IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 January 2025, accepted 20 January 2025, date of publication 22 January 2025, date of current version 29 January 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3532832

RESEARCH ARTICLE

Using Attention for Improving Defect Detection in Existing RC Bridges

SERGIO RUGGIERI[1], ANGELO CARDELLICCHIO[2], ANDREA NETTIS[1], VITO RENÒ[2], AND GIUSEPPINA UVA[1]

[1]DICATECH Department, Polytechnic University of Bari, 70126 Bari, Italy
[2]Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, 70126 Bari, Italy

Corresponding author: Angelo Cardellicchio (angelo.cardellicchio@stiima.cnr.it)

The work of Sergio Ruggieri was supported by the Italian Ministry of University and Research through the project "PON-Ricerca e Innovazione 2014–2020 (D.M. 10/08/2021, n. 1062)" under Grant CUP D95F21002140006. The work of Andrea Nettis and Giuseppina Uva was supported by the Centro Nazionale Sustainable Mobility Center, within the framework of "MOST" project, under CUP D93C22000410001.

ABSTRACT In the constantly growing need for sustainable mobility and transportation, on-site inspections of existing reinforced concrete (RC) bridges are critical in ensuring the safety of such infrastructures. However, surveying RC bridges presents several challenges, such as the high costs and effort required by the surveyors, the subjectivity in assessing identified defects, and the possible lapses of attention when inspections are systematically repeated on different bridges. Hence, traditional methods of on-site inspection can be enhanced by leveraging digital innovations and by developing new instruments that support road management companies in ensuring the safety of the existing infrastructure. Among the new technologies, deep learning-based object detection systems provide promising and effective solutions. As such, this research proposes a new, simple, intuitive and efficient tool to support engineers and surveyors in assessing the health state of existing RC bridges. To this end, domain experts gathered and labelled a dataset of real images containing typical defects found in existing RC bridges. Consequently, an improved version of YOLO11, embedding attention mechanisms to allow the network to focus on the most relevant details in each image, was trained, tested, and validated on the provided dataset, showing an overall improvement of quantitative metrics such as precision and recall, while retaining enough computational efficiency to allow real-time implementation on constrained devices. Visual explanations achieved via the Eigen-CAM algorithm were also exploited to evaluate the reliability of the predictions. The model was finally embedded in an end-to-end tool offering a graphical user interface (GUI) to allow an effective interaction between the domain expert and the machine. Overall, the proposal revealed its potential to improve the effectiveness of the survey, lowering the burden on surveyors and engineers and providing a reliable method to improve the overall security in large RC bridges portfolios.

INDEX TERMS Existing bridges, surface defects, deep-learning, object detection, practice-oriented tool.

I. INTRODUCTION

The study of the risks to which existing infrastructures are daily subjected has not always attracted the attention of public institutions, which for years ignored the real capacity of the built stock to face different sources of hazard. Nevertheless, this uncompromising approach was disproved by the occurrence of several disasters, bringing losses that are sometimes even priceless. The matter becomes more complex for bridges and viaducts, where the losses related to hazardous events can exponentially grow, especially when looking at the crucial role that each bridge covers within a given road network. Taking as reference Italy, several bridge collapsed for different sources of hazard, such as the Polcevera Bridge for absence of manutention [1]; the Albiano Magra Bridge for slow kinematic motions [2]; the bridge

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

on the Tronto River as a consequence of the Central Italy Earthquake, 2016 [3]. When talking about risks for existing bridges, it is not possible to only refer to a specific hazard. A combination of risk sources shall be considered (e.g., seismic actions, floods, geological and geotechnical motions, and natural decay of structural materials). Therefore, it is clear that the safety of existing bridges should be investigated as a multi-risk problem, and specific protocols are essential for driving road management companies to elaborate reliable risk mitigation plans. With this goal in mind, the Italian Ministry of Transportation, supported by the scientific community, drafted new guidelines for the management and evaluation of the safety of existing bridges [4]. The recently introduced prescriptions deal with the definition of a specific framework to apply to the entire national stock of existing bridges, emphasizing the often-neglected monitoring and maintenance phases. The proposed approach is articulated by six consecutive assessment levels, each characterized by different degrees of accuracy and analytical complexity. The first three levels (Levels 0, 1, and 2) were developed to carry out a preliminary screening of all the bridges assigned to each management company and to elaborate a first risk prioritization plan. The remaining levels (Levels 3, 4, and 5) drive the actions to take on critical bridges subjected to significant risk, as identified in the prior phases. Within the initial trio of levels, the most consistent phase regards Level 1, which consists of performing on-site surveys of bridges to (a) derive an overall score reflecting the health state of the bridge through a detailed visual inspection of all structural elements (e.g., decks, girders, piers, bearing devices, abutments); (b) assess potential interference from different risk sources, such as traffic, floods, landslides, and earthquakes. According to the outcomes of this phase, Level 2 can be employed for assigning a bridge-specific ''risk class'' encompassing all aforementioned risk sources. For the sake of completeness, the guidelines [4] prescribe five levels of risk classes, ranging from low to high, and assigned according to a predefined logical operators scheme. For bridges identified with high-risk classes, further investigations (according to Levels 3, 4, and 5) should be performed to evaluate, for specific limit-states, the actions to undertake to ensure the safety of the bridge and the overall road network (e.g., sensor-based structural monitoring, retrofitting).

For the scope of this paper, the main interest is to characterize the activities related to Level 1, where, in general, a team of trained surveyors is deployed to perform on-site inspections to record defects according to a specific form (i.e., by assigning to each defect scores about intensity and extent), and to take photographs to support defect detection and for tracking their temporal evolution. Although the most direct and expedient method for detecting defects on existing bridges is the on-site visual inspection, several issues arise and can influence the reliability of the final evaluation. One primary issue regards the subjective interpretation of defects by surveyors that, despite the initial training, can be influenced by human factors, such as lapses in attention and mental weariness (especially when the number of bridges to inspect increases). Such human factors can lead to biased scores attributed to detected defects, ultimately varying the overall vulnerability estimate. In addition, external factors can also introduce uncertainties to the score to be assigned to each defect, such as weather conditions, lighting, and distance between the surveyor and the inspected element. Furthermore, it is worth considering the real inaccessibility of some parts of the inspected bridges, such as bearing supports, which may not have been inspected several times and, in this case, without an assigned score. Ultimately, the elevated time required to perform an accurate visual inspection (and to record all defects) and the related high costs should not be neglected.

The problems mentioned above, first-hand experienced by the authors of this article during on-site inspections of existing bridges, demonstrate the imperative need to develop new tools to assist and support surveyors in this crucial phase [5]. To this end, the recent advances in computer vision technologies can be exploited by leveraging the possible benefits of automated defect recognition and detection in this field. This paper aims to provide a tool, named BRIDE-YOLO (acronym of *BRIdge DEfects detection via YOLO*), to deal with the above necessities. The tool aims to automatically recognize and detect typical defects in existing reinforced concrete (RC) bridges by only exploiting the information content of images taken from real structures. The choice of a deep-learning-based model is motivated by two fundamental advantages provided by these architectures in the specific application of object detection. First, deep-learning-based models can exploit the representation learning paradigm [6], using the capabilities of deep neural networks to automatically extract embedding to represent relevant and complex characteristics in heterogeneous images, overcoming the traditional limitations of identifying features invariant to phenomena such as occlusions, lighting variations, or changes in pose and angle of view. Second, deep learning models for object detection are not subjected to the constraints posed by classic cascade detectors [7], specifically, the requirement to provide predefined templates to match the structure of the objects to be identified within the image. These advantages pose an obvious trade-off, as deep neural networks are composed of many parameters that should be properly trained and, therefore, require adequate datasets and computational capacity to provide meaningful performance. This trade-off must be, therefore, addressed when selecting those tools, as described in Section III.

With this regard, it is worth mentioning some attempts that authors experienced in past research using existing Convolutional Neural Networks (CNNs) (e.g., [8]) and different typologies of object detectors (e.g., [9], [10], [11]). Nevertheless, based on the latest version of the data set, BRIDE-YOLO offers a real practice-oriented tool ready to be used during or after the on-site inspection of existing

RC bridges. In detail, the tool was developed by exploiting the labelling of defects carried out by authors on more than 6500 images. Afterwards, the base architecture of YOLO11 was improved with attention mechanisms and used for training, testing, and validation, and the evaluation metrics of each object detector were critically analyzed. The Eigen-CAM visual explainability methodology was employed to assess the predictions' reliability. The pipeline and the graphical user interface (GUI) of BRIDE-YOLO were also presented. The functionalities and the strength points of the tool were shown, highlighting how to implement the tool in real-life applications along with some practical insights for future applications. Finally, it is worth clarifying the choice of the acronym BRIDE-YOLO, as the name states, can become the inseparable life partner of surveyors involved in bridge inspections.

II. RELATED WORKS: OBJECT DETECTION FOR CIVIL INFRASTRUCTURES

Over recent years, new approaches based on computer vision (CV) and artificial intelligence (AI) emerged and strongly contaminated the scientific research, promoting the growth of digital innovation also in traditional research fields as civil and structural engineering. One of the recent trends regards the possibility of collecting information about structures and infrastructures from images, as they are the most readily available data sources on structures and infrastructures. In this view, different goals were pursued by the scientific community for retrieving data from images, such as the extraction of structural typological information (e.g., [12], [13]), the damages identification and quantification after a hazardous event (e.g., [14], [15]), or the survey of common defects due to natural decay (e.g., [16], [17]). The latter are of great interest when moving to the field of periodic visual inspection of existing bridges, for which CV techniques can support traditional resource-demanding practices. In particular, to extract information from images (or also from objects and pixels), it is possible to adopt different approaches, such as classification, segmentation, feature detection, and object detection [17].

Classification categorizes objects or patterns in data, such as images, according to defined classes. The main aim is to assign a label or a class to an input image or a part of it. Different works employed classification techniques in the field of bridge inspections, such as Liang et al. [18], which proposed a three-level image-based approach for performing fault classification, component-level detection, and damage localization in post-disaster inspections of RC bridges. Fotouhi et al. [19] used CNNs to quantitatively classify different types of in-service damages in laminated composite structures. Segmentation consists of subdividing an image into multiple segments, simplifying the identification of regions of interest within the image. Through different operations, segmentation allows for locating objects and defining boundaries for the investigated images. As an example, following this approach, Saleem et al. [20] performed instance segmentation using deep learning to localize and classify cracks in bridge inspection images. For the sake of synthesis, other recent works using segmentation are following listed, i.e., Ding et al. [21], Sajedi and Liang [22], La et al. [23], Hoskere et al. [24], Kim et al. [25], Wang et al. [26], Wang et al. [27], Zhang and Lin [28]. Concerning feature detection, this approach consists of identifying and locating patterns (i.e., features) within images, going from simple lines to complex figures, to investigate the variations of the considered feature over time. In the field of bridge inspection, Jana et al. [29], [30] used feature detection to monitor the variation of cable tension measurement by retrieving information on a camera motion. Analogously, Kromanis and Kripakaran [31] employed feature detection tracking to determine structural displacements from images captured by multiple cameras.

Still, the main focus of this work is on object detection, which consists of identifying and localizing, within the image, specific objects of interest. Two main types of object detectors based on deep neural networks are widely used nowadays. The first is represented by *two-stage object detectors*, which base their operation on the presence of a couple of neural networks, that is, a *Region Proposal Network*, whose purpose is to identify candidate regions within the image which are likely to hold an object of interest, and a classification network, whose goal is to classify the content of the proposed regions. The other type of object detector is *single-stage* detectors, which use a single network to perform both localization and classification.

Examples of two-stage object detectors are R-CNN [32] and its successors, such as Fast R-CNN [33] and Faster R-CNN [34] (where R stands for Region). In the realm of two-stage detectors for defect detection in civil structures, Cha et al. [35] employed Faster R-CNN to recognize five typologies of surface damage on concrete and steel structures on a dataset composed of 2355 images, demonstrating good accuracy with a mean average precision of 87.8%. Li et al. [36] introduced a modified version of Faster R-CNN for identifying three types of concrete defects, showing a detection accuracy of 80.7% and a localization accuracy of 86% per image. Deng et al. [37] predisposed a Faster R-CNN on a dataset of around 5000 images to detect cracks in RC bridges despite interferences as handwriting automatically. Results were compared with those obtained using YOLOv2, showing the proposed approach's advantages. Deng et al. [38] employed Faster R-CNN to detect three typologies of defects (i.e., cracks, concrete delamination, and steel reinforcement exposure) trained on a dataset created by using four lasers (structured lights) and a depth camera, achieving low percentage of errors and high value of the F1 score (i.e., 83%).

Even though two-stage detectors often provide outstanding performance in object detection, the use of two neural networks implies a high computational burden, undermining their usability on constrained devices, such as unmanned aerial vehicles. Furthermore, the larger number of parameters

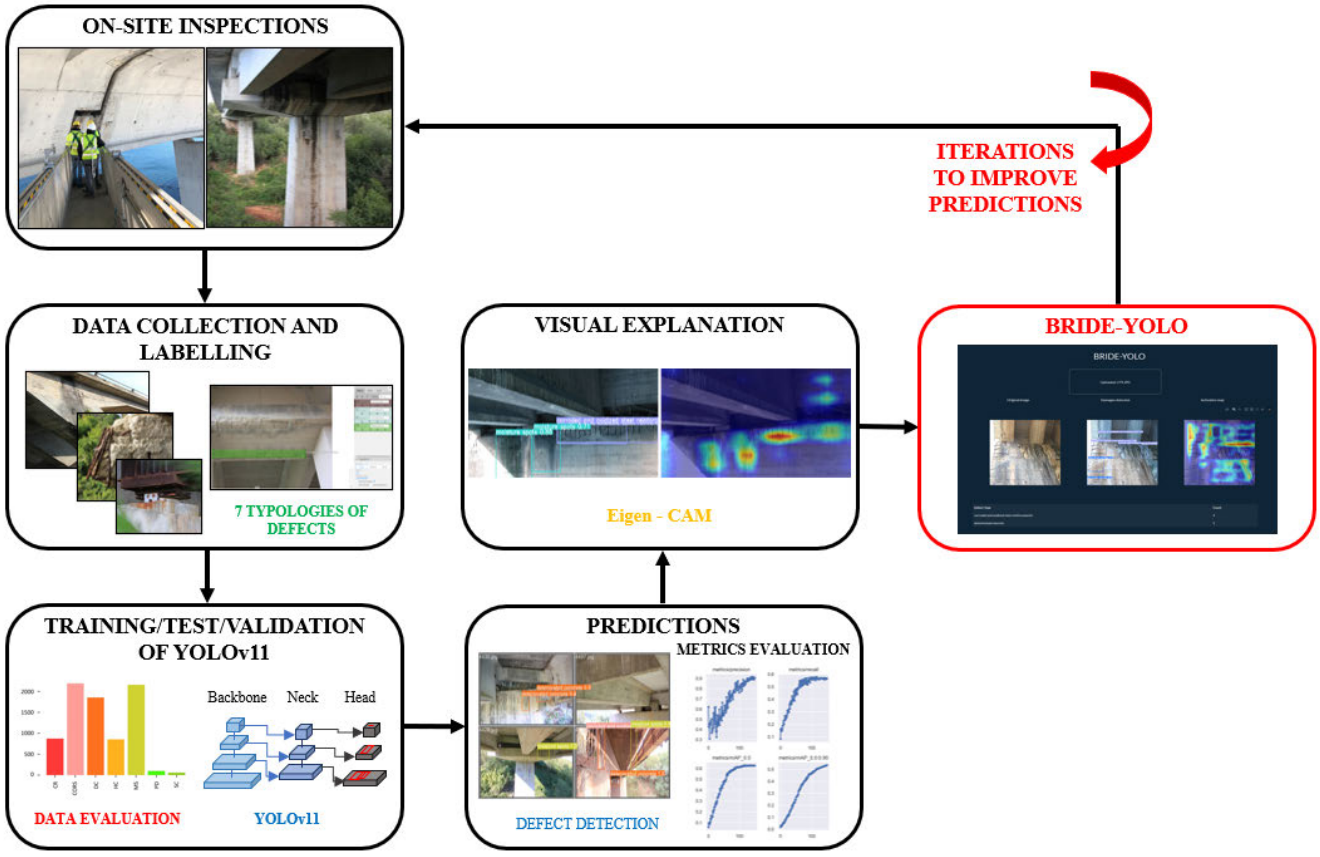


FIGURE 1. The proposed framework for BRIDE-YOLO.

generally used by these networks requires more data for proper training, posing a greater burden on domain experts during data gathering and labelling. One-stage detectors use smaller networks and provide limited accuracy if compared to two-stage detectors; however, the gap was filled with the adoption of more sophisticated approaches over time. Therefore, one-stage detectors are currently often preferred over two-stage detectors [39].

The main exponent of one-stage detectors is *YOLO* [40], an acronym which stands for *You Only Look Once* (even if other popular examples exist, such as Shot MultiBox Detector [41], [42]). YOLO detectors are composed of three different sections:

- *Backbone*: the backbone is a CNN-based architecture used by the model to extract relevant feature maps from images. These maps are then fed directly to the neck.
- *Neck*: the neck is the section that fuses the feature maps extracted by the backbone to reduce the loss of details that may occur during the feature extraction and add context information.
- *Head*: the head consists of several convolutional layers that output a vector containing the bounding box and category information of the targets in the input image.

Different examples of YOLO application for defect detection in existing bridges could be mentioned, such as the work by Maeda et al. [44], which developed a YOLO-based method to identify damages on a dataset of about 9000 images depicting road damages captured through a smartphone from a car, from which about 15500 instances were identified. The proposed approach achieved a maximum accuracy of 95%. Zhang et al. [45] combined YOLOv3 and a transfer learning algorithm with pre-trained weights for detecting four typologies of defects on RC bridge surfaces (i.e., crack, pop-out, spalling, and exposed rebar). Although good results were achieved, with an accuracy of about 80%, the main hurdle was represented by localization errors. Park et al. [46] employed YOLOv3 for real-time defect detection of cracks in concrete structures, aiming at defining the position and the size of the focused defect. Simulations and experimental tests showed good results with an accuracy and a precision of 94% and 98%, respectively. For the sake of conciseness, other important works employing YOLO can be mentioned, as the works by Arya et al. [47], Jiang et al. [48], Yu et al. [49], and Qiu and Lau [50]. Although different methodologies of object detection exist in the literature and the reliability of YOLO and its variants was several times assessed, BRIDE-YOLO presents some novelties with respect to other methods proposed in the field of structural health management.

FIGURE 2. Some samples of the original images acquired during the survey. Each image shows a different structural element of the bridge, with defects of various sizes and severity.

- First, a practice-oriented tool was proposed and made operative for supporting inspection operations to respond to the need for a real-time, easy-to-use, and portable decision support system. The tool can be easily deployed via a web service, and its code will be published on GitHub.
- The authors collected an extensive database during several on-site inspections, and seven classes of defects were manually labelled, exploiting a consensus-based procedure. This was designed to fill the need for a comprehensive dataset of heterogeneous bridge defects in real-world scenarios.
- The dataset was used to assess the capability of YOLO, along with specifically tailored architectural modifications, specifically the use of attention-based mechanisms, to support the decisions of the domain experts in the classification and localization of defects during a survey.
- Finally, the results achieved by the identified models were physically assessed by exploiting an eXplainable Artificial Intelligence (XAI) approach, specifically, Eigen-CAM, evaluating the coherence of the predictions performed by the trained neural network.

All the above aspects make BRIDE-YOLO a practical and, at the same time, scientific-based tool oriented to simplify the life of bridge inspectors.

III. MATERIALS AND METHODS

The overall framework for the proposed tool is reported in Figure 1, in which all the steps to realize BRIDE-YOLO were reported in a logic flow. Briefly, through real on-site inspections, a dataset of photos was collected and labelled according to some specific defect typologies. According to available data, different versions of YOLO11 were trained and compared through specific metrics to select the version providing the best results. Moreover, a visual explanation approach (i.e., Eigen-CAM) was used to explain the obtained prediction. The results were then used by means of the BRIDE-YOLO GUI, which allowed the upload of images

FIGURE 3. Example of manual labelling through CVAT [43].

and the detection of the considered defects. Although BRIDE-YOLO can be used "*as it is*" for supporting surveyors in the phase of bridge inspection, it was designed and developed to be subsequently upgraded, allowing the increment of the number of images and labels at the base of object detection training, which can then be iteratively re-proposed to improve the reliability of the tool itself and its prediction. A detailed description of each phase is reported in the next Sections.

A. DATASET

The dataset of collected images used in this work was initially proposed by [9] and further refined in [10]. The dataset comprises 6580 images, some shown in Figure 2, each one relative to one or more structural elements of existing RC bridges (e.g., derived from girders, decks, piers, pier caps, and abutment), presenting several types of defects of different extent and intensity. Data were gathered during on-site surveys and after labelled by domain experts following a subset of defects prescribed by the abovementioned Italian guidelines [4]. The images constituting the dataset were collected by authors over a large time period and under different environmental conditions. For the scope of the paper, this is an advantage, because the training of the network can be characterized by adequate generalization capabilities, accounting for different phenomena, such as occlusions and lighting variations.

The annotation was manually performed by domain experts exploiting the *Computer Vision Annotation Tool* (CVAT) [43]. To ensure accuracy and consistency in the annotated data, the ground truth was labelled by three different independent domain experts. A reliability check was then performed using a consensus procedure. In more detail, the dataset was first split into batches of 100 images each, with the latest composed of 80 images. For each image, one domain expert was tasked to detect each defect, highlighting its extent and typology through a box. Multiple boxes could be provided per image if more than one defect could be found. Once the labelling was complete, the second domain expert was asked to validate the provided labelling. If the two experts agreed, no more actions were necessary, and the labelling was validated. Otherwise, the second expert was tasked with proposing an alternative labelling, which was proposed, along with the first one, to the third domain expert. This would be validated if the third domain expert agreed with one of the proposals. Otherwise, the process was performed another time, with each domain expert having to consider the contributions provided by the others. This process was performed for each batch of images, randomly selecting the role each domain expert had to cover for each batch. A sample result of this labelling procedure is provided in Figure 3, where the annotation logic is shown.

Regarding the considered defects, according to [8], the following typologies were considered:

- *Corroded steel reinforcements*: refers to exposed steel reinforcement rebars that may be corroded after the spalling of the cover layer.
- *Cracks*: denotes thin or thick cracks resulting from degradation phenomena or static deficiencies.
- *Deteriorated concrete*: involves the superficial degradation of the concrete surface, such as swelling or scaling, induced by aggressive environmental conditions.

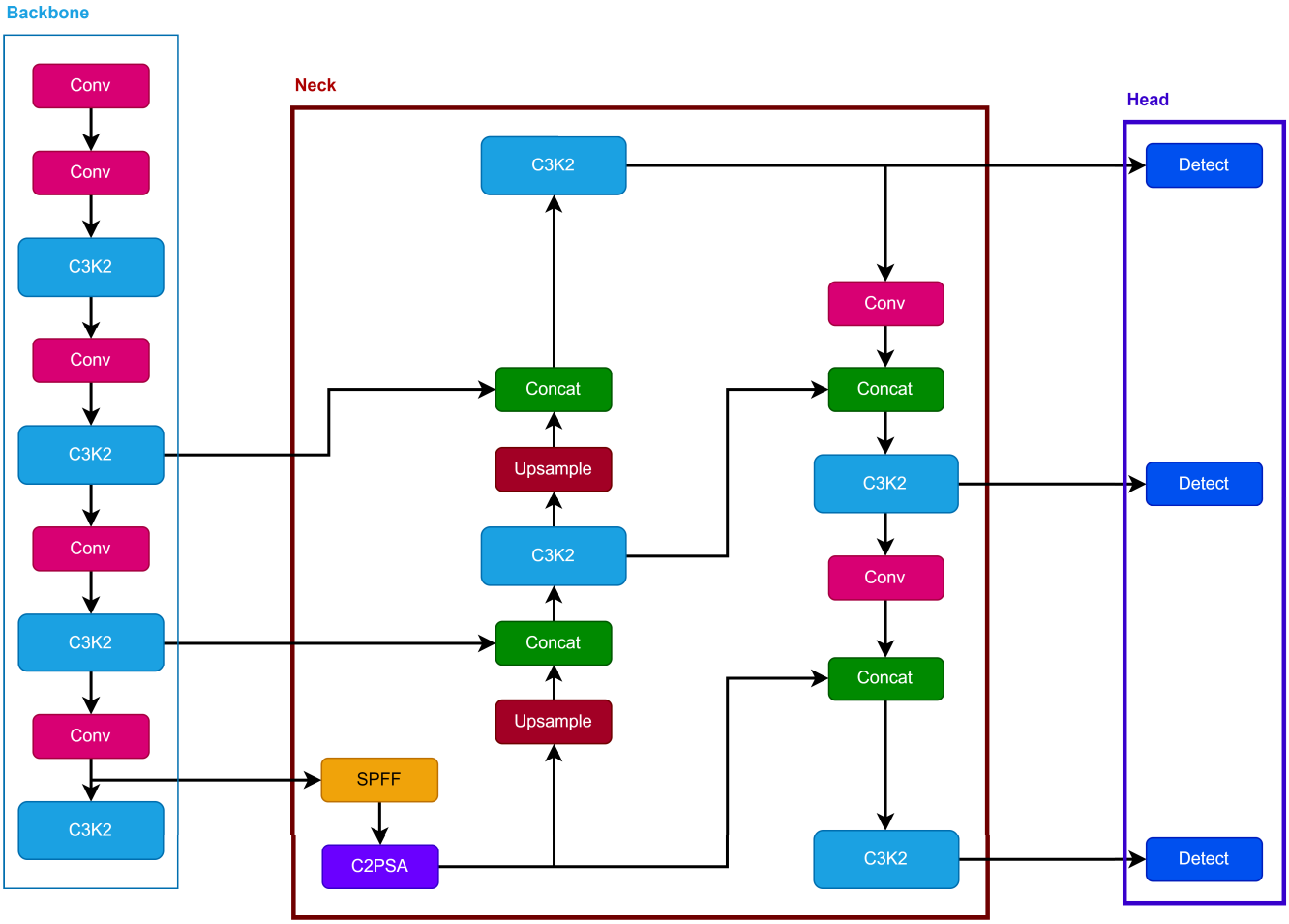


FIGURE 4. The base architecture of YOLO11.

- *Honeycombs*: describes casting errors leading to non-homogeneous areas with visible aggregates.
- *Moisture spots*: encompasses traces of drainage water and infiltrations on the concrete surface.
- *Shrinkage (Crazing) cracks*: pertains to the spread of thin cracks on concrete due to the drying out of moisture during the construction phase.
- *Pavement degradation*: involves defects (e.g., cracks, holes) affecting the asphalt layer of the road surface.

It is worth underlining that the Italian guidelines [4] propose a more refined taxonomy of defect typologies but, for the case at hand, the (relatively) small amount of available data for specific defect classes led to an under-representation of some of those classes. For this reason, visually related defect classes were grouped. For example, cracks were considered independently on their orientation. Vertical, horizontal, and diagonal cracks were all represented under the (generic) *crack* class. After these evaluations, the final dataset was characterized by seven typologies of defects, for a total of 10831 labels, as summarized in Table 1, which reports a specific acronym for each defect along with the total number of labels.

B. YOLO11 FOR OBJECT DETECTION

For BRIDE-YOLO development, the latest available model in the YOLO family was employed, that is, YOLO11 [51]. It is important to underline the rationale behind the selection of single-stage detectors and, specifically, of the latest iteration of the YOLO family. First, as already stated in Section II, single-stage detectors provide reliable performance with inference speeds which can be used for real-time implementation, as demonstrated by several benchmarks provided on foundational datasets [52]. As real-time processing of data streams is a matter of fundamental importance, both for during on-situ surveys and offline evaluations, using fast and non-resource-intensive models is mandatory in this specific application. Furthermore, the YOLO family achieved several important goals throughout its evolution, making it more appealing for selection as the base architecture for this work. First, starting from YOLOv8, the models could provide anchor-free detection, meaning that no predefined anchors were used for matching bounding boxes in the achieved results. Consequently, this choice allowed us to provide faster performance while retaining reliability. Furthermore, the constant evolution of YOLO

TABLE 1. Labels distribution for the collected dataset of 6,580 images.

Defect	Acronym	Number of labels
Cracks	CR	1138
Corroded and oxidized steel reinforcements	CORS	2928
Deteriorated concrete	DC	2448
Honeycombs	HC	1165
Moisture spots	MS	2962
Pavement degradation	PD	119
Shrinkage cracks	SC	71

models allowed state-of-the-art advancements in aspects such as the backbone or specific improved layers, which demonstrated superior performance over classic approaches such as SSD in construction assessment [53].

In detail, the release of YOLO11 made available five different versions, namely YOLO11n (nano), YOLO11s (small), YOLO11m (medium), YOLO11l (large), and YOLO11x (extra-large), each one with an increasing number of parameters, starting from the less (YOLO11n) to the most dense (YOLO11x). Each of the densities was tested in this study. When compared to its predecessors, YOLO11 provides three main improvements.

1) *Improved convolution mechanisms*: The C3k2 block is a direct evolution of the Cross Partial Stage (CSP) block used in the CSP bottleneck with two convolutions (C2f) block in YOLOv8 [54]. Specifically, the C3k2 block shares with the C2f block the same underlying architecture with the noticeable difference of using two smaller convolutions, therefore lowering the overall computational burden.
2) *Improved neck feature fusion*: The Cross Stage Partial with Spatial Attention (C2PSA) block embedded attention mechanisms to allow the neck to focus on relevant parts of the image
3) *Refined detection*: The use of Convolution - BatchNorm - SiLU (CBS) layers in the detection head allowed YOLO11 to refine the features used for normalization and used a sigmoid linear unit function for the activation of the final layer.

The base architecture of YOLO11 is shown in Figure 4.

1) CONVOLUTIONAL BLOCK ATTENTION MODULE

To further enhance the results of the bare YOLO11 object detector, this work modified the base architecture using the *Convolutional Block Attention Module* (CBAM). This attention mechanism was first proposed in [55] to model both *channel* (i.e., *which feature in the image is the most informative?*) and *spatial* (i.e., *what part of the image is the most informative?*) attention. Combining these attention mechanisms can be useful to identify the most discriminative parts and channels of an image.

Specifically, given an intermediate feature map denoted as $F_i \in \mathbb{R}^{C\times H\times W}$, with C number of filters, and H and W height and width of the attention map, respectively, the CBAM block provides a *channel refined* feature F_{CR} starting from the inferred single channel attention map $M_C \in \mathbb{R}^{C\times 1\times 1}$.

$$F_{CR} = M_C(F_i) \otimes F_i \tag{1}$$

In Equation 1 and in the following, the symbol $\otimes$ represents the element-wise multiplication. Let us underline how $M_C(F)$ is computed as the element-wise summation of the descriptors extracted by a Global Average Pooling (*GAP*) and a Global Max Pooling (*GMP*) to aggregate the information via a multilayer perception *MLP*, provided as an input to a sigmoid function.

$$M_C(F) = \sigma\left[MLP(GAP(F)) \oplus MLP(GMP(F))\right] \tag{2}$$

In Equation 2 the $\oplus$ symbol represents the element-wise summation. Once F_{CR} is known, the final refined feature F_{FR} can be computed as follows:

$$F_{FR} = M_S(F_{CR}) \otimes F_{CR} \tag{3}$$

As for M_S, it is computed using a 7×7 convolutional layer, as described in Equation 4.

$$M_S(F) = \sigma\left[f^{7\times 7}(GAP(F) \circ GMP(F))\right] \tag{4}$$

The overall output feature map of the CBAM block is then given by the element-wise summation of the input feature maps and its final refined version.

$$F_o = F_i \oplus F_{FR} \tag{5}$$

The structure of the CBAM module is shown in Figure 5.

The base YOLO11 model was improved by adding attention specifically to the neck. This was motivated by the need to provide more contextual attention during feature fusion, therefore increasing the capability of the network to fuse features at different levels and focus on traits relevant to the localization and classification of defects. Figure 6 shows the proposed architectural modifications.

C. VISUAL EXPLAINATIONS

Explainability is a matter of extreme importance in assessing the performance of complex deep learning models. In other words, as these architectures involve several complex non-linear transformations of the input to extract the output,

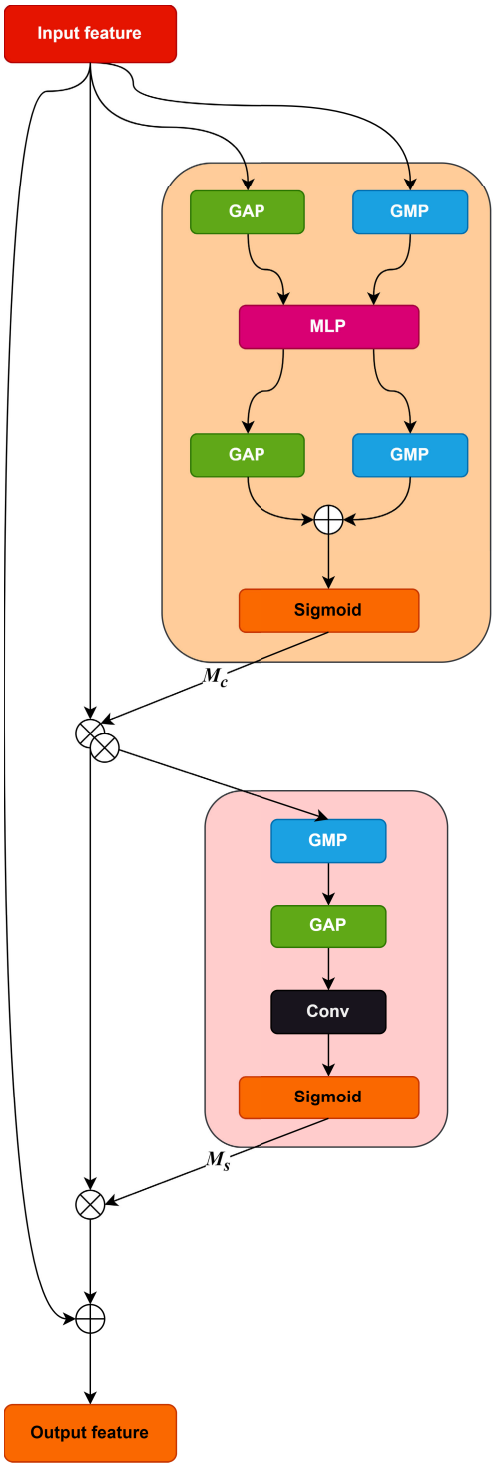


FIGURE 5. The CBAM block scheme.

it is difficult to provide a clear, simple, and straightforward interpretation of why the model yielded a certain decision without specific tools. In the specific framework of convolutional neural networks, visual explanations in the form of *class activation maps* (CAMs) are the most common way to provide a comprehensive and satisfactory interpretation of the results achieved by a model. Several methods exploits CAMs, including GradCAM [56], GradCAM++ [57], and Smooth-CAM [58]. An example of GradCAM and GradCAM++ application in the field of defects identification in RC bridges is provided by [8], in which visual explanations were used to evaluate qualitatively the accuracy of the employed CNNs.

Methods for extracting CAMs are usually divided into two groups, that is, methods which heavily rely on gradient computations, such as GradCAM and GradCAM++, and gradient-free methods, such as EigenCAM [59] or AblationCAM [60]. Specifically, gradient-free methods were developed in response to a common criticism of gradient-based methods, which were demonstrated to be afflicted by the problem of gradient saturation and, consequently, by the poor quality of visualization [60]; furthermore, gradient-based methods are often heavily reliant on the specific architecture and should be adapted accordingly. On the other hand, common criticisms of gradient-free methods lie in their computational burden, which is significantly higher than gradient-based techniques [61].

This work proposed a comparison between EigenCAM and its gradient-based variant, EigenGradCAM, which integrates the gradient computation used in GradCAM by considering a multiplicative term associated with gradients. This allowed the assessment of the qualitative performance of the models while still retaining the possibility to compare the effectiveness of two different approaches to CAM computation. The main idea behind EigenCAM was to assess the relevance of the features through the convolutional layers. To this end, EigenCAM considers the principal component of the learned representation, which explains most of the variance in the original data. The idea is that relevant features lie in this component, while non-relevant features can be filtered out.

Let I represents the image under investigation, whose dimensions are (i, j) pixels, and let W_k be the combined weight matrix of the first k layers of the neural network. The output of the projection of image I on the k-th layer is given as follows.

$$O_k = W_k^T I \tag{6}$$

Let us suppose that the size of O_k is (m, n) due to the change in dimensionality caused by convolutional layers. To identify principal components, Single Value Decomposition (SVD) can be used:

$$O_k = U \Sigma V^T \tag{7}$$

In Equation 7, U is an $m \times m$ orthogonal matrix, whose columns are the left singular vectors, Σ is a diagonal matrix of size $m \times n$, with singular values along the diagonal, and V is an $n \times n$ orthogonal matrix. Hence, the activation map using the Eigen-CAM method can be defined as

$$CAM_{Eigen} = O_k V_1 \tag{8}$$

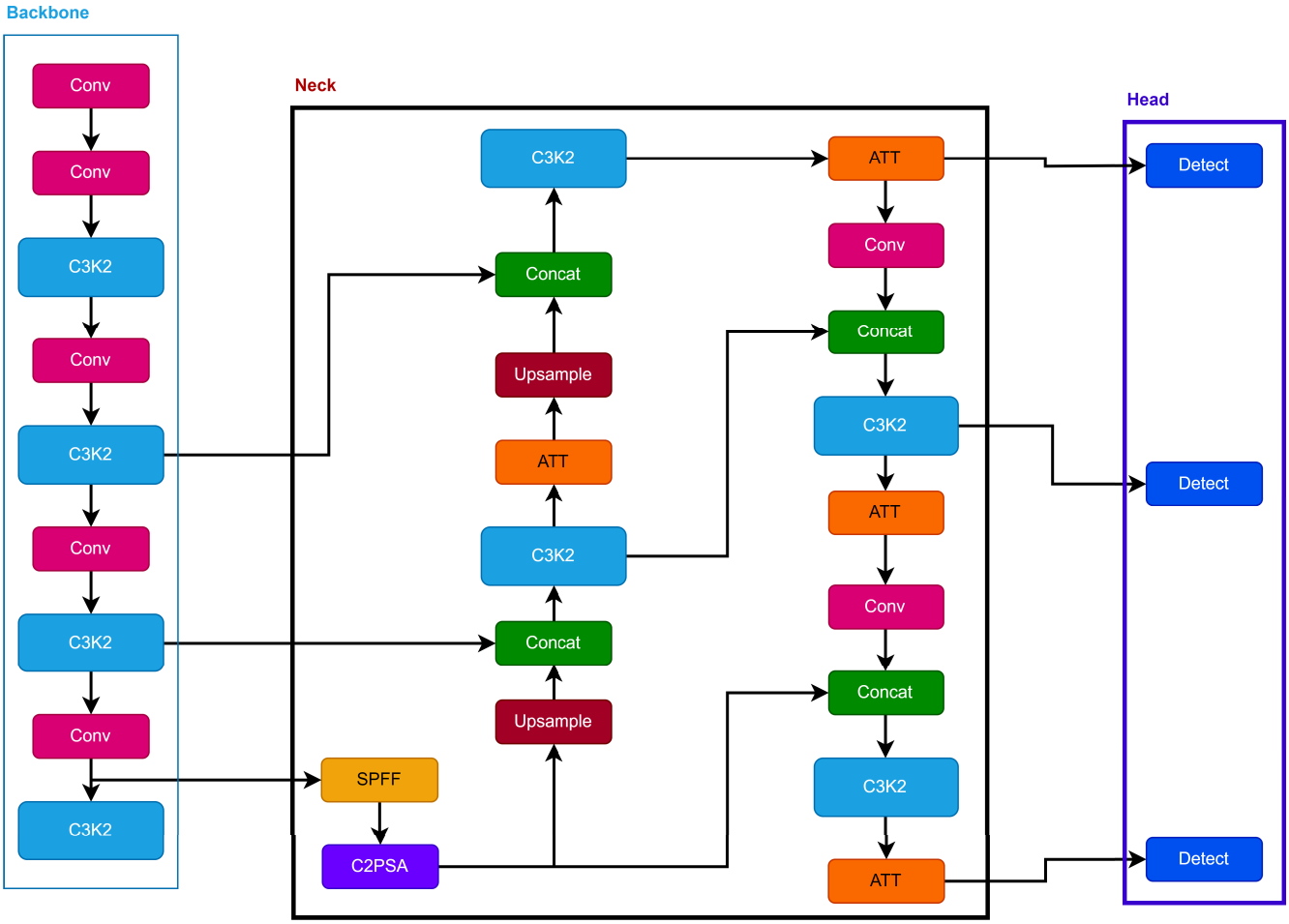


FIGURE 6. The architecture of YOLO11 with the integration of the attention mechanisms.

where V_1 is the first eigenvector in the V matrix. This activation map can then be projected onto the original image, highlighting the salient parts of the image as the ones with the higher activation values.

According to the evidences shown in [59] and [62], Eigen-CAM demonstrates robustness and reliability in providing consistent visual explanations. Furthermore, it only requires the learned representations at the final convolution layer, independently of other classification layers.

D. EVALUATION METRICS

The results were evaluated using four metrics tailored for object detectors [63], that is, *precision* (P), *recall* (R), *F1 score* (F1) and *mean average precision* (mAP). Specifically, these metrics measure the similarity between the bounding boxes predicted by the detector and those provided in the ground truth, providing independent scores for the location and the class of the object, which are synthesized in a single value. To this end, it is important to define the *Intersection over Union* (IoU) metric, a coefficient based on the Jaccard index [64]. Given a ground truth bounding box, B_{gt}, and its correspondent prediction, B_p, the IoU is defined as:

$$IoU = \frac{B_p \bigcap B_{gt}}{B_p \bigcup B_{gt}} \tag{9}$$

In other words, Equation 9 expresses the IoU as the ratio between the intersection and the union of the predicted bounding box and the ground truth bounding box. Therefore, a perfect match is given when $IoU = 1$, while if $IoU = 0$ the model completely misses the prediction. A visual interpretation of the IoU metric is provided in Figure 7.

Usually, the evaluation of object detectors implies using a threshold on IoU, which is commonly set above 0.5, meaning that the predicted bounding box overlaps the ground truth by more than 50%. However, this threshold can also be set according to the experimental setup: for example, if the ground truth bounding boxes are of small size, the threshold can be relaxed to consider lower IoU values, which do not significantly affect the validity of the predictions, as shown in [65]. In the proposed scenario, only thresholds above 0.5 were considered, mainly due to the relevant size of the ground truth bounding boxes.

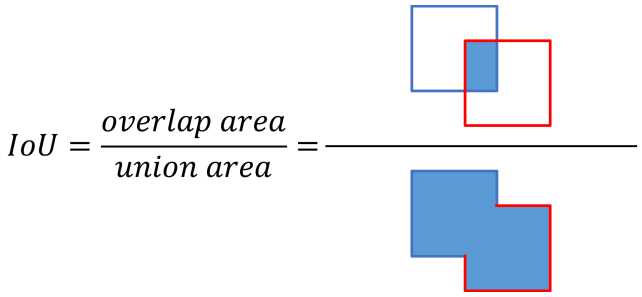


FIGURE 7. Intersection over Union.

Precision is generally defined as the ability of the model to localize and classify objects. To be quantified, one shall compute the ratio between correctly identified objects and the overall number of predictions. As for the recall, this metric represents the ability of the model to provide coherent and complete predictions and is computed as the ratio between the correct predictions and the total number of ground truth bounding boxes available. As such, to compute precision and recall, each detection must be categorized as follows:

- **True Positive**, when the ground-truth bounding box is correctly predicted.
- **False Positive**, when the predicted bounding box does not exist within the ground truth, or it is completely misplaced (and, therefore, $IoU = 0$).
- **False Negative**, when the ground-truth bounding box has no predictions associated.

Starting from these value, precision can be computed as follows:

$$P = \frac{TP}{TP + FP} \tag{10}$$

$$R = \frac{TP}{TP + FN} \tag{11}$$

where TP is the total number of true positives, FP the total number of false positives, and FN the overall number of false negatives.

From the definition of P and R, the *F1 score* can be derived, quantifying the accuracy of the statistical test, and evaluated according to the following expression:

$$F1 = 2\frac{P \cdot R}{P + R} \tag{12}$$

These metrics provide a quantitative evaluation of the model's effectiveness in localization and classification. However, object detectors also provide a confidence score for each detection. Consequently, it is possible to adjust precision and recall by this value, hence considering positive predictions whose confidence scores are above a certain threshold τ, and negative otherwise. Consequently, Equations 10 and 11 can be rewritten in the following form:

$$P(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau)} \tag{13}$$

$$R = \frac{TP(\tau)}{TP(\tau) + FN(\tau)} \tag{14}$$

In Equations 13 and 14, the values for $TP(\tau)$, $FP(\tau)$ and $FN(\tau)$ are the values of TP, FP and FN adjusted according to the threshold τ. Clearly, both $TP(\tau)$ and $FP(\tau)$ are inversely proportional to τ, while $FN(\tau)$ has a direct proportionality with the threshold. From the above considerations, the *average precision* (AP) can be computed as the area under the $P(\tau) - R(\tau)$ curve at k different values of τ. The expression for AP is the following:

$$AP = \frac{1}{N}\sum_{n=1}^{N} P_{int}(R_r(n)) \tag{15}$$

In Equation 15, $P_{int}(R_r(n))$ is a mathematical function modelling the interpolation between n points of precision and recall pairs at different values of τ. In contrast, $R_r(n)$ is the set of reference recall values for the n selected points. As a single AP can be computed for each one of the classes within the dataset, a synthetic index which considers the *mean* between all these classes is computed as the *mean Average Precision* metric:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{16}$$

where C is the total number of classes contained within the dataset. The mAP can be further extended by using different thresholds for the IoU metric: for example, $mAP0.5$ considers a threshold of 0.5 for the IoU, while $mAP0.5-0.95$ considers an average over all the IoU values in the range [0.5, 0.95] sampled at a step of 0.05.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

The experiments were performed on a machine equipped with an NVIDIA RTX 4090 GPU with 24 GB of video RAM, 64 GB of RAM, and an Intel Core i9-14900HK CPU. The language used for performing experiments was Python with the support of the Pytorch [66] and Ultralytics [54] libraries.

Experiments were designed to identify the most suitable version of YOLO11 and evaluate whether the insertion of architectural modification and processing improvements had a significant impact. To this end, a baseline was established using the base version of YOLO11 at all the different available densities. Furthermore, two fixed image resolutions were considered, that is, *low resolution* (i.e., 640×640 pixels), and *high resolution* (i.e., 960×960). This variability was introduced to test the robustness of trained models when different image densities are considered.

The hyperparameters were tuned using a mutation and crossover approach, mainly inspired by genetic algorithms [67]. Specifically, the default set of hyperparameters was randomly mutated by applying small random iterations to maximize the evaluation metrics considered. The resulting hyperparameters were reported in Table 2.

Finally, to provide statistical significance and avoid overfitting, all the reported results were averaged over a k-fold cross-validation procedure, with $k = 10$.

TABLE 2. Hyperparameters value selected after optimization.

Parameter	Optimized value
Initial learning rate	0.00998
Final learning rate	0.00986
Momentum	0.9134
Weight decay	0.00066
Warmup epochs	3.13215
Warmup momentum	0.70808
Box loss	6.62038
Class loss	0.49984
DFL loss	1.66513

B. OBJECT DETECTION RESULTS

1) METRICS EVALUATION

A total of ten experiments were performed to set the baseline, each with 5 iterations involving 100 training epochs. To ensure a fair comparison between the baseline and the architecture improved using CBAM, the weights of the networks were trained from scratch, i.e., the models were not individually pre-trained on the COCO dataset.

Table 3 describes the results achieved using the base architecture of YOLO11. Specifically, the best result is reported in blue, the second-best in green, and the third in red. The analysis of the baseline highlighted how the best-performing network was YOLO11x using the high resolution. This was mainly related to two different aspects.

The first was the number of provided labels, which was adequate to allow the denser model to train its larger number of parameters properly. This was confirmed by the fact that, generally, the models improved their behaviour when the density increased, except for the Nano model, which exhibited a sort of ''sweet-spot'' behaviour, possibly caused by a local optimum in the selection of data used for the specific training.

The second aspect was related to the resolution of the provided images, which allowed the network to capture fine-grained details on the defects of interest. In this case, it was clear that each model, independently from the density, benefited from the higher resolution, with an improvement in terms of $F1$-score ranging from 17.85% for the Nano model to 5.35% for the Medium one.

Table 4 describes the results achieved using the improved version of YOLO11 using the CBAM attention module. With this insertion, detection results were generally more predictable, with the best-performing models being the largest, confirming that higher resolutions yielded improved metrics values. Specifically, the Extra model with high resolution was confirmed to provide the best performance, with an overall improvement of 2.38% in terms of F1 score, 2.41% in terms of mAP 0.5, and 2.05 in terms of mAP 0.95.

Still, the most interesting results are probably related to the models fed with low-resolution images. Let us focus on the Extra model: the attention module allowed to yield an improvement of 9.61% in terms of $F1$-score, of 6.69% in terms of mAP 0.5, and 7.89% for the mAP 0.95, effectively bridging the gap highlighted by the baseline models in terms of quantitative metrics. An interesting point could be raised if the inference processing speeds were also considered. Specifically, from Table 3, the Extra model fed with high-resolution images required a processing time of 11.73 milliseconds, resulting in an average of 85.25 FPS on the reference implementation. If the CBAM block was inserted, and the resulting Extra model was fed with low-resolution images, an average processing time of 9.14 milliseconds was required per image, resulting in a 109.40 average FPS on the reference implementation. Therefore, the insertion of the CBAM attention mechanism allowed the improvement of the overall quantitative metrics provided by the model while also reducing the computation time and, therefore, increasing the feasibility of deploying the model directly in the field on constrained hardware.

2) CONFIDENCE BOUNDARIES

A proper quantitative evaluation of the results achieved by the models cannot disregard the evaluation of the confidence boundaries to which the provided predictions refer. Conceptually, object detectors from the YOLO family provide two types of confidence scores. The first type is box confidence, that is, a measure of the confidence of the model in establishing that a predicted bounding box contains an object of interest. In other words, the box confidence combines the certainty of the model in stating that a bounding box contains an object with the IoU between the prediction and the provided ground truth. The other type of confidence is class confidence, which expresses how certain the model is that the detected object belongs to a specific class according to a conditional probability. These two scores are combined, providing a single conference score, which is heavily related to the selected value of IoU. Therefore, by evaluating how precision, recall, and F1 score metrics behave at different confidence scores, it is possible to have a further in-depth quantitative assessment of the performance of the model. Figure 8 shows the behaviour of precision, recall, and F1 score for YOLO11x (Figures 8a, 8b, 8c) and YOLO11x after the addition of CBAM layers (Figures 8d, 8e, 8f) at different levels of confidence score. Overall, the analysis highlighted how the insertion of the CBAM mechanism impacted all the proposed metrics, providing a higher value of precision, recall, and F1 score at higher confidence scores. From Figure 8, it appears this is mainly related to a better characterization of the pavement degradation and shrinkage cracks objects, which also are the less represented types of object within the dataset, mainly in terms of recall. This could be confirmed for shrinkage cracks by considering the confusion matrices in Figure 9, computed using a standard confidence score of 0.5. In other words, using CBAM attention improves the detection performance for this class. Interestingly, the same effect cannot be inferred for pavement degradation, as in this

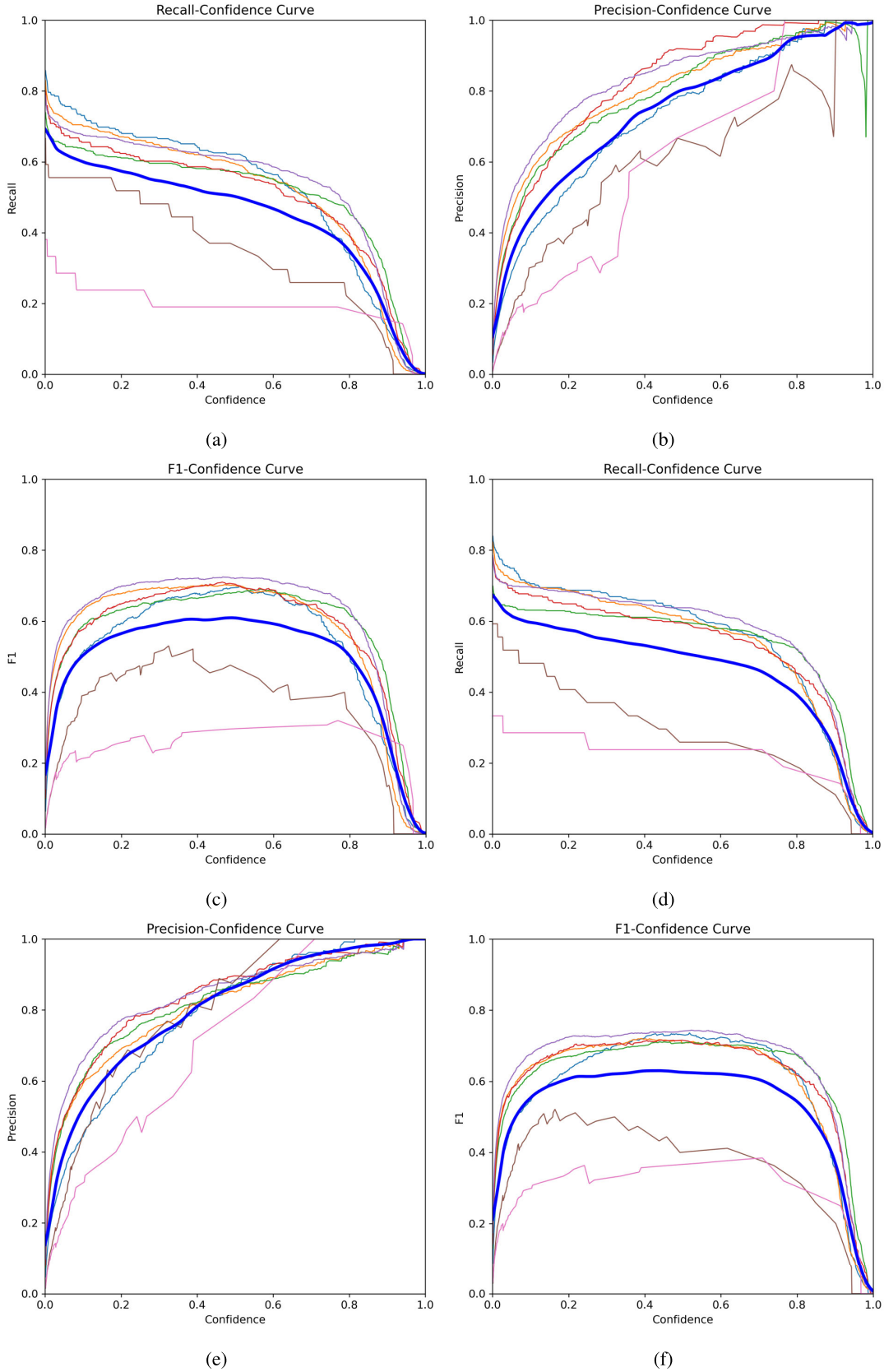


FIGURE 8. Variation of recall, precision, and F1 score at different confidence scores for YOLO11x (Figures 8a, 8b, 8c) and YOLO11x after the use of CBAM layers (Figures 8d, 8e, 8f). The light blue line is relative to CR, the orange to CORS, the green to DC, the red to HC, the purple to MS, the brown to PD, the pink to SC, and the heavy blue is the average for all classes.

TABLE 3. Results achieved using the baseline version of YOLO11.

Density	Resolution	P (%)	R (%)	F1 (%)	mAP 0.5 (%)	mAP 0.5 − 0.95 (%)	Speed (ms)
Nano	Low	46.66	37.86	41.80	38.97	20.28	2.06
Nano	High	**74.29**	**49.83**	**59.65**	**54.33**	**35.00**	2.42
Small	Low	46.44	35.82	40.45	35.96	18.36	2.24
Small	High	62.35	40.73	49.27	45.99	27.43	3.24
Medium	Low	61.70	43.70	51.16	47.23	27.29	3.11
Medium	High	**71.86**	47.35	**57.08**	**52.58**	**33.38**	5.75
Large	Low	69.57	41.38	51.89	46.22	27.15	3.26
Large	High	65.49	**49.62**	56.46	51.95	32.90	6.49
Extra	Low	67.46	45.35	54.24	50.80	31.83	5.49
Extra	High	**79.93**	**50.62**	**61.98**	**56.97**	**39.81**	11.73

case, the use of the attention layer decreases the performance at lower confidence scores. However, attention mechanisms appear to provide a better response to higher confidence scores: in other words, attention allows the network to provide a more reliable and confident answer, that is, predict bounding boxes which are more aligned to the ground truth, thanks to the combined effect of evaluating the effect of space and channel on the overall predicted boxes. Another interesting effect that can be inferred from attention is that all the observed curves are smoother, avoiding sudden drops in precision and recall at high confidence scores. This implies a stabilization effect provided by the CBAM attention layers, which improves the overall confidence and reliability of the model.

3) IN-DEPTH CLASS RESULT EVALUATION

Let us focus on the per-class results achieved by the Extra model with and without the use of CBAM attention. The results were reported in Figure 9 in the form of confusion matrices.

The confusion matrices clearly show how the network, without the use of attention (Figure 9a), provided a higher number of mismatches (represented by non-zero values outside the diagonal). Furthermore, the overall number of missed detections, represented by the percentages on the bottom row, was slightly lower in each class for the network embedding the CBAM mechanism, except for pavement degradations. This was also valid for the rate of false positives, represented by the rightmost column in the representation. Overall, the analysis highlighted how the network modified using the attention mechanism outperformed the baseline version in identifying each type of defect, except for pavement degradations.

It is worth noting that, from the perspective of a safety-critical scenario, the false negatives are usually considered the most undesirable errors, as the model misses the detection of potentially safety-critical situations. About this latter aspect, most of the false negatives were observed on SC defects, which were the most under-represented classes. This is an additional quality check for the proposed methodology that, as expected, highlighted the necessity of more data for training the network (especially for the under-represented classes) to work properly under all required circumstances.

Still, the comparison shown in Figure 10 between the ground truth and the predictions demonstrated the capability of the network to detect most of the labelled defects, failing in very few situations, especially when small objects were considered.

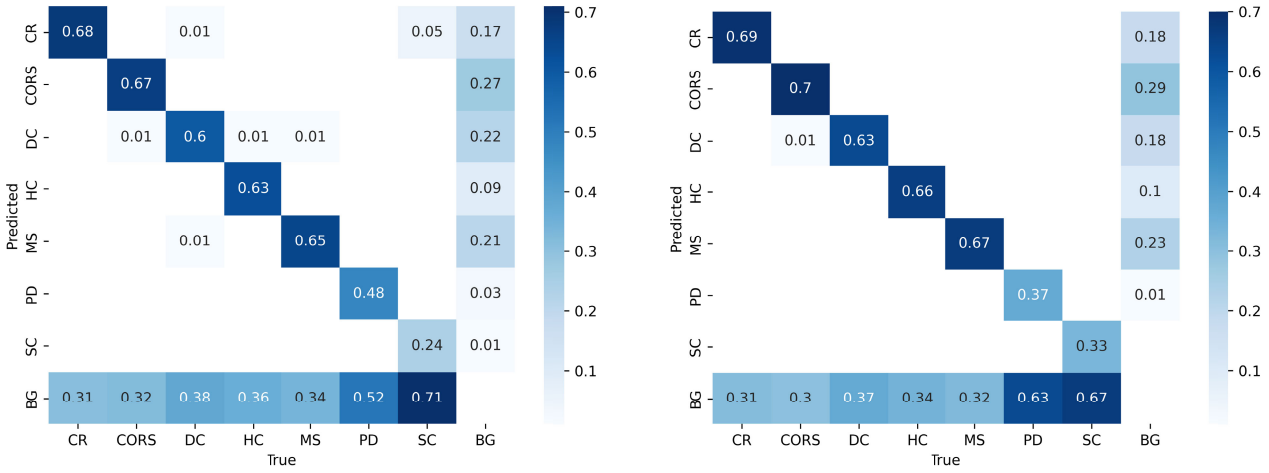
Finally, it is important to properly discuss the causes behind false positives and false negatives, as shown in the confusion matrices. Specifically, object detectors provide two types of false positives and negatives.

The first type is related to the misclassification of objects in the provided samples, obviously extended to a multiclass scenario. The confusion matrices describe this effect as misclassification of the bounding boxes and represent a scenario where the model is correctly localizing an object but incorrectly assigning it a class label. The reasons behind this misclassification are to be found in the visual appearance of the object, which implies that the model was not able to characterize meaningful features in the backbone properly and, therefore, provides high confidence scores to wrong class labels, compromising the overall classification of the localized bounding boxes. From the analysis provided in Figure 9, it was clear how both YOLOv11x and its modified version with the CBAM attention layer yielded good results in terms of the correctness of the classification, with a few misclassified samples, especially when visually related defects, such as DC, CORS, HC and MS, were considered. This effect was further reduced when the CBAM attention mechanisms were considered.

As for the other types of false positives and negatives, these are commonly referred to as *background* false positives and *background* false negatives. The background false positives describe situations where the model finds objects during validation with no ground truth associated. Background false positives are not necessarily an indication of poor performance. Instead, given the proper representational capability of the model, it is most likely that background

TABLE 4. Results achieved using improving the YOLO11 base architecture by inserting the CBAM block.

Density	Resolution	P (%)	R (%)	F1 (%)	mAP 0.5 (%)	mAP 0.5 − 0.95 (%)	Speed (ms)
Nano	Low	51.95	26.64	35.22	25.27	11.35	2.40
Nano	High	37.79	31.88	34.58	27.45	12.51	2.48
Small	Low	62.49	37.77	47.08	40.74	21.84	2.56
Small	High	52.87	45.05	48.65	45.44	25.57	4.42
Medium	Low	67.88	**53.19**	59.64	55.04	33.47	4.65
Medium	High	75.50	47.88	58.60	54.75	34.48	9.83
Large	Low	76.83	48.56	59.51	54.51	34.92	5.26
Large	High	**77.02**	50.82	**61.24**	**56.37**	**35.46**	10.88
Extra	Low	**81.82**	**52.36**	**63.85**	**57.49**	**39.72**	9.14
Extra	High	**82.87**	**52.61**	**64.36**	**59.38**	**41.86**	19.22



(a) (b)

FIGURE 9. Confusion matrices for YOLO11x without (on the left) and with (on the right) the integration of CBAM attention blocks in the detection head.

false positives are related to missing labels in the dataset, which are likely to occur due to the effort required by domain experts to process large-size datasets. As for background false negatives, these are associated with the need for the model to gather more data to represent specific classes of objects. For example, by inspecting Figure 9, it could be found that the model is biased in identifying defects such as PDs or SCs. This was mainly related to the fact that these defects were less represented and should be addressed in future versions of the tool by including more instances of these defects in the gathered dataset.

C. INTERPRETABILITY OF THE DETECTION MODEL

As described in Section III-C, EigenCAM and EigenGradCAM were exploited to provide an interpretation of the results achieved. Specifically, Figure 11 shows the activation maps from the latest C3k2 layer of YOLO11x with and without using the CBAM block when both a gradient-free and a gradient-based method were considered. Let us underline how "warmer" areas in the heatmap (i.e., regions whose colour is shifted towards the red tone) indicate higher activation for the network. Overall, two concurring effects could be observed from these results.

- First, the CAMs provided by gradient-free methods, shown in Figures 11a, 11e, 11c, 11g were usually much more diffused with respect to the ones provided by gradient-based methods. This implied that gradient-free methods considered activations characterizing the capability of the network to fuse features at different scales, therefore integrating information about different contexts at different scales.
- Second, the CAMs related to models using attention mechanisms provided activation boundaries more defined and focused than those provided by the base model. This is especially true when comparing Figure 11d to Figure 11b and Figure 11h to Figure 11f, which showed the activation maps gathered using EigenGradCAM with attention (Figures 11d and 11h) and without attention (Figures 11b and 11b). As shown, the activations of YOLO11x with CBAM are much more

(a) Ground truth

(b) Predictions

FIGURE 10. A comparison between the ground truth provided to the network during validation (shown in Figure 10a) and the predictions provided by the network (shown in Figure 10b).

focused than the ones provided by YOLO11x without CBAM.

This provided an effective indication of how the insertion of attention mechanisms could help the network in focusing

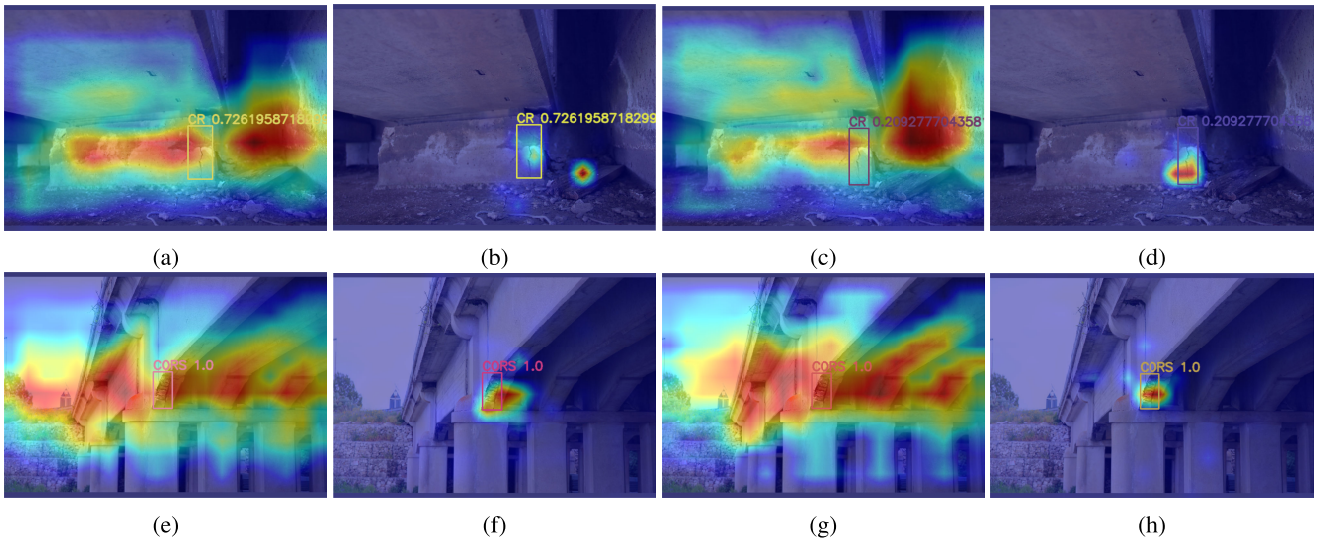


FIGURE 11. Interpretability of a small subset of predictions using EigenCAM and EigenGradCAM on YOLOv11x and its modified version with CBAM attention layers. Specifically, the results of CAM computation using both algorithms for two images were shown, with Figures 11a, 11e, 11b and 11f depicting the results achieved by the bare version of YOLO11x, and Figures 11c, 11g, 11d and 11h the results achieved using the modified version of YOLO11x. Overall, it was evident how the CAMs computed using EigenCAM (i.e., Figures 11a, 11e, 11b, 11f) looked at the overall context of the image to provide the prediction, while CAMs computed using gradient-based methods were directly focused on the prediction, providing useful hints on the distribution of the gradients throughout backpropagation.

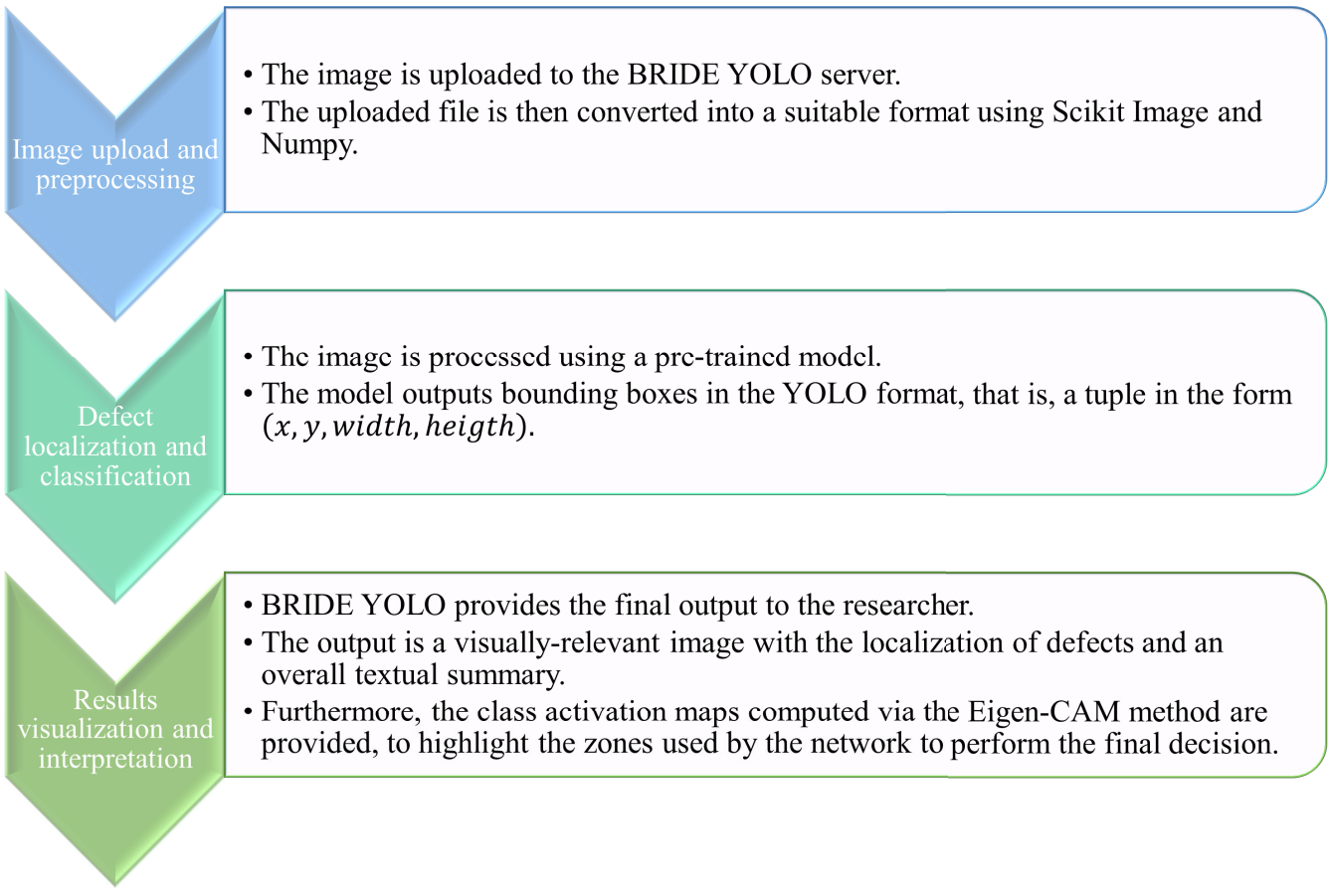


FIGURE 12. The processing pipeline followed by the BRIDE-YOLO tool.

the overall detection capabilities towards the most informative parts of the image.

It is worth noting that the use of a CAM method, besides confirming the performance in object detection,

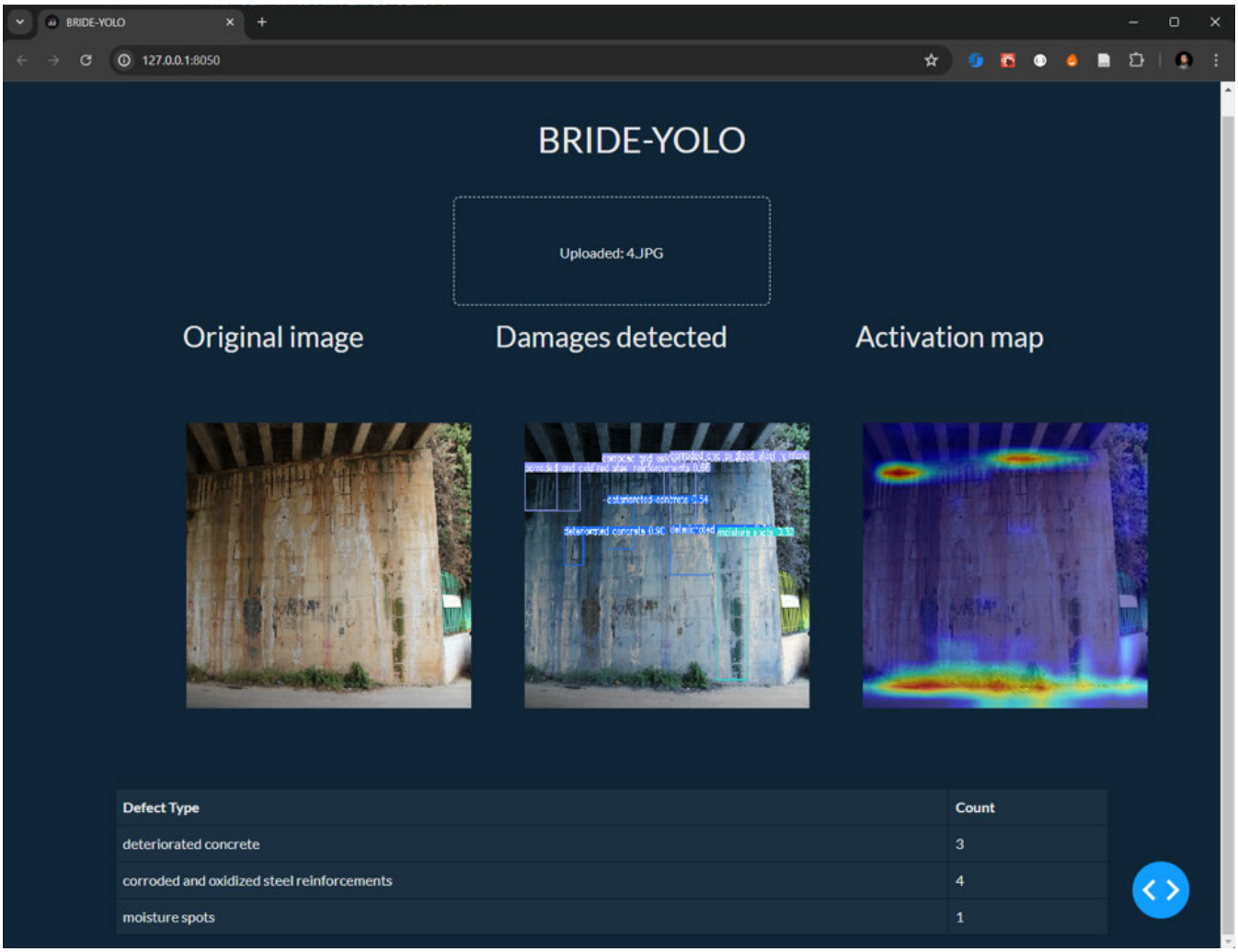


FIGURE 13. The GUI of BRIDE-YOLO. The interface shows the original image, as uploaded by the user, the detections provided by the model, and the activation map computed using EigenGradCAM. It is important to underline how the detections were not only shown in terms of counting results, but also directly on the original image, allowing for a rapid visual assessment by the surveyor.

can directly support through a graphic result the tasks of surveyors, especially when dealing with several defects (of different typologies) characterizing the focused image and catching the surveyor's attention on specific parts of the image.

V. BRIDE-YOLO: TOOL DESCRIPTION AND USAGE

Using the results highlighted in the precedent Sections, the main aim of the paper is to propose the practice-oriented tool named BRIDE-YOLO, which stands for BRIdge DEfects detection via YOLO. BRIDE-YOLO provides an easy-to-use visual tool for automatically detecting (i.e., localizing and classifying) defects, starting from raw data associated with an existing bridge (i.e., images). It is worth specifying that the proposed tool was idealized, designed, and proposed specifically for bridges. If from one hand, defects like corrosion can be found in other structures such as buildings, on the other hand, some defects (e.g., honeycombs) are typical of bridges. Hence, as a rule of thumb, if varying the defect types and the type of structures, it should be necessary perform another labelling and collect different data. Figure 12 provides a synthetic sequential block diagram describing the working principle of the proposed tool as well as the interaction between the user and the GUI of BRIDE-YOLO.

The workflow on which BRIDE-YOLO is based is composed of three main phases: (a) image upload and pre-processing, (b) defect localization and classification, and (c) results visualization and interpretation. As for step (a), the user can upload a raw image through a specific interface provided by the underlying web service. Still, the raw image is converted into a numerical array using the computing engine provided by the Scikit Image [68] and Numpy libraries. After this initial pre-processing, the input is ready to be processed (step (b)). In detail, using a model previously trained on the proposed dataset, the image is processed according to the components described

in Section III-B, and the expected output is returned to the user. This latter is returned in a compatible form with YOLO, i.e., a four-term tuple composed by the centre of the bounding boxes, expressed as the normalized width and height of the original image (in percentage), along with their effective width and height (in pixels). Ultimately, the output of step (c) is provided to the user with a comprehensive visual representation. This latter comprises the originally uploaded image, the localization of the defects, the related typologies, and the activation maps (obtained through Eigen-CAM) of the zones used by the network to perform the final decision.

The GUI available to users is shown in Figure 13, which shows the specific interface commands to upload the raw image and the visual feedback provided regarding bounding boxes and activation maps. The GUI was designed and developed using an agile methodology, where the first step was to identify the minimum set of functional requirements via interaction with domain experts, that is, surveyors who would use the system directly in the field. The functional requirements were mainly focused on the simplicity of usage, the responsiveness of the system, and the possibility of using it without reliable network connections, which could be unavailable in certain situations when working on-site. Afterwards, a simple interface was designed and implemented using specific Python libraries such as Dash and Plotly [69]. The first implementation was then tested by gathering the feedback of the surveyors via a simple Questionnaire for User Interface Satisfaction (QUIS), whose main aim was to assess whether the readability, organization of information and learning curve were adequate for the usage of the tool on devices with constrained screens, such as tablets, directly on the field, i.e., under challenging environmental conditions. The GUI was refined until a sufficient score was achieved in these tests. Some limitations remain, such as responsive design, smartphone adaptability, and capabilities for the system to operate in a federated environment, which will be addressed in future releases of the tool.

Finally, another feature of BRIDE-YOLO is the count of the detected defects, which is reported in the table in the bottom part of the GUI (see Figure 13). In particular, the table indicates the number of detected boxes and the class associated with each box, providing an easy-to-use and comprehensive interpretation of the status of the bridge. Obviously, as for the tool operation, BRIDE-YOLO requires the use of a model already trained on a dataset related to the field of the problem under investigation (for the case at hand, the YOLO11x version with CBAM attention shown in section IV-B was selected).

VI. CONCLUSION

The paper presents BRIDE-YOLO, a deep-learning-based tool that can be employed during or after on-site inspections to automatically detect typical defects in existing reinforced concrete (RC) bridges. The tool was developed based on an initial dataset of 6580 real images, which were labelled by domain experts. More than ten thousand instances of defect were obtained by looking at seven typologies of recurrent defects in existing RC bridges, which were used for training different scaled models of the last version of YOLO (i.e., YOLO11), as one of the most famous single-stage object detectors. After training, testing, and validation, the best performance was achieved through YOLO11x by integrating the CBAM attention module in the neck. The outcomes of this latter were also assessed through a visual explanation method based on class activation maps (CAMs), namely Eigen-CAM, which showed the reliability of the proposed network in the tasks of defect localization and classification. Finally, the pipeline and the first version of the graphical user interface (GUI) of BRIDE-YOLO were presented, and all functionalities provided by the tool were described, showing a simple, straightforward, and effective interaction between the user and the tool itself. The main aim of BRIDE-YOLO is to support engineers and practitioners involved in on-site inspections of RC bridges by reducing the high costs and time required for detailed assessment and overcoming the main limitations characterizing this phase, such as subjectivity and lapses in attention.

Several advantages are offered by a tool like BRIDE-YOLO. Firstly, it represents a practice-oriented tool directly usable in inspection phases. In addition, it is trained on a proper database, accounting for seven classes of typical defects in existing RC bridges. Finally, the obtained results are physically explained through an explainability method such as Eigen-CAM.

Future developments of the tool will aim to improve the model, first by expanding the proposed dataset, and subsequently to increase the current classes of identified defects, according to the current guidelines [4]. Still, additional efforts will be required to characterize the features of the defects, i.e., intensity and extent, and for differentiating the nature of the defects (e.g., differences between chloride-induced and carbonation corrosion). Afterwards, more architectural improvements will be assessed, including the impact of using different attention modules in various parts of the architecture or evaluating the use of advanced techniques proposed by cutting-edge advancements, such as improved versions of the CSPNet architecture in the backbone or the adoption of methods to minimize the impact of NMS on predicted bounding boxes.

These aspects could lead BRIDE-YOLO to act as a decision support system for driving users in further decisions (e.g., retrofit interventions). Still, the tool could be implemented in framed platforms at the disposal of road management companies, which could compare the results obtained at different inspections over time to assess the decay evolution of the inspected structural elements. To this scope, other information could be considered for ensuring an overall evaluation, such as sensor data, bridge structural information, and any other data aimed at providing a vision on the current and future structural performance. In the end,

BRIDE-YOLO could be implemented in other means to perform bridge inspections, such as unnamed aerial vehicles (UAVs), to consistently reduce the time and costs of on-site inspections.

REFERENCES

[1] F. Bazzucchi, L. Restuccia, and G. A. Ferro, "Considerations over the Italian road bridge infrastructure safety after the polcevera viaduct collapse: Past errors and future perspectives," *Frattura ed Integrità Strutturale*, vol. 12, no. 46, pp. 400–421, Sep. 2018.

[2] E. Farneti, N. Cavalagli, M. Costantini, F. Trillo, F. Minati, I. Venanzi, and F. Ubertini, "A method for structural monitoring of multispan bridges using satellite InSAR data with uncertainty quantification and its pre-collapse application to the Albiano-Magra bridge in Italy," *Struct. Health Monitor.*, vol. 22, no. 1, pp. 353–371, Jan. 2023.

[3] M. G. Durante, L. Di Sarno, P. Zimmaro, and J. P. Stewart, "Damage to roadway infrastructure from 2016 central Italy earthquake sequence," *Earthq. Spectra*, vol. 34, no. 4, pp. 1721–1737, Nov. 2018.

[4] *Linee Guida Per la Gestione Del Rischio Dei Point Esistenti e Delle Istruzioni Operative Per L'Applicazione Delle Linee Guida Stesse*, MIT, Cambridge, MA, USA, 2021.

[5] B. F. Spencer, V. Hoskere, and Y. Narazaki, "Advances in computer vision-based civil infrastructure inspection and monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, Apr. 2019.

[6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[7] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[8] A. Cardellicchio, S. Ruggieri, A. Nettis, V. Renò, and G. Uva, "Physical interpretation of machine learning-based recognition of defects for the risk management of existing bridge heritage," *Eng. Failure Anal.*, vol. 149, Jul. 2023, Art. no. 107237.

[9] A. Cardellicchio, S. Ruggieri, A. Nettis, C. Patruno, G. Uva, and V. Renò, "Deep learning approaches for image-based detection and classification of structural defects in bridges," in *Image Analysis and Processing* (Lecture Notes in Computer Science), P. L. Mazzeo, E. Frontoni, S. Sclaroff, and C. Distante, Eds. Cham, Switzerland: Springer, 2022, pp. 269–279.

[10] A. Cardellicchio, S. Ruggieri, A. Nettis, N. Mosca, G. Uva, and V. Renò, "On the use of YOLOv5 for detecting common defects on existing RC bridges," in *Multimodal Sensing and Artificial Intelligence: Technologies and Applications III*, vol. 12621. Bellingham, WA, USA: SPIE, 2023, pp. 134–141.

[11] S. Ruggieri, A. Cardellicchio, A. Nettis, V. Renò, and G. Uva, "Using machine learning approaches to perform defect detection of existing bridges," in *Proc. 19th ANIDIS Conf., Seismic Eng. Italy Proc. Struct. Integrity*, vol. 44, Jan. 2023, pp. 2028–2035.

[12] S. Ruggieri, A. Cardellicchio, V. Leggieri, and G. Uva, "Machine-learning based vulnerability analysis of existing buildings," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103936.

[13] A. Cardellicchio, S. Ruggieri, V. Leggieri, and G. Uva, "View VULMA: Data set for training a machine-learning tool for a fast vulnerability analysis of existing buildings," *Data*, vol. 7, no. 1, p. 4, Dec. 2021.

[14] M. Hamidia, S. Mansourdehghan, A. H. Asjodi, and K. M. Dolatshahi, "Rapid post-earthquake structural damage assessment using convolutional neural networks and transfer learning," *Measurement*, vol. 205, 2022, Art. no. 112195.

[15] P. D. Ogunjinmi, S.-S. Park, B. Kim, and D.-E. Lee, "Rapid post-earthquake structural damage assessment using convolutional neural networks and transfer learning," *Sensors*, vol. 22, no. 9, p. 3471, 2022.

[16] J. Guo, P. Liu, B. Xiao, L. Deng, and Q. Wang, "Surface defect detection of civil structures using images: Review from data perspective," *Autom. Construct.*, vol. 158, Feb. 2024, Art. no. 105186.

[17] M. Zakaria, E. Karaaslan, and F. N. Catbas, "Advanced bridge visual inspection using real-time machine learning in edge devices," *Adv. Bridge Eng.*, vol. 3, no. 1, Dec. 2022, Art. no. 105186.

[18] X. Liang, "Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 34, no. 5, pp. 415–430, May 2019.

[19] S. Fotouhi, F. Pashmforoush, M. Bodaghi, and M. Fotouhi, "Autonomous damage recognition in visual inspection of laminated composite structures using deep learning," *Compos. Struct.*, vol. 268, Jul. 2021, Art. no. 113960.

[20] M. R. Saleem, J.-W. Park, J.-H. Lee, H.-J. Jung, and M. Z. Sarwar, "Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network," *Struct. Health Monitor.*, vol. 20, no. 4, pp. 1760–1777, Jul. 2020.

[21] W. Ding, H. Yang, K. Yu, and J. Shu, "Crack detection and quantification for concrete structures using UAV and transformer," *Autom. Construct.*, vol. 152, Aug. 2023, Art. no. 104929.

[22] S. O. Sajedi and X. Liang, "Uncertainty-assisted deep vision structural health monitoring," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 36, no. 2, pp. 126–142, Feb. 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12580

[23] H. M. La, T. H. Dinh, N. H. Pham, Q. P. Ha, and A. Q. Pham, "Automated robotic monitoring and inspection of steel structures and bridges," *Robotica*, vol. 37, no. 5, pp. 947–967, May 2019.

[24] V. Hoskere, Y. Narazaki, and B. F. Spencer, "Physics-based graphics models in 3D synthetic environments as autonomous vision-based inspection testbeds," *Sensors*, vol. 22, no. 2, p. 532, Jan. 2022.

[25] H. Kim, J. Yoon, and S. Sim, "Automated bridge component recognition from point clouds using deep learning," *Struct. Control Health Monitor.*, vol. 27, no. 9, p. e2591, Sep. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2591

[26] X. Wang, C. Demartino, Y. Narazaki, G. Monti, and B. F. Spencer, "Rapid seismic risk assessment of bridges using UAV aerial photogrammetry," *Eng. Struct.*, vol. 279, Mar. 2023, Art. no. 115589.

[27] R. Wang, Y. Shao, Q. Li, L. Li, J. Li, and H. Hao, "A novel transformer-based semantic segmentation framework for structural condition assessment," *Struct. Health Monitor.*, vol. 23, no. 2, pp. 1170–1183, Mar. 2024.

[28] Y. Zhang and W. Lin, "Computer-vision-based differential remeshing for updating the geometry of finite element model," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 37, no. 2, pp. 185–203, Feb. 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12708

[29] D. Jana and S. Nagarajaiah, "Computer vision-based real-time cable tension estimation in Dubrovnik cable-stayed bridge using moving handheld video camera," *Struct. Control Health Monitor.*, vol. 28, no. 5, p. e2713, May 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2713

[30] D. Jana, S. Nagarajaiah, and Y. Yang, "Computer vision-based real-time cable tension estimation algorithm using complexity pursuit from video and its application in Fred-Hartman cable-stayed bridge," *Struct. Control Health Monitor.*, vol. 29, no. 9, p. e2985, Sep. 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/stc.2985

[31] R. Kromanis and P. Kripakaran, "A multiple camera position approach for accurate displacement measurement using computer vision," *J. Civil Struct. Health Monitor.*, vol. 11, no. 3, pp. 661–678, Jul. 2021.

[32] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2013, *arXiv:1311.2524*.

[33] R. Girshick, "Fast R-CNN," 2015, *arXiv:1504.08083*.

[34] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*.

[35] Y. Cha, W. Choi, G. Suh, S. Mahmoudkhani, and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 9, pp. 731–747, Sep. 2018.

[36] R. Li, Y. Yuan, W. Zhang, and Y. Yuan, "Unified vision-based methodology for simultaneous concrete defect detection and geolocalization," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 7, pp. 527–544, Jul. 2018.

[37] J. Deng, Y. Lu, and V. C. Lee, "Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 4, pp. 373–388, Apr. 2020.

[38] H. Bang, J. Min, and H. Jeon, "Deep learning-based concrete surface damage monitoring method using structured lights and depth camera," *Sensors*, vol. 21, no. 8, p. 2759, Apr. 2021.

[39] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, Dec. 2016, pp. 21–37.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2016, pp. 1–10.
[41] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, "Deep learning–based automated detection of sewer defects in CCTV videos," *J. Comput. Civil Eng.*, vol. 34, no. 1, Jan. 2020, Art. no. 4019047.
[42] M.-T. Cao, Q.-V. Tran, N.-M. Nguyen, and K.-T. Chang, "Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources," *Adv. Eng. Informat.*, vol. 46, Oct. 2020, Art. no. 101182.
[43] *Computer Vision Annotation Tool*, CVAT, Palo Alto, CA, USA, 2024. [Online]. Available: https://www.cvat.ai
[44] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018.
[45] C. Zhang, C. Chang, and M. Jamshidi, "Concrete bridge surface damage detection using a single-stage detector," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 4, pp. 389–409, Apr. 2020.
[46] S. E. Park, S.-H. Eem, and H. Jeon, "Concrete crack detection and quantification using deep learning and structured light," *Construct. Building Mater.*, vol. 252, Aug. 2020, Art. no. 119096.
[47] D. Arya, H. Maeda, S. K. Ghosh, D. Toshniwal, A. Mraz, T. Kashiyama, and Y. Sekimoto, "Deep learning-based road damage detection and classification for multiple countries," *Autom. Construct.*, vol. 132, Dec. 2021, Art. no. 103935.
[48] Y. Jiang, D. Pang, and C. Li, "A deep learning approach for fast detection and classification of concrete damage," *Autom. Construct.*, vol. 128, Aug. 2021, Art. no. 103785.
[49] Z. Yu, Y. Shen, and C. Shen, "A real-time detection approach for bridge cracks based on YOLOv4-FPM," *Autom. Construct.*, vol. 122, Feb. 2021, Art. no. 103514.
[50] Q. Qiu and D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," *Autom. Construct.*, vol. 147, Mar. 2023, Art. no. 104745.
[51] G. Jocher and J. Qiu, "YOLO11 (version 11.0.0)," Ultralytics, Madrid, Spain, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics
[52] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, "Evaluating the evolution of YOLO (you only look once) models: A comprehensive benchmark study of YOLO11 and its predecessors," 2024, *arXiv:2411.00201*.
[53] D. Demetriou, P. Mavromatidis, P. M. Robert, H. Papadopoulos, M. Petrou, and D. Nicolaides, "Real-time construction demolition waste detection using state-of-the-art deep learning methods; single–stage vs two-stage detectors," *Waste Manage.*, vol. 167, pp. 194–203, Jun. 2023.
[54] G. Jocher and J. Qiu, "YOLOv8 (version 8.0.0)," Ultralytics, Madrid, Spain, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[55] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 3–19.
[56] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[57] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.
[58] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smooth grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models," 2019, *arXiv:1908.01224*.
[59] M. B. Muhammad and M. Yeasin, "Eigen-CAM: Class activation map using principal components," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.
[60] S. Desai and H. G. Ramaswamy, "Ablation-CAM: Visual explanations for deep convolutional network via gradient-free localization," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 972–980.
[61] S.-Y. Byun and W. Lee, "Recipro-CAM: Fast gradient-free visual explanations for convolutional neural networks," 2022, *arXiv:2209.14074*.
[62] M. B. Muhammad and M. Yeasin, "Eigen-CAM: Visual explanations for deep convolutional neural networks," *Social Netw. Comput. Sci.*, vol. 2, no. 1, pp. 1–14, Feb. 2021.
[63] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021.
[64] C. Jaccard, "Étude théorique et expérimentale des propriétés électriques de la glace," Ph.D. dissertation, ETH Zurich, Zürich, Switzerland, 1959.
[65] A. Cardellicchio, F. Solimani, G. Dimauro, A. Petrozza, S. Summerer, F. Cellini, and V. Renò, "Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors," *Comput. Electron. Agricult.*, vol. 207, Apr. 2023, Art. no. 107757.
[66] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019.
[67] H.-P. P. Schwefel, *Evolution and Optimum Seeking: The Sixth Generation*. Hoboken, NJ, USA: Wiley, 1993.
[68] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. J. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2012.
[69] *Collaborative Data Science*, Plotly Technol., Montreal, QC, Canada, 2015.

SERGIO RUGGIERI was born in 1989. He received the bachelor's, master's, and Ph.D. degrees from the Polytechnic University of Bari, Italy. He is currently an Assistant Professor in structural engineering with the Polytechnic University of Bari. He is the author or co-author of over 50 peer-reviewed papers in scientific journals and national and international conferences as a result of several national and international collaborations. His research activities mainly concern the static and seismic vulnerability of new and existing structures and infrastructures, proposing studies on specific problems on existing structures (torsion and in-plane deformability), problems related to seismic engineering (acceleration demand of non-structural elements), and large-scale fragility and vulnerability analyses of different building typologies (RC residential and school buildings, masonry residential buildings and aggregate, and churches). In recent years, he worked in the field of digital innovation applied to structural engineering, in order to use new technologies for improving the prediction and management of risk of the built heritage. He is a member of editorial boards for some international scientific journals and has co-organized some special sessions for national and international conferences.

ANGELO CARDELLICCHIO was born in Taranto, Italy, in 1985. He received the master's degree in computer engineering from the Polytechnic University of Bari and the Ph.D. degree in electrical and information engineering from the Polytechnic University of Bari, in 2019. He defended the thesis "Smart sensor systems for environmental monitoring: implications and Applications." His professional experiences range from web and mobile development to data analysis in the energy, environmental, vulnerability, and industrial domains. He has also been a contract Professor with the University of Bari and the University of Foggia. He is currently a contract Professor with the Polytechnic University of Bari. He has with the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council of Italy, since 2021. He has authored several contributions to international peer-reviewed journals, conferences, and book chapters.

ANDREA NETTIS was born in Bari, Apulia, Italy, in 1992. He received the degree in building engineering-architecture from the Polytechnic University of Bari, in 2017, with a thesis on the seismic assessment of existing buildings characterized by a mixed steel-reinforced concrete structure, and the Ph.D. degree in structural engineering from DICATECH, Polytechnic University of Bari, in 2021, with a final dissertation concerning the seismic risk assessments of existing bridges combining innovative methods for data collection to be used at both structure- and network-scale. During the Ph.D., he carried out research collaborations with the Universitat Politecnica de Valencia (Valencia, Spain, and University College London, London, U.K. Currently, he is a Postdoctoral Researcher with DICATECH, Polytechnic University of Bari and an Adjunct Lecturer in structural design courses. His research topics concern innovative procedures for structural and seismic risk assessment of existing bridges and viaducts considering the use of artificial intelligence techniques and remote-sensing approaches (unmanned aerial vehicles and satellite data) for bridge structural health monitoring. He serves as a reviewer for several journals of international relevance in the field of civil engineering.

VITO RENÒ was born in Bari, Italy, in 1988. He received the master's degree (Hons.) in computer engineering from the Polytechnic University of Bari, in 2011, defending a thesis about computer vision and robust background modeling, and the Ph.D. degree in electrical and information engineering from the same University in 2017, defending the thesis "3D modeling, reconstruction and analysis of environments assisted by multi-sensorial data processing." He is currently a Researcher with CNR STIIMA, involved in research activities in the fields of computer vision and pattern recognition. He is the co-author of more than 70 scientific papers and one international patent. He is deeply curious and enthusiast about artificial intelligence, with a pinch of multi-disciplinary and synergic applications.

GIUSEPPINA UVA was born in 1969. She received the Ph.D. degree in computational mechanics from the University of Calabria. She is currently a Full Professor in structural engineering with the Polytechnic University of Bari, Italy. She is the Head of the seismic and structural engineering area, carried out both theoretical, applicative, and numerical research in the cultural areas of structural and seismic design, nonlinear dynamics of structures, computational mechanics, seismic risk and vulnerability, areas in which she is the author of about 150 publications. In recent years, her main research topics have concerned the static and dynamic nonlinear modeling and analysis of masonry and reinforced concrete structures, risk mitigation and resilience of the built environment, residential heritage, historical and monumental buildings, strategic buildings (in particular school buildings), bridges and viaducts, monitoring and SHM, application of data mining, and enrichment and machine learning techniques to the evaluation of existing structures and infrastructures. About these topics, she is responsible for several agreements, research projects, funding for research activities and contracts.