Available online at www.eccomasproceedia.org
Eccomas Proceedia COMPDYN (2025) 1448-1457

ECCOMAS
Proceedia

COMPDYN 2025
10th ECCOMAS Thematic Conference on
Computational Methods in Structural Dynamics and Earthquake Engineering
M. Papadrakakis, M. Fragiadakis (eds.)
Rhodes Island, Greece, 15-18 June 2025

DETECTING AND IDENTIFYING CRACKS IN RC BRIDGES VIA SINGLE-STAGE OBJECT DETECTORS

Angelo Cardellicchio[1], Vito Renò[1], Vincenzo Mario Di Mucci[2], Andrea Nettis[2], Sergio Ruggieri[2], and Giuseppina Uva[2]

[1]Institute for Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA), National Research Council of Italy
Via Amendola 122 D/O, Bari, 70126, Italy
e-mail: firstname.lastname@stiima.cnr.it

[2] DICATECH Department, Politecnico di Bari
Via Orabona 4, Bari, 70126, Italy
e-mail: firstname.lastname@poliba.it

Abstract. *The maintenance of existing reinforced concrete (RC) bridges is a primary issue addressed by public institutions to ensure the safety of infrastructures. Bridges are usually subjected to multiple sources of hazards related to, for example, seismic actions, floods, or the natural decay of materials. As such, the assessment of the safety of existing bridges must be approached via a multi-risk strategy following specific protocols. The six-level framework implemented by the Italian Public Administration requires a heavy involvement of domain experts, whose evaluation can be subject to several sources of biases, such as fatigue or inexperience. These problems can be stressed when dealing with cracks, which represent a type of defect that can strongly influence the residual life of the structure. Specifically, the surveyors shall count the number of cracks, define their location and orientation, and characterize an overall pattern for each structural element to plan further assessment and retrofit strategies to ensure structural safety. To address this issue, this work proposes an automatic tool that takes advantage by new technological advancements via a processing pipeline composed of several steps to automatically identify crack patterns in existing RC bridges. First, a single-stage object detector automatically locates cracks within the image. Second, a specific convolutional neural network (CNN) is used to frame the problem of crack segmentation as a binary classification problem, achieving sub-optimal performance and allowing for assessing and estimating damage extension. The results are calibrated on an existing dataset of RC bridge damages acquired during real surveys and tested on a real-case scenarios.*

Keywords: Cracks, Bridge structural monitoring, Deep Learning, Segmentation, Damage Detection.

ISSN:2623-3347
Peer-review under responsibility of the organizing committee of COMPDYN 2025.
doi: 10.7712/120125.12501.24667

1 INTRODUCTION

Ensuring safety of infrastructures represents one of the main goals of all the Public Administration worldwide. This is especially true in the case of critical infrastructures, such as existing bridges, which may pose a threat to public safety if not properly monitored and managed, given their exposure to several hazard sources, going from seismic actions to environmental agents, which accelerate the natural decay of materials. An example of the consequences of mismanagement was shown in Italy, when the Polcevera Viaduct collapsed, causing 43 deaths and 566 displaced people. To respond to this tragic event, the Italian Government released a series of specific guidelines to manage and evaluate the safety of the existing bridge portfolio [8], providing a multi-level approach for assessment and intervention. The first three levels focused on preliminary screening, while the last ones addressed to preventive actions on bridges presenting risks.

Interestingly, one of the most effort-intensive phases is the first level, in which onsite inspections should be performed, assigning an overall health score based on detailed visual inspections of main structural elements. This allows domain experts to assess the potential impact of various risk sources, e.g., traffic, floods, landslides, and earthquakes, and assign a status score to sensible elements. The inspections should also be extended over time, tracking the evolution of the bridge status. However, it is important to underline how the human factor poses serious limitations to the coherence of the assessment due to factors such as inexperience, subjectivity, or fatigue. Another aspect concerns the most relevant damages, which should be assessed carefully during the inspection. One of the most important types of damage is the occurrence of cracks, which should be properly quantified and evaluated in terms of extent, shape, and orientation, to determine the status of the structural element and infer some information about the underlying cause.

Dealing with this issue requires introducing novel methodologies to ease the human tasks. Interestingly, the automatic characterization of some of the most relevant damages, such as cracks, was extensively discussed by the research community. Several methodologies were proposed for this task over the last decade, based on sensors such as laser scanners and stereo systems for three-dimensional reconstruction or ultrasonic waves for identifying sub-superficial anomalies [5]. These sensors provided raw data to be interpreted by computer vision (CV) approaches, which can be categorized into classification, segmentation, feature detection, and object detection [4]. Overall, the two most relevant categories for the task are object detection and segmentation. First, object detection allows localization and classification of objects of interest, and for the case at hand, damages within the image. For example, this approach was followed by [1], which exploited a CNN and a sliding window technique to detect cracks in a dataset of 40.000 images. Deep learning (DL) based damage detectors were also proposed by several subsequent works, including [2], which exploited a two-stage detector (Faster R-CNN) to detect cracks in RC bridges, or by [9], which exploited YOLOv3, a single stage detector, for the same goal. The authors of this paper used also different versions of YOLO for multidefect detection (see [12], [11]). The second interesting approach is segmentation, which allows the identify the location of the exact boundaries of objects of interest in the focused image. For example, [7] proposed a pixel-based multi-defect detection using a CNN. At the same time, other deep learning methods, such as transformers, are explored in [13] and [5] using unnamed aerial vehicles (UAVs). For example, three-dimensional data were also exploited by [3], who used this mapping type to localize cracks based on depth maps.

Although the continuos efforts in this direction, there is a lack of automated and up-to-date

end-to-end processing pipelines able to provide the location of cracks within structural elements of reinforced concrete (RC) structures and assess their extension, orientation, and shape. This work proposes a two-step processing pipeline to fill this specific gap, developed as follow:

- *Damage localization* tool identifies damages automatically within a structural element of a bridge.
- *Crack assessment* tool assesses the shape, orientation, and extent of the cracks identified by the DL tool.

By combining the two above approaches in a composed pipeline, this work aims to provide a preliminary assessment of the status of the RC structure under investigation, therefore providing an effective decision and management support system to the domain expert.

The paper is structured as follows. In Section 2, the theoretical foundations of the work are described. In Section 3, the results are provided accounting for an use case. Finally, Section 4 provides the conclusions and the perspectives on future works.

2 MATERIALS AND METHODS

The main steps characterizing the proposed approach following listed:

1. **Data collection**: The human expert should gather pictures in a specific and standardised way to avoid unnecessary biases.
2. **Labelling**: The collected data must be labelled to be fed to the processing pipeline stages, identifying the location of the damages and reconstructing the shape of the cracks.
3. **Features identification**: The boxes containing relevant damage (i.e., cracks) are counted using the object detector discussed in Section 2.2, and segmented using the approach discussed in Section 2.3.

2.1 Data collection

Two datasets were collected for the analysis. The first dataset was gathered using a UAV and was composed of 150 images representing single elements of a bridge. The second dataset was gathered during onsite inspections and was constituted by 7.000 images of structural elements. Both datasets were labelled by human experts using the Label Studio tool. Data were divided into two sets for both datasets in a $70 - 30$ proportion for training and test, respectively.

2.2 Object detection

This work compared the performance of a single stage and an end-to-end detector for object detection. Specifically, as for the single-stage detector, the latest iteration of YOLO, i.e., YOLOv12 [14], was used, while for the end-to-end detector, RT-DETR [15] was used.

On the one end, YOLOv12 was the latest iteration of the YOLO family and introduced several different key aspects, mainly focused on attention mechanisms, including a new approach to reduce the computational effort of the self-attention mechanism while retaining the same receptive field, or a revamping of the standard ELAN mechanism called Residual ELAN, which exploits residual connections to improve detection capabilities. Overall, the main focus of the authors of the architecture was on streamlining the attention mechanisms already introduced in previous versions of YOLO, such as YOLOv11, and optimizing the computational efficiency

to yield high-quality results while minimizing computational complexity. On the other hand, RT-DETR is based on an entirely different paradigm: the use of transformers to perform object detection. Specifically, RT-DETR proposes modifications to the encoder used over previous iterations, namely DETR (Detection Transformer), exploiting an efficient hybrid encoder composed of two parts for decoupling intra-scale interaction and cross-scale functions, along with an optimisation of the query selection based on information related to the intersection over the union. Overall, this architecture is strongly focused on optimising the performance of other transformer-based detectors, removing unnecessary steps such as non-maximum suppression and anchor detection to streamline and optimise the detection process further.

Results were evaluated through four metrics, usually employed for object detection: *precision* (P), *recall* (R), *F1 score* (F1), and *mean average precision* (mAP). These metrics are computed starting from the *Intersection over Union* metric, defined as follows:

$$IoU = \frac{B_p \bigcap B_{gt}}{B_p \bigcup B_{gt}} \tag{1}$$

The IoU metric evaluates the overlapping between the *predicted* B_p and *ground truth bounding box* B_{gt}. If the value of IoU is 1, the predicted box perfectly overlaps the ground truth box. A common threshold used for the IoU is 0.5. Therefore, given this threshold, the following quantities can be defined in the context of object detection:

- **True Positive** (TP), when the IoU is above the imposed threshold.
- **False Positive** (FP), when the predicted bounding box does not exist within the ground truth.
- **False Negative** (FN), when the ground truth bounding box has no predictions associated.

Starting from these metrics, precision, recall, and F1 score can be computed as follows:

$$P = \frac{TP}{TP + FP} \tag{2}$$

$$R = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2\frac{P \cdot R}{P + R} \tag{4}$$

It is worth underlining that object detectors also provide confidence scores associated to each detection. It is therefore possible to adjust precision and recall by considering a specific threshold value τ:

$$P(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau)} \tag{5}$$

$$R = \frac{TP(\tau)}{TP(\tau) + FN(\tau)} \tag{6}$$

Hence, the *average precision* (AP) can be computed as the area under the $P(\tau) - R(\tau)$ curve at k different values for the threshold τ:

$$AP = \frac{1}{N} \sum_{n=1}^{N} P_{int}(R_r(n)) \tag{7}$$

In Equation 7, $P_{int}(R_r(n))$ 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 C classes within the dataset, a synthetic index which considers the *mean* between all these classes is computed as the *mean Average Precision* metric:

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \tag{8}$$

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.

2.3 Segmentation

A pixel-based approach was developed from scratch to segment cracks, aiming to exploit lightweight models that are available even when small amounts of data are available. Specifically, a standard CNN was used to differentiate areas whose centres *belonged to cracks* and areas whose centres *did not belong to cracks*. This allowed three improvements over existing approaches.

1. Data availability was greatly improved thanks to the proposed data extraction method. Specifically, given a segmented bounding box of $M * N$ pixels, a total of $\left(M - \frac{W}{2}\right) \cdot \left(M + \frac{W}{2}\right)$ square patches were extracted with W the side of the patch.
2. Each extracted patch was labelled as a *positive* patch, i.e., a patch whose centre was contained within a crack or otherwise *negative* patch.
3. The adoption of lightweight models allowed the implementation of the proposed approach even on constrained devices.

The network scheme proposed for this work is shown in Figure 1. The proposed architecture is extremely simple and accepts a $W \cdot W$ patch fed to three convolution layers. The resulting feature map is downsampled using a max pool layer, and a sigmoid layer makes the final decision.

The sigmoid layer outputs a probability value and, therefore, a threshold ϕ, assumed equal to $\phi = 0.5$ was used for the proposed experiments. In the context of the segmentation problem, the evaluation metrics were derived from the following values:

- **True Positives** (**TP**), the number of patches whose central pixel represents a crack and is correctly classified as positive samples.
- **True Negatives** (**TN**), the number of patches whose central pixel do not represent a crack and is correctly classified as negative samples.
- **False Positives** (**FP**), i.e., the number of patches whose central pixel does not represent a crack and is incorrectly classified as positive samples.

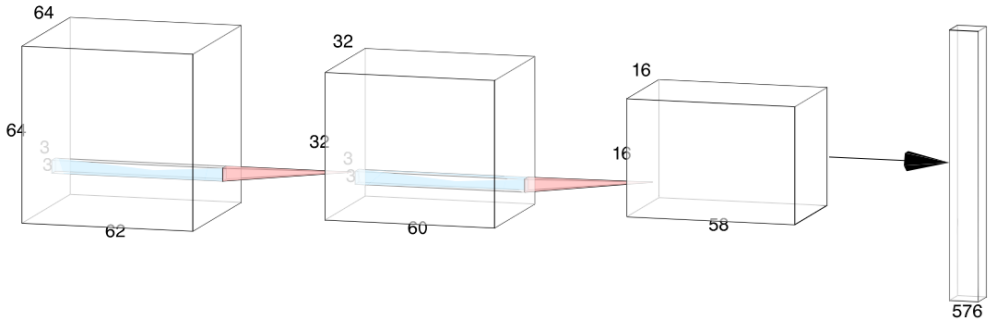

Figure 1: The architecture of network used for crack segmentation.

- **False Negatives** (**FN**), i.e., the number of patches whose central pixel represents a crack and is incorrectly classified as negative samples.

Once trained, the network can be used in inference via a sliding window approach on the totality of the pixels composing the original image. Consequently, each pixel will be labelled as belonging to the crack or not, and the overall damage will be properly tracked.

3 EXPERIMENTAL SECTION

This section discusses the experiments performed to validate the proposed methodologies. All the experiments were performed on a machine equipped with an Intel Core i9-14900HK, 64 GB of RAM, and an NVIDIA GeForce 4090 with 24 GB of VRAM. The software was developed using the Ultralytics [6] and PyTorch [10] libraries.

3.1 Training and evaluation of object detection

As already highlighted in Section 2.2, two main types of object detectors were compared in this work, i.e., YOLO12 and RT-DETR. The motivation behind this choice was to compare two approaches based on DL approaches for object detection, establishing the best viable option for the task under investigation. All models were trained for 300 epochs. The results are shown in Table 1, accounting for the different available versions of the two algorithms.

Table 1: Comparison of the object detectors used to identify cracks within the proposed images.

Network	P	R	F1	mAP 0.5	mAP 0.95
YOLO12n	89.43	57.09	69.69	63.31	50.16
YOLO12s	88.95	**58.58**	70,64	67.39	58.28
YOLO12m	91.18	58.57	71.33	70.23	63.09
YOLO12l	**93.80**	58.54	**72.09**	71.19	64.48
YOLO12x	90.46	57.85	70.56	**72.42**	**67.50**
RT-DETR-l	88.91	57.45	69.80	61.85	47.23
RT-DETR-x	88.12	58.26	70.14	62.73	48.25

Interestingly, YOLO12 outperforms RT-DETR for each one of the considered metrics. This may be related to different issues, including the amount of available data, which can be considered adequate for convolution-based networks, such as YOLO, but may be inadequate for

Figure 2: Four samples of the images used in the segmentation dataset. The first two on the left belong to positive samples, images whose centre belongs to a crack. The two patches on the left instead represent negative samples.

Table 2: Values achieved by the evaluation metrics for the network with $W = 65$.

Augmentation	P	R	F1
Yes	99.95%	98.38%	99.16%
No	99.94%	98.54%	99.24%

transformer-based networks. Among the different densities, the large one achieves the best results in terms of precision and F1 score, while the extra version achieves the best results in terms of mAP. This can be mainly related to two aspects: first, the number of parameters compared with the size of the dataset, which appears to provide an optimum for training the second most dense version of the network. Still, the extra network demonstrated higher representational capabilities regarding confidence score, given the higher values achieved for the mAP metrics.

3.2 Training and evaluation of crack segmentation

The crack segmentation network was trained using a dataset composed of 195.971 positive samples and 198.656 negative ones. A small sample of the dataset is shown in Figure 2. It is worth underlining that three versions of the dataset were available, according to the specific value of W, specifically 65, 129 or 257. For the sake of simplicity, only the version with $W = 65$ was tested.

As for the training already described in Section 2.3, it was performed using SGD as the optimization algorithm, binary cross-entropy as loss function, 0.01 for the learning rate, and 25 epochs of training. To improve the robustness of the results, data augmentation was applied and tested, including procedures such as random flips, random rotations, and HSI manipulation. It is worth mentioning that other methods, such as random crops or mosaic augmentation, were discarded to avoid situations where the centre of the image did not belong to a crack.

The results achieved by the network are reported in Table 2, including results with and without data augmentation. Overall, it can be seen that the network provides high values for all the provided metrics with and without data augmentation.

3.3 Discussion on the feasibility study on a real use case

Once trained, the object detector was used in inference to extract relevant damages (i.e., cracks) on a specific use case, a beam of an RC bridge for which a near-complete image was retrieved, as shown in Figure 3, to provide a perspective on the feasibility of the approach. Specifically, it can be noted how the object detector also provided satisfactory results in situations with high background noise, for example, for cracks visible over the moisture spots on the structural element.

Afterwards, the segmentation network was used to perform crack segmentation over the

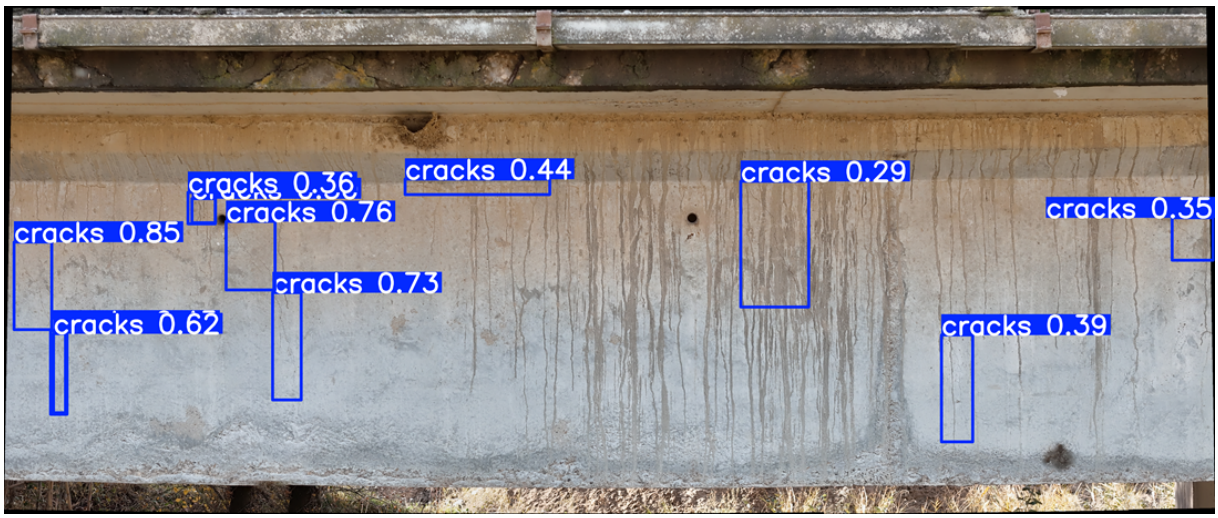

Figure 3: Results of crack identification on the proposed use case.

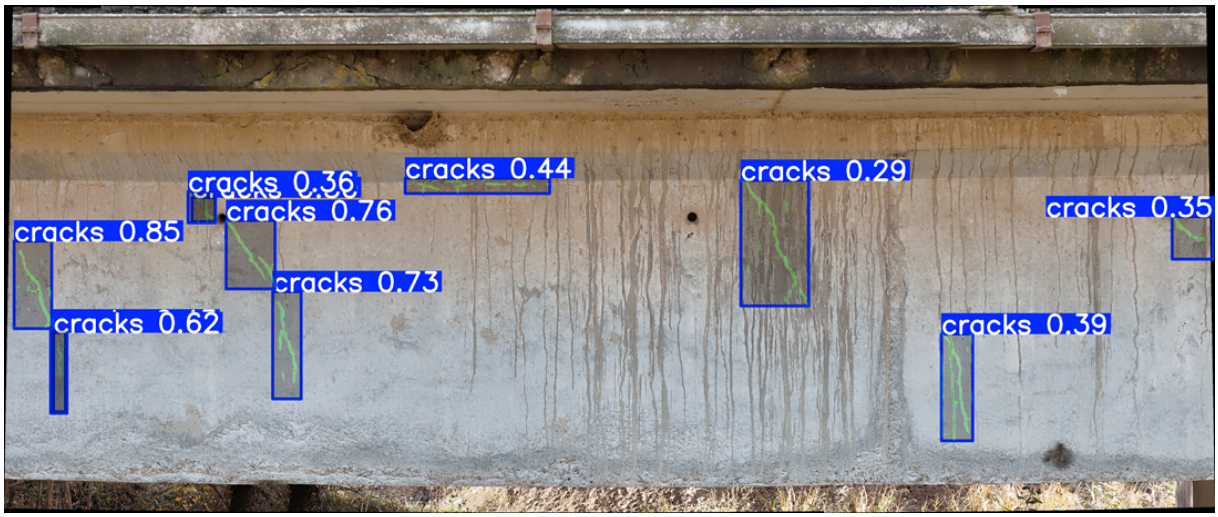

Figure 4: Results of crack segmentation on the proposed use case.

bounding boxes identified by the detector. The results are shown in Figure 4. One of the most interesting aspects is that the segmentation network could perfectly follow even the smallest cracks when overlapped with moisture spots. This provided a satisfactory preliminary assessment on the feasibility of the approach.

4 CONCLUSIONS

This work proposes a processing pipeline composed of two steps: damage detection and crack segmentation. The processing pipeline was successfully tested against a real use case, and a first application of the approach was provided, demonstrating its feasibility. Specifically, as for the second part of the pipeline, an approach using a binary classification and a simple CNN was proposed, and its effectiveness was tested. After assessing the approach, it is worth noting its potentiality in providing to domain experts a rapid assessment tool to count and quantify structural damage in a standardized and streamlined fashion.

Still, several challenges have yet to be overcome. First, one of the most important issues to solve is to provide an end-to-end framework that can estimate the cracks orientation and the

related overall extension, which are essential information for structural assessment. Afterwards, the pipeline should be properly implemented and optimized to work on constrained hardware, such as embedded UAV systems. To this end, possible modifications of the proposed detectors, which preserve their ability to properly identify damages while reducing the overall computational burden, will be proposed and implemented in future revisions.

REFERENCES

[1] Young-Jin Cha, Wooram Choi, and Oral Büyüköztürk. Deep learning-based crack damage detection using convolutional neural networks. *Computer-Aided Civil And Infrastructure Engineering*, 2017.

[2] J Deng, Y Lu, and V. C-S. Lee. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Computer-Aided Civil and Infrastructure Engineering*, 35:373–388, 2020.

[3] L. Deng, T. Sun, L. Yang, and R. Cao. Binocular video-based 3d reconstruction and length quantification of cracks in concrete structures. *Automation in Construction*, 148:105719, 2023.

[4] Vincenzo Mario Di Mucci, Angelo Cardellicchio, Sergio Ruggieri, Andrea Nettis, Vito Renò, and Giuseppina Uva. Artificial intelligence in structural health management of existing bridges. *Automation in Construction*, 167:105719, 2024.

[5] Wei Ding, Han Yang, Ke Yu, and Jiangpeng Shu. Crack detection and quantification for concrete structures using UAV and transformer. *Automation in Construction*, 152:104929, August 2023.

[6] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024.

[7] S. Li, X. Zhao, and G. Zhou. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Computer-Aided Civil And Infrastructure Engineering*, 2019.

[8] MIT. Linee guida per la gestione del rischio dei point esistenti e delle Istruzioni operative per l'applicazione delle Linee Guida stesse, 2021.

[9] S.E. Park, S.-H. Eem, and H. Jeon. Concrete crack detection and quantification using deep learning and structured light. *Construction and Building Materials*, 252:119096, 2020.

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[11] Sergio Ruggieri, Angelo Cardellicchio, Andrea Nettis, Vito Renò, and Giuseppina Uva. Using machine learning approaches to perform defect detection of existing bridges. *Procedia Structural Integrity*, 44:2028–2035, 2023. XIX ANIDIS Conference, Seismic Engineering in Italy.

[12] Sergio Ruggieri, Angelo Cardellicchio, Andrea Nettis, Vito Renò, and Giuseppina Uva. Using attention for improving defect detection in existing rc bridges. *IEEE Access*, 13:18994–19015, 2025.

[13] Muhammad Rakeh Saleem, Jong-Woong Park, Jinhwan Lee, Hyung-Jo Jung, and Muhammad Sarwar. Instant bridge visual inspection using an unmanned aerial vehicle by image capturing and geo-tagging system and deep convolutional neural network. *Structural Health Monitoring*, July 2020.

[14] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[15] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16965–16974, June 2024.