolution residual networks (FRRNs) [139] utilize the two streams to fuse the multi-scale global contextual information with the pixel-level local information. The first stream carries information at full resolution to achieve accurate segmentation of the boundaries of various shapes. The second stream utilizes a series of max-pooling operations to obtain the high-level feature for recognition. The FRRNs [139] couples these two streams and finally provide a hierarchically fused segmentation map. In our work, we have adapted the original FRRNs [139] for a better multi-stage fusion of hierarchical features based on our design shown in Figure 6. For the crack segmentation, a certain segmentation threshold (denoted as Thres on Table 9) require to be chosen for obtaining the final binary segmentation map. We have chosen the threshold for various tested segmentation methods based on their original implementation in their original paper.

We have successfully integrated our proposed attentional pyramid network architecture shown in Fig. 6 with current SOTA deep network models for semantic segmentation, such as the DeepLab V3+ [144], DenseNet [147], and FRRNs [139]. As shown in Fig. 7, it can be demonstrated that integrated with our proposed method [141], the DeepLab V3+ [144], DenseNet [147], and FRRNs [139] all show superior performance when encountered with the diverse background if the network training parameters are fine-tuned. We have shown the typical detection results under diverse complicated circumstances. Our approach can realize very accurate segmentation of the cracks under different complex backgrounds, with large shadows, road greenings, and roadside bricks. It can be seen that the shadows can be properly handled. Although shadow pixels do not have sharp contrasts with the background pixels of road surface, the network will not mistakenly recognize the high-contrast shadows as cracks. From the last column, we have also shown the object detection results with the non-maximum suppression (NMS)-based post-processing approach. The details of the NMS algorithm are shown in Algorithm 1. It is demonstrated that it can suppress the redundant bounding boxes obtained from the semantic segmentation, and select the most typical detection bounding box result based on selective search for object detection [140]. Also, it can be seen that the selective search methods can find candidates with excellent efficiency and robustness. The object detection follows the semantic segmentation results to do more accurate crack object detection, which demonstrates the effectiveness and robustness of our proposed method [141]. The final results of crack semantic segmentation performance are also summarized in Table 9. It can be demonstrated that our proposed method can be successfully integrated with SOTA methods and shows consistently better performance compared with other ones. The performance increment can be ascribed to the effective network design of attentional transformer module and effective multi-stage fusion strategies. We have integrated our proposed network architecture in Figure 6 with three typical segmentation backbone networks. As shown in Table 9, it can be seen that the three proposed networks all achieve SOTAs semantic segmentation performance in terms of mIoU. Also, the provided comprehensive comparisons between existing SOTA algorithms for crack classification and segmentation can provide a solid baseline for future research in industrial infrastructural defects inspection.

5. Suggestions on datasets and methodology

5.1. The suggestions on constructing a defect dataset

5.1.1. Classification-oriented dataset

Classification task is the basic building block to the detection and segmentation task. To build up a high-quality dataset for defect classification, the defect categories should be firstly defined according to the government inspection guidelines. Then, the data collection procedure should be conducted and recorded in a controlled environment by strictly following inspection guidelines. The data collection system should be developed or chosen for specific application scenarios. The accuracy and robustness of the object classification algorithm face several challenges posed by object viewpoint variation, intraclass variation (e.g., the same type of crack but with a different background or color intensity), the difficulty of identifying fine-grained categories (e.g., various types of the crack), background clutter, illumination changes, deformation, and occlusion. The dataset can wittingly incorporate images with the challenges mentioned above to improve the accuracy and robustness. Besides, it should be noticed that there exist conflicts between the labeling results of different annotators. The effect of annotation conflict can be alleviated by introducing a self-checking mechanism during the labeling process or utilizing label smoothing techniques during the network training process [15]. Moreover, data augmentation (e.g., crop and flip) can be adopted to increase the data volume.

5.1.2. Segmentation- and detection-oriented dataset

The dataset should be recorded in a standardized way. The corresponding infrastructure type, material type, defect type, data type, sensor specifications, data collection procedure, and geometric properties of the defects should be recorded. The dataset should have sufficient data and defect diversity to train a superior defect detector. Traditional data augmentation and GAN-based data augmentation (e.g., Defect-GAN [151]) can be used to increase the data volume. The context level of the dataset also matters. The pixel-level context is conducive to the network training process, while the object-level context is beneficial to localizing the defects, relating the defects to the structure, and further evaluating the hazard level of defects. The scene-level context can increase the generalization ability of the trained model in real

applications. It is promising to build up a multi-modal defect dataset (e.g., SDNET2021 [30]). RGB images are conducive to detecting surface defects, while IRT images, IE signals, and GPR signals reveal subsurface defects. It should be noted that there are conflicts between different annotation results (even when annotated by experts) [15], which will influence the training result.

5.2. The suggestions on defect visual inspection methodologies

5.2.1. Developing advanced methods and algorithms

In real industrial applications, the specific infrastructure to be inspected can not be easily accessed. Although intelligent industrial robots such as UAVs or UGVs with sensing capacity have been developed, complicated autonomous localization, navigation, and planning algorithms should be developed to collect high-quality data on the target infrastructure to be inspected. In most cases, merely limited high-quality data for the inspected target can be collected, and the labeling process is time-consuming and cumbersome. Therefore, to train and deploy an effective crack recognition model for modern industrial applications, firstly, the efficient labeling strategy should be further explored to achieve highly efficient labeling, which we have discussed in detail in Table 7. Secondly, the domain gap should be considered in establishing the dataset. Domain adaptation is a great method that can expand the applications of the crack recognition model across different domains. From our experience, effective domain adaptation in crack detection and segmentation can be achieved if we take the intrinsic information into consideration and formulate them into the optimization of the deep network model. The intrinsic information in the images includes depth and edge information. For crack recognition, the edges reveal the most likely pixels that belong to cracks. Moreover, the drastic change in depth information can also indicate the change in the 3D geometric structures. Therefore, they all play an essential role in finding the intrinsic feature representations of cracks and can be well utilized to improve the generalization capacity of the learning-based deep neural network models.

5.2.2. Using more advanced 3D sensors

Intrinsically, the defects such as crack and spalling are structural damages. And the geometric patterns of them can be captured very easily by 3D sensors. Therefore, advanced sensors, such as the 3D industrial cameras, the advanced high-precision industrial LiDAR sensors, and the industrial laser scans should be incorporated to better enhance the 3D geometrical information, which is just complementary to the 2D visual information. Subsequently, to better enhance the performance in defects recognition, the fusion networks or mechanisms should be further developed to boost the performance by utilizing the complementary characteristics of multiple sensors.

5.2.3. Constructing high-quality database to boost the performance of SOTA Methods

The algorithm design and the database are complementary to each other. The highly effective algorithms and high-quality datasets can both boost the final defects recognition performance in a mutually beneficial way. According to our experiments, various SOTA networks have nearly equal performance in the task of crack semantic segmentation. From our experience, the issue that matters most in achieving highly accurate industrial defects recognition lies in two aspects regarding the constructed dataset. The first is the amount of the training data, and the second is the quality of the data. To be more specific, firstly, the amount of the training data should be sufficient enough to support various types of defects, such as the most typical infrastructural damages with varying patterns including crack and spalling. Also, the domain gap between the training set and the on-site captured test images of the infrastructures to be inspected should be as small as possible. Secondly, the quality of the training data should also be guaranteed, which means the geometric patterns of various defects are largely covered in the established dataset. When faced with real industrial applications, the quantity and the quality of the dataset should be evaluated carefully to guarantee robustness in inspections. When evaluating the performance, a high-quality large-scale dataset will also be beneficial to the robust and fair comparisons between diverse learning-based defects identification approaches.

5.2.4. Algorithms illustration and recommendation for weakly-supervised defect recognition without sufficient labeling for industrial applications

In real industrial applications, according to our experiments, it can be seen that the fully supervised defects classification, detection, and segmentation approaches have an upper bound in recognition accuracy, even with a fully labeled training set and no domain gap between the training and test set. Moreover, their actual performance depends more on the effectiveness of the learned model from limited labeled data. The detection accuracy may also experience a considerable drop when there is a large domain gap between the source labeled datasets and target unlabeled defects to be inspected. Therefore, this subsection discusses several promising weakly supervised algorithm approaches to alleviate the data hunger problem in defect recognition.

For **semantic segmentation**, many weakly or semi-supervised approaches have been proposed to reduce the demand for large-amount of annotated datasets, such as weakly supervised image segmentation methods with image-level labels [152]. Attention mechanism with a transformer-based network design can be used to extract the semantic affinity between various contextual objects, using the affinity from attention (AFA) module to refine and improve the quality of the pseudo labels. For the semi-supervised semantic segmentation, U2PL [153] has been proposed to make better use of the unreliable samples in the unlabeled data. Because a large amount of unlabeled data contains a great deal of

meaningful information in both low-level geometry and high-level semantics, the U2PL can make full use of the unlabeled data of low reliability as negative samples to boost the performance of the semantic segmentation models.

For **object detection**, the remarkable work Dense-Teacher [154] with a newly defined teacher-student model can be adopted to improve the performance of the single-stage object detector. The threshold-based object detection models also rely on NMS and therefore depend on accurate semantic segmentation results as shown in our experiments. However, choosing an inappropriate threshold will result in noisy pseudo labels. The teacher model gives a dense model of the whole feature map and proposes the quality focal loss to supervise the output of the student model. Using the mean-teacher scheme, the DTG-SSOD [155] can provide dense supervision for the teacher model with iterative NMS clustering and rank match strategies. Therefore, more abundant features and information on the unlabeled data are utilized.

In addition, a single model of multi-task learning can be utilized to handle the **object classification, semantic segmentation, and object detection** simultaneously for real applications. It has great potential to enhance real-time performance and construct memory-efficient learned models with real-time performance for multiple tasks. In 2D multi-task learning, classical works have formulated it as a multi-objective optimization problem and jointly optimized every target with network training [156]. The cracks and other structural defects can be regarded as 3D geometric changes, with point clouds captured by LiDAR sensors or RGB-D cameras. For multi-task learning based on 3D point clouds in a weakly supervised setting, the approach proposed in [157] can tackle the 3D scene understanding problem with limited labels, and can be integrated seamlessly with different neural network backbones to achieve 3D scene perception with multiple down-streams tasks. There is still considerable room for improving 2D/3D multi-task learning for defect recognition given their data characteristics and advantages.

6. Conclusions and outlook

This paper summarizes 40 publicly available defect datasets for deep learning-based classification, segmentation, and detection tasks. The architectures are suggested for the task of classification and semantic segmentation, while multiple deep learning-based models are trained, validated, and tested, and the performances are compared in a very detailed way. Critical remarks on the review and comparison results as well as future research directions are summarized as below.

6.1. Remarks on review and comparison results

Based on the comprehensive review and systematic comparison in this paper, major findings with deep learning-based defect inspection are presented as follows:

(1) The **quantity** of the summarized defect datasets: The volume of summarized publicly available datasets reaches around 13.38 M, with approximately 13.25 M, 0.061 M, 0.064 M for classification, segmentation, and detection respectively. The quantity of the dataset shrinks dramatically when the inspection task transfers from classification to high-level segmentation and detection, because segmentation and detection tasks require further annotation exhausting resources of the researchers. Considering the significant impact of the labor-intensive labeling process on productivity, SOTA labeling tools are summarized and compared in Subsection 3.4. The labeling process with a semi-automatic labeling tool is found to save about 33% of the time compared with a manual labeling tool. Furthermore, to alleviate data scarcity, the inspection research community is enriched with our self-established defect dataset, which contains more than 15,000 and 11,000 images for defect classification and semantic segmentation, respectively.

(2) The **diversity** of the summarized defect datasets: The dataset diversity lies in the defect type, infrastructure type, material type, and image context. The reviewed datasets cover more than 5 most common and vital defect types including crack, spalling, delamination, corrosion, and efflorescence, as well as more than 5 civil infrastructures including the pavement, bridge, building, tunnel, and dam. 5 material types including concrete, asphalt, steel, masonry, and wood are targeted with 3 image context levels (i.e., pixel, object and scene). The diversity in material types and image context level is essential since deep learning-based defect inspection algorithms depend highly on the diverse content and context features to generalize effectively.

(3) The **difficulty** for constructing a high-quality defect dataset: As to defect classification, 7 challenges get in the way of developing accurate and robust classification algorithms, including viewpoint variation, intra-class variation, difficulty of identifying fine-grained categories, background clutter, illumination changes, defect deformation, and occlusion. The established dataset needs to wittingly contain images with the aforementioned features to adapt the deep learning-based algorithms with higher accuracy and robustness. Regarding defect segmentation and detection, an additional main difficulty lies in annotating the defect image accurately and efficiently. Some attempts to utilize machine learning algorithms for automatic labeling is identified to get preliminary labeling results which can be then manually refined for accurate labeling in a much shorter time [119, 120, 121].

(4) The **feasibility** of the data collection platforms for defect inspection: Among the summarized 40 visual defect datasets, 10 of them are collected via cameras installed on ground vehicles, 6 of them are acquired by cameras on UAV platforms, 13 of them are obtained via hand-held cameras, and the rest are crawled from the internet. The ground vehicle is preferred as a data collection platform for pavement inspection due to its stability, accessibility, and long-duration ability. The UAV platform

is preferred as a feasible and cost-effective solution to conduct defect inspection of bridges and high-rise buildings. It is noteworthy that a valuable dataset collected by the UAV, named "Highway-crack dataset [59]", contains highway crack images taken just after a 6.4-level earthquake in China, revealing UAV's rapid response capability. Hand-held cameras are the most common data collection tools but their field of view (FOV) is limited, accompanied by image occlusion and perspective distortion, resulting in the incorrect recognition of defects and their geometric properties.

(5) The **scalability** for establishing a large-quantity defect dataset: Data augmentation methods, composed of basic image manipulations (e.g., kernel filters, geometric transformations, random erasing, and color space transformations) and deep learning approaches (based on adversarial learning, neural style transfer, and GAN) [158], are identified to efficiently expand the data volume of the defect dataset [88, 151].

(6) The **superiority** of our proposed algorithms: We have proposed the multi-branch self-attention module and multi-stage-fused attentional pyramid network architecture. As to crack classification, the multi-branch self-attention module is successfully integrated into the Swin Transformer [122] to get the adapted Swin Transformer-MB network, which achieves 88.0% accuracy slightly better than the original Swin Transformer with 87.7% accuracy and ranks in the second place out of 11 SOTA classification networks. For crack semantic segmentation, the multi-stage-fused attentional pyramid network architecture is successfully combined with SOTA segmentation networks such as DeepLab V3+ [144], DenseNet [147], and FRRNs [139]. The resulting models achieve satisfactory performances among 12 SOTA segmentation networks, with 77.8%, 77.6%, 76.9% mIoU respectively and an acceptable efficiency on the modern graphic processing unit.

(7) The **criticality** of algorithm comparison results: We have systematically compared 11 SOTA classification networks in terms of the accuracy and efficiency and 12 SOTA segmentation networks in terms of the widely-accepted accuracy metrics and efficiency. Based on the comparison results, suggestions are provided regarding the model deployment on robotic platforms and the development of semi-supervised algorithms for defect inspection. A good starting point is set up for the follow-up researchers and practitioners.

6.2. Outlook for automatic defect inspection

Following concluding marks, potential research topics are proposed as below for defect inspection:

(1) **Establish a multi-modal benchmark dataset**: A large-scale multi-modal dataset containing data collected from multiple sensors, such as optical cameras, IRT cameras, depth cameras, IE, GPR, ultrasonic sensors, and industrial LiDAR, will be conducive to defect localization and quantifying. One such dataset called "SDNET-2021 [30]" is identified for detecting the subsurface defects of the bridge decks and benchmarking advanced deep learning models. Advanced data fusion methods will be required to tackle the defect inspection more accurately with the established multi-modal dataset.

(2) **Standardize the summarized visual defect dataset**: The research community lacks a widely-accepted large-scale benchmark dataset for advancing and fairly comparing deep learning algorithms for visual defect inspection. Despite the systematically summarized 40 publicly available defect datasets, enormous efforts are still needed to standardize all the datasets into a unified benchmark defect dataset.

(3) **Develop datasets and algorithms for evaluating defect hazard level and predicting structure deterioration**: The core objective of the defect inspection is to quantify the hazard level of the defect. Except for the CCSSS [61] dataset, no publicly available visual dataset is found to evaluate defect hazard levels. Besides, the prediction of structure deterioration needs more research attention for the estimation of optimal rehabilitation measures [159, 160, 161].

(4) **Develop autonomous robotic platforms**: Most robotic platforms for defect data collection still rely on manual or remote control, requiring at least one operator to be exposed to uncomfortable and dangerous environments. An autonomous data collection platform [162, 163] can not only enhance the operation safety, but also accelerate the inspection process with improved objectivity and accuracy, providing a better reference for follow-up maintenance decisions and rehabilitation measures.

(5) **Develop automated defect inspection pipelines**: The community still lacks an integral defect inspection pipeline, which can automatically register the corresponding information (e.g., localization, quantity, hazard level, and tendency) of the defect to civil infrastructure management systems. Some attempts have been made in [164] and [165], where the 2D defect inspection results are mapped to reconstructed 3D model from LiDAR data and further registered to the building information modeling (BIM) system [165] or geographic information system (GIS) [166]. The digital twin (DT) system can also benefit the data storage and analysis process [167].

CRediT authorship contribution statement

Guidong Yang: Conceptualization, Investigation, Formal analysis, Writing - Original Draft. **Kangcheng Liu:** Conceptualization, Investigation, Methodology, Formal analysis, Writing - Original Draft. **Jihan Zhang:** Investigation, Formal analysis, Writing - Original Draft. **Benyun Zhao:** Investigation, Formal analysis, Writing - Original Draft. **Zuoquan Zhao:** Investigation, Formal analysis, Writing -

Original Draft. **Xi Chen:** Conceptualization, Resources, Supervision, Writing - Review & Editing, Project administration. **Ben M. Chen:** Conceptualization, Funding acquisition, Resources, Supervision, Writing - Review & Editing, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reproted in the paper.

Acknowledgements

This project is supported in part by the Research Grants Council of Hong Kong SAR (Grant No: 14209020 and Grant No: 14206821) and in part by the Hong Kong Centre for Logistics Robotics (HKCLR).

References

[1] Wai-Kiong Chong and Sui-Pheng Low. Assessment of defects at construction and occupancy stages. *Journal of Performance of Constructed facilities*, 19(4):283–289, 2005.

[2] Wai-Kiong Chong and Sui-Pheng Low. Latent building defects: causes and design strategies to prevent them. *Journal of performance of constructed facilities*, 20(3):213–221, 2006.

[3] Yuqing Gao and Khalid M Mosalam. PEER Hub ImageNet: A large-scale multiattribute benchmark data set of structural images. *Journal of Structural Engineering*, 146(10):04020198, 2020.

[4] Yasser El Masri and Tarek Rakha. A scoping review of non-destructive testing (NDT) techniques in building performance diagnostic inspections. *Construction and Building Materials*, 265:120542, 2020.

[5] Mahesh Yumnam, Hina Gupta, Debdutta Ghosh, and Jayaprakash Jaganathan. Inspection of concrete structures externally reinforced with FRP composites using active infrared thermography: A review. *Construction and Building Materials*, 310:125265, 2021.

[6] Ali Akbar Shirzadi Javid, Parviz Ghoddousi, Gholamreza Ghodrati Amiri, and Khalil Donyadideh. A new photogrammetry method to study the relationship between thixotropy and bond strength of multi-layers casting of self-consolidating concrete. *Construction and Building Materials*, 204:530–540, 2019.

[7] Junzhi Zhang, Jin Huang, Chuanqing Fu, Le Huang, and Hailong Ye. Characterization of steel reinforcement corrosion in concrete using 3D laser scanning techniques. *Construction and Building Materials*, 270:121402, 2021.

[8] Wei Jiang, Youjun Xie, Jianxian Wu, and Guangcheng Long. Influence of age on the detection of defects at the bonding interface in the CRTS III slab ballastless track structure via the impact-echo method. *Construction and Building Materials*, 265:120787, 2020.

[9] Mezgeen Rasol, Jorge C. Pais, Vega Pérez-Gracia, Mercedes Solla, Francisco M. Fernandes, Simona Fontul, David Ayala-Cabrera, Franziska Schmidt, and Hossein Assadollahi. GPR monitoring for road transport infrastructure: A systematic review and machine learning insights. *Construction and Building Materials*, 324:126686, 2022.

[10] Liang Yang, Bing Li, Wei Li, Zhaoming Liu, Guoyong Yang, and Jizhong Xiao. Deep concrete inspection using unmanned aerial vehicle towards CSSC database. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 24–28, 2017.

[11] Devdatt P Purohit, NA Siddiqui, Abhishek Nandan, and Bikarama P Yadav. Hazard identification and risk assessment in construction industry. *International Journal of Applied Engineering Research*, 13(10):7639–7667, 2018.

[12] Mateusz Żarski, Bartosz Wójcik, and Jarosław Adam Miszczak. KrakN: Transfer learning framework for thin crack detection in infrastructure maintenance. *arXiv preprint arXiv:2004.12337*, 2020.

[13] Markus Eisenbach, Ronny Stricker, Daniel Seichter, Karl Amende, Klaus Debes, Maximilian Sesselmann, Dirk Ebersbach, Ulrike Stoeckert, and Horst-Michael Gross. How to get pavement distress detection ready for deep learning? a systematic approach. In *2017 international joint conference on neural networks (IJCNN)*, pages 2039–2047. IEEE, 2017.

[14] Ronny Stricker, Markus Eisenbach, Maximilian Sesselmann, Klaus Debes, and Horst-Michael Gross. Improving visual road condition assessment by extensive experiments on the extended gaps dataset. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[15] Ronny Stricker, Dustin Aganian, Maximilian Sesselmann, Daniel Seichter, Marius Engelhardt, Roland Spielhofer, Matthias Hahn, Astrid Hautz, Klaus Debes, and Horst-Michael Gross. Road surface segmentation-pixel-perfect distress and object detection for road assessment. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1789–1796. IEEE, 2021.

[16] Sattar Dorafshan, Robert J. Thomas, and Marc Maguire. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, 186:1031–1045, 2018.

[17] Ruoxu Ren, Terence Hung, and Kay Chen Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, 2018.

[18] Jun Kang Chow, Kuan fu Liu, Pin Siang Tan, Zhaoyu Su, Jimmy Wu, Zhaofeng Li, and Yu-Hsing Wang. Automated defect inspection of concrete structures. *Automation in Construction*, 132:103959, 2021.

[19] Qiuchen Zhu and Quang Ha. A bidirectional self-rectifying network with bayesian modelling for vision-based crack detection. *IEEE Transactions on Industrial Informatics*, 2022.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[21] Yuzhi Zhao, Lai-Man Po, Tingyu Lin, Xuehui Wang, Kangcheng Liu, Yujia Zhang, Wing-Yin Yu, Pengfei Xian, and Jingjing Xiong. Legacy photo editing with learned noise prior. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2103–2112, 2021.

[22] Kangcheng Liu, Yuzhi Zhao, Zhi Gao, and Ben M Chen. WeakLabel3D-Net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[23] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Conv: Large-scale lidar point clouds understanding leveraging feature correlation mining and geometric-aware modeling. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12896–12902. IEEE, 2021.

[24] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Net: A fast and accurate framework for large-scale lidar point cloud understanding. *IEEE Transactions on Cybernetics*, 2022.

[25] Kexin Guo, Zhirong Qiu, Cunxiao Miao, Abdul Hanif Zaini, Chun-Lin Chen, Wei Meng, and Lihua Xie. Ultra-wideband-based localization for quadcopter navigation. *Unmanned Systems*, 04(01):23–34, 2016.

[26] Murad Al Qurishee, Weidong Wu, Babatunde Atolagbe, Joseph Owino, Ignatius Fomunung, and Mbakisya Onyango. Creating a dataset to boost civil engineering deep learning research and application. *Engineering*, 12(3):151–165, 2020.

[27] Narges Kheradmandi and Vida Mehranfar. A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, 321:126162, 2022.

[28] Sandra Pozzer, Ehsan Rezazadeh Azar, Francisco Dalla Rosa, and Zacarias Martin Chamberlain Pravia. Semantic segmentation of

defects in infrared thermographic images of highly damaged concrete structures. *Journal of Performance of Constructed Facilities*, 35(1):04020131, 2021.

[29] Sattar Dorafshan and Hoda Azari. Deep learning models for bridge deck evaluation using impact echo. *Construction and Building Materials*, 263:120109, 2020.

[30] Eberichi Ichi and Sattar Dorafshan. SDNET2021: Annotated NDE dataset for Structural Defects. 2021.

[31] Kexin Guo, Zhirong Qiu, Cunxiao Miao, Abdul Hanif Zaini, Chun-Lin Chen, Wei Meng, and Lihua Xie. Ultra-wideband-based localization for quadcopter navigation. *Unmanned Systems*, 04(01):23–34, 2016.

[32] P. Huethwohl. Cambridge bridge inspection dataset. *Online*, 2017.

[33] Hongyan Xu, Xiu Su, Yi Wang, Huaiyu Cai, Kerang Cui, and Xiaodong Chen. Automatic bridge crack detection using a convolutional neural network. *Applied Sciences*, 9(14):2867, 2019.

[34] Philipp Hüthwohl, Ruodan Lu, and Ioannis Brilakis. Multi-classifier for reinforced concrete bridge defects. *Automation in Construction*, 105:102824, 2019.

[35] Mingpeng Li. Concrete-crack-detection dataset. 2020.

[36] C. F. Özgenel and Arzu Gönenç Sorguç. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC)*, pages 693–700, July 2018.

[37] Sattar Dorafshan, Robert J Thomas, and Marc Maguire. Sdnet2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. *Data in brief*, 21:1664–1668, 2018.

[38] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, 2005.

[39] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[40] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[41] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

[42] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[49] Yun Wang, Ju Zhang, Jing Liu, Yin Zhang, Zhi Chen, Chun Li, Kai He, and Rui Yan. Research on crack detection algorithm of the concrete bridge based on image processing. *Procedia Computer Science*, 154:610–616, 01 2019.

[50] Sylvie Chambon and Jean-Marc Moliard. Automatic road pavement assessment with image processing: review and comparison. *International Journal of Geophysics*, 2011, 2011.

[51] Qin Zou, Yu Cao, Qingquan Li, Qingzhou Mao, and Song Wang. Cracktree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, 33(3):227–238, 2012.

[52] Rabih Amhaz, Sylvie Chambon, Jérôme Idier, and Vincent Baltazart. Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2718–2729, 2016.

[53] Yong Shi, Limeng Cui, Zhiquan Qi, Fan Meng, and Zhensong Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[54] Lei Zhang, Fan Yang, Yimin Daniel Zhang, and Ying Julie Zhu. Road crack detection using deep convolutional neural network. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3708–3712, 2016.

[55] Fan Yang, Lei Zhang, Sijia Yu, Danil Prokhorov, Xue Mei, and Haibin Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1525–1535, 2019.

[56] Qipei Mei, Mustafa Gül, and Md Riasat Azim. Densely connected deep neural network considering connectivity of pixels for automatic crack detection. *Automation in Construction*, 110:103018, 2020.

[57] Qipei Mei and Mustafa Gül. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Construction and Building Materials*, 256:119397, 2020.

[58] Qipei Mei, Mustafa Gül, and Nima Shirzad-Ghaleroudkhani. Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles. *Journal of Civil Structural Health Monitoring*, 10:653–665, 2020.

[59] Zhonghua Hong, Fan Yang, Haiyan Pan, Ruyan Zhou, Yun Zhang, Yanling Han, Jing Wang, Shuhu Yang, Peng Chen, Xiaohua Tong, et al. Highway crack segmentation from unmanned aerial vehicle images using deep learning. *IEEE Geoscience and Remote Sensing Letters*, 2021.

[60] Xiao-Wei Ye, T Jin, ZX Li, SY Ma, Y Ding, and YH Ou. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *Journal of Structural Engineering*, 147(11):04721008, 2021.

[61] Eric Bianchi and Matthew Hebdon. Corrosion Condition State Semantic Segmentation Dataset. 12 2021.

[62] Eric Bianchi and Matthew Hebdon. Labeled Cracks in the Wild (LCW) Dataset. 10 2021.

[63] Yahui Liu, Jian Yao, Xiaohu Lu, Renping Xie, and Li Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation. *Neurocomputing*, 338:139–153, 2019.

[64] Y Bai, Bing Zha, Halil Sezen, and Alper Yilmaz. Deep cascaded neural networks for automatic detection of structural damage and cracks from images. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:411–417, 2020.

[65] Dimitris Dais, Ihsan Engin Bal, Eleni Smyrou, and Vasilis Sarhosis. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, 125:103606, 2021.

[66] Yupeng Ren, Jisheng Huang, Zhiyou Hong, Wei Lu, Jun Yin, Lejun Zou, and Xiaohua Shen. Image-based concrete crack detection in tunnels using deep fully convolutional networks. *Construction and Building Materials*, 234:117367, 2020.

[67] Christian Benz, Paul Debus, Huy Khanh Ha, and Volker Rodehorst. Crack segmentation on UAS-based imagery using transfer learning. In *2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–6, 2019.

[68] khanhha. Crack segmentation. https://github.com/khanhha/crack_segmentation, 2020.

[69] Yongsheng Bai, Halil Sezen, and Alper Yilmaz. Detecting cracks and spalling automatically in extreme events by end-to-end deep learning frameworks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2:161–168, 2021.

[70] Eric Bianchi and Matthew Hebdon. Concrete Crack Conglomerate Dataset. 10 2021.

[71] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[72] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[73] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[74] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[75] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[76] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[77] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[78] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[79] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[80] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[81] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[82] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

[83] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6688–6697, 2019.

[84] Yongsheng Bai, Halil Sezen, and Alper Yilmaz. End-to-end deep learning methods for automated damage detection in extreme events at various scales. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6640–6647. IEEE, 2021.

[85] Bo Wang. Aerialcrackdataset: Towards object detection with dataset. https://github.com/arasharchor/AerialCrackDetection_Keras, 2017.

[86] Hui Li and Billie F. Spencer Jr. the first International Project Competition for Structural Health Monitoring. *Journal of Structural Engineering*, 2020.

[87] Hiroya Maeda, Yoshihide Sekimoto, Toshikazu Seto, Takehiro Kashiyama, and Hiroshi Omata. Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, 33(12):1127–1141, 2018.

[88] Hiroya Maeda, Takehiro Kashiyama, Yoshihide Sekimoto, Toshikazu Seto, and Hiroshi Omata. Generative adversarial network for road damage detection. *Computer-Aided Civil and Infrastructure Engineering*, 36(1):47–60, 2021.

[89] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, and Yoshihide Sekimoto. RDD2020: An annotated image dataset for automatic road damage detection using deep learning. *Data in brief*, 36:107133, 2021.

[90] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Alexander Mraz, Takehiro Kashiyama, and Yoshihide Sekimoto. Transfer learning-based road damage detection for multiple countries. *CoRR*, abs/2008.13101, 2020.

[91] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto. Global road damage detection: State-of-the-art solutions. *CoRR*, abs/2011.08740, 2020.

[92] Hamed Majidifard, Peng Jin, Yaw Adu-Gyamfi, and William G. Buttlar. Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. *Transportation Research Record: Journal of the Transportation Research Board*, 2674(2):328–339, Feb 2020.

[93] Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11196–11205, 2019.

[94] Xiaoming Lv, Fajie Duan, Jia-jia Jiang, Xiao Fu, and Lin Gan. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors*, 20(6), 2020.

[95] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[96] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[97] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[98] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, page 313–318, New York, NY, USA, 2003. Association for Computing Machinery.

[99] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[100] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[101] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[102] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *arXiv preprint arXiv:1611.02167*, 2016.

[103] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[104] Donald Kossmann and Bing Liu. Road damage detection and classification challenges. https://cci.drexel.edu/bigdata/bigdata2018/BigDataCupChallenges.html, 2018.

[105] Deeksha Arya, Hiroya Maeda, Sanjay Kumar Ghosh, Durga Toshniwal, Hiroshi Omata, Takehiro Kashiyama, and Yoshihide Sekimoto. Global road damage detection: State-of-the-art solutions. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5533–5539. IEEE, 2020.

[106] Kechen Song and Yunhui Yan. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science*, 285:858–864, 2013.

[107] Ybat. Ybat. https://github.com/drainingsun/ybat.

[108] LabelImg. Labelimg. https://github.com/tzutalin/labelImg.

[109] LabelMe. Labelme. https://github.com/wkentaro/labelme.

[110] VIA. Via. https://www.robots.ox.ac.uk/~vgg/software/via.

[111] VoTT. Vott. https://github.com/microsoft/VoTT#build-and-run-from-source.

[112] PixelAnnotationTool. Pixelannotationtool. https://github.com/abreheret/PixelAnnotationTool.

[113] CVAT. Cvat. https://github.com/openvinotoolkit/cvat.

[114] RectLabel. Rectlabel. https://rectlabel.com/.

[115] Labelbox. Labelbox. https://labelbox.com/product/platform/annotate.

[116] V7 Darwin. V7 darwin. https://www.v7labs.com/.

[117] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

[118] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[119] Yizheng Chen, Jia Liang, Xingyu Gu, Qipeng Zhang, Hanyu Deng, and Shuwei Li. An improved minimal path selection approach with new strategies for pavement crack segmentation. *Measurement*, 184:109877, 2021.

[120] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[121] Paul Voigtlaender, Lishu Luo, Chun Yuan, Yong Jiang, and Bastian Leibe. Reducing the annotation effort for video object segmentation datasets. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3060–3069, January 2021.

[122] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *International Conference on Computer Vision (ICCV)*, 2021.

[123] Mitchell J Hallee, Rebecca K Napolitano, Wesley F Reinhart, and Branko Glisic. Crack detection in images of masonry using cnns. *Sensors*, 21(14):4929, 2021.

[124] Dimitrios Loverdos and Vasilis Sarhosis. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*, 140:104389, 2022.

[125] Amir Rezaie, Radhakrishna Achanta, Michele Godio, and Katrin Beyer. Comparison of crack segmentation using digital image correlation measurements and deep learning. *Construction and Building Materials*, 261:120474, 2020.

[126] Eric Bianchi. *COCO-Bridge: Common Objects in Context Dataset and Benchmark for Structural Detail Detection of Bridges*. PhD thesis, Virginia Tech, 2019.

[127] Kangcheng Liu, Yanbin Qu, Hak-Man Kim, and Huihui Song. Avoiding frequency second dip in power unreserved control during wind power rotational speed recovery. *IEEE transactions on power systems*, 33(3):3097–3106, 2017.

[128] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

[129] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[130] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021.

[131] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[132] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[133] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[134] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[135] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 6848–6856, 2018.

[136] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[137] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[138] Qin Zou, Zheng Zhang, Qingquan Li, Xianbiao Qi, Qian Wang, and Song Wang. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3):1498–1512, 2018.

[139] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4151–4160, 2017.

[140] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[141] Kangcheng Liu, Xiaodong Han, and Ben M Chen. Deep learning based automatic crack detection and segmentation for unmanned aerial vehicle inspections. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 381–387. IEEE, 2019.

[142] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[143] Weiwei Sun and Ruisheng Wang. Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with dsm. *IEEE Geoscience and Remote Sensing Letters*, 15(3):474–478, 2018.

[144] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[145] Zhenyu Zhang, Shouwei Gao, and Zheng Huang. An automatic glioma segmentation system using a multilevel attention pyramid scene parsing network. *Current Medical Imaging*, 17(6):751–761, 2021.

[146] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable

convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[147] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[148] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[149] Kangcheng Liu, Zhi Gao, Feng Lin, and Ben M Chen. FG-Net: Fast large-scale lidar point cloudsunderstanding network leveraging correlatedfeature mining and geometric-aware modelling. *arXiv preprint arXiv:2012.09439*, 2020.

[150] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[151] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2524–2534, 2021.

[152] Lixiang Ru, Yibing Zhan, Baosheng Yu, and Bo Du. Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16846–16855, 2022.

[153] Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4248–4257, 2022.

[154] Hongyu Zhou, Zheng Ge, Songtao Liu, Weixin Mao, Zeming Li, Haiyan Yu, and Jian Sun. Dense teacher: Dense pseudo-labels for semi-supervised object detection. *arXiv preprint arXiv:2207.02541*, 2022.

[155] Gang Li, Xiang Li, Yujie Wang, Yichao Wu, Ding Liang, and Shanshan Zhang. DTG-SSOD: Dense teacher guidance for semi-supervised object detection. *arXiv preprint arXiv:2207.05536*, 2022.

[156] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

[157] Kangcheng Liu, Yuzhi Zhao, Zhi Gao, and Ben M Chen. WeakLabel3D-Net: A complete framework for real-scene lidar point clouds weakly supervised multi-tasks understanding. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5108–5115. IEEE, 2022.

[158] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[159] Monica Santamaria Ariza, Ivan Zambon, Hélder S. Sousa, Jose Antonio Campos e Matos, and Alfred Strauss. Comparison of forecasting models to predict concrete bridge decks performance. *Structural Concrete*, 21(4):1240–1253, 2020.

[160] Shouyan Jiang, Linxin Zhao, and Chengbin Du. Structural deformation prediction model based on extreme learning machine algorithm and particle swarm optimization. *Structural Health Monitoring*, page 14759217211072237, 2022.

[161] Mohammed Alsharqawi, Tarek Zayed, and Saleh Abu Dabous. Integrated condition rating and forecasting method for bridge decks using visual inspection and ground penetrating radar. *Automation in Construction*, 89:135–145, 2018.

[162] Elisabeth Menendez, Juan G Victores, Roberto Montero, Santiago Martínez, and Carlos Balaguer. Tunnel structural inspection and assessment using an autonomous robotic system. *Automation in Construction*, 87:117–126, 2018.

[163] Jacob J Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting. *Journal of Computing in Civil Engineering*, 35(2):04020064, 2021.

[164] Jacob J Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge inspection with aerial robots: Automating the entire pipeline of visual data capture, 3D mapping, defect detection, analysis, and reporting. *Journal of Computing in Civil Engineering*, 35(2):04020064, 2021.

[165] Jun Kang Chow, Kuan-fu Liu, Pin Siang Tan, Zhaoyu Su, Jimmy Wu, Zhaofeng Li, and Yu-Hsing Wang. Automated defect inspection of concrete structures. *Automation in Construction*, 132:103959, 2021.

[166] Kaiwen Chen, Georg Reichard, Abiola Akanmu, and Xin Xu. Geo-registering UAV-captured close-range images to GIS-based spatial model for building façade inspections. *Automation in Construction*, 122:103503, 2021.

[167] Jihan Zhang, Ruoyu Wang, Guidong Yang, Kangcheng Liu, Chuanxiang Gao, Yu Zhai, Xi Chen, and Ben M. Chen. Sim-in-real: Digital twin based uav inspection process,. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 784–801. IEEE, 2022.

August 28, 2022

Regarding the comments from editors and reviewers on our paper entitled by

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

We would like to provide our responses, as follows:
All changes/additions are highlighted in blue in the revised manuscript.

Reviewers' comments:

(The author should review the suggested papers by reviewers and may include references to them if they find them helpful. It is not mandatory to do so.)

There are two minor corrections still to be made:

1- abbreviation "SDNET" refers to Structural Defects NET and not to an author's name

2- it should be noted that the accuracy metric can be significantly misleading if there's a class imbalance

We sincerely appreciate your time and comments very much.

For the suggested papers by reviewers, we found these papers are helpful to the review of the methods in our paper. We have systematically reviewed the suggested papers in the first two paragraphs of Section 4 on Page 12, these papers are very useful to further improve our manuscript. We sincerely appreciate the suggestions from reviewers again.

1. For the abbreviation mistake, we have fixed it.

2. For the task of crack classification, we follow the ImageNet large-scale visual recognition dataset [1] to define the metric accuracy. We have also formulated it in our manuscript according to the suggestions from reviewers. In our training dataset, the crack and non-crack images are balanced. After we divide the training set into 120×100 sub-images for training, the crack images compose 41.3%, while the non-crack ones compose 58.7%. Note that in the classification task of more than two thousand images, the indeterminacy caused by the class imbalance can be neglected if the training set is not highly imbalanced. We directly follow DeepCrack [2] to define the evaluation metrics for the average precision in crack segmentation.

[1] Russakovsky, Olga, et al. "ImageNet large scale visual recognition challenge." International journal of computer vision 115.3 (2015): 211-252.
[2] Zou, Qin, et al. "DeepCrack: Learning hierarchical convolutional features for crack detection." IEEE Transactions on Image Processing 28.3 (2018): 1498-1512.

To emphasize the problem of class imbalance, we further add an illustration of the class proportion in our dataset and offers several techniques to tackle the class imbalance emerging in the image classification. The added illustration can be found in the last paragraph of Section 4.2.1 on Page 14:

It is noteworthy that the accuracy metric can be misleading if there exists a class imbalance in the dataset. In our training dataset for crack classification, crack and non-crack images are balanced with the proportion of 41.3% and 58.7%, respectively. In the image classification task of more than 2,000 images, the indeterminacy caused by the class imbalance can be neglected if the training set is not extremely imbalanced. Various techniques [129] can be adopted encountering class imbalance, such as undersampling, oversampling, merging similar classes, and data augmentation.

[129] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. Learning from imbalanced data sets, volume 10. Springer, 2018.

We appreciate sincerely again for your time and comments.

Prepared by:

Guidong Yang, Kangcheng Liu, Jihan Zhang, Benyun Zhao, Zuoquan Zhao, Xi Chen, Ben M. Chen

Highlights

- Review of the datasets for deep learning-based visual defect inspection
- Comparison of the algorithms for defect classification, segmentation and detection
- Proposed deep learning-based network architectures for defect inspection
- Suggestions on developing defect datasets and defect inspection algorithms

CRediT author statement

Guidong Yang: Conceptualization, Investigation, Formal analysis, Writing - Original Draft. **Kangcheng Liu**: Conceptualization, Investigation, Methodology, Formal analysis, Writing - Original Draft. **Jihan Zhang**: Investigation, Formal analysis, Writing - Original Draft. **Benyun Zhao**: Investigation, Formal analysis, Writing - Original Draft. **Zuoquan Zhao**: Investigation, Formal analysis, Writing - Original Draft. **Xi Chen**: Conceptualization, Resources, Supervision, Writing - Review & Editing, Project administration. **Ben M. Chen**: Conceptualization, Funding acquisition, Resources, Supervision, Writing - Review & Editing, Project administration.

Datasets and Processing Methods for Boosting Visual Inspection of Civil Infrastructure: A Comprehensive Review and Algorithm Comparison for Crack Classification, Segmentation, and Detection

Guidong Yang, Kangcheng Liu*, Jihan Zhang, Benyun Zhao, Zuoquan Zhao, Xi Chen* and Ben M. Chen

Department of Mechanical and Automation Engineering,
The Chinese University of Hong Kong, Shatin, NT, Hong Kong, China

* Corresponding author e-mail: kcliu@mae.cuhk.edu.hk; xichen002@cuhk.edu.hk

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: