Engineering Failure Analysis 149 (2023) 107237

Contents lists available at ScienceDirect

Engineering Failure Analysis

journal homepage: www.elsevier.com/locate/engfailanal

Physical interpretation of machine learning-based recognition of defects for the risk management of existing bridge heritage

Angelo Cardellicchio [a], Sergio Ruggieri [b], Andrea Nettis [b], Vito Renò [a], Giuseppina Uva [b,*]

[a] Institute for Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA), National Research Council of Italy, Via Amendola, 122 D/O, Bari, 70126, Italy
[b] DICATECH Department, Polytechnic University of Bari, Via Orabona, 4, Bari, 70126, Italy

ARTICLE INFO

Keywords:
Civil engineering
Existing RC bridges
Reinforced concrete
Defect detection
Machine-learning
Degradation
Material defect

ABSTRACT

The challenge of the research work presented in the paper is to combine the growing interest in monitoring the health condition of existing bridge heritage through systematic and periodic visual inspections with automated recognition of typical bridge defects, which can greatly facilitate the assessment of defect evolution over time. The study focused on the automated identification of defects in existing Reinforced Concrete (RC) bridges exploiting different Deep Learning (DL) approaches and techniques to interpret the obtained predictions. Ensuring the safety of infrastructures is typically a technical and economic issue. Still, in the case of the engineering infrastructure heritage, there are existing bridges and viaducts with a high historical, cultural, and symbolic value. For them, accurate knowledge and characterization of possible degradation processes become particularly important in order to define intervention strategies that combine safety and conservation requirements. With the aim to develop systematic and non-invasive investigation protocols for continuous and effective control of defects and their evolution, a database of existing RC bridge defect images was collected, and the most recurrent defect typologies were classified by domain experts. Some existing Convolutional Neural Networks (CNNs) algorithms were applied to the dataset for automatically recognizing all defects, but the specific novel contribution of the research work is the interpretation of the obtained results in a form that is humanly explainable and directly implementable in new tools for bridge inspections. To interpret the results, Class Activation Maps (CAMs) approaches were employed within available eXplainable Artificial Intelligence (XAI) techniques, which allow to observe the activation zones and nearly perfectly highlight the type of specific defect in a given image. The obtained results, besides suggesting which network works better than others and if the specific defect is effectively recognized, have been evaluated through a quasi-quantitative procedure that compared a qualitative assessment of the CNNs models reliability with two novel indexes representing new explaining metrics of the obtained results. In the end, the outcomes of the proposed study were observed also in a real-life case study. The proposed discussion opens new scenarios in the application of these techniques for supporting road management companies and public organizations in the evaluation of the road networks health state.

* Corresponding author.
E-mail address: giuseppina.uva@poliba.it (G. Uva).

https://doi.org/10.1016/j.engfailanal.2023.107237
Received 11 March 2023; Received in revised form 27 March 2023; Accepted 1 April 2023
Available online 5 April 2023

(a) (b) (c)

Fig. 1. Examples of masonry Roman bridges: (a) Bridge of Augustus, (b) Milvio Bridge, (c) Bridge of Emilius.

1. Introduction and motivations of the study

After several recent disastrous events, public authorities and private owners have focused on the management of the structural safety of the existing built heritage. For example, taking into account existing buildings, several cases could be mentioned, for which the occurrence of unexpected events as earthquakes highlights the high vulnerability of the built heritage, as stated in several studies proposed by the existing scientific literature [1,2]. Analogously, in the context of transportation, managers must ensure the structural safety of bridges and viaducts, which represent critical components of existing networks. In this case, the losses related to hazardous events can become particularly severe considering economic and technical aspects. The issue of safety management of existing bridges, therefore, should be performed not only with regard to service loads (e.g., [3,4]) but also considering natural hazards such as earthquakes (e.g., [5,6]), floods (e.g., [7]), subsidence and landslides (e.g., [8,9]).

In some cases, besides the aforementioned aspects, there is also another kind of intangible loss that should be considered. There are numerous bridges and viaducts, both ancient and modern, whose importance is rather related to their historical, cultural, and symbolic value, and safety requirements, here, go hand in hand with conservation. In Italy, for example, when speaking of historical infrastructures, masonry bridges immediately come to mind and particularly Roman bridges, which formed a consistent part of the road network built within the Roman Empire. In order to give an idea of the amount of works dating back to Roman times (for more information, see [10]), Roman roads extended for about 80.000 km under the Diocletian Empire, where some very important bridges can be found, such as the ”Bridge of Augustus”, the ”Milvio Bridge” or the ”Bridge of Emilius”, all built around 200 years Before Christ and characterized by stone piers and arches (see Fig. 1). They are today considered monuments and are object of great attention from public institutions and the scientific community.

In the last decades, several research studies have investigated the behavior of masonry bridges through different approaches, such as limit analysis [11] or finite element modeling [12]. It is not only monumental masonry bridges that form an important part of the infrastructure heritage, but also many historical RC bridges are of significant importance, such as the RC bridges and viaducts designed and built by glorious engineers who marked the history of structural engineering. For example, in Italy, we can remember Riccardo Morandi or Sergio Musmeci, who left us one-of-a-kind works, built more than 50 years ago and using innovative techniques for the time of construction. Going further back in time, we cannot forget the technologies developed for bridge construction, such as the Hennebique, widely applied in Italy by the Porcheddu Company from the early 1900s [13]. A significant expression of the Hennebique technique can be found in the authors’ city of work, namely the ”Viaduct of Corso Italia”, shown in Fig. 2. Built in 1915, it was one of the first European railway bridges built with the Hennebique technique, showing a high level of technological and formal excellence in the field of concrete construction. This Viaduct can be considered a true work of art, showing undeniable elegance and simplicity, besides embodying great historical and cultural value. Obviously, like other infrastructural works, also this bridge has suffered from several decay phenomena over the years (e.g., concrete degradation and steel corrosion), which led the management company to abandon the entire rail branch. The example of this bridge, as well as other existing cases, highlights the necessity of urgent maintenance and rehabilitation actions for heritage bridges, both masonry and RC ones, as well as the necessity to develop systematic and non-invasive investigation protocols to carry out effective controls of degradation and its evolution.

Standardized procedures are necessary for routing road management companies to develop reliable risk mitigation plans. With this regard, it is worth mentioning the risk-mitigation model developed in Italy after the well-known collapse of the Polcevera Viaduct (or Morandi bridge, [14,15]). Particularly, the Italian Ministry of Transportation, in collaboration with the scientific community, developed specific guidelines for managing the structural safety of existing bridges [16]. The new guidelines propose a multi-level safety evaluation procedure to be applied on the entire national existing bridge stock. This framework, similarly with respect to strategies applied in other countries [17,18], requires periodic onsite surveys to evaluate the health condition of a given bridge considering degradation effects induced by environmental aggressive agents or relevant load condition experienced by the bridge. The survey is based on detailed visual inspections of all the structural components (i.e., decks, girders, piers, bearing devices, and abutments). These inspections are performed by a team of trained surveyors that take note of the observed defects and degradation signs, evaluating the corresponding intensity and extension, by means of specific forms and photographs. The guidelines [16] proposed a list of defects that should be checked for each structural component of the bridge. Classification, indeed, depends on the specific training and on-field experience of the surveyor. Other variables, such as weather and light conditions, the distance of the surveyor from the inspected element, can influence the score assigned to a defect. Furthermore, potential

Fig. 2. An example of historical RC bridge made with Hennebique technique: Viaduct of Corso Italia, Bari, Italy.

loss of attention, e.g., tiredness, can induce misjudgments. The reliability of the visual inspection can be also affected by the actual accessibility to some specific parts of the bridge (e.g., bearing supports), which can be unreachable or hidden by natural or artificial obstacles. In this case, unmanned aerial vehicles (i.e., drones) equipped with appropriate sensors may be used to reach inaccessible bridge components [19] and collect defect images that can be later classified. The use of such technologies presents several advantages, such as (i) reducing inspection time and traffic disruption; (ii) increasing the safety of operators.

Motivated by these reasons, there has been a great effort towards the development of new methodologies and tools to support road management companies and improve the defects survey techniques for better safeguarding the heritage infrastructure asset. A very interesting option is represented by the growth of novel technologies in the field of computer vision, which exploits Machine-Learning (ML) and Deep-Learning (DL) algorithms for identifying specific objects in images with a certain reliability degree. Within this framework, the paper presents a tool for automatically recognizing defects on existing reinforced concrete (RC) bridges, given a database of real defects. The proposed methodology is based on two main steps: (a) training, test, and validation of some Convolutional Neural Networks (CNNs) for a set of typical bridge defects; (b) interpretation of the obtained results through Class Activation Maps (CAMs) methods, within the available eXplainable Artificial Intelligence (XAI) techniques. The final evaluation was performed in two ways: (i) by visually observing the obtained results and assigning a qualitative score to assess the trustworthiness of the employed training models; (ii) by quantifying the results through two novel indexes, capable to provide a sort of explainability measure. The combination of the two interpretations is herein referred to as quasi-quantitative and allows to highlight the pros and cons of automated defect recognition, offering practical suggestions for future applications. In the end, an application on the Viaduct of Corso Italia reported in Fig. 2 is proposed.

2. Background

In Section 2.1, a state-of-the-art about ML method applied to defect detection in bridges is presented, while a theoretical background about XAI methods is reported in Section 2.2, giving an overview of the main field of application of XAI and of the possible extensions to structural engineering.

2.1. Use of ML methods for defect detection on structures

Over the last few years, as discussed within several state-of-the-art reviews on the topic [20,21], ML has contaminated several application fields of structural engineering, such as earthquake engineering, structural health monitoring, damage identification and detection, and structural design. Models such as generalized linear models and neural networks have been widely used to predict different outputs, such as performance states for a stock of buildings [22] and building-specific structural responses under specific loads [23].

Generally, a recurrent pipeline is followed by most approaches: (1) exploratory data analysis is performed once data are gathered through simulations or experiments; (2) one or more ML models are trained on such data; (3) the best one is used for prediction and inference. Other ML methods run side-by-side with structural engineering, proposing an interdisciplinary approach that entails topics such as Human–Machine Interaction and Computer Vision. Regarding the latter, some applications can be mentioned where the aim is to gather data from images, which are the most accessible and informative data sources about structures. For example, some of the authors of this paper have already developed an experience with these methods (see Ruggieri et al. [24]), proposing an ML-based tool named VULMA for the automatic extraction of data from images and generation of a simplified seismic vulnerability index for existing buildings. In detail, the tool is made by four sub-modules performing different tasks: download of georeferenced photos to create a proper dataset (for more information, see [25]); labeling by domain experts; training and test of several CNNs models; definition of a vulnerability index based on the identified structural features.

Researchers have proposed several methods to automatically detect and classify damage types from images. Earlier approaches were based on standard image processing algorithms, such as image registration, morphological image processing, and texture/color analysis via multi-resolution wavelet [26], and were mainly focused on defects related to steel and RC structures [27,28]. An end-to-end framework for cracks prediction based on image rectification was proposed by Yang et al. [29], where authors performed displacement and deformation analysis on an unfolded version of certain regions of interest, identifying cracks as thin as 0.2 pixels on a full-scale RC bridge. Li et al. [30] developed a three-step method for long-distance image acquisition and crack identification. First, the authors extracted existing cracks using image processing techniques, such as clipping, smoothing, rotation, and segmentation. Afterward, a distance algorithm computed the width of each crack, and, finally, image segmentation was performed based on a C-V model. Authors tested their approach on 1000 pictures, achieving a *misclassification rate* of 6.58%. Chen et al. [31] used several image processing techniques along with self-organizing maps, a specific type of unsupervised neural network, to perform crack detection, achieving an accuracy of 89% in terms of crack recognition on a dataset composed by 216 images. Perry et al. [32] used image processing, specifically edge detection, to identify defects on two bridges. Afterward, a 3D representation of the original bridge acquired using LiDAR and photogrammetry was used to localize damages. Image processing has also been applied by devices directly used on the field: as an example, Potenza et al. used in [33] color-based image processing algorithms to perform automatic bridge inspection using Unmanned Aerial Vehicles (UAVs), as for example developed by [19].

With regard to ML methods, the basis for most of the algorithms is template matching. For example, Prasanna et al. [34] proposed STRUM, a three-steps tool for localizing potential crack regions. STRUM consists of a line segment detector based on RANSAC, a spatially-tuned multiple-feature computation tool, and a machine learning classifier. The best results were achieved using random forest, with a top accuracy of 92%. In Abdelkader et al. [35], authors proposed a self-adaptive two-tier method for detecting noises and restoration of bridge defects in images. The overall accuracy achieved by the model was 95.28% on the proposed dataset, outperforming several other ML models. Hoang [36] proposed a data-driven method for automatically detecting pitting corrosion in bridges. Specifically, the authors combined LSHADE and SVM, achieving a classification accuracy of 91.8% on the proposed dataset. ML techniques have also been used on non-strictly visual data. As an example, Montaggioli et al. [37], used the thermal gradient between damaged and undamaged parts of structural elements in bridges to feed an ML classifier with the aim of recognizing potential sub-superficial damages.

Deep neural networks working on image data, such as CNNs, have been often used in crack detection and damage segmentation. To overcome the limitations of available datasets, transfer learning, and fine-tuning are used. In [38], the authors proposed the use of transfer learning on a CNN based on the InceptionV3 architecture, achieving an overall accuracy of 97.8% on a small dataset containing 1180 images representing six types of different defects. In [39], the authors proposed the use of transfer learning, performing a complete comparison of transfer learning and fine-tuning on several available architectures on a dataset gathered by real-world inspections, achieving a top accuracy of about 94% through InceptionV3. Finally, to highlight the constraints and the advantages of the use of transfer learning, in [40], the authors proposed a comprehensive evaluation of in-domain and cross-domain transfer learning for damage detection in bridges, comparing the results achieved on six different open datasets. Another approach was proposed in [41], where a three-stage classifier was used to identify concrete defects in a dataset composed by both images gathered by authors and publicly available datasets. With this hierarchical approach, authors claimed an overall 83.5% F1 score. CNNs for crack detection are also trained from scratch. In [42], the authors used a CNN on a dataset composed of more than 40.000 images achieving an accuracy of about 98%. In [43], the authors proposed a multi-scale feature fusion, using binary classification to identify cracks in a dataset composed by 67.200 images and achieving an F1-Score above 52.0% in the best-case scenario. Another network configuration is proposed by Yang in [44], where a fully convolutional network was used to perform crack segmentation and identification at a pixel level, estimating features as width and length. The authors claimed a maximum accuracy of 97.96%, despite a relatively low F1 score of 79.95%, probably due to data imbalance. Another class of DL models is the one used for object identification. These models are often based on U-shaped networks or single-stage/two-stage detectors. As an example of the first type of network, Deng et al. [45] proposed a bridge damage detection algorithm based on a module named *atrous spatial pyramid pooling*, tested on a dataset composed by 732 images containing two defect classes and non-damaged samples. Their proposed model, named LinkASPP, is able to slightly outperform UNet-based models, achieving an overall F1 score of 93.46% on non-defective samples, and 65% on average on defective samples. As for two-stage detectors, Cha et al. [46] used Faster R-CNN to identify five types of structural surface damages on concrete and steel surfaces, achieving a mean average precision of 87.8% on a dataset composed of 2.355 images. A modified version of Faster R-CNN was proposed by Li et al. [47] to identify three types of concrete defects. While two-stage detectors usually achieve good accuracy, they employ an additional computational overhead, severely undermining their real-world applicability. To overcome this issue, single-stage detectors [48] and YOLO [49] were proposed, and used in some approaches, such as the one proposed by Maeda et al. [50] to detect damages on a dataset composed by 9.053 images of road damage captured with a smartphone on a car, and 15.435 instances of road damage, achieving a maximum accuracy of 95%.

2.2. XAI approach: overview and introduction to structural engineering problems

Artificial intelligence and ML methodologies facilitate the solution of complex engineering problems by tracing accurate relationships between a series of inputs and outputs. Despite the evident advances and the invincible advantages of ML-based systems, there is still a nightmare for analysts due to the black-box nature of the mathematical model to be employed in the problems under investigation. Entrusting specific metrics such as goodness indices of a given solution is possible, the real joke resides behind the physics. In general, one can mathematically establish a near-perfect relationship between some inputs and outputs through the most complex existing numerical model. At the same time, that relationship could not have any physical foundation. With this issue in mind, in the last years several researchers wondered whether a method exists to explain the adopted ML models. This is the fairy tale introducing XAI, as the superhero that can help users to interpret the adopted model, including a guarantee certificate during the learning process, improving system robustness by fighting against an evil called uncertainty and ensuring the inference of the output with the essential features, without distractions. The real challenge is to quantify the advantages and disadvantages of the ML model by estimating the trade-off between performance and explainability [51]. When talking about the explainability of ML approaches, XAI techniques allow explaining (qualitatively or quantitatively) how a specific ML model connects inputs and outputs and, to summarize the possible applications, three different levels can be mentioned: (1) input explainability, with regard to the input dataset; (2) model explainability, to describe ML model by nature; (3) output explainability, concerning to the output and the capacity of the ML model to make correct predictions. Input explainability refers to data processing methods for extracting insights from the input, which can be performed by employing statistical techniques of data management (e.g., data analysis, summary, and transformation). Model explainability allows explaining how the model uses variables to produce predictions.

In general terms, an effective interpretation of the model can be achieved when it presents three features: (a) simulatability, where a simple alternative model can complement the main one; (b) decomposability, where each part of the used model can be intuitively explained; (c) transparency, where the learning algorithm can be explained by means other mathematical methods. Output explainability is perhaps the best way to interpret the result of the ML analysis by using the efficiency of the black-box model through a series of techniques that analysts can use to judge the prediction. Some model-agnostic techniques can be employed for simple ML algorithms, while model-specific approaches are used for complex algorithms, such as CNNs. Analysts can opt for visualization, textual justification, simplification, and feature relevance in both approaches. Visualization and textual justification produce detailed explanations as images or text. Simplification allows providing a reduced-order version of the main model (or a part of it) with similar performance and easier understanding. Feature relevance provides the importance that each model variable covers within the model (e.g., sensitivity).

The scientific community provided several explainability methods, each suited for a specific class of models, including CNNs. An elegant summary of such techniques was provided by [52], which compared a deep visual representation approach on several XAI visualization methods to provide a qualitative measure of the attribution models. Recently popular visualization methods are the activation visualization [53], the heatmaps [54], the class activation methods, such as the gradient-weighted class activation mapping (*GradCAM*) [55], and the saliency maps [56]. All the above visualization methodologies exploit the gradient information for some specific convolutional layers of the network to identify the areas used by the network for the prediction of the target label.

XAI techniques have also found some applications in structural engineering problems. Naser provided in [57] an overview on the use of such techniques in structural engineering on structured data, interpreting the results obtained by using two tree-based algorithms and a deep neural network in the prediction of the maximum capacity of fiber-reinforced polymer strengthened beams and their propensity to collapse by using SHAP and Partial Dependence Plots (PDPs). PDPs are also used by Tapeh and Naser in [58] to investigate the spalling phenomenon in concrete mixture induced by fire from a heuristic point of view, along with feature importance on tree-based algorithms, providing a graphical interpretation of the feature importance in the overall classification. Other applications of the SHAP interpretation method were proposed by Somala et al. [59], which estimated the fundamental vibration period of RC infilled structures through ML methods, providing local and global explanation of the autcomes. Analogously, Mangalathu et al. [60] employed a random forest algorithm for predicting failure modes of RC columns and shear walls, performing a sensitivity analysis by defining SHAP dependencies among key parameters.

Nevertheless, when switching to CNNs for visually identifying defects in existing structures, the GradCAM method was usually employed. A large overview of automated methodologies and techniques of computer vision in the visual inspection of critical infrastructure assets was provided by Spencer et al. [61]. Still, few works explained the adopted CNNs and the input/output data interpretation. Among the few available examples, Bukhsh et al. [40] provided a visual explanation of the employed CNN (VGG16) on the existing dataset through GradCAM, while Bush et al. [62] used VGG16 and MobileNet on a dataset for recognizing corrosion, crack and spalling in structural elements, using GradCAM to assess the performance of the network.

3. Materials and methods

3.1. Dataset

The reference dataset used to characterize RC bridge defects was created by collecting 10.779 labeled images from real surveys on existing RC bridges in Southern Italy and was used to evaluate the effectiveness of transfer learning on bridge defects classification [39]. Although many defect categories affecting RC structural elements of bridges can be identified in manuals such as [16], only seven different defect typologies were considered for this study to maximize the data throughput and, as a consequence, the effectiveness of the investigation. Among the selected typologies, six are related to concrete surfaces and elements, while one is related to the asphalt road surface. In order to define uniquely the assumed defects, a brief description for each one is reported in:

Table 1
Number of images per each defect type contained in the reference dataset.

Defect type	Tag	N. of images
Cracks	C	1134
Corroded steel reinforcements	CRS	2910
Deteriorated concrete	DC	2439
Honeycombs	HC	1159
Moisture spots	MS	2951
Pavement degradation	PD	115
Shrinkage cracks	SC	71

- *Corroded steel reinforcements* refers to steel bars that are exposed and possibly corroded, after spalling of the cover layer.
- *Cracks* refers to generic thin or thick cracks which can be related by degradation phenomena or static deficiencies.
- *Deteriorated concrete* includes superficial degradation of the concrete surface (e.g. swelling, scaling) induced by environmental aggressive conditions.
- *Honeycombs* refers to casting errors leading to non-homogeneous areas with visible aggregates.
- *Moisture spots* accounts for both traces of drainage water incorrectly removed from the deck surface and water infiltration from concrete surfaces
- *Shrinkage (Crazing) cracks* concerns spread thin cracks on concrete due to drying out of moisture during construction phase
- *Pavement degradation* concerns defects (e.g., cracks, holes) affecting the asphalt layer of the road surface.

Further research developments can be oriented to include other defect categories or consider a more refined classification of defects with proper intensity proxies (e.g., exposed steel bars/corroded steel bars). Based on this first classification, in Table 1, the dimension of the dataset are reported, specifying the available number of images. For each image, the authors, as domain experts, performed a labeling work in which the presence of each defect was manually registered. With this regard, it is worth noting that some of the real photos of existing bridge parts can contain multiple defects (e.g., moisture spots and corroded steel bars). In these cases, domain experts selected the main visible defect in the images and labeled it according to one of the items of the above-mentioned list. Moreover, only a single damage box (later named as *red box*) was traced within each image, identifying the main part reporting the specific defect. In some cases, real-life photos can report the same defect in more parts. If more defects can be observed in a single image, the defect labeling was driven by the one that covers the larger part of the image. In this Section, no defect image belonging to the original dataset has been reported, but some can be found later in the text.

3.2. Use of CNNs for features extraction and recognition in images

The automated definition and recognition of defects in existing bridges can be achieved using CNNs. In particular, according to continuous advances in computer vision techniques, it is currently possible to analyze visual imagery and extract relevant features. A comprehensive description of the working principles of CNNs was reported by authors in [24], while herein, a gentle summary of the method was provided. In general, a typical layer in a CNN is ruled by three characters: (i) a set of convolutions, (ii) a nonlinear function, and (iii) a pooling step. Briefly, the convolution, indicated as *(i,j)*, can be mathematically expressed as:

$$(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \tag{1}$$

where I represents the image to be investigated, m and n identify the pixels of the image, and K indicates a kernel (usually bi-dimensional). The nonlinear function is used to model real-world effects, which are inherently non-linear and is applied after the convolution step. Usually, the adopted function is the Rectified Linear Units (ReLU), which can be mathematically expressed as

$$ReLU(x_i) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \tag{2}$$

Where x_i is a generic output value that is accounted as input of the next layer only if it is positive (or at least equal to 0). Finally, the pooling layer allows reducing data dimensions among subsequent layers, exploiting a different representation, i.e., max or average value.

Still, there are some aspects and details which should be considered while training a CNN. First of all, this kind of architecture can be subjected to overfitting, and to overcome it, it is necessary to involve a regularization layer named *dropout*. This type of layer reduces overfitting by training the network on different randomly selected subsets of the original data, as extensively reported in [63]. Another commonly exploited technique to improve the CNNs prediction is the use of *residual blocks* [64], which slightly reduces the effect of the vanishing gradient, which can be observed when several convolutional layers are stacked in an architecture. Another layer that is worth mentioning is the *inception layer* [65], which is a slight modification over classic convolution layers, which tries to learn spatial correlations and cross-channel correlations (that is, the correlation over different pixels or image channels) by employing a stacked set of kernels, leveraging on different dimensional components, such as height, depth, and width of the image.

Some remarks should be provided with regard to the usage of CNNs, which can provide the near-optimal prediction only if two conditions are available: (a) a large dataset, for which a labeling process must be carried out beforehand, and (b) an adequate computational power (for instance, see [66]). Should any of these two conditions not be fully satisfied, as shown in this paper, the combination of two approaches can be employed: (i) transfer learning; (ii) fine-tuning. The first approach exploits the internal structure of a pre-trained CNN, whose low-level layers (that is, the ones near the input) extract generic features, such as shapes and edges, while high-level layers extract features specific to the problem under investigation. Hence, transfer learning aims to train on the limited quantity of data available only in high-level layers, specializing the general-purpose network on the problem under investigation. Fine-tuning is consequent to transfer learning: in this case, the whole network is re-trained on the small dataset with a low learning rate to refine achieved results. In this work, we combine these approaches, mainly due to the relatively small size of the dataset used within our experiments.

3.3. Explaining CNNs prediction: Class Activation Maps

One of the main limitations of ML and DL models, which prevents their extensive use in real-world applications, is the *black box* effect [57]. Specifically, complex and non-linear ML and DL models are often seen as a black box, with limited insight into their internal operation. Consequently, these models are not *interpretable*. Then, it is extremely difficult to understand *how* the model reached a certain conclusion. As such, the lack of interpretability can cause a general lack of *trust* by the user in the results achieved by the model. To overcome this issue, several methodologies can be employed. As for CNNs, the most well-known are the *Class Activation Maps* (CAMs) [55], which provide a visual explanation of the prediction of a CNN by highlighting the parts of the image I that are considered relevant for the final prediction.

The CAM value $M_c(x, y)$ can be expressed as a function of an image of class c and a specific pixel in position (x, y) in the input image. Thus, the following equation can be defined:

$$M_c(x, y) = ReLU\left(\sum_k w_k^c A_k(x, y)\right) \tag{3}$$

where $A_k(x, y)$ represents the activation value of the kth feature map in the last convolutional layer at position (x, y), while w_k^c is the weight associated to the kth feature map and class c. In other words, a high CAM value at position (x, y) results from an average high activation value of the feature maps A_k.

The CAM algorithm requires feature maps to directly precede the final layer of the network, which is in charge of the classification of the single image. To overcome this constraint, the *GradCAM* algorithm [55] proposes a modification of the original CAM using the gradient flowing into the deepest convolutional layer of the CNN to assign importance values to each neuron for a particular decision. Specifically, the GradCAM value for each class c and for the position of pixel (x, y) is computed as

$$M_{Gradcam}^c(x, y) = ReLU\left(\sum_k \alpha_c^k A^k(x, y)\right) \tag{4}$$

where a_k^c represents the *neuron importance weights*.

GradCAM++ [67] is a generalization over GradCAM, aimed at better localizing multiple class instances and capturing more complex objects. To this end, GradCAM++ applies a weighted average for the partial derivatives, covering a wider portion of the original object. This results in a more complex formulation of the neuron importance weights α_c^k, which is expressed as

$$\alpha_k^c = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\delta Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}} ReLU\frac{\partial Y^c}{\partial A_{ij}^K} \tag{5}$$

In this work, both GradCAM and GradCAM++ were used to explain the experimental results achieved by general-purpose CNNs after transfer learning.

3.4. CAMs generation and evaluation

As stated in Section 3.3, CAM-based algorithms highlight which zones of an image are the most relevant for a CNN in providing the final classification results. According to the processing pipeline shown in Fig. 3, activation maps can be used to evaluate whether the model responds to the defect itself or if the surrounding context determines its response. Another interesting possibility is to compare the response of different models via a quantitative index, as described in the next section.

3.5. Quantitative analysis via activation maps

In this section, a quantitative analysis using activation maps is proposed. Let us recall that CAMs are characterized by values that fall within the range $[0, 1]$, each directly proportional to the activation of a specific neuron. If $M^c(x, y) \to 1$, the region (x, y) of the image maximally contributes to the classification results from the model. Consequently, an ideal damage classification network should be influenced mainly by areas containing the defect. Otherwise, the network may observe either side effects caused by the defect or other details that do not directly relate to it.

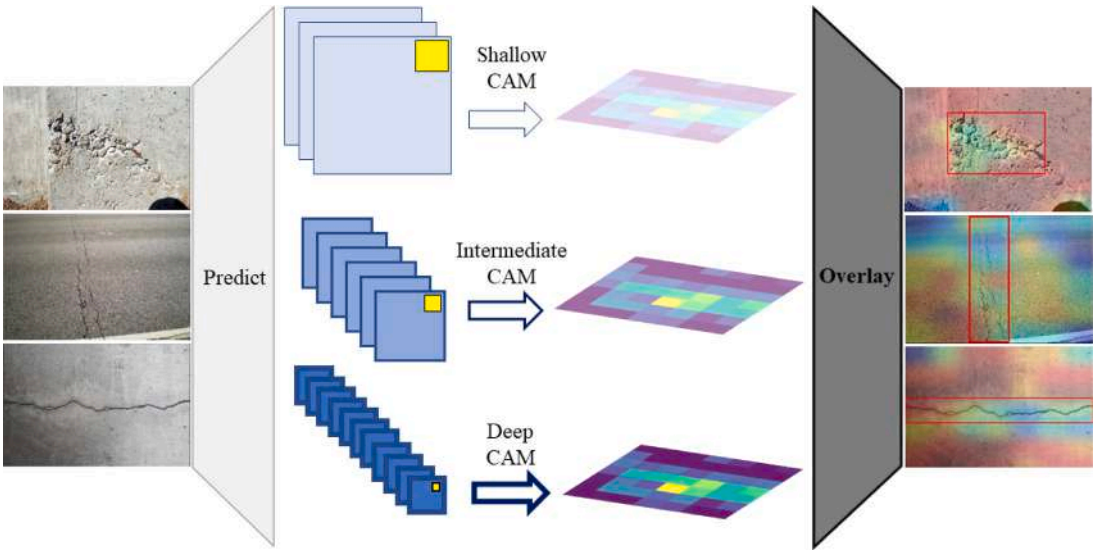

Fig. 3. The processing pipeline used to extract CAMs and obtain an overlapped image.

Hence, this study proposes two metrics to evaluate how much the damage "impacts" on the final result provided by the network. The first metric is defined as

$$I_{ctx} = \frac{Act_{db}}{Act_{tot}} \tag{6}$$

where I_{ctx} is the *context influence index*, given by the ratio between the total number of pixels activated that lie within the damage box, Act_{db}, and the total number of pixels activated in the whole image, Act_{tot}. Hence, if $I_{ctx} \to 0$, most of the activated pixels lie outside the damage box or the activated pixels in the damage box are few, and, therefore, the network mostly looks at the context of the image rather than the damage box. Otherwise, if $I_{ctx} \to 1$, most of the activated pixels lie within the damage box, few pixels are activated in the surrounding context, and the network mainly looks at the damage itself. The second metric is defined as

$$I_{db} = \frac{Act_{db}}{S_{db}} \tag{7}$$

where I_{db} is the *damage box index*, provided by the ratio between Act_{db} and the overall size of the damage box, S_{db}. Therefore, if $I_{db} \to 0$, the model looks at side effects or other details within the image for classification. If $I_{db} \to 1$, the network considers most of the damage relevant for the classification. It is worth noting that there is no direct relationship between the values of I_{db} and I_{ctx}. The network could be entirely relying on a small activated area within the damage box and, in this case, $I_{db} \to 0$ and $I_{ctx} \to 1$. Conversely, if most of the areas of the image are activated, I_{ctx} would be low and $I_{db} \to 1$. Additionally, if most salient pixels are outside the damage box, I_{db} will tend to 0, and I_{ctx} will tend to 0. The following experimental evaluation will account for both indexes, assuming that high values of both will highlight that the network is looking at the part of the image considered relevant by humans to provide its classification result.

3.6. CAM activation threshold

The metrics described in Section 3.5 cannot be directly inferred from the activation value of each pixel. Instead, a piece of binary information is required, that is, whether the pixel is activated or not. Hence, a binarization step using thresholding is used. Specifically, given a threshold value $\sigma \in [0, 1]$, the activation $Act_{(i,j)}$ for the pixel in position (x, y) is provided by

$$Act_{(x,y)} = \begin{cases} 1, & if\ M^c(x,y) > \sigma \\ 0, & otherwise \end{cases} \tag{8}$$

where $M(x, y)$ is the generic CAM value for the pixel (x, y). By increasing the value of σ, $Act_{db} \to 0$, as fewer pixels will be above the threshold and, as a consequence, considered active. Hence, both I_{db} and I_{ctx} will decrease. As the logic behind fixing a threshold value is fuzzy, the experiments were performed using an incremental variation of the activation threshold, with values of $\sigma \in [0.1, \ldots, 0.9]$ with a fixed step of 0.1, to search for hints on the most active parts of the model.

4. Experimental results

A sample of 15 images, shown in Fig. 4, was randomly selected from the initial dataset to describe the experimental results performed on the entire dataset. Specifically, two images were selected per each type of defect, while three images were selected only for moisture spots (assuming that it summarizes the defects of efflorescence, drainage traces, and active humidity).

Fig. 4. Images used for the experimental evaluation. Each image characterizes a specific defect, as reported in Table 1; the defect notation, along with a progressive identifier, is used to identify each image univocally.

4.1. Defects classification via transfer learning

For the proposed approach, the method of transfer learning was used, which allows training a large network with a relatively small number of samples, by exploiting the generalization capability of the shallow layers of the CNN, hence, training data only on the deepest and specific layers. For the purposes of the experimental campaign, eight different network architectures were used, that is, MobileNetV3 [68], both in its Small and Large configurations, EfficientNetV2 [69], both in its B0 and L configurations, DenseNet [70], ResNetV2 [71], Xception [72], and InceptionResNetV2 [73].

Table 2
Accuracy, precision, and recall achieved by different neural network models after training.

Model	Accuracy	Precision	Recall
DenseNet121	35.79%	65.58%	7.17%
EfficientNetV2B0	53.67%	78.68%	32.37%
EfficientNetV2L	35.22%	55.14%	10.88%
InceptionResNetV2	54.14%	70.21%	33.76%
MobileNetV3Large	45.47%	50.36%	37.54%
MobileNetV3Small	63.46%	69.97%	55.42%
ResNet152V2	28.02%	29.82%	1.31%
Xception	55.83%	77.81%	27.05%

The networks were selected for the following reasons:

- **MobileNetV3Small**, **MobileNetV3Large**: are characterized by limited numbers of parameters (2.9 million for the Small configuration, and 5.4 million for the Large configuration). Consequently, these networks can be easily implemented on resource-constrained devices (i.e., smartphones, tablets, or even drones and surveillance cameras), which have the highest probability of being directly deployed on the field.
- **EfficientNetV2 B0**, **EfficientNetV2 L**: are able to provide an efficient scaling method for models such as MobileNet and ResNet, improving the overall accuracy and efficiency of the network. In addition, the first network includes 29 million trainable parameters, while the second one contains 479 million parameters. This aspect allows establishing whether the size of the original networks somehow influences classification performance on such small datasets.
- **DenseNet**: able to evaluate the effectiveness of dense layer-to-layer connections over the proposed dataset.
- **ResNetV2**, **InceptionResNetV2**, **Xception**: able to evaluate the effectiveness of inception and residual blocks (or the combination of both) on the proposed dataset.

The training was performed using a machine equipped with a Core i9-13900HK CPU, 64 GBs of RAM, and an NVIDIA GeForce 3090 GPU with 24 GBs or VRAM. As for training parameters, each session was performed for 50 epochs, using a standard learning rate of 0.001, Adam [74] as optimization algorithm, and cross-entropy as loss function. Furthermore, an early stopping criterion was employed to avoid overfitting, which consisted of stopping the network training once the validation accuracy was not improved over three consecutive epochs. As for training/testing/validation percentages, standard sizes of 60%, 20%, and 20% were adopted, respectively (obviously, different percentages could be considered, depending on the quality and quantity of data, as for example by increasing the training data and decreasing the validation ones). Validation results are shown in Table 2, where the percentage values of accuracy, precision, and recall were provided for each neural network model after the training.

As shown in Table 2, the model with the best overall accuracy was MobileNetV3Small, with a validation accuracy of 63.46%, and a validation recall of 55.42%. As for the precision, the better performing model is EfficientNetV2B0, with a precision of 78.68%. From these outcomes, two considerations arise:

1. Models with lower parameters are capable of achieving better results on problems with challenging domains.
2. Overall performance is mainly affected by low recall.

As for point 1, this is somehow expected, as a larger number of parameters require a larger dataset, especially when the domain of the problem under investigation is far from the original domain. Let us recall that ImageNet is a wide and general-purpose dataset, with 1000 different classes of objects that are visually different from the domain under analysis, as seen in the 15 image samples shown in Fig. 4. Specifically, while objects in ImageNet are clearly defined in terms of color and edge appearances, the damages in the proposed dataset are defined by irregular shapes and colors, which can change according to several factors, such as the severity of the damage and the underlying and surrounding material. This makes the proposed dataset extremely challenging, as shown by the results in Table 2. Consequently, point 2 is a direct consequence of point 1. It can be derived from the consideration that several defects, such as corroded steel reinforcements, deteriorated concrete, and moisture spots, are often related and visually similar. As a consequence, predictions by the CNN can present a high number of mismatches in terms of false negatives, which negatively affect the overall recall of the network.

4.2. Interpretation of results via activation maps

In this Section, a qualitative evaluation of the predictions provided by the CNN using activation maps is described, comparing results achieved using both GradCAM and GradCAM++. As stated in Section 3.3, the activation maps provided by GradCAM and GradCAM++ highlight which zones of the images are the most relevant for the final prediction outcome. These activation maps are different for each type of network. Therefore, a comprehensive evaluation was performed and indexes I_{db} and I_{ctx} were extracted for each image shown in Fig. 4. Both interpretability methods were applied to all the figures of the sample dataset in Fig. 4, and heatmaps were generated through the process graphically displayed in Figs. 5–7. In detail, accounting for two randomly selected defects, $C1$ and $H1$, and using the best prediction model, MobileNetV3Small, Fig. 5 shows the heatmaps for GradCAM (Figs. 5(a) and 5(b)) and GradCAM++ (Figs. 5(c) and 5(d)). The size of the heatmaps is the same for both visualization methods, and strictly

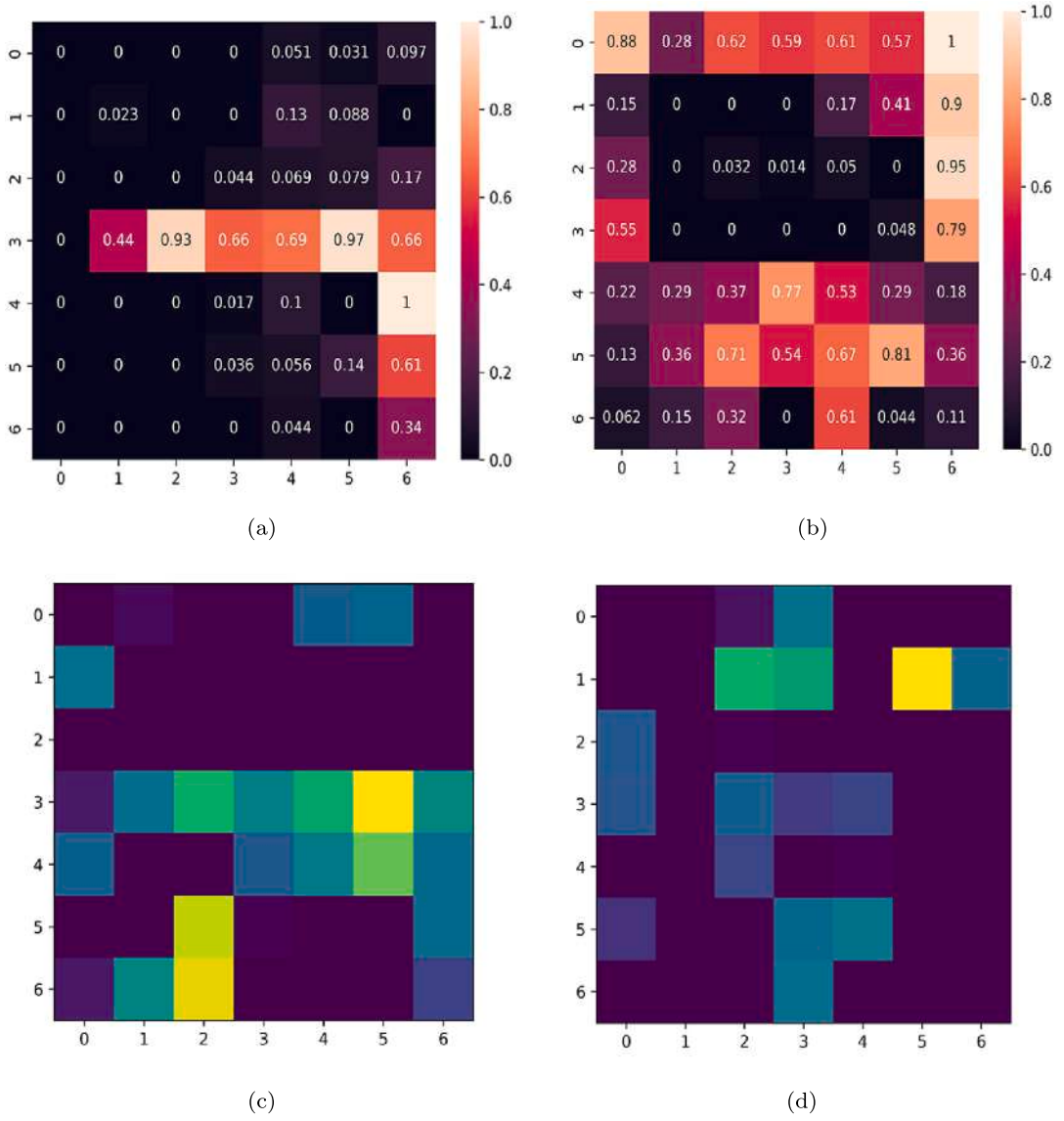

Fig. 5. Heatmap in matrix form for defects (a) $C1$ and (b) $H1$ by using MobileNetV3Small model and GradCAM; Heatmap in matrix form and original colors (to overlap on original images) for defects $C1$ and (b) $H1$ by using MobileNetV3Small model and GradCAM++ (the scale of colors goes from the blue, with activation value equal to 0, to yellow, with activation value equal to 1).

depends on the architecture under analysis. Specifically, if the activation maps are being extracted from a convolutional layer whose input is $n \times n$, the size of the resulting activation map will be $n \times n$. The activation maps are real-valued matrices, where the element in position (i, j) expresses the activation value of the corresponding area of the original image and lies within the range $[0, 1]$. In the proposed representation, the darkest cells were associated with the image parts with low activation values. In contrast, the brightest cells were associated with the parts of the image which mostly activate the network.

An analogous concept is shown in the heatmap activation for GradCAM++, where the colormap ranges from blue (i.e., an activation value close to 0) to yellow (i.e., an activation value close to 1). For visualization purposes, these heatmaps are re-scaled, interpolated, and overlapped on the original image, as shown in Fig. 5. An example of this process is shown in Figs. 6 and 7. The most activated parts of the original images are displayed in blue, while the less activated parts are displayed in red. Note that one may expect that most activated parts should be displayed always in red, but the real coloring output depends on the CNN model employed.

It is worth noting that Figs. 6(c) and 7(c) report the thresholded masks for defects *C1* and *H1*, respectively, for a specific value of the threshold σ (0.5 for the case at hand). Let us recall that these masks are provided by the thresholded pixel activations, as defined in Eq. (8). Consequently, the thresholded mask T_I highlights the parts of the image where the pixel activation values are

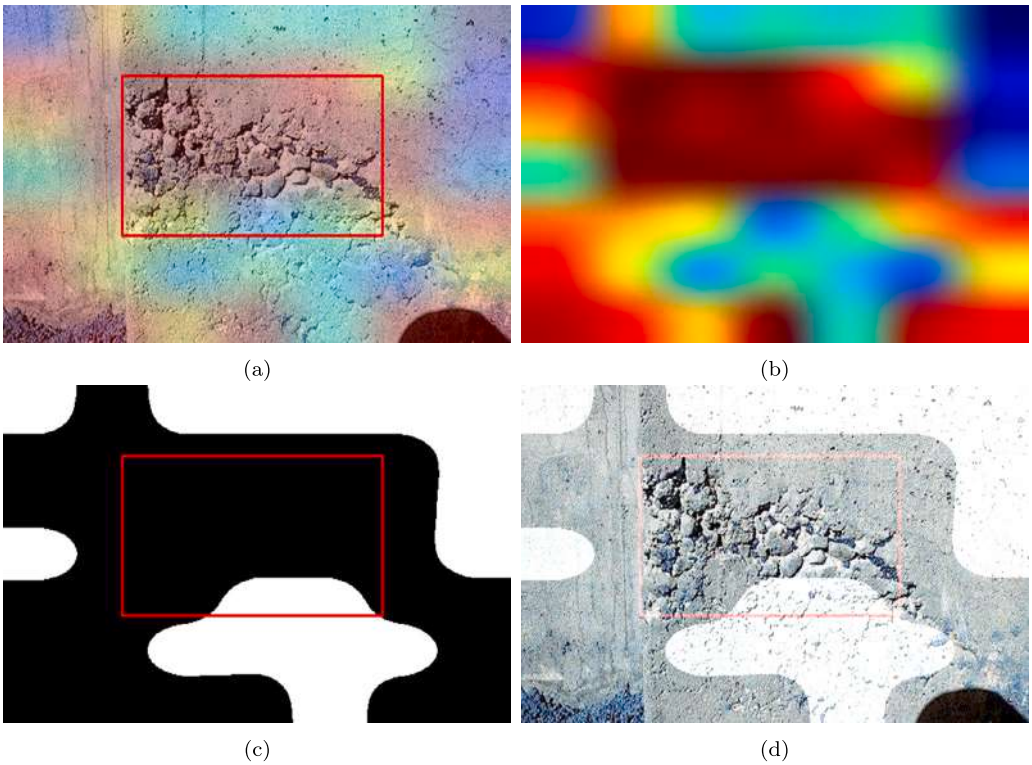

Fig. 6. (a) Overlapped heatmap on the original figure; (b) rescaled heatmap; (c) activation zone through segmented mask; (d) overlapped segmented mask on the original figure; for defect $H1$ by using MobileNetV3Small model and GradCAM for a threshold σ equal to 0.5.

above σ, darkening the remaining parts. The effect of overlapping such thresholded masks on the original images was shown in Figs. 6(d) and 7(d).

Let us underline that while the CAM can be used to explain the behavior of the CNN directly, the threshold σ can be interpreted as the *severity* of a filter applied on the activation map. Therefore, if σ increases, the number of pixels having $M^c > \sigma$ decreases, and the activated zones shrink. The threshold selection assumes a relevant significance only on the human- and quantitative-based defect visualization rather than the behavior of the CNN model itself. By considering the defect *C1* investigated through MobileNetV3Small, in Figs. 8 and 9, the visual effects of the threshold σ (values equal to 0.2, 0.5 and 0.8) increment are shown for GradCAM and GradCAM++ methods, respectively. At this point, two aspects can be denoted. Firstly, the white part of the activation mask sensibly reduced for higher values of σ, and in some cases (Figs. 8(d) and 9(d)), it darkened almost completely. Secondly, GradCAM and GradCAM++ techniques provided different results, with different values of Act_p for the same threshold.

For the other defects shown in Fig. 4, the results of the activation maps provided by the MobileNetV3Small model were shown in Figs. 10 and 11 for GradCAM and GradCAM++, respectively. Red boxes indicated the labels defined by the domain expert, for which CNNs models were trained. The analogous elaborations made through the other employed CNNs models are reported in the Supplementary Material file (indications are provided in Appendix).

To quantify the reliability of the CNNs models according to the presented methodology, the proposed metrics in Eqs. (6) and (7) can be evaluated, regarding to the indexes I_{ctx} and I_{db}, respectively. Particularly, in Figs. 12 and 13, the values of I_{db} and I_{ctx} were shown for defects *C1* and *H1*, respectively. The trend of the two indexes was reported, by varying the value of the threshold σ from 0.1 to 0.9 and accounting for all employed training models (lines of different colors) and for GradCAM and GradCAM++ methods (continuous and dotted lines, respectively). Analogous elaborations for other defects were reported in Appendix, for the sake of brevity.

5. Discussion of the results

5.1. Quasi-quantitative interpretation

From the obtained results, some general outcomes can be provided. Looking at the results in Table 2 and the corresponding visual attentions (e.g., Figs. 10 and 11), the experiments have not been completely satisfactory, since in some cases low values of accuracy and precision have been measured. As a matter of fact, the lower values of precision and accuracy obtained for some networks suggest that the capacity of some CNN models, trained, tested, and validated with the collected database, is not currently

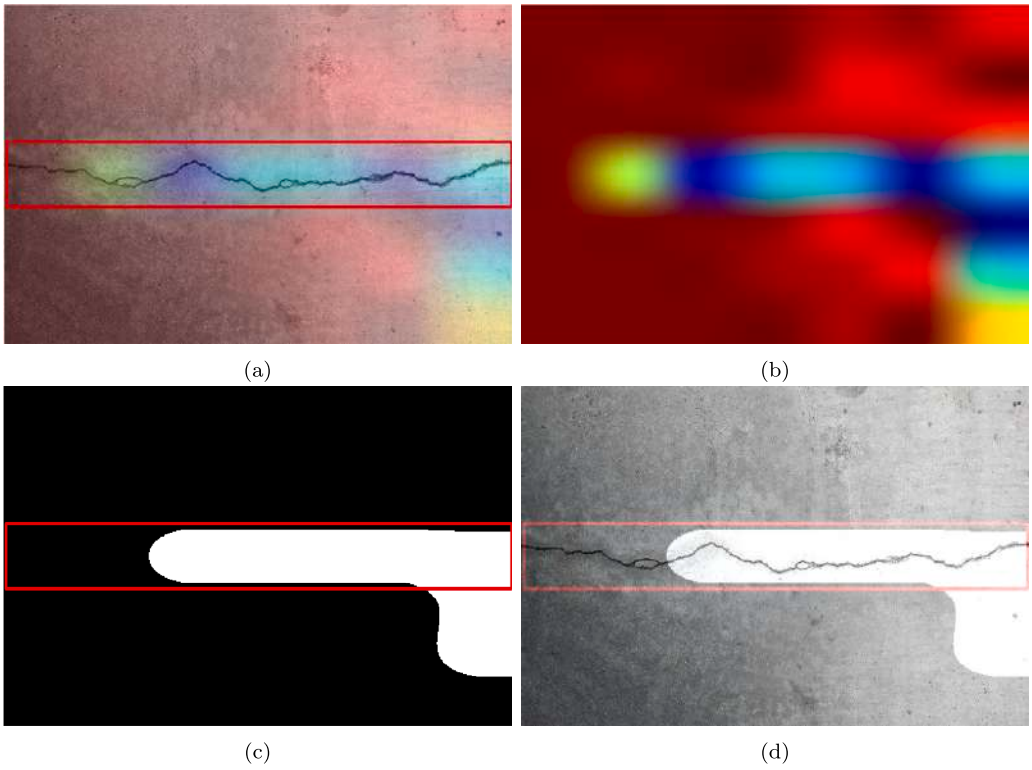

Fig. 7. (a) Overlapped heatmap on the original figure; (b) rescaled heatmap; (c) activation zone through segmented mask; (d) overlapped segmented mask on the original figure; for defect $C1$ by using MobileNetV3Small model and GradCAM for a threshold σ equal to 0.5.

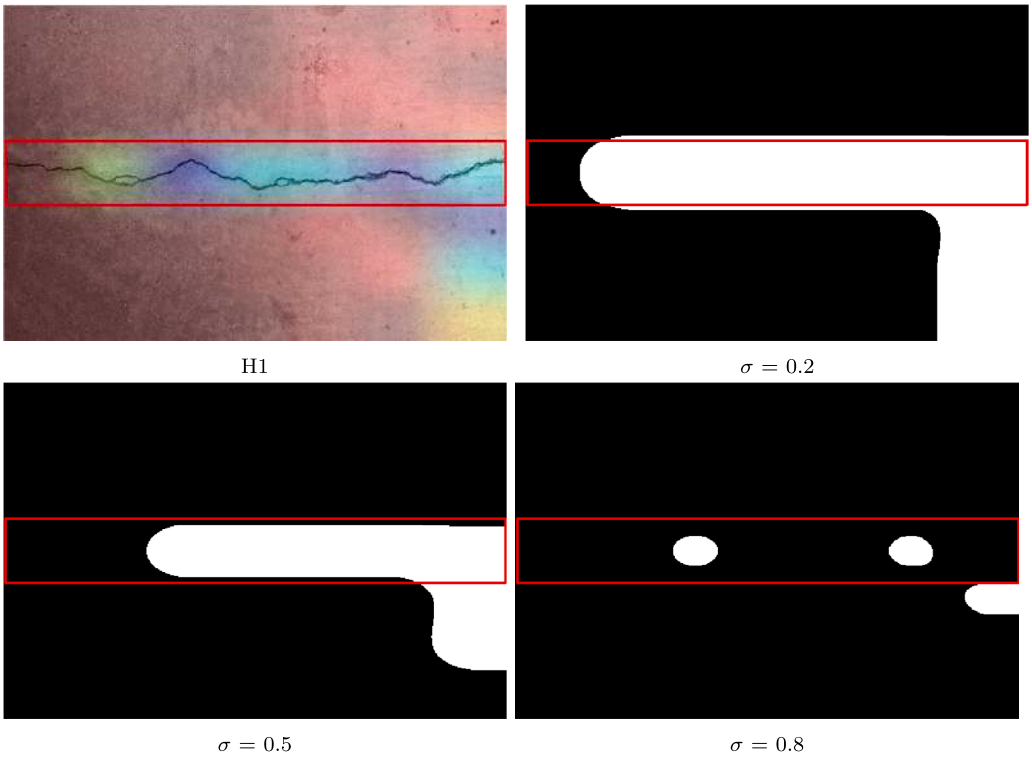

Fig. 8. Activation zones through (a) overlapped heatmap on the original figure and segmented masks using MobileNetV3Small model and GradCAM for value of σ equal to 0.2 (b), 0.5 (c), and 0.8 (d).

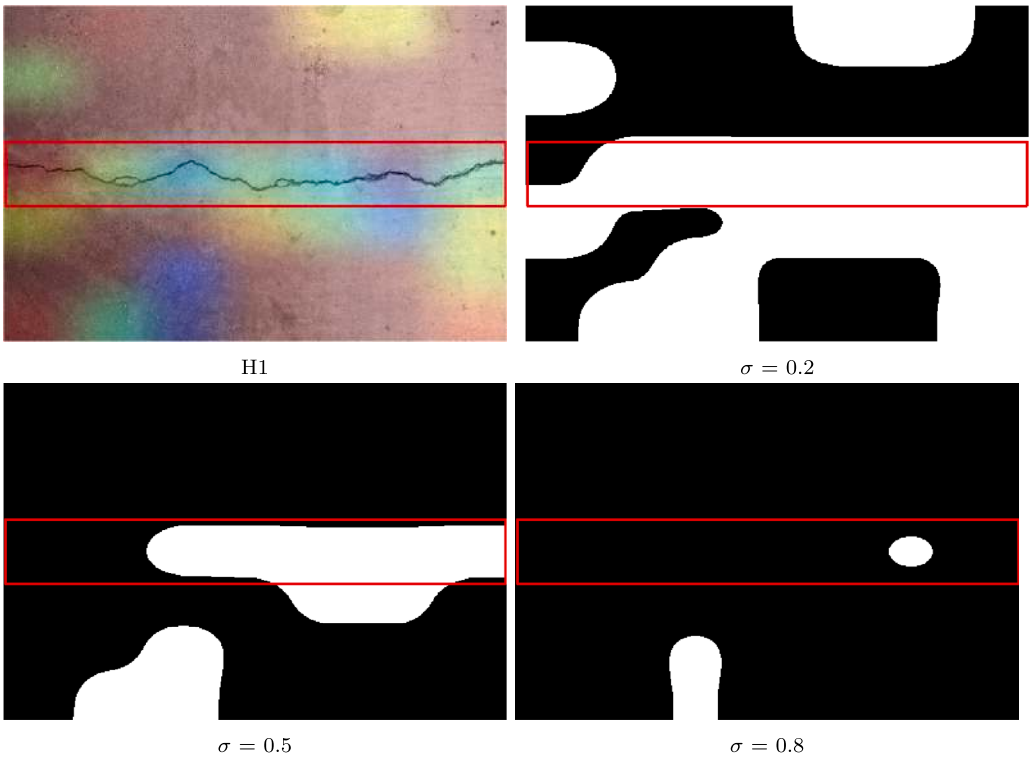

Fig. 9. Activation zones through (a) overlapped heatmap on the original figure and segmented masks using MobileNetV3Small model and GradCAM++ for values of σ equal to 0.2 (b), 0.5 (c), and 0.8 (d).

ready to be used for the purpose of automatic defect recognition. On the other hand, it is worth mentioning two aspects: (a) some improvements can be introduced in the future developments of the work (as later stated for the used database); (b) the aim of the paper is not only to understand which network works better but also to provide a new explaining method to assess the reliability of the predictions. It has been shown that a quasi-quantitative interpretation can be provided, aiming at physically explaining the reliability of the employed prediction models. With this regard, it is worth noting that the provided comments were defined as *quasi-quantitative*, but additional insights can be deduced from the quantification of I_{db} and I_{ctx}. The main conclusions on the CNNs efficiency can be derived qualitatively, that is, by observing the figures containing defects and by visually comparing the activation zones provided by the CNNs for the red boxes drawn from the expert domain. With this goal in mind, given a specific defect, a qualitative score was assigned to each image, as following indicated:

- **G**, indicating a good or near-good matching between the activation zone and the red box (i.e., all the activated zones in the image are in the red box, and very few image portions outside the damage box are activated on the considered defect).
- **M-G**, indicating an average satisfactory matching between the activation zone and the red box (i.e., the activated zones partially fill the red box in the image, and few image portions outside the damage box are activated on the considered defect).
- **M-B**, indicating a near-average satisfactory matching between the activation zone and the red box (i.e., the activated zones are partially within the red box and partially outside of it, and these latter are not representing the considered defect).
- **B**, indicating a bad matching between the activation zone and the red box (i.e., all the activated zones in the image are out of the red box, or no activation is observed).

A summary of the qualitative evaluation is reported in Tables 3 and 4. The above scores were assigned to each defect by observing results obtained by each CNN model and each CAM model. The first observation deduced from the qualitative analysis is about the prediction capacity of each CNNs concerning the specific defects.

In Tables 3 and 4, more negative judgments (B and MB) than positive ones (G and MG) can be observed, and this is due to several reasons, among which a relatively small dataset for the CNNs models used (probably slightly biased and unbalanced with respect to the different defect typologies); the existence of different defects in every single image (this could provide errors in the specific defect recognition); the presence of a single damage box in each image, and probably also the assignment of restrictive qualitative criteria. Looking in detail at the results, the best predictions were obtained for the cracks, honeycombs, and moisture spot defects, and acceptable predictions were obtained for corroded steel reinforcements and deteriorated concrete defects. Instead, unsatisfactory predictions were observed for pavement degradation and shrinkage crack defects. Observing the CNNs models, the results shown in

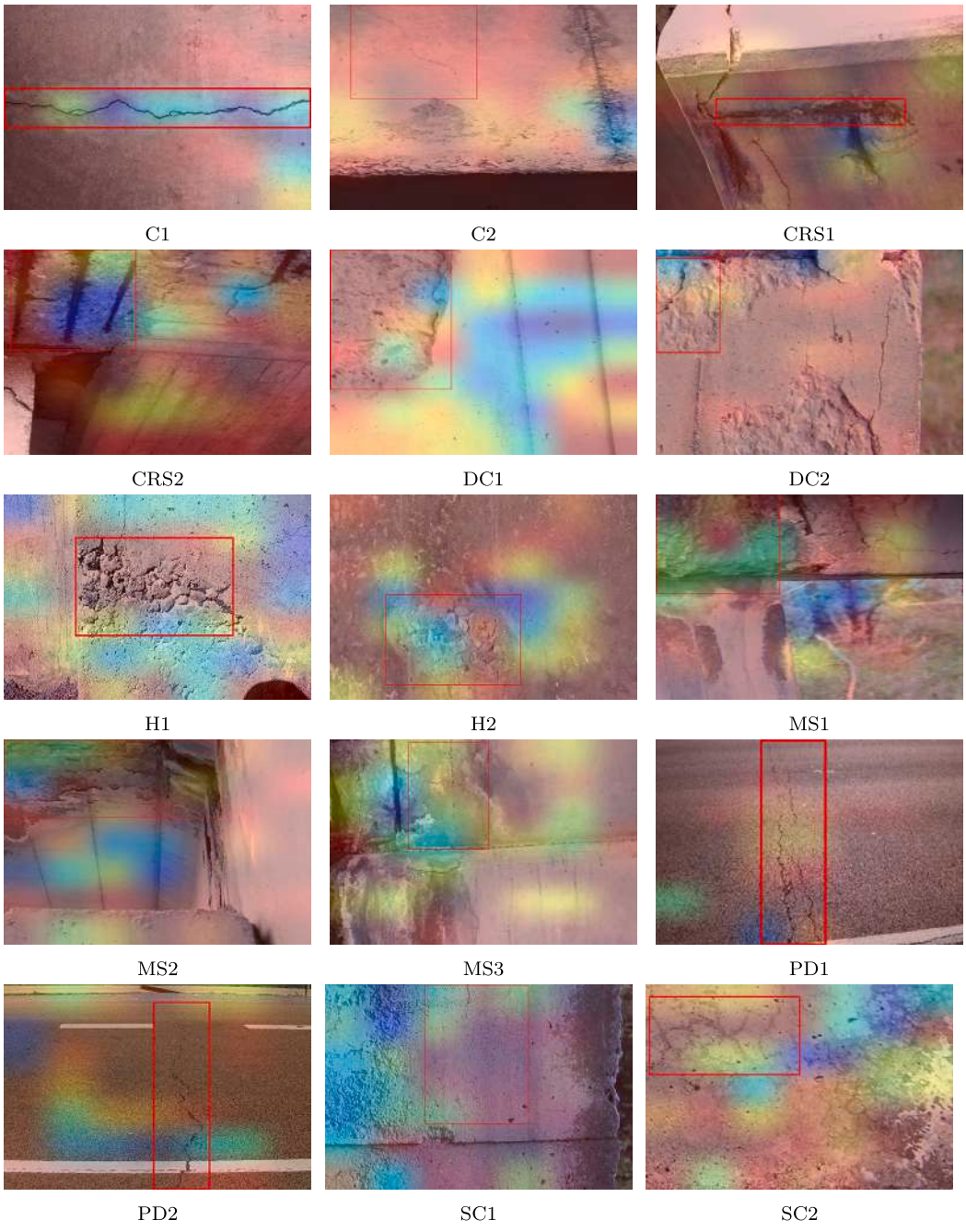

Fig. 10. Sample of images from the dataset with the overlapped GradCAM feature map activation by using MobileNetV3Small model.

Table 2 were not always adequately confirmed, with a good prediction capacity for some specific defects and bad performance for others. For example, the best model, MobileNetV3Small, has predicted in a good way cracks, honeycombs, and moisture spot defects, but not other defects. Models EfficientNetV2B0, MobileNetV3Large, and Xception have shown a good prediction for honeycombs and moisture spot defects. DenseNet121 has provided some good chances for honeycombs, while InceptionResNetV2 has given good feedback for cracks. As expected, EfficientNetV2L and ResNet152V2 have provided a bad prediction capacity for defects.

Interesting observations can be derived from applying either GradCAM or GradCAM++ approaches. In particular, we would expect GradCAM++ to be a better gradient-based visualization technique than GradCAM, since it is aimed to localize multiple class features (as for the case at hand). Observing the results in Tables 3 and 4, this was true in some cases (e.g., cracks in EfficientNetV2B0) and false in other ones (e.g., cracks in InceptionResNetV2). In other words, the improvement given by

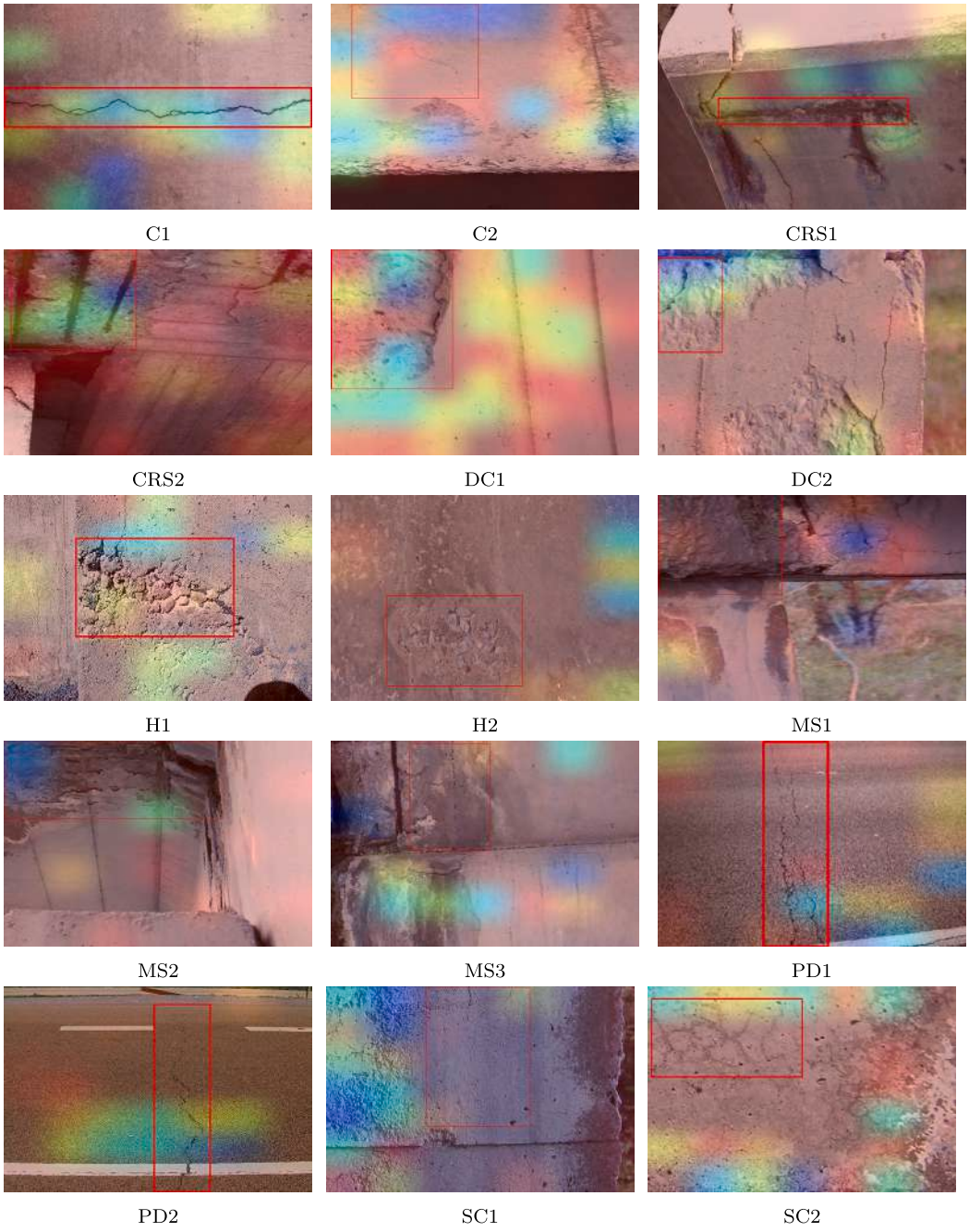

Fig. 11. Sample of images from the dataset with the overlapped GradCAM++ feature map activation by using MobileNetV3Small model.

GradCAM++ is not evident, which leads to the impossibility of generalizing the above-given statement (for some defects, GradCAM dominated GradCAM++).

To quantify all the above-reported insights given by qualitative observations, it is now necessary to provide a quantitative explanation of the obtained results, and it is time to call back the metrics defined in Section 3.5, I_{db}, and I_{ctx}. In particular, the role of the above indexes should be to measure in quantitative terms the visual attention in Tables 3 and 4, by confirming or disproving the attributed scores.

In Tables 5 and 6 the values of I_{ctx} are reported, while Tables 7 and 8 list the values of I_{db}. All the indexes are quantified for a fixed value of σ equal to 0.5, according to the qualitative evaluation carried out as previously described.

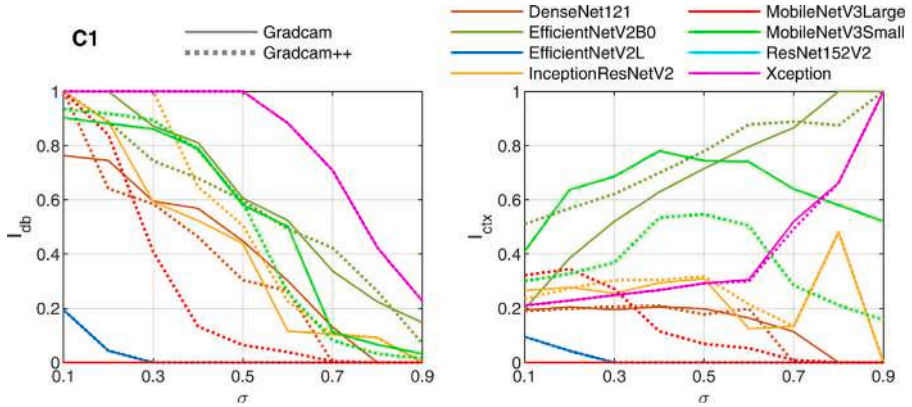

Fig. 12. I_{db} and I_{ctx} values for defect $C1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

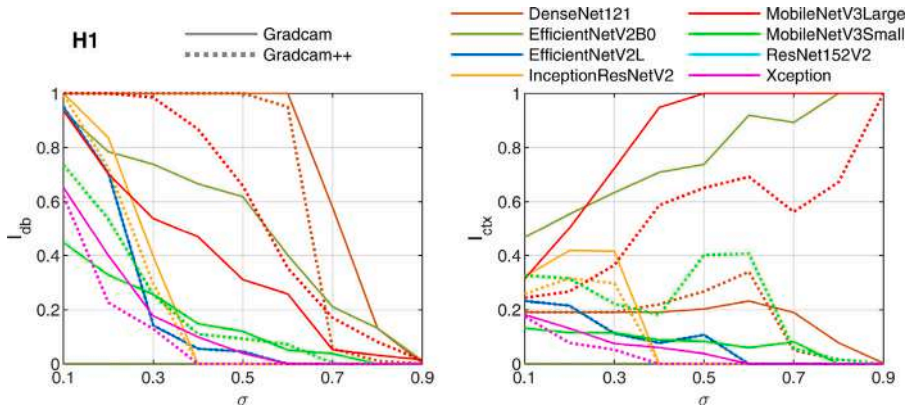

Fig. 13. I_{db} and I_{ctx} values for defect $H1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

Table 3
Qualitative scores assigned from the visual comparison between the activation zones provided by the CNNs models and the red boxes. Models 1, 2, 3, and 4 correspond to DenseNet121, EfficientNetV2B0, EfficientNetV2L, and InceptionResNetV2, respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively. The observed images are obtained by using σ equal to 0.5.

Defect	Model 1		Model 2		Model 3		Model 4	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	MB	MB	MG	G	B	B	MG	MG
C2	MB	B	B	G	B	B	G	B
CRS1	MB	MB	B	B	B	B	MB	MB
CRS2	B	MB	MG	B	B	B	MB	MB
DC1	B	B	B	MG	B	B	MB	MB
DC2	B	B	B	B	B	B	B	B
H1	MG	MG	MG	B	B	B	B	B
H2	MB	MB	MG	G	MB	MB	B	B
MS1	MB	MB	B	B	MB	MB	MB	MB
MS2	MB	MB	MB	MB	MB	MB	MB	MB
MS3	B	MB	MB	MG	B	B	MB	MB
PD1	B	B	MG	B	MB	MB	B	B
PD2	B	B	MB	B	B	B	B	B
SC1	MB	MB	MB	B	B	B	B	B
SC2	B	B	B	B	B	B	B	MB

The I_{db} and I_{ctx} values overall confirmed the qualitative evaluations by presenting low values of both indexes for defects with scores B and MB, and medium-high values for defects with score MG and G. Regarding the index I_{ctx}, it was observed that defects with values equal to 0 were more times marked as B and equal to 1 were more times marked with G. For defects qualitatively classified as G and MG, in most of the cases, I_{ctx} goes from 0.3 to higher values, while for defects qualitatively classified as B

Table 4
Qualitative scores assigned from the visual comparison between the activation zones provided by the CNNs models and the red boxes. Models 5, 6, 7, and 8 correspond to MobileNetV3Large, MobileNetV3Small, ResNet152V2, and Xception, respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively. The observed images are obtained by using σ equal to 0.5.

Defect	Model 5		Model 6		Model 7		Model 8	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	B	B	MG	MG	B	B	MG	MG
C2	B	B	B	MB	B	B	B	B
CRS1	B	B	B	B	B	B	B	MB
CRS2	B	B	MG	G	B	B	B	B
DC1	MG	MB	B	MG	B	B	B	B
DC2	B	B	MB	MB	B	B	B	B
H1	G	MG	B	MB	B	B	B	B
H2	G	G	MG	B	B	B	B	G
MS1	B	B	MB	B	B	B	B	B
MS2	MG	MG	MB	MG	B	B	B	MB
MS3	B	MB	MG	B	B	B	MB	B
PD1	B	B	MB	B	B	B	B	B
PD2	MG	MB	MB	MB	B	B	B	MB
SC1	B	MG	B	B	B	B	B	MG
SC2	MG	MB	B	B	B	B	B	B

Table 5
Values of I_{ctx} by using σ equal to 0.5. Models 1, 2, 3, and 4 correspond to DenseNet121, EfficientNetV2B0, EfficientNetV2L, and InceptionResNetV2, respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively.

Defect	Model 1		Model 2		Model 3		Model 4	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	0.198	0.177	0.715	0.779	0.000	0.000	0.309	0.317
C2	0.179	0.000	0.000	1.000	0.000	0.000	1.000	0.047
CRS1	0.113	0.134	0.000	0.000	0.000	0.000	0.092	0.083
CRS2	0.195	0.198	0.567	0.000	0.000	0.000	0.288	0.287
DC1	0.025	0.025	0.130	0.738	0.000	0.000	0.244	0.166
DC2	0.038	0.099	0.000	0.000	0.000	0.000	0.000	0.044
H1	0.303	0.368	0.738	0.000	0.107	0.107	0.000	0.000
H2	0.212	0.206	0.386	1.000	0.273	0.273	0.066	0.000
MS1	0.369	0.369	0.000	0.000	0.237	0.237	0.193	0.193
MS2	0.142	0.155	0.272	0.295	0.223	0.223	0.245	0.297
MS3	0.167	0.173	0.245	0.324	0.083	0.083	0.100	0.155
PD1	0.074	0.081	0.363	0.000	0.139	0.139	0.125	0.126
PD2	0.005	0.186	0.195	0.000	0.000	0.000	0.092	0.090
SC1	0.003	0.000	0.121	0.000	0.000	0.000	0.126	0.173
SC2	0.000	0.000	0.000	0.000	0.000	0.000	0.014	0.114

Table 6
Values of I_{ctx} by using σ equal to 0.5. Models 5, 6, 7, and 8 correspond to MobileNetV3Large, MobileNetV3Small, ResNet152V2, and Xception respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively.

Defect	Model 5		Model 6		Model 7		Model 8	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	0.000	0.069	0.745	0.547	0.000	0.000	0.392	0.392
C2	0.079	0.000	0.000	0.292	0.000	0.000	0.000	0.054
CRS1	0.000	0.000	0.058	0.000	0.000	0.000	0.068	0.241
CRS2	0.000	0.004	0.662	0.996	0.000	0.000	0.045	0.076
DC1	0.748	0.114	0.096	0.524	0.000	0.000	0.000	0.000
DC2	0.037	0.000	0.234	0.299	0.000	0.000	0.000	0.000
H1	1.000	0.651	0.083	0.232	0.000	0.000	0.038	0.000
H2	0.988	0.877	0.622	0.000	0.000	0.000	0.060	1.000
MS1	0.088	0.000	0.238	0.000	0.000	0.000	0.127	0.004
MS2	0.386	0.578	0.153	0.957	0.000	0.000	0.000	0.236
MS3	0.286	0.245	0.437	0.000	0.000	0.000	0.286	0.178
PD1	0.000	0.000	0.282	0.213	0.000	0.000	0.088	0.098
PD2	0.317	0.000	0.170	0.286	0.000	0.000	0.102	0.269
SC1	0.098	0.535	0.060	0.000	0.000	0.000	0.123	0.936
SC2	0.309	0.109	0.074	0.112	0.000	0.000	0.105	0.104

and MB, in most of the cases, I_{ctx} goes from 0.3 to lower values. Similar outcomes were observed concerning the index I_{db}, with values of the index closer to the 0 for B and MB scores and higher for G and MG votes (even if in some cases MB values have

Table 7
Values of I_{db} by using σ equal to 0.5. Models 1, 2, 3, and 4 correspond to Models 1, 2, 3, and 4 correspond to DenseNet121, EfficientNetV2B0, EfficientNetV2L, and InceptionResNetV2, respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively.

Defect	Model 1		Model 2		Model 3		Model 4	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	0.248	0.304	0.606	0.597	0.000	0.000	0.439	0.505
C2	0.296	0.000	0.000	0.303	0.000	0.000	0.357	0.036
CRS1	1.000	1.000	0.000	0.000	0.000	0.000	1.000	1.000
CRS2	1.000	0.827	0.407	0.000	0.000	0.000	1.000	1.000
DC1	0.081	0.081	0.080	0.353	0.000	0.000	0.176	0.312
DC2	0.127	0.178	0.000	0.000	0.000	0.000	0.000	0.194
H1	1.000	1.000	0.618	0.000	0.046	0.046	0.000	0.000
H2	1.000	0.794	0.345	0.307	0.194	0.194	0.114	0.000
MS1	1.000	1.000	0.000	0.000	0.143	0.143	1.000	1.000
MS2	1.000	0.929	0.174	0.247	0.065	0.065	0.385	0.475
MS3	1.000	0.974	0.000	0.000	0.748	0.748	0.379	0.336
PD1	0.940	0.940	0.578	0.000	0.000	0.000	0.213	1.000
PD2	0.846	0.785	0.100	0.000	0.061	0.061	0.159	0.127
SC1	0.008	0.090	0.121	0.000	0.000	0.000	0.100	0.000
SC2	0.000	0.000	0.000	0.000	0.000	0.000	0.145	0.329

Table 8
Values of I_{db} by using σ equal to 0.5. Models 5, 6, 7, and 8 correspond to MobileNetV3Large, MobileNetV3Small, ResNet152V2, and Xception, respectively; GC and GC++ correspond to GradCAM and GradCAM++, respectively.

Defect	Model 5		Model 6		Model 7		Model 8	
	GC	GC++	GC	GC++	GC	GC++	GC	GC++
C1	0.000	0.064	0.580	0.589	0.000	0.000	1.000	1.000
C2	0.062	0.000	0.000	0.316	0.000	0.000	0.000	0.020
CRS1	0.000	0.000	0.015	0.000	0.000	0.000	0.098	0.327
CRS2	0.000	0.002	0.394	0.414	0.000	0.000	0.054	0.054
DC1	0.115	0.093	0.225	0.486	0.000	0.000	0.000	0.000
DC2	0.086	0.000	0.146	0.143	0.000	0.000	0.000	0.000
H1	0.411	0.658	0.090	0.129	0.000	0.000	0.039	0.000
H2	0.465	0.311	0.396	0.000	0.000	0.000	0.151	0.430
MS1	0.184	0.000	0.279	0.000	0.000	0.000	0.098	0.001
MS2	0.630	0.328	0.072	0.248	0.000	0.000	0.000	0.629
MS3	0.000	0.214	0.285	0.000	0.000	0.000	0.174	0.039
PD1	0.000	0.000	0.056	0.084	0.000	0.000	0.202	0.226
PD2	0.270	0.000	0.169	0.141	0.000	0.000	0.175	0.067
SC1	0.163	0.536	0.061	0.000	0.000	0.000	0.188	0.301
SC2	0.341	0.000	0.067	0.082	0.000	0.000	0.105	0.105

been quantified with higher values, as occurs for DenseNet121). Despite this evaluation being made for a specific threshold value (and then, not fully generalizable), the proposed indexes quantify with a numerical explanation the human observed results. Finally, observing the trend of I_{db} and I_{ctx} in Figs. 12 and 13, it is worth noting that the I_{db} decreases when threshold increases, while I_{ctx} tends to increase in some cases and to decrease in other cases when threshold increases. This observation suggests that while I_{db} is an absolute index that describes the capacity of the network for the purpose of prediction, I_{ctx} is a relative index, and it depends on different factors. The results obtained by the proposed $quasi-quantitative$ evaluation suggest a good capacity for a physical explanation of the CNNs prediction, even if, in some specific cases, some discrepancies occurred. From the obtained outcomes, an insight for further development is the adoption of a univocal metric that summarizes the two novel indexes, computed as, for example, by combining the two indexes to obtain a synthetic and overall measure for a qualitative and quantitative explanation.

5.2. Application of the methodology for automatic defect recognition to the Viaduct of Corso Italia

The case of the Corso Italia Viaduct in Bari, as already mentioned, is among the examples of infrastructural engineering works characterized by a significance that is not only functional and strategic but also related to historical and cultural aspects. For these reasons, in fact, in 2006, an in-depth experimental study and investigation campaign was realized on the bridge [13]. Besides the detailed geometric survey, collection of existing historical design documents and drawings (as reported in Fig. 14), a full diagnostic program was carried out, to provide the data necessary for the evaluation of the structural vulnerability (for more details, see the different chapter devoted to the viaduct in [13]). First, visual inspections were performed, collecting a complete photographical survey and mapping of the degradation state of the viaduct, as shown in Fig. 15.

This phase required the fieldwork of 2 teams of 3 staff members each and lasted about one month. In the second phase, a program of on-site destructive and non-destructive tests was performed, including extraction of concrete samples, rebound-hammer, and ultrasonic tests; measurement of carbonation-depth; measurement of corrosion potential.

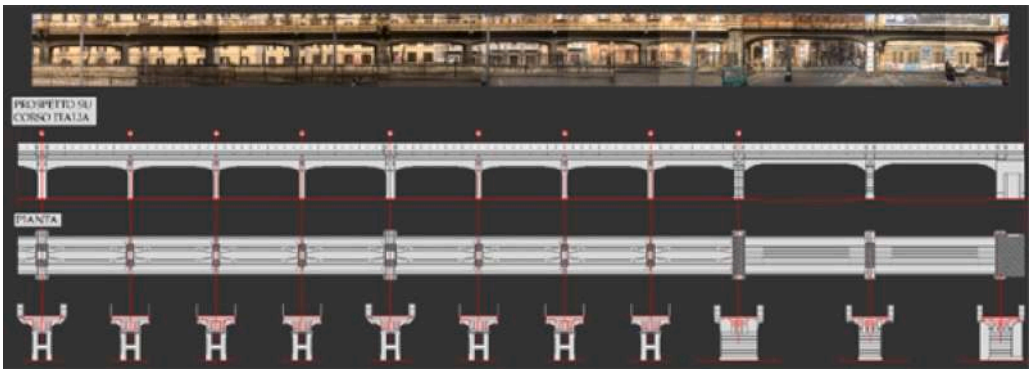

Fig. 14. Full photographic and geometric survey of the Viaduct of Corso Italia.

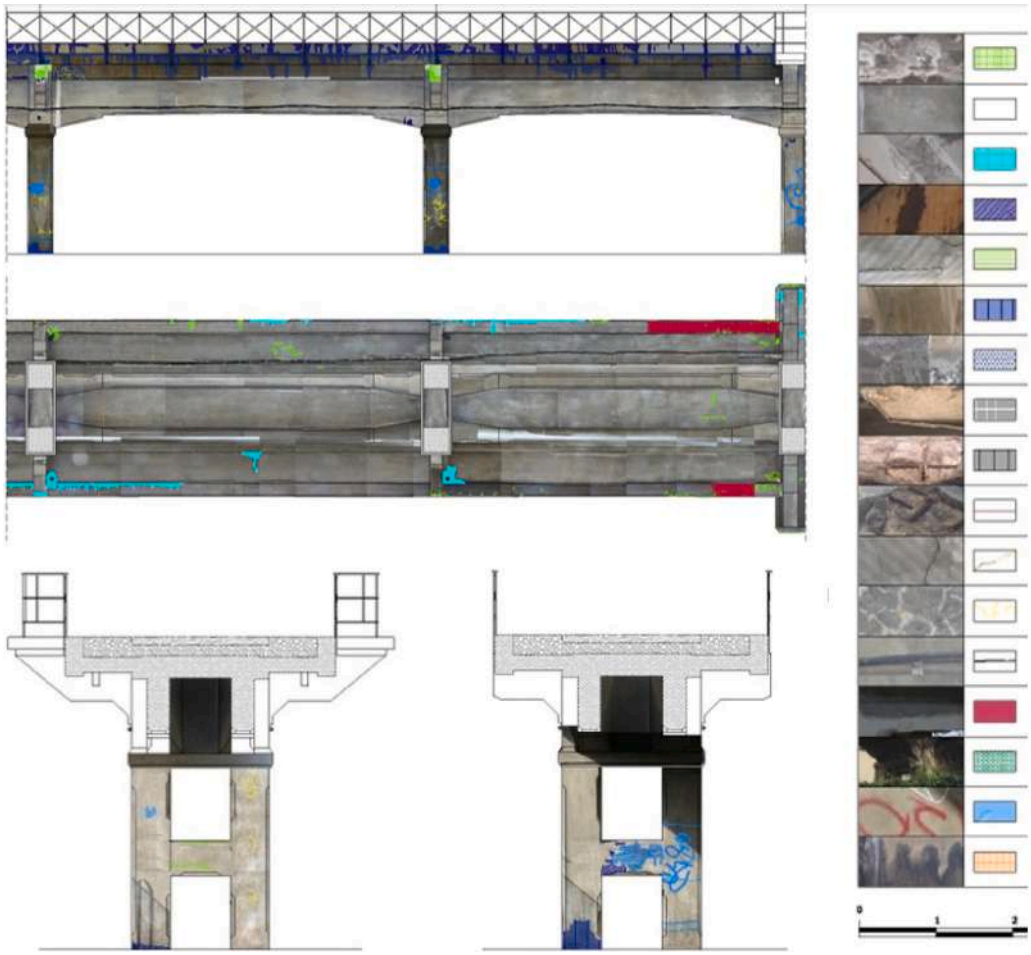

Fig. 15. Degradation map of structural elements of the Viaduct of Corso Italia.

The methodology for automatic defect recognition presented in the paper has been now applied in an exploratory way to this case study, which is well-known to the authors, in order to evaluate the current performance of the method and test the possibility of supporting degradation diagnosis by reducing time and manpower. According to the procedure, the images taken during the surveys have been processed for a bay of the Viaduct. In particular, in Fig. 16, the selected bay and the defects manually detected through photographs are indicated.

Fig. 16. Typical defects manually detected on a bay of Viaduct of Corso Italia. Red boxes indicate the damage box for each image.

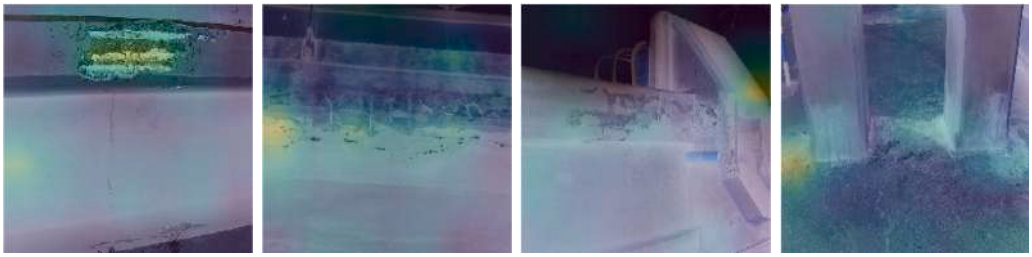

Fig. 17. Application of MobileNetV3Small and GradCAM on the defects detected on a bay of Viaduct of Corso Italia.

For the defects reported in Fig. 16, identifying two kinds of corroded steel reinforcements and moisture spots on the deck, and shrinkage cracks on the pile, the better CNN model was employed, that is, MobileNetV3Small and GradCAM was used to physically explain the automated prediction. The results of the application are shown in Fig. 17, where the visual attentions are reported by using a specific value of the threshold for each image ($\sigma = 0.7$ for defect in Fig. 17(a); $\sigma = 0.4$ for defect in Fig. 17(b); $\sigma = 0.2$ for defect in Fig. 17(c); $\sigma = 0.4$ for defect in Fig. 17(d)).

The performed prediction reveals overall good qualitative results, which encourage the use of the adopted means for defect recognition. However, the quantitative evaluation can be provided, where defect in Fig. 17(a) reports I_{db} equal to 0.909 and I_{ctx} equal to 0.276; defect in Fig. 17(b) reports I_{db} equal to 1.000 and I_{ctx} equal to 0.165; defect in Fig. 17(c) reports I_{db} equal to 0.981 and I_{ctx} equal to 0.143; defect in Fig. 17(d) reports I_{db} equal to 1 and I_{ctx} equal to 0.059. Observing the values of the proposed metrics, for all photos high values of I_{db} were obtained, which means that defects in the bounding boxes were effectively recognized. Instead, different values of I_{ctx} were obtained, which indicate that other parts of the images were activated by the employed CNN model (as for example occur in Fig. 17(d)).

6. Conclusions, practical suggestions, and further developments

In this paper, a study on the topics of the automatic recognition of defects in existing RC bridges was presented, by using DL techniques to perform prediction and XAI approaches to physically explain the obtained results. The importance of the existing infrastructural stock is not only related to economic and technical aspects but also to intangible aspects since some masonry and RC bridges represent monumental evidence of the historical evolution of the technology construction and the grandeur of our ancestors. The true challenge that public institutions, road management companies, and the scientific community are currently facing is preserving the existing heritage infrastructural stock through periodic and systematic inspections, aimed at prioritizing risks and planning further interventions. With the aim to support these operations, it is possible to use innovative digital techniques borrowed from the field of automation and artificial intelligence, in order to improve the currently available procedures. In this framework, the present study pursues the goal of the automatic recognition of defects in existing RC bridges and explains the obtained results.

Given a database of images describing seven types of real defects on RC bridges, eight different CNNs were trained using transfer learning starting from weights pre-trained on ImageNet. Afterward, two CAM-based techniques for visualization were used: GradCAM and GradCAM++, to visually explain the achieved results. The evaluation was performed by observing the activation maps provided by both methodologies and by evaluating the correspondence of the activated zones with a damage box defined by domain experts.

Following this comparison, two novel indexes were defined, to quantitatively evaluate how the network was visually activated by either the damage or the surrounding context. The achieved results showed the utility of the proposed metrics (quantitative), considering that for some specific defects, models with higher accuracy did not always return a human explainable prediction (qualitative), and vice versa. The proposed methodology was also tested on a real-life case study, for which the survey of defects was carried out by hand. Results highlighted the importance of innovative means for a fast but reliable inspection, in addition to provide the possibility to explain the predictions.

As the main practical suggestion derived from the observed results, for the purpose of developing tools aimed at automatically recognizing defects in existing RC bridges, accurate preparatory work should be carried out in the phase of data selection, attempting to increase as much as possible the number of images explained for each defect in order to avoid biases and poor data balancing. With this regard, the proposed novel indexes can represent an easy evaluation metric to provide a quasi-quantitative interpretation of the predictions given by the networks, and to drive analysts in the selection of the best strategies in the field of data collection.

Further developments will be focused on providing a univocal synthetic index to physically explain the overall results achieved by the prediction models, e.g., on combining the proposed metrics using a certain criterion, with the aim to provide a sub-optimal threshold, specifically suitable for generic defect recognition on existing RC bridges. On the structural side, some improvements should be proposed in terms of automatic recognition, by increasing the classes and typologies of defects and better identifying specific structural symptoms, such as the orientation of the cracks, that could indicate different types of decay conditions for the monitored structure.

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 reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The second author acknowledges funding by Italian Ministry of University and Research, within the project ‘PON-Ricerca e Innovazione 2014–2020, (D.M. 10/08/2021, n. 1062) CUP CODE: D95F21002140006.

Appendix. Values of I_{db} and I_{ctx}

In Appendix, the values of I_{db} and I_{ctx} obtained from GradCAM and GradCAM++ at the variation of σ, for all CNNs models and for defects not reported in the main text were shown in Figs. A.18–A.30. Supplementary material file can be anonymously accessed at the following link: bit.ly/3mnXg3d.

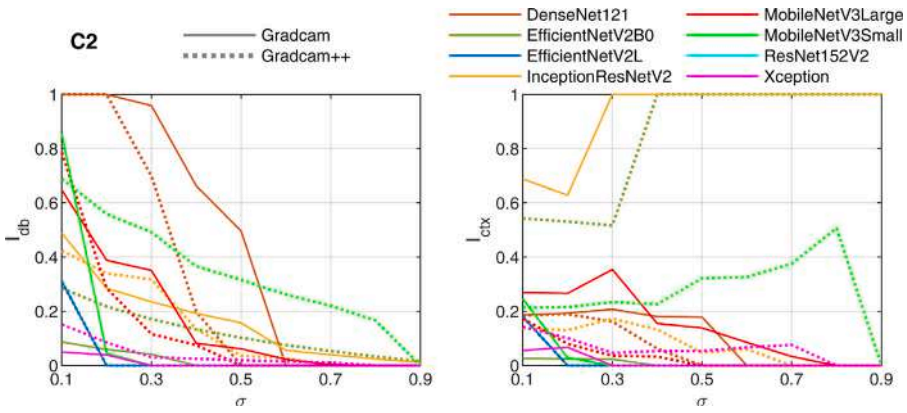

Fig. A.18. I_{db} and I_{ctx} values for defect $C2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

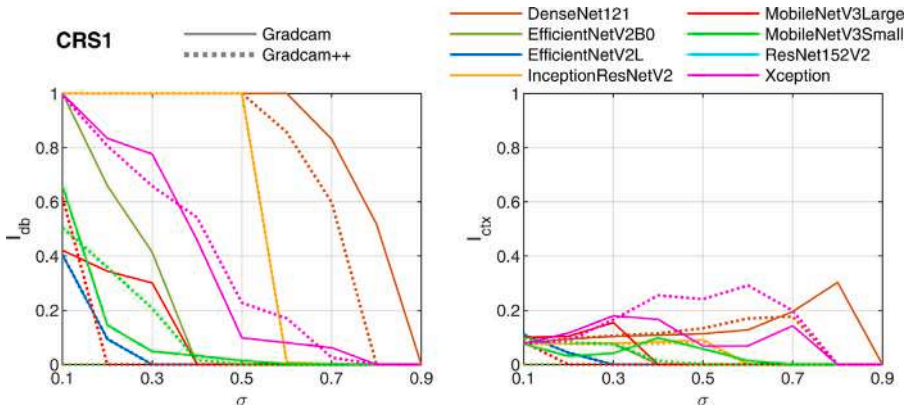

Fig. A.19. I_{db} and I_{ctx} values for defect $CRS1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

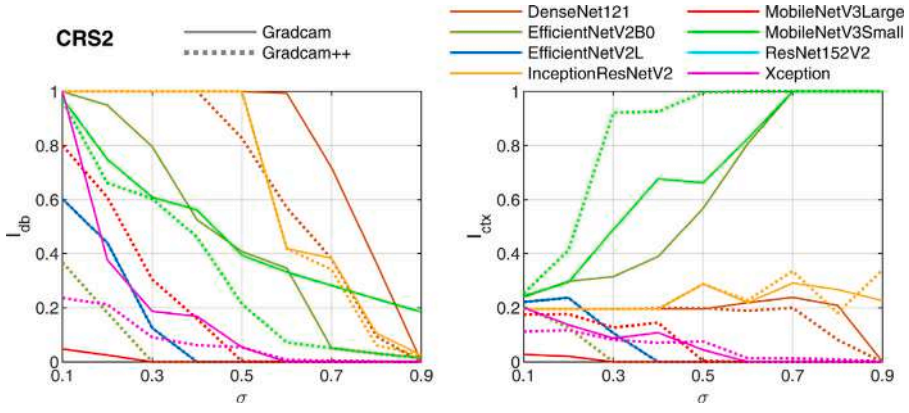

Fig. A.20. I_{db} and I_{ctx} values for defect $CRS2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

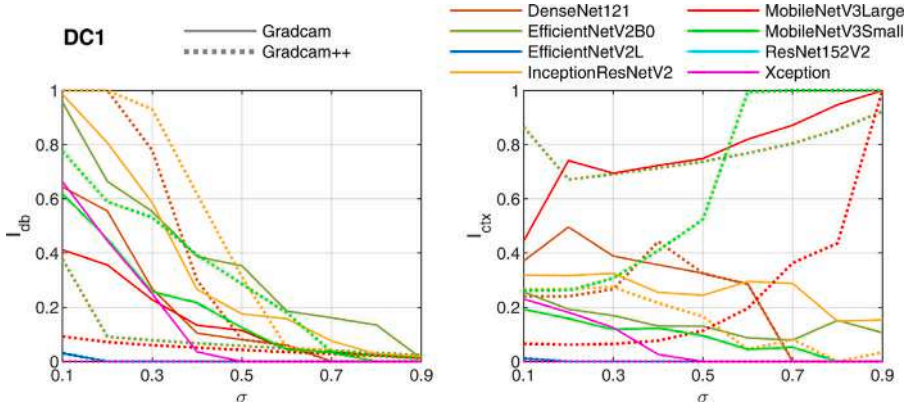

Fig. A.21. I_{db} and I_{ctx} values for defect $DC1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

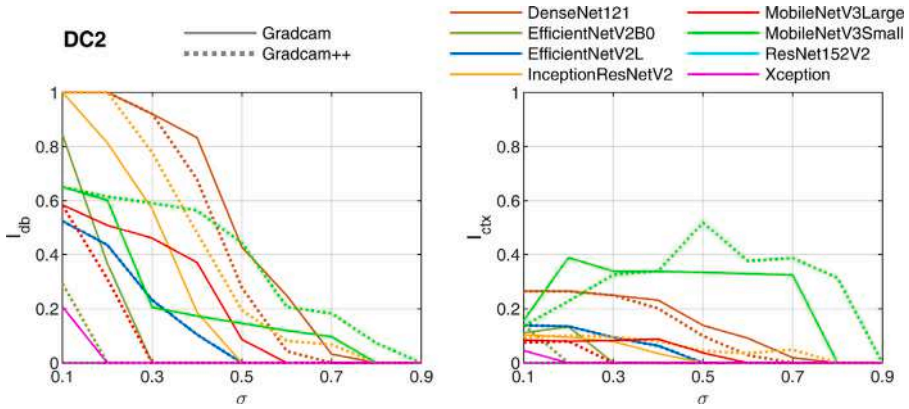

Fig. A.22. I_{db} and I_{ctx} values for defect $DC2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

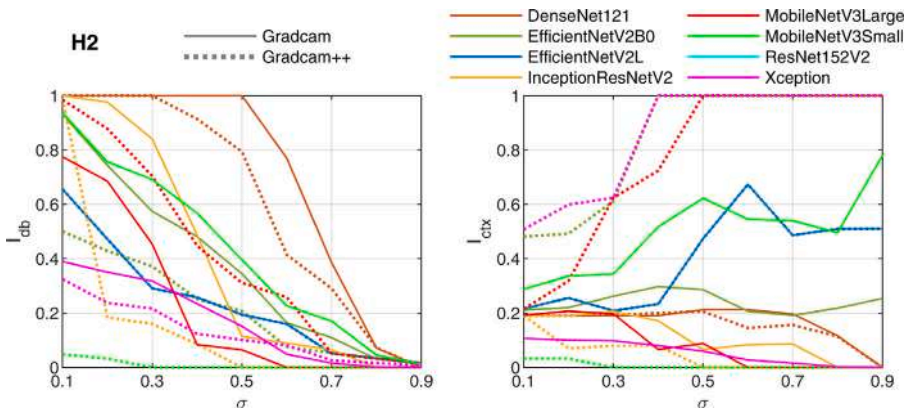

Fig. A.23. I_{db} and I_{ctx} values for defect $H2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

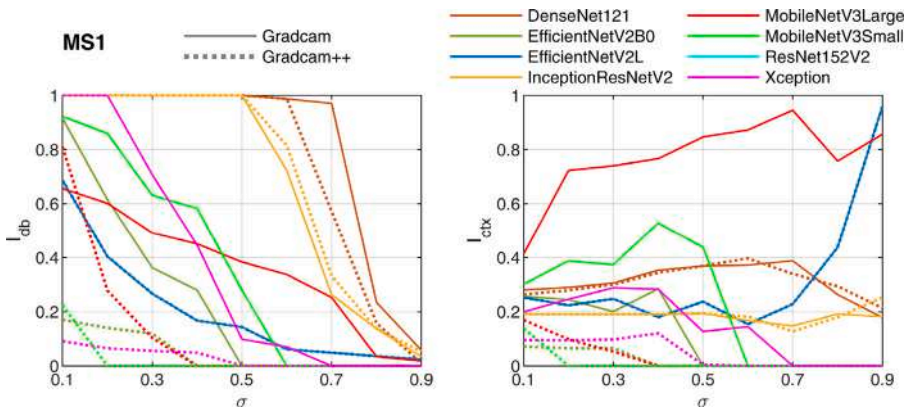

Fig. A.24. I_{db} and I_{ctx} values for defect $MS1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

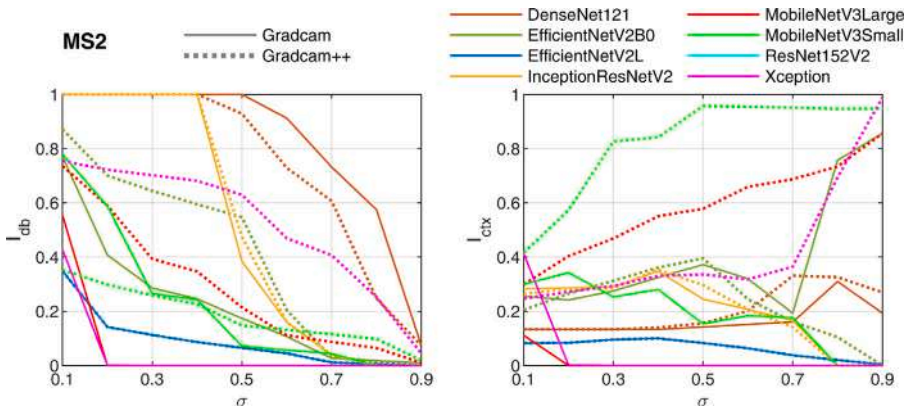

Fig. A.25. I_{db} and I_{ctx} values for defect $MS2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

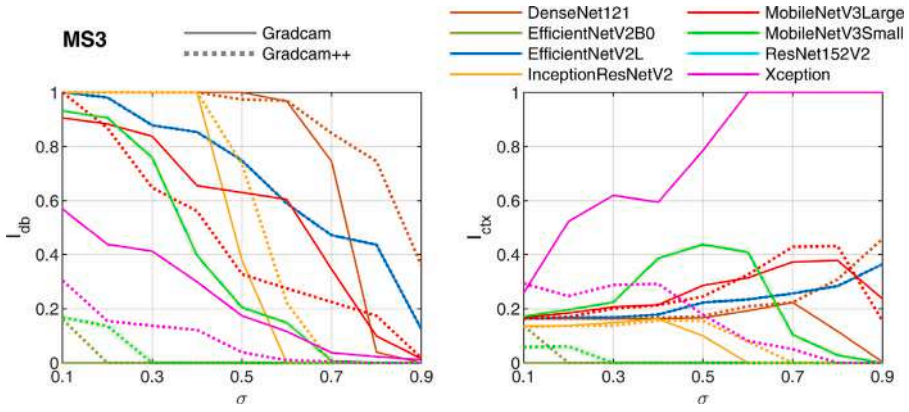

Fig. A.26. I_{db} and I_{ctx} values for defect $MS3$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

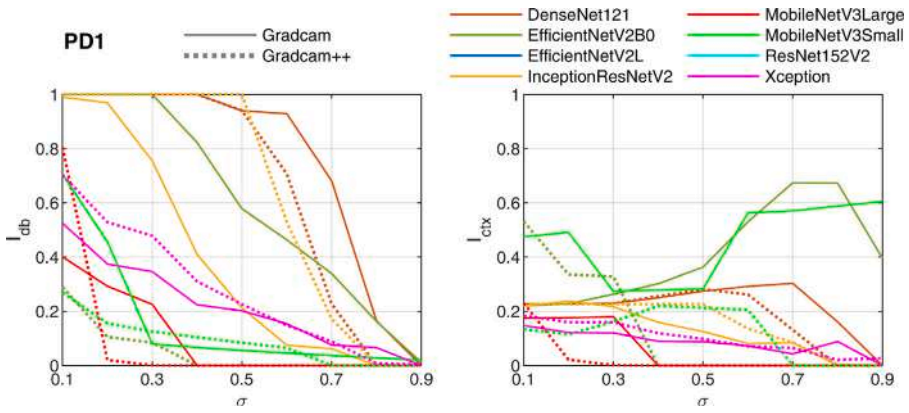

Fig. A.27. I_{db} and I_{ctx} values for defect $PD1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

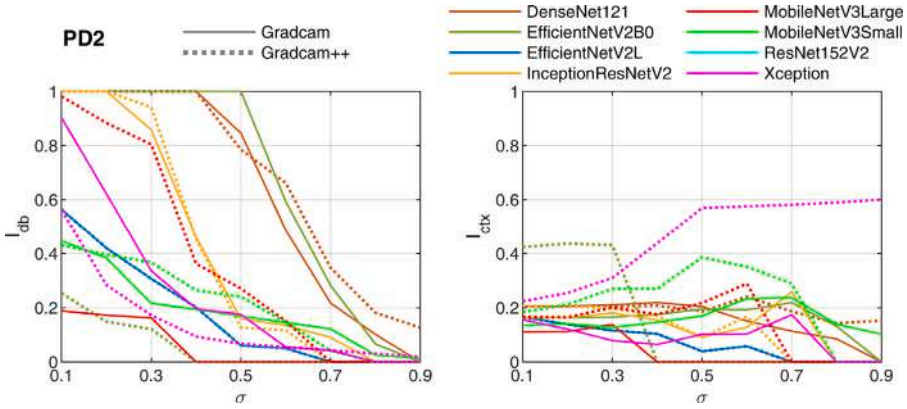

Fig. A.28. I_{db} and I_{ctx} values for defect $PD2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

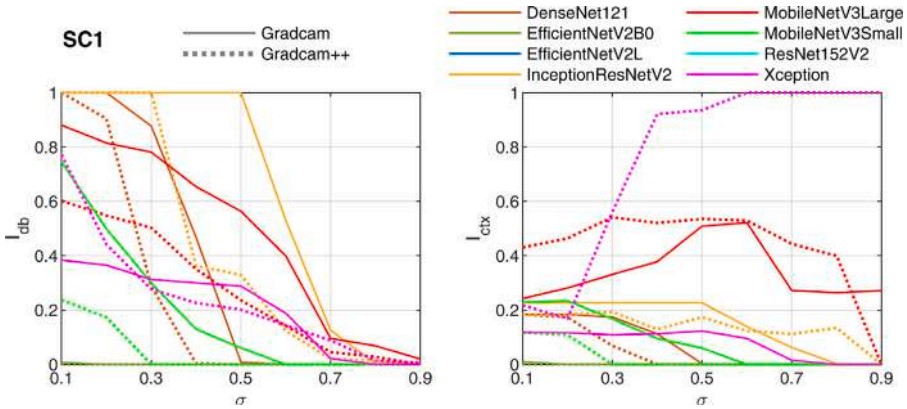

Fig. A.29. I_{db} and I_{ctx} values for defect $SC1$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

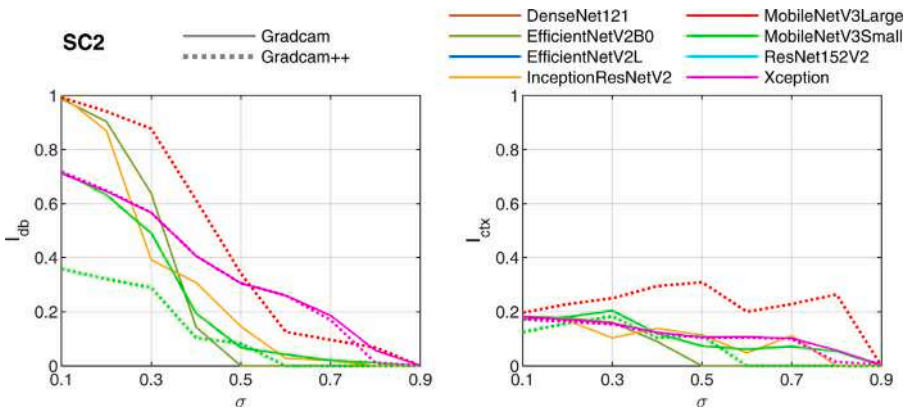

Fig. A.30. I_{db} and I_{ctx} values for defect $SC2$ and all CNNs models, obtained from (left to right) GradCAM and GradCAM++.

References

[1] M. Mosoarca, I. Onescu, E. Onescu, A. Anastasiadis, Seismic vulnerability assessment methodology for historic masonry buildings in the near-field areas, Eng. Fail. Anal. (2020) http://dx.doi.org/10.1016/j.engfailanal.2020.104662.

[2] I. Apostol, M. Mosoarca, V. Stoian, Modern consolidation solutions for buildings with historical value. part 1: Reinforced concrete structures, in: 16th National Technical-Scientific Conference on Modern Technologies for the 3rd Millennium, 2017.

[3] G. Miluccio, D. Losanno, F. Parisi, E. Cosenza, Traffic-load fragility models for prestressed concrete girder decks of existing Italian highway bridges, Eng. Struct. 249 (2021) 113367, http://dx.doi.org/10.1016/j.engstruct.2021.113367.

[4] V. Sangiorgio, A. Nettis, G. Uva, F. Pellegrino, H. Varum, J.M. Adam, Analytical fault tree and diagnostic aids for the preservation of historical steel truss bridges, Eng. Fail. Anal. 133 (2022) 105996, http://dx.doi.org/10.1016/j.engfailanal.2021.105996.
[5] B. Borzi, P. Ceresa, P. Franchin, F. Noto, G.M. Calvi, P.E. Pinto, Seismic vulnerability of the Italian roadway bridge stock, Earthq. Spectr. (2015) https://journals.sagepub.com/doi/full/10.1193/070413EQS190M.
[6] A. Nettis, P. Iacovazzo, D. Raffaele, G. Uva, J.M. Adam, Displacement-based seismic performance assessment of multi-span steel truss bridges, Eng. Struct. 254 (2022) 113832, http://dx.doi.org/10.1016/j.engstruct.2021.113832.
[7] A. Anisha, A. Jacob, R. Davis, S. Mangalathu, Fragility functions for highway RC bridge under various flood scenarios, Eng. Struct. 260 (2022) http://dx.doi.org/10.1016/j.engstruct.2022.114244.
[8] D. Peduto, F. Elia, R. Montuori, Probabilistic analysis of settlement-induced damage to bridges in the city of amsterdam (The Netherlands), Transp. Geotech. 14 (2018) 169–182, http://dx.doi.org/10.1016/j.trgeo.2018.01.002.
[9] A. Nettis, V. Massimi, R. Nutricato, D.O. Nitti, S. Samarelli, G. Uva, Satellite-based interferometry for monitoring structural deformations of bridge portfolios, Autom. Constr. (2023) http://dx.doi.org/10.1016/j.autcon.2022.104707.
[10] E. Bertolesi, G. Milani, F.D. Lopane, M. Acito, Augustus bridge in narni (Italy): Seismic vulnerability assessment of the still standing part, possible causes of collapse, and importance of the roman concrete infill in the seismic-resistant behavior, Int. J. Archit. Herit. 11 (5) (2017) 717–746, http://dx.doi.org/10.1080/15583058.2017.1300712.
[11] T. Papa, N. Grillanda, G. Milani, Three-dimensional adaptive limit analysis of masonry arch bridges interacting with the backfill, Eng. Struct. 248 (2021) 113189, http://dx.doi.org/10.1016/j.engstruct.2021.113189.
[12] G. Milani, P.B. Lourenço, 3D non-linear behavior of masonry arch bridges, Comput. Struct. 110 (2012) 133–150, http://dx.doi.org/10.1016/j.compstruc.2012.07.008.
[13] M. Mezzina, G. Uva, R. Greco, Sicurezza e conservazione delle prime costruzioni in calcestruzzo armato, CittàStudi Editor, 2008 (in Italian).
[14] G.M. Calvi, M. Moratti, G.J. O'Reilly, N. Scattarreggia, R. Monteiro, D. Malomo, P.M. Calvi, R. Pinho, Once upon a time in Italy: The tale of the morandi bridge, Struct. Eng. Int. 29 (2019) 198–217, http://dx.doi.org/10.1080/10168664.2018.1558033.
[15] F. Bazzucchi, L. Restuccia, G.A. Ferro, Considerations over the Italian road bridge infrastructure safety after the polcevera viaduct collapse: past errors and future perspectives., Frattura E Integrita Strutturale 12 (2018).
[16] Ministero delle Infrastrutture e dei Trasporti, Linee Guida per la Classificazione e Gestione del Rischio, la Valutazione della Sicurezza ed il Monitoraggio dei Ponti Esistenti, Ministero delle Infrastrutture e dei Trasporti Consiglio Superiore dei Lavori Pubblici, 2020, https://www.mit.gov.it/comunicazione/news/mit-approvate-le-linee-guida-per-la-sicurezza-dei-ponti. (Accessed 22 October 2022).
[17] N.J. Bertola, E. Brühwiler, Risk-based methodology to assess bridge condition based on visual inspection, Struct. Infrastructure Eng. 19 (4) (2023) http://dx.doi.org/10.1080/15732479.2021.1959621.
[18] Federal Highway Administration (FHWA), National bridge inspection standards, 2018, URL https://www.fhwa.dot.gov/bridge/nbis.cfm.
[19] M. Mandirola, C. Casarotti, S. Peloso, I. Lanese, E. Brunesi, I. Senaldi, F. Risi, A. Monti, C. Facchetti, Guidelines for the use of unmanned aerial systems for fast photogrammetry-oriented mapping in emergency response scenarios, Int. J. Disaster Risk Reduct. 58 (2021) http://dx.doi.org/10.1016/j.ijdrr.2021.102207.
[20] H. Sun, H.V. Burton, H. Huang, Machine learning applications for building structural design and performance assessment: State-of-the-art review, J. Build. Eng. 33 (2021) 101816, http://dx.doi.org/10.1016/j.jobe.2020.101816.
[21] Y. Xie, M.E. Sichani, J. Padgett, R. DesRoches, The promise of implementing machine learning in earthquake engineering: a state-of-the-art review, Earthq. Spectr. 36 (4) (2020) 1769–1801, http://dx.doi.org/10.1177/8755293020919419.
[22] J. Kiani, C. Camp, S. Pezeshk, On the application of machine learning techniques to derive seismic fragility curves, Comput. Struct. 218 (2019) 108–122, http://dx.doi.org/10.1016/j.compstruc.2019.03.004.
[23] M.K. Almustafa, M.L. Nehdi, Machine learning model for predicting structural response of RC columns subjected to blast loading, Int. J. Impact Eng. 162 (2022) 104145, http://dx.doi.org/10.1016/j.ijimpeng.2021.104145.
[24] S. Ruggieri, A. Cardellicchio, V. Leggieri, G. Uva, Machine-learning based vulnerability analysis of existing buildings, Autom. Constr. 132 (2021) 103936, http://dx.doi.org/10.1016/j.autcon.2021.103936.
[25] A. Cardellicchio, S. Ruggieri, V. Leggieri, G. Uva, View VULMA: Data set for training a machine-learning tool for a fast vulnerability analysis of existing buildings, Data 7 (1) (2022) 4, http://dx.doi.org/10.3390/data7010004.
[26] M.R. Jahanshahi, J.S. Kelly, S.F. Masri, G.S. Sukhatme, A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures, Struct. Infrastructure Eng. 5 (6) (2009) http://dx.doi.org/10.1080/15732470801945930.
[27] S. Lee, L.-M. Chang, M. Skibniewski, Automated recognition of surface defects using digital color image processing, Autom. Constr. 15 (2006) http://dx.doi.org/10.1016/j.autcon.2005.08.001.
[28] R.S. Adhikari, O. Moselhi, A. Bagchi, Image-based retrieval of concrete crack properties for bridge inspection, Autom. Constr. 39 (2014) 180–194, http://dx.doi.org/10.1016/j.autcon.2013.06.011.
[29] Y.-S. Yang, C.-M. Yang, C.-W. Huang, Thin crack observation in a reinforced concrete bridge pier test using image processing and analysis, Adv. Eng. Softw. 83 (2015) 99–108, http://dx.doi.org/10.1016/j.advengsoft.2015.02.005.
[30] G. Li, S. He, Y. Ju, K. Du, Long-distance precision inspection method for bridge cracks with image processing, Autom. Constr. 41 (2014) 83–95, http://dx.doi.org/10.1016/j.autcon.2013.10.021.
[31] J.-H. Chen, M.-C. Su, R. Cao, S.-C. Hsu, J.-C. Lu, A self organizing map optimization based image recognition and processing model for bridge crack inspection, Autom. Constr. 73 (2017) 58–66, http://dx.doi.org/10.1016/j.autcon.2016.08.033.
[32] B.J. Perry, Y. Guo, R. Atadero, J.W. van de Lindt, Streamlined bridge inspection system utilizing unmanned aerial vehicles (UAVs) and machine learning, Measurement 164 (2020) 108048, http://dx.doi.org/10.1016/j.measurement.2020.108048.
[33] F. Potenza, C. Rinaldi, E. Ottaviano, V. Gattulli, A robotics and computer-aided procedure for defect evaluation in bridge inspection, J. Civil Struct. Health Monit. 10 (2020) 471–484, http://dx.doi.org/10.1007/s13349-020-00395-3.
[34] P. Prasanna, K.J. Dana, N. Gucunski, B.B. Basily, H.M. La, R.S. Lim, H. Parvardeh, Automated crack detection on concrete bridges, IEEE Trans. Autom. Sci. Eng. 13 (2016) 591–599, http://dx.doi.org/10.1109/TASE.2014.2354314.
[35] E. Mohammed Abdelkader, M. Marzouk, T. Zayed, A self-adaptive exhaustive search optimization-based method for restoration of bridge defects images, Int. J. Mach. Learn. Cybern. 11 (2020) 1659–1716, http://dx.doi.org/10.1007/s13042-020-01066-x.
[36] N.-D. Hoang, Image processing-based pitting corrosion detection using metaheuristic optimized multilevel image thresholding and machine-learning approaches, Math. Probl. Eng. 2020 (2020) e6765274, http://dx.doi.org/10.1155/2020/6765274.
[37] G. Montaggioli, M. Puliti, A. Sabato, Automated damage detection of bridge's sub-surface defects from infrared images using machine learning, 2021, http://dx.doi.org/10.1117/12.2581783.
[38] J. Zhu, C. Zhang, H. Qi, Z. Lu, Vision-based defects detection for bridges using transfer learning and convolutional neural networks, Struct. Infrastructure Eng. 16 (7) (2020) 1037–1049, http://dx.doi.org/10.1080/15732479.2019.1680709.
[39] A. Cardellicchio, S. Ruggieri, A. Nettis, C. Patruno, G. Uva, V. Renò, Deep learning approaches for image-based detection and classification of structural defects in bridges, in: P.L. Mazzeo, E. Frontoni, S. Sclaroff, C. Distante (Eds.), Image Analysis and Processing. ICIAP 2022 Workshops, in: Lecture Notes in Computer Science, vol. 13373, Springer International Publishing, 2022, pp. 269–279, http://dx.doi.org/10.1007/978-3-031-13321-3-24.

[40] Z.A. Bukhsh, N. Jansen, A. Saeed, Damage detection using in-domain and cross-domain transfer learning, Neural Comput. Appl. 33 (2021) 16921–16936, http://dx.doi.org/10.1007/s00521-021-06279-x.
[41] P. Hüthwohl, R. Lu, I. Brilakis, Multi-classifier for reinforced concrete bridge defects, Autom. Constr. 105 (2019) 102824, http://dx.doi.org/10.1016/j.autcon.2019.04.019.
[42] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep learning-based crack damage detection using convolutional neural networks, Comput.-Aided Civ. Infrastruct. Eng. 32 (2017) http://dx.doi.org/10.1111/mice.12263.
[43] Y. Xu, Y. Bao, J. Chen, W. Zuo, H. Li, Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images, Struct. Health Monit. (2018) 1–22, http://dx.doi.org/10.1177/1475921718764873.
[44] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, X. Yang, Automatic pixel-level crack detection and measurement using fully convolutional network, Comput.-Aided Civ. Infrastruct. Eng. 33 (2018) http://dx.doi.org/10.1111/mice.12412.
[45] W. Deng, Y. Mou, T. Kashiwa, S. Escalera, K. Nagai, K. Nakayama, Y. Matsuo, H. Prendinger, Vision based pixel-level bridge structural damage detection using a link ASPP network, Autom. Constr. 110 (2020) 102973, http://dx.doi.org/10.1016/j.autcon.2019.102973.
[46] Y.J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Büyüköztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, Comput.-Aided Civ. Infrastruct. Eng. 33 (2018) http://dx.doi.org/10.1111/mice.12334.
[47] R. Li, Y. Yuan, W. Zhang, Y. Yuan, Unified vision-based methodology for simultaneous concrete defect detection and geolocalization, Comput.-Aided Civ. Infrastruct. Eng. 33 (7) (2018) 527–544, http://dx.doi.org/10.1111/mice.12351.
[48] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: Single shot MultiBox detector, 2016, http://dx.doi.org/10.1007/978-3-319-46448-0-2.
[49] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2016, http://arxiv.org/abs/1506.02640.
[50] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, H. Omata, Road damage detection and classification using deep neural networks with smartphone images, Comput.-Aided Civ. Infrastruct. Eng. 33 (12) (2018) 1127–1141, http://dx.doi.org/10.1111/mice.12387.
[51] D. Minh, H.X. Wang, Y.F. Li, T.N. Nguyen, Explainable artificial intelligence: a comprehensive review, Artif. Intell. Rev. 55 (2022) 3503–3568, http://dx.doi.org/10.1007/s10462-021-10088-y.
[52] B. Zhou, D. Bau, A. Oliva, A. Torralba, Interpreting deep visual representations via network dissection, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2019) 2131–2145, http://dx.doi.org/10.1109/TPAMI.2018.2858759.
[53] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, 2015, http://dx.doi.org/10.48550/arXiv.1506.06579.
[54] C. Payer, D. Štern, H. Bischof, M. Urschler, Integrating spatial configuration into heatmap regression based CNNs for landmark localization, Med. Image Anal. 54 (2019) 207–219, http://dx.doi.org/10.1016/j.media.2019.03.007.
[55] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision, ICCV, (ISSN: 2380-7504) 2017, pp. 618–626, http://dx.doi.org/10.1109/ICCV.2017.74.
[56] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, Adv. Neural Inf. Process. Syst. 31 (2018).
[57] M.Z. Naser, An engineer's guide to explainable artificial intelligence and interpretable machine learning: Navigating causality, forced goodness, and the false perception of inference, Autom. Constr. 129 (2021) 103821, http://dx.doi.org/10.1016/j.autcon.2021.103821.
[58] A.T.G. Tapeh, M.Z. Naser, Discovering graphical heuristics on fire-induced spalling of concrete through explainable artificial intelligence, Fire Technol. 58 (2022) 2871–2898, http://dx.doi.org/10.1007/s10694-022-01290-7.
[59] S.N. Somala, K. Karthikeyan, S. Mangalathu, Time period estimation of masonry infilled RC frames using machine learning techniques, Structures 34 (2021) 1560–1566, http://dx.doi.org/10.1016/j.istruc.2021.08.088.
[60] S. Mangalathu, S.-H. Hwang, J.-S. Jeon, Failure mode and effects analysis of RC members based on machine-learning-based shapley additive explanations (SHAP) approach, Eng. Struct. 219 (2020) 110927, http://dx.doi.org/10.1016/j.engstruct.2020.110927.
[61] B.F. Spencer, V. Hoskere, Y. Narazaki, Advances in computer vision-based civil infrastructure inspection and monitoring, Engineering 5 (2019) 199–222, http://dx.doi.org/10.1016/j.eng.2018.11.030.
[62] J. Bush, T. Corradi, J. Ninić, G. Thermou, J. Bennetts, Deep neural networks for visual bridge inspections and defect visualisation in civil engineering, 2021.
[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[64] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, 2016, pp. 630–645.
[65] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, 2014, http://dx.doi.org/10.48550/arXiv.1409.4842, cs.
[66] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2012) 1097–1105.
[67] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, 2018, pp. 839–847, http://dx.doi.org/10.1109/WACV.2018.00097.
[68] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al., Searching for mobilenetv3, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.
[69] M. Tan, Q.V. Le, EfficientNetV2: Smaller models and faster training, 2021, http://dx.doi.org/10.48550/arXiv.2104.00298.
[70] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[71] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1, 2017.
[72] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer, 2016, pp. 630–645.
[73] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1251–1258.
[74] D. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, ArXiv Preprint ArXiv, 1412.6980.