



## Research paper

## Post-earthquake structural damage detection with tunable semi-synthetic image generation



Piercarlo Dondi <sup>a</sup>\*, Alessio Gullotti <sup>a</sup>, Michele Inchingolo <sup>a,b</sup>, Ilaria Senaldi <sup>b</sup>, Chiara Casarotti <sup>b</sup>, Luca Lombardi <sup>a</sup>, Marco Piastra <sup>a</sup>

<sup>a</sup> Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via Ferrata 5, Pavia, 27100, Italy

<sup>b</sup> EUCENTRE Foundation, European Centre for Training and Research in Earthquake Engineering, Via Ferrata 1, Pavia, 27100, Italy

## ARTICLE INFO

Dataset link: <https://www.eucentre.it/>

## Keywords:

Structural damage detection  
Deep learning  
Data augmentation  
Three-dimensional modeling  
Tracking  
Post-earthquake survey

## ABSTRACT

In the aftermath of an earthquake, conducting rapid structural safety assessments is essential. A Deep Learning-based damage detector capable of automatically analyzing videos from Unmanned Aircraft Systems (UAS) surveys would be highly beneficial for this purpose. Despite significant advancements in object detection using Deep Convolutional Neural Networks (DCNNs), developing an effective post-earthquake damage detector remains challenging due to the scarcity of large, annotated image datasets. In this work, we present a system to create a large number of images where artificial damage instances are applied to real-world three-dimensional (3D) models of buildings and bridges. We defined such images as semi-synthetic. The proposed method relies on the definition, made by human experts, of meta-annotations from which a variety of damage instances can be generated in a controlled way. Semi-synthetic images are designed to augment real-world datasets, enhancing the training process of a DCNN-based damage detector. This semi-synthetic image augmentation can be iteratively refined to target the most critical cases. Experiments conducted on the 'Image Database for Earthquake damage Annotation' (IDEA) dataset shown that a detector trained on a combination of real and semi-synthetic images performs better than one trained on real images alone. A damage detector trained using the proposed strategy was then incorporated into a system that analyzes and tracks multiple damage instances in UAS-acquired videos, generating concise summaries of the findings. The effectiveness of the system was validated by the analysis of post-earthquake UAS videos and the production of reports that were reviewed by structural engineering experts.

## 1. Introduction

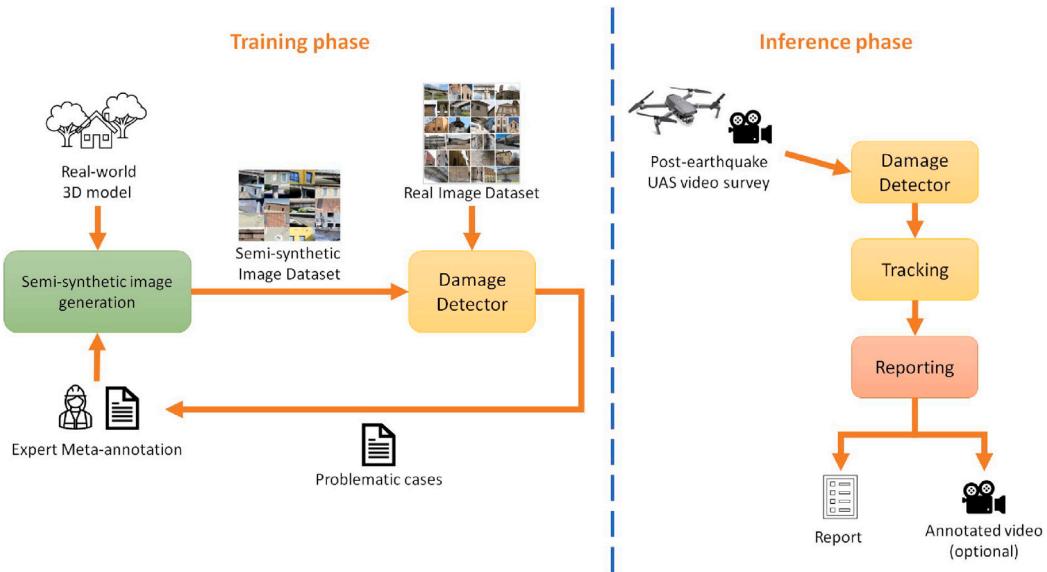
Fast structural safety assessment is a crucial task after a disastrous event like an earthquake. In recent years, Unmanned Aircraft Systems (UAS) have gained increasing acceptance for structural surveys due to their ability to acquire high-resolution images and videos from various viewpoints, even in otherwise inaccessible areas (Seo et al., 2018; Xu and Turkan, 2020). The innovative use of diverse sensors and unique aerial perspectives provided by UAS, combined with data processing through Artificial Intelligence (AI) methods, has proven highly effective for surveying the built environment (Rakha and Gorodetsky, 2018; Halder and Afsari, 2023; Bayomi and Fernandez, 2023).

Traditionally, in emergencies, human experts perform safety assessments by directly inspecting images and videos acquired on the field, which is a complex and very time-consuming activity. This inspection

is especially critical in the aftermath of an earthquake, when many structures may require rapid inspection, to identify potential risks to inhabitants and first responders alike. Computer Vision (CV) and Deep Learning (DL) techniques have proven to be valuable tools in assessing the safety of buildings and civil structures (Sony et al., 2021). In particular, automated damage detection algorithms can be used to rapidly screen a large number of images and videos, thus reducing the amount of data that experts will need to inspect and validate directly. However, most existing systems are designed for routine maintenance monitoring (Flah et al., 2021), which, in general, reveals less severe damage on areas for which some previous information is available. Developing a post-earthquake structural damage detector introduces additional challenges, like the necessity to inspect large, unknown areas, with the requirement to analyze a vast amount of images and

\* Corresponding author.

E-mail addresses: [piercarlo.dondi@unipv.it](mailto:piercarlo.dondi@unipv.it) (P. Dondi), [alessio.gullotti.pro@gmail.com](mailto:alessio.gullotti.pro@gmail.com) (A. Gullotti), [michele.inchingolo01@universitadipavia.it](mailto:michele.inchingolo01@universitadipavia.it) (M. Inchingolo), [ilaria.senaldi@eucentre.it](mailto:ilaria.senaldi@eucentre.it) (I. Senaldi), [chiara.casarotti@eucentre.it](mailto:chiara.casarotti@eucentre.it) (C. Casarotti), [luca.lombardi@unipv.it](mailto:luca.lombardi@unipv.it) (L. Lombardi), [marco.piastra@unipv.it](mailto:marco.piastra@unipv.it) (M. Piastra).



**Fig. 1.** Overview of the proposed approach.

videos on site with limited computational resources. Furthermore, developing an effective damage detector based on Deep Convolutional Neural Networks (DCNNs) requires large datasets of annotated images, which, at present, are scarcely available. In fact, most of the existing datasets are from routine monitoring, especially for bridges, and show less severe damage than those that can occur with an earthquake, often focusing on specific structures and damage types. Only a small fraction includes post-disaster images and even fewer include bounding box annotations for multiple damage classes (Bianchi and Hebdon, 2022; Yang et al., 2022). Likewise, in most methods cited above, the implemented detectors are working on either whole-image classification or segmentation, applied to close-up images or small patches thereof. This kind of solutions are effective for inspecting areas that are known in advance, but less suitable for surveys of previously unknown areas. In addition, while various data augmentation approaches have been proposed in literature (Li et al., 2024; Xu et al., 2023; Zhao et al., 2024), to the best of authors' knowledge there is no augmentation technique specifically designed for a post-earthquake scenario, in which the generated images should include a multitude of severely damaged structures, framed at different distance and under different meteorological condition.

This data scarcity is surely the major challenge in this scenario, and the one on which we mainly focus on this paper. The approach presented here is intended to help fill the gap in data availability by introducing a tunable augmentation strategy, that aims to produce coverage of various damage situations and conditions that may occur after an earthquake. We focus on images and videos taken at low altitude, with the UAS flying at a mid-low distance, in the range 2–7 m, from damaged structures (Mandriola et al., 2022). On the other hand, we did not consider UAS surveys at high altitude, such as Hu et al. (2024), which require completely different approaches and datasets. More precisely, our approach includes:

- A *semi-synthetic image generation* method that can produce a large number of images of damaged structures. The method takes three-dimensional (3D) models of real-world buildings and bridges as input, where experts manually place and define *meta-annotations*. These meta-annotations guide the automatic generation of a variety of damage instances. We refer to the resulting images as *semi-synthetic* because only the damage is artificial, while the structures are real. Semi-synthetic images are intended to augment real images during the training process of a DCNN. This method refines our previously published crack image generator (Dondi et al., 2024) and extends it to three new damage

classes: spalling, corrosion/exposed rebars and leaching. The effect of damage on the internal layers of the structures considered has been added, too.

- A *tunable data augmentation* strategy designed to increase the effectiveness of the training process of a DCNN-based damage detector. This strategy consists in iteratively tuning the parameters of the meta-annotations, so that the features of the resulting semi-synthetic images can be tailored to match the cases that are problematic for the detector.
- A *damage tracking and reporting* system, for field use, which integrates a detector trained with the proposed strategy to analyze UAS-acquired videos. The system can detect and track the four classes of damage considered and provide a concise summary of all damage occurrences as aid for human experts.

**Fig. 1** shows a high-level summary of the procedure. Detailed schemes of the main steps will be provided in Section 3.

Our solution was validated on the Image Database for Earthquake damage Annotation (IDEA) dataset created by the EUCENTRE Foundation. To the best of the authors' knowledge, this is currently the most complete available dataset of post-earthquake survey images with bounding box annotations. Differently from other datasets available in literature (Bianchi and Hebdon, 2022; Yang et al., 2022) IDEA is characterized by the presence of multiple damage classes on various structural typologies, such as, for example, reinforced concrete bridges, reinforced concrete infilled frames, unreinforced brick and/or stone masonry buildings. Such heterogeneity makes the IDEA dataset ideal for developing a DL-based structural damage detector for post-seismic event inspections.

As DCNN for our experiments, we employed the You Only Look Once (YOLO) framework developed by Ultralytics (Jocher et al., 2023), testing various configurations and architectures. YOLO networks are among the best performing single-stage object detection models and are currently employed in numerous fields (Terven et al., 2023). They offer rapid inference time while achieving detection accuracy comparable to, or better than, slower two-stage methods, like Region based Convolutional Neural Networks (RCNN) (Diwan et al., 2023). These characteristics make YOLO very suitable in the scenario considered, in which multiple videos and images need to be quickly analyzed using only the computational resources available on the field (e.g., a laptop).

The remaining of the article is structured as follow: Section 2 discusses the state-of-the-art in the field; Section 3 describes the proposed approach; Section 4 illustrates the datasets used in the experiments;

Section 5 presents and discusses the achieved results; finally, Section 7 draws the conclusions and proposes possible future developments.

## 2. Previous works

Machine and Deep Learning techniques are nowadays commonly employed in structural engineering (Thai, 2022), especially within the context of Structural Health Monitoring (Flah et al., 2021; Sony et al., 2021). In the following we will provide a brief overview of recent damage detection solutions and datasets.

### 2.1. Structural damage detection methods

Structural damage detection techniques are generally categorized into two groups: vibration-based (Avci et al., 2021) and image-based methods, on which we will focus.

In 2017, Cha et al. (2017) designed a DCNN for detecting cracks on concrete structures. This method was improved the following year with a faster region-based CNN model to identify different classes of surface damage, including cracks, delamination, and corrosion (Cha et al., 2018). In 2018, Yang et al. (2018) proposed a DCNN-based semantic segmentation approach for detecting and measuring cracks. That same year, Davoudi et al. (2018a,b) focused on detecting flexure and shear crack of Reinforced Concrete (RC) beams and slabs using SVM-based models. Ye et al. (2019) also studied crack segmentation in RC beams and proposed a customized DCNN called Ci-Net. Li et al. (2019) adopted a DCNN model to identify different classes of damage (cracks, spalling and efflorescence) in concrete structures. Dung et al. (2019) employed a different DCNN architecture to detect cracks on steel gusset plates. Mundt et al. (2019) proposed a meta-learning DCNN approach to identify different classes of damage (crack, spalling, efflorescence, exposed bars, corrosion) on concrete bridges.

Various semantic segmentation methods have been proposed in 2020, mainly for detecting cracks (Choi and Cha, 2020; Kang et al., 2020; Zheng et al., 2020). Hoskere et al. (2020) developed MaDnet, a DCNN able to identify different classes of structural damage (crack, exposed rebar, spalling and corrosion) on different materials (concrete, steel, asphalt). A different approach was proposed in the same year by Chow et al. (2020), who treated the problem of the identification of cracks and spallings on concrete structures as an anomaly detection task, by using a convolutional autoencoder. An End-to-end Defect Detection Network (EDDN) for metal defect detection was proposed by Lv et al. (2020). Between 2020 and 2021, specific CNN-based crack detectors have been developed for masonry walls (Dais et al., 2021) and road pavements (Yang et al., 2020; Nguyen et al., 2021).

In 2023, Mondal et al. (2023) proposed a semantic segmentation approach that combines color and depth data for the monitoring of some common types of damage (spalling, spalling with exposed rebars, and severely buckled rebars) on reinforced concrete buildings. In the same year, Bai et al. (2023) designed and tested a series of DL automatic methods for segmentation and classification of damage on structures impacted by extreme events. Considered damage classes include cracks, spalling, and collapse type (partial or entire). The classifier was also able to identify the damage sub-type (flexural, shear, or combination) and intensity, as well as the structural part affected (beam, column or wall). Both Wan et al. (2023) and Gao et al. (2023) focused on damage detection on bridges, the former developd a variation of the Detection Transformers (DETR), called Bridge Detection Transformers (BR-DETR), while the latter proposed a few-shot learning approach to identify cracks. A similar problem was also studied by Xu et al. (2023), who focused on crack detection for dams.

In 2024, Canchila et al. (2024) proposed a Bayesian approach to optimize the hyperparameters of a DCNN segmentation algorithm for cracks, while Bhatta and Dang (2024) used a novel Quantum CNN (QCNN) to detect post-earthquake damage on RC buildings. Agyemang et al. (2024) proposed DetectorX, a novel framework for structural

damage detector that combines two DCNN models employing an event-based reward reinforcement learning and spiral pooling. Finally, Su et al. (2024) provided a comparison among various DCNN and Vision Transformer (ViT) approaches for crack detection.

Literature analysis indicates that most studies focus on either whole-image classification or segmentation techniques applied to close-up images or small patches thereof. Methods of this kind are effective for inspecting areas that are well known in advance. This is not the case of post-earthquake surveys in which inspected areas are almost completely unknown. Specifically, in this work we focus on inspections conducted by UAS flying at mid-low distance from the affected structures, to quickly gather comprehensive information. Therefore, in this context, object detection methods may be more effective.

### 2.2. Datasets and data augmentation

A substantial challenge for training a post-earthquake structural damage detector is the scarcity of specific and adequately large available datasets. At present, many of the available datasets contain images obtained from routine structural monitoring, in particular for bridges (Bianchi and Hebdon, 2022; Yang et al., 2022). The level of damage that can be detected in these is generally less severe than that caused by earthquakes. More specifically, existing datasets often focus on a particular structural type (e.g., asphalt roads in RDD2020 Arya et al. (2021), or steel bridges in COCO-Bridge 2021+ Bianchi and Hebdon (2021)), and/or a specific damage class (e.g., spalling Yeum et al. (2018) or crack Ye et al. (2021)), and/or consist of patches extracted from larger images (Xu et al., 2019; Bai et al., 2021; Dais et al., 2021). In many cases, annotations are provided as image-wise classification labels, such as in PEER Hub ImageNet (Gao and Mosalam, 2020). Just a small fraction of available datasets includes images from post-disaster reconnaissance survey (Bianchi and Hebdon, 2022; Yang et al., 2022). In addition, only a few datasets include bounding box annotations for multiple damage classes and, typically, they contain only a few thousand images like CODEBRIM (Mundt et al., 2019) and the IDEA dataset used in this work.

To compensate such data scarcity, various data augmentation strategies have been explored in the literature. Some of them rely on traditional image processing techniques, such as geometrical and color transformation (Polovnikov et al., 2021) or random damage displacements (Zhao et al., 2024). Others have delved into synthetic image generation methods, with Generative Adversarial Networks (GAN) being the most common solution (Dunphy et al., 2022; Jain et al., 2022; Maeda et al., 2021; Branikas et al., 2023; Li et al., 2024). Customized DCNNs (Kim and Lee, 2023) and 3D modeling techniques (Hoskere et al., 2019; Boikov et al., 2021) have been tested as well. However, most of these approaches focus on generating close-up images, which are of limited usefulness in the case considered here. To the best of our knowledge, only a few methods, like the crack generator for dam developed by Xu et al. (2023) and our previous crack generator for buildings and bridges (Dondi et al., 2024), address mid-distance image generation.

## 3. Methodology

The proposed solution involves a novel procedure for training an effective DCNN-based structural damage detector by using a combination of real and semi-synthetic images. Once trained, the damage detector is integrated into a processing pipeline that, given a UAS-acquired input video, performs a complete analysis and produces a concise report for final validation by human experts.

In the following we will describe in detail all the stages of the proposed procedure.

**Table 1**

Types of structures to which the proposed semi-synthetic image generation method is applicable.

Structure	Type
Buildings	Unreinforced masonry building with exposed random rubble stone masonry
	Unreinforced masonry building with plastered walls
	Unreinforced masonry building with exposed stretcher/flemish bond brick/block masonry
	Reinforced concrete frame buildings with plastered external wall surface and internal stretcher bond block masonry infills
Bridges	Reinforced concrete girder bridge
	Reinforced concrete arch bridge
	Unreinforced masonry arch bridge

### 3.1. Semi-synthetic image generation

The core idea of our *semi-synthetic image generation* method is using 3D models of real-world building and infrastructures, possibly undamaged, on which synthetic damage instances are applied through parametric *meta-annotations*, positioned and specified by human experts. The method was designed in a strict cooperation between computer scientists and structural engineers, to guarantee the realism of the outcome. Special attention, as required by structural engineers, was given to replicating complex scenarios, which are common after an earthquake, where multiple types of damage occur simultaneously. For example, a long crack on a wall combined with extensive spalling, leading to the appearance of multiple inner layers.

In our intention, this approach should produce more realistic results than those achievable from fully artificial CAD models. On the other hand, in comparison to generative AI methods such as diffusion models, our approach is intended to allow a more explicit degree of expert-based control over the generation of damage instances than that achievable by generative network conditioning.

As said in the Introduction, this method extends our previous crack generator (Dondi et al., 2024) to include three other damage classes: spalling, corrosion/exposed rebars and leaching. The current approach allows also to simulate various kinds of internal layers of the structures considered, in order to reproduce more faithfully the effects of severe damage occurrences. Our method can work with various categories of buildings and bridges, as listed in Table 1.

The four main steps of the proposed semi-synthetic image generation method are:

1. **3D model setup** – the input real-world 3D model is examined, defects in the mesh are fixed, the areas to be damaged are chosen, and the inner/outer layers are specified.
2. **Meta-annotation phase** – meta-annotations are positioned in selected areas of the 3D model and completed with parameter specifications.
3. **Damage generation** – damage instances are generated from the corresponding meta-annotations, by randomizing actual values from the specified parameter ranges, and applied to the 3D model.
4. **Image rendering** – the rendering sequence is set up, including lighting and meteorologic conditions and UAS flight-simulation camera paths, and the final images are rendered together with the corresponding bounding box annotations.

Steps one and two above require a manual setup from a human expert, whereas step three and four are fully automatic. The software tools supporting the method described were implemented in Blender (version 3.6 LTS) (Blender Team, 2023) via a series of node-based workflows and add-on Python scripts.

**Table 2**

Main parameters of meta-annotations for each class of damage.

Class	Parameter	Meaning
All	Length	Percentage of the line to be covered (range)
	Appearance	Probability of occurrence
Crack	Roughness	Perturbation of the line, as low and high frequency (ranges)
	Thickness	Crack width (range)
	Depth	Crack inner depth (range)
Spalling	Thickness	Spalling width (range)
	Depth	Spalling inner depth (range)
	Number	Number of spallings instances along the line (range)
Leaching	Texture	Leaching texture, used as basis
	Frequency	Frequency of the percolation (range)
	Extension	Vertical length of the percolation (range)
	Transparency	Transparency of the texture (range)

#### 3.1.1. 3D model setup

The first step involves the initial processing of a real-world 3D model. Since such input model typically contains multiple buildings and/or structures (e.g., it could be a large scan of a village neighborhood or of the surroundings of a bridge), specific relevant *components* need to be selected. The remaining parts of the model will be used as background for the rendering.

On each selected component, one or more *Area-of-Damage* (AoD) will be chosen for the application of synthetic damage. If required, 3D mesh defects in the AoD are fixed at this stage before further processing. An example of AoD is shown in Fig. 2(a), namely the facade of a building.

While the input 3D model reproduces faithfully the external surface of a structure, the application of substantial damage might reveal also the underlying layers (e.g., masonry bricks, reinforced concrete beams and columns or steel rebars). Thus, at this stage, the specification of the internal layers is added to each AoD.

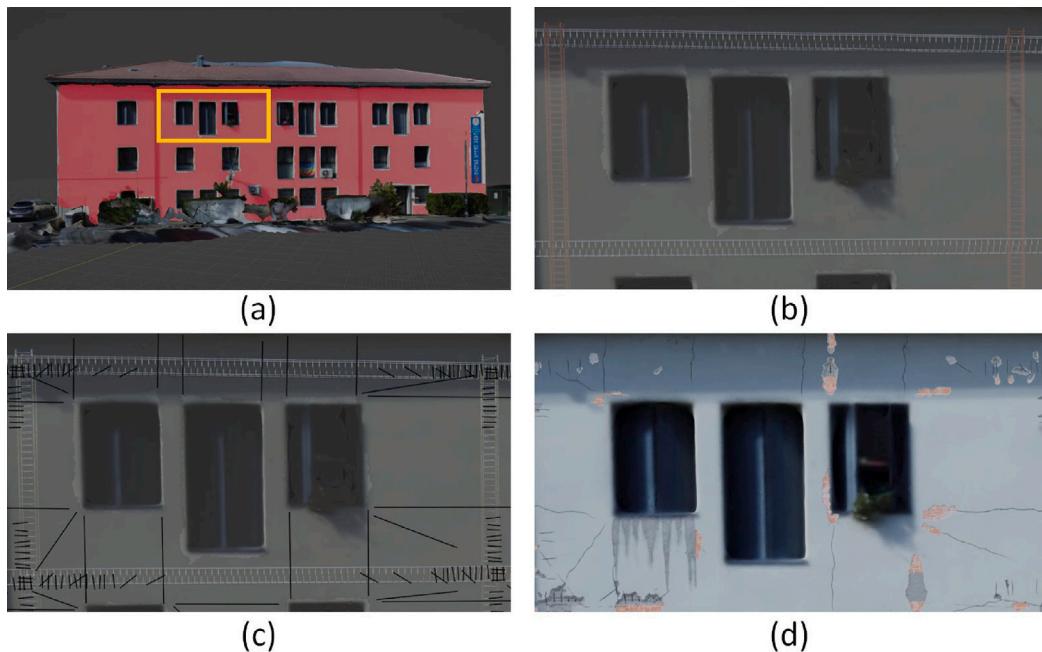
In practice, with reinforced concrete (RC) structures, steel rebars are to be defined explicitly and placed in specific locations (Fig. 2(b)). To speed up this operation, a set of predefined steel reinforcements primitives have been included in the software tool. Such primitives can be combined, modified, and fitted at will (Fig. 3(a)(b)). Other inner layers are specified parametrically as *inner layer types* (e.g., a brick/stone masonry) and will be fitted automatically during the damage generation phase.

The outer surface textures of the AoD are already present in the input 3D model. As the last stage of this step, such textures could be changed, for instance, to turn a plastered wall into exposed masonry. This operation is intended to allow to increase the variability of the simulated conditions.

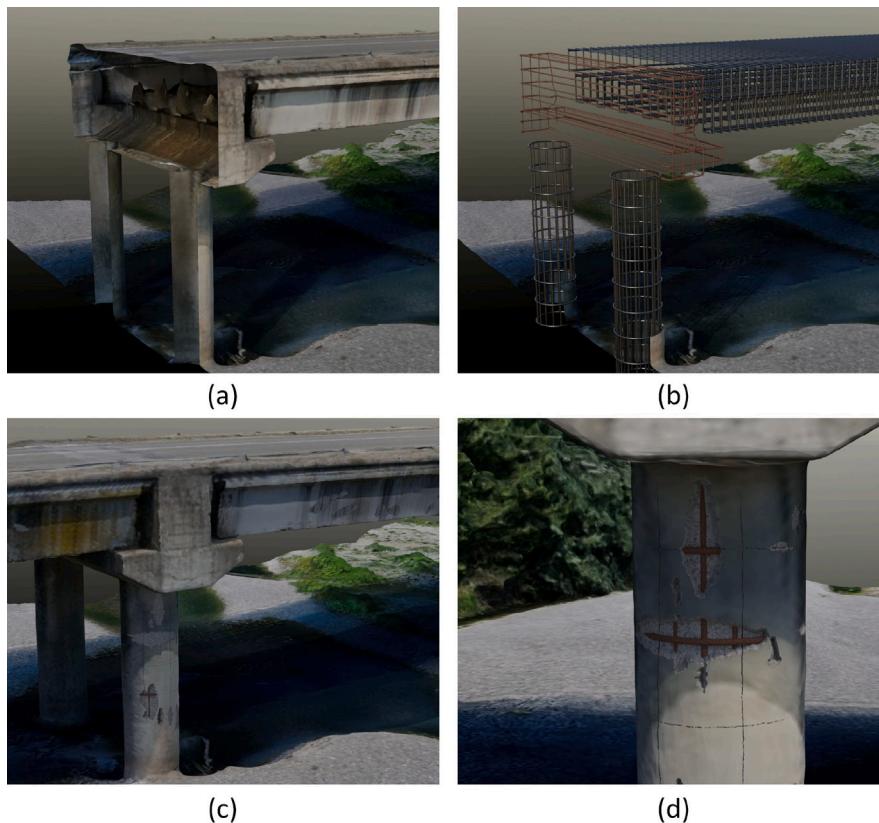
#### 3.1.2. Meta-annotation phase

Meta-annotations are intended to specify the positioning and general constraints that guide the creation of the actual damage instances. The fundamental graphical primitive for defining a meta-annotation is a set of two points connected by a straight line, as shown in Fig. 2(c). Each such primitive is associated with a class of damage and a specific set of parameters, summarized in Table 2.

Since, in the real world, cracks and spallings may overlap, the two classes share the same meta-annotations, including both sets of parameters. This means that the resulting damage instance has a certain probability to be a crack, a spalling or a combination of both. Moreover, multiple small spallings may appear on the same line. Depending on the position, size and depth of the generated damage instances, corroded steel bars may be exposed. Thus, the class corrosion/exposed rebars is not described via a specific meta-annotation, but it is considered as a possible outcome of the crack/spalling damage instances. Finally,



**Fig. 2.** Various stages of the proposed method: (a) selection of an Area of Damage (AoD - in red) on a component, in this case the facade of a building; (b) detail of the addition of the inner steel reinforcements, corresponding to the section of the AoD highlighted in yellow in (a); (c) same detail section with meta-annotations; (d) generated damage instances after parameter randomization.



**Fig. 3.** Example of the inner structure specification for a RC bridge: (a) section of the bridge from the input 3D model; (b) addition of the steel reinforcements; (c) final rendering of the bridge, including the application of damage that makes the steel reinforcements to become visible; (d) detail of the final rendering.

the leaching class is simulated through the application of a texture in keeping with the parameters that specify its aspect. Fig. 2(c)(d) show an example of meta-annotations and corresponding generated damage instances.

Meta-annotations can also be copied and pasted in different positions or grouped together to share a common set of parameter values. For example, grouped meta-annotations can be structured to apply different levels of damage on different positions of an AoD.



**Fig. 4.** Examples of semi-synthetic images with various types of generated damage instances, on different surfaces: top line, buildings; bottom line, bridges.



**Fig. 5.** Different rendered images obtained from the same set of meta-annotations, with randomized parameter values and varying ambient conditions.

### 3.1.3. Damage generation

During this step, actual damage instances are generated by sampling random values within the meta-annotation parametric ranges and by applying the corresponding geometrical transformations and visual effects to the 3D model. Specifically, for crack/spalling, the meta-annotation line is transformed into a 3D object that is shaped according to the random parameter values and then subtracted from the 3D mesh of the AoD. Appropriate corrections are applied to the borders to obtain a smoother blending of the generated instance with the original surface. The shape of each damage instance will be produced in keeping with the type of outer most layer specified. For example, on a masonry wall, cracks will follow the spaces between the bricks. In case where the damage severity is such that the underlying layers are exposed (e.g., due to a large spalling), the inner surfaces of the AoD will be added automatically as a function of the damage depth and the type of materials selected during the inner layers' specification phase. If present, also steel reinforcements will be exposed, coherently merged with the other inner layers (see Fig. 3(c)(d)). Finally, for leaching, the meta-annotation line specifies the starting point of the percolation. Morphological and chromatic transformations are then applied to the selected texture before it is placed on the AoD.

The randomness of the generation process allows us to obtain a wide variety of damage instances from the same set of meta-annotations. This also helps in reproducing more faithfully real-world damage instances, that obviously may vary significantly in shape, extent, and severity, with a limited amount of effort. Fig. 4 shows some examples of the results that could be achieved on various types of structures and surfaces.

### 3.1.4. Image rendering

This final step aims to render a vast amount of semi-synthetic images with bounding box annotations. The simulation of virtual UAS flights can be achieved by defining sequences of camera paths and

angles in Blender. These camera movements around the 3D model will produce images of the same structure from different perspectives and distances. This process aims to replicate the actual perspective that can occur during an UAS flight, including abrupt changes.

World parameters, such as lighting and meteorological conditions, can also be set for each simulated flight. To do this, we took advantage of a specific Blender add-on, called “Sun Position”, which can simulate various lighting and meteorological conditions at a given geographic location and at specific time and date. This makes it possible to reproduce the same conditions of the day of acquisition of the original input 3D model, thus helping to match shadows that are already present in it.

The setup of an entire rendering sequence may include multiple virtual flights, with different levels of damage, lighting, and meteorological conditions. Fig. 5 shows some possible results.

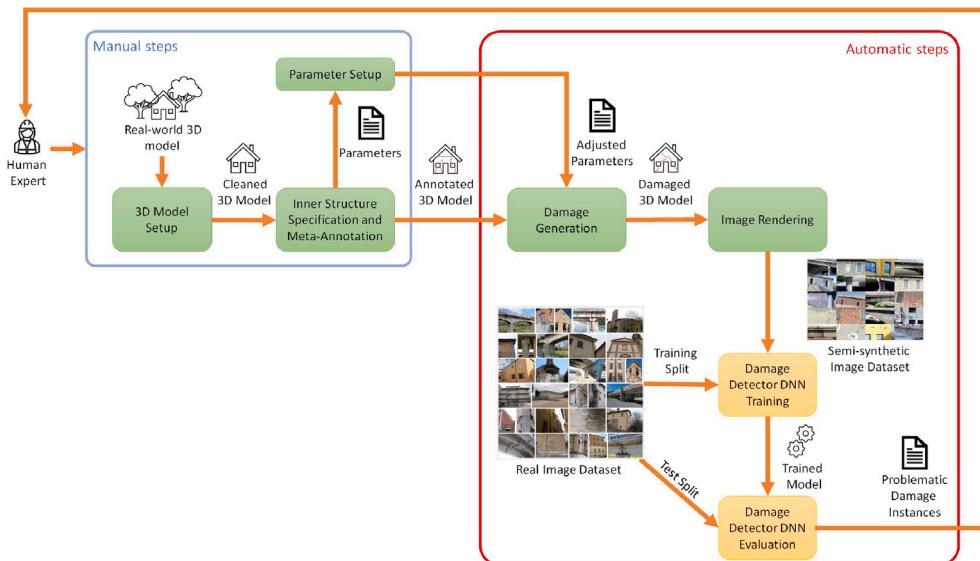
For each frame, aside from the rendered image, the system also creates a segmentation mask that describes the areas corresponding to damage instances, with a specific color for each class. From the latter, bounding box annotations will be generated and then translated into an XML annotation file in Pascal VOC format (Everingham et al., 2015). As a further option, intended for debugging, the same generated images can be produced with overlapping bounding boxes. Fig. 6 shows some examples of rendered images before/after the application of damage, and without/with bounding boxes, respectively.

### 3.2. Tunable data augmentation

One of the main objectives of the method outlined above is enabling a controlled variability in the generation of large datasets of semi-synthetic images. More precisely, by adjusting a limited number of parameter values and/or reusing existing meta-annotations, a semi-synthetic dataset can be iteratively regenerated or extended.



**Fig. 6.** Two examples of the semi-synthetic image generation process, respectively for a church (top row) and a bridge (bottom row): rendered image without damage (left column); rendered image with damage applied (middle column); rendered image with damage applied and overlapping bounding boxes with colors corresponding to damage classes (right column).



**Fig. 7.** Workflow of the overall iterative semi-synthetic image generation method for tunable data augmentation.

The overall procedure, summarized in Fig. 7, involves the iterative training of a DCNN-based damage detector. At each training, semi-synthetic images are used as augmentation of the real ones. Then, by carefully inspecting the results of the detection on real images, specific problematic cases, in which the detector struggles to recognize particular damage, can be identified. For instance, the detector may properly identify cracks on RC bridges yet miss them on masonry bridges. Consequently, the image generation process can be reoriented to increase the representation of those problematic conditions in the training set.

In experimental practice, we adopted the Ultralytics YOLO framework (Jocher et al., 2023) for the damage detector.

### 3.3. Damage tracking and reporting system

A complete and trained damage detector will be integrated into a further software system designed to analyze input videos from UAS surveys on field. The goal is to produce concise summaries that describe all the findings of the detector in a way that could effectively support the analysis by human experts. This can be achieved by first tracking damage instances across different frames, and then selecting meaningful *key images* that could concisely describe all damage occurrences. Overall, the objective of this work was to provide a system capable of processing entire UAS-acquired videos in one run, performed offline.

In contrast, we did not consider the possibility of on-board real-time damage detection, although such an extension could be a possible direction of further investigation. The main phases of the procedure are highlighted in Fig. 8.

An initial tracking of bounding boxes across the sequence of frames is performed using the BoT-SORT algorithm (Aharon et al., 2022). This tracking technique is integrated in the Ultralytics YOLO framework and is designed to track multiple instances at a time, with little overhead in computation. Having been developed for pedestrian tracking, BoT-SORT is not necessarily optimal for the scenario considered. Specifically, in the case of damage detection, bounding boxes may vary highly even across subsequent frames. This aspect is not critical when damage instances are few and well separated (e.g., two cracks at the opposite edges of a wall). In contrast, in severely damaged sections of a structure, several problems may arise. For example, the same damage instance may be erroneously given different IDs in consecutive frames or being associated with several bounding boxes that might have eluded non-maximum suppression (NMS). Moreover, variations in the position of the camera due to abrupt changes in the UAS flight direction may also contribute to mislabeling and redundant detections.

We thus applied some temporal filters as post-processing corrections to refine the outcome of the tracker.

A merging filter checks for the presence of the same class bounding boxes that becomes excessively overlapping during the sequence. Depending on a threshold, a new bounding box that replaces the original

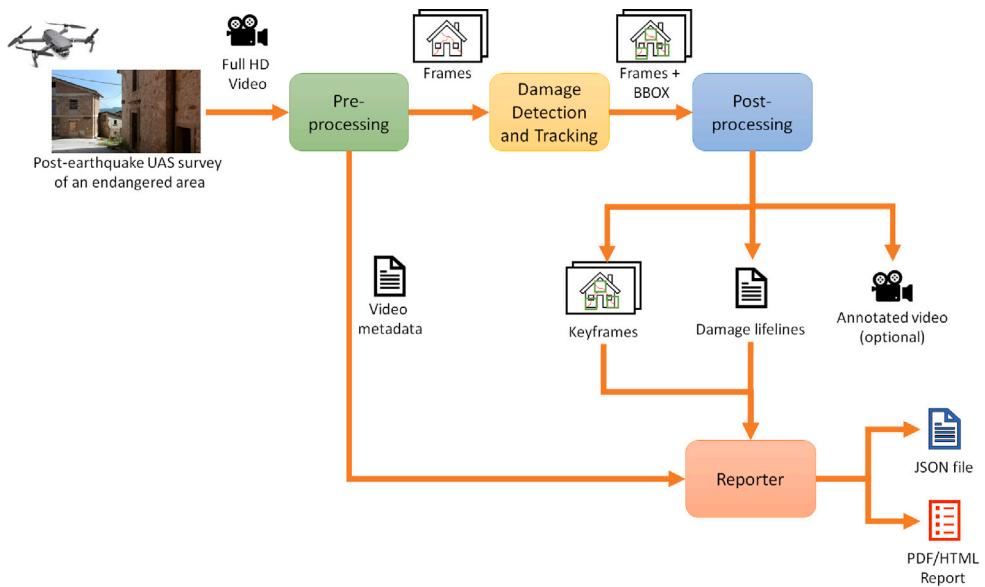


Fig. 8. Overview of the proposed damage tracking and reporting system.

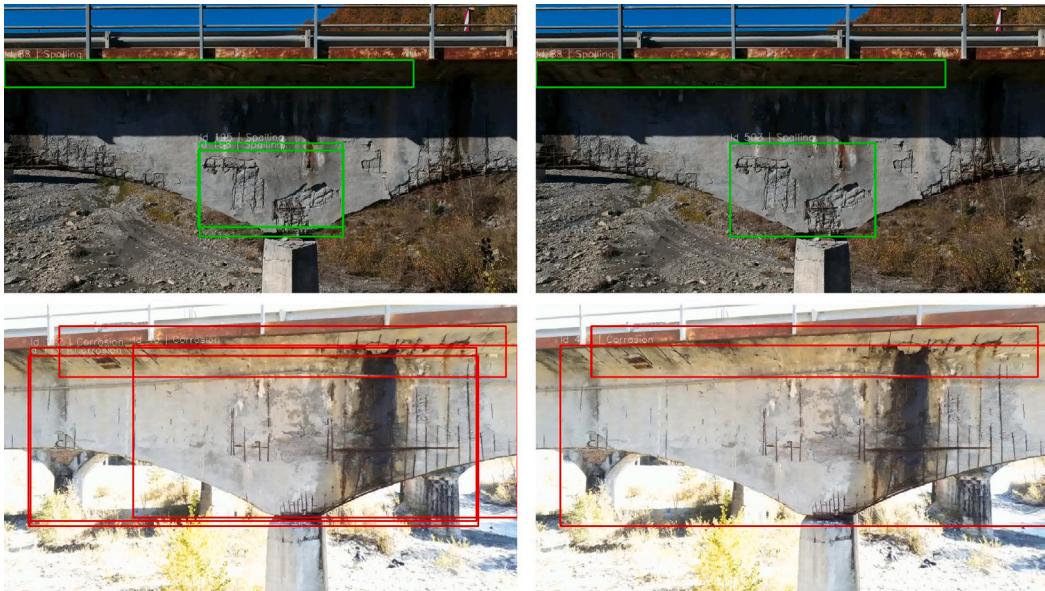


Fig. 9. Examples of the application of the merging filter: original detection (left column) and after correction (right column). For clarity, damage of only one class is shown in each image, although multiple types of damage are present.

ones is created, and a unique ID is assigned to it. Fig. 9 shows some examples of the effect of this correction.

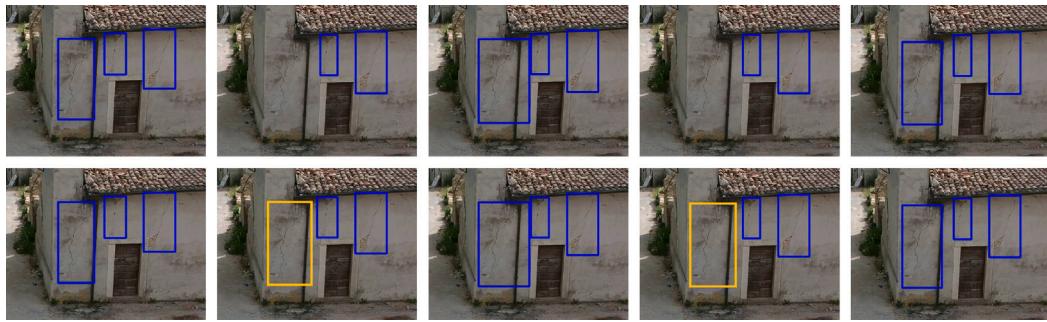
An *interpolation* filter deals with possible bounding box flickering, namely damage instances that keep appearing and disappearing in consecutive frames. In this case, the gaps are filled by creating new bounding boxes through interpolation. These corrections produce a more stable outcome and a cleaner identification of the damage lifelines. An example of interpolation is shown in Fig. 10.

Both merging and interpolation filters operate on bounding boxes only, thus having a very low computational footprint. After the application of these filters, the system identifies the lifelines of each damage instance in the video considered. Each lifeline is associated with a start and finish timestamp. Further on, analyzing the lifelines, it is possible to identify relevant intersections among them, namely a reduced subset of key frames that include all the damage occurrences at least

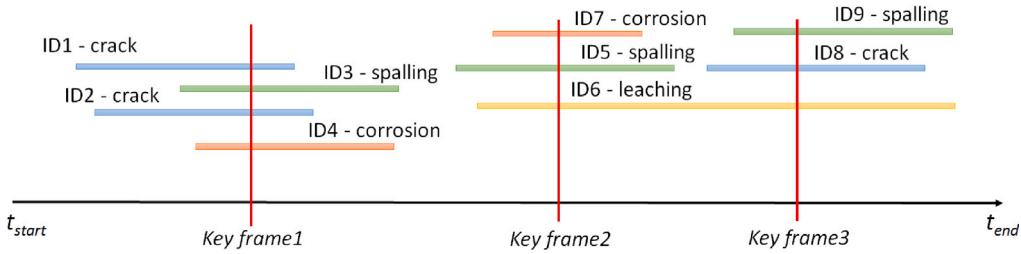
once. Fig. 11 plots an example of lifelines with the position of some key frames. Optionally, at the end of this post-processing stage, an annotated video in MP4 format can be produced, as well.

As the last step, the system collects and summarizes all the detection findings, together with video metadata, and saves them in a JSON file for further processing. To simplify the consultation by human experts, a report, in PDF or HTML format, is also produced. This report includes:

- An initial page with metadata coming from the original video (e.g., date and time of acquisition, geolocation, etc.) together with survey notes redacted by experts.
- One page per key frame, showing its timestamp, the frame image with and without bounding boxes, the IDs of all the damage occurrences being shown.
- A final page, containing a summary table of all the damage instances together with their corresponding lifelines.



**Fig. 10.** Example of the application of the interpolation filter to consecutive frames: original detection sequence (top row), after correction (bottom row). For clarity, interpolated bounding boxes are shown in yellow, even if they belong to the same class.



**Fig. 11.** Example of damage lifelines and key frames.

## 4. Datasets

### 4.1. IDEA dataset

IDEA (Image Database for Earthquake damage Annotation) is a dataset created by the EUCENTRE Foundation. It contains high-resolution, real-world, images acquired during post-earthquake surveys following three recent major events in Italy: L'Aquila (2009), Emilia (2012) and Central Italy (2016–2017). Images in this dataset have been acquired in a variety of lighting and meteorological conditions, different points of view and geometrical scales (structural elements, structural portions, or structure as a whole). Images also frame a variety of structural typologies such as reinforced concrete bridges, reinforced concrete frame buildings, unreinforced brick and/or stone masonry buildings.

At present, the collection and the labeling process of the IDEA dataset is continuously ongoing at EUCENTRE. At the time of the experiments, it contained 4946 annotated images (2317 framing damaged buildings/bridges and 2629 non-damaged buildings/bridges). Damage instances were labeled by human experts by applying bounding boxes associated to class labels. As discussed in Dondi et al. (2024) a correction in these bounding boxes was needed to address possible artifacts. In fact, since actual damage instances have imprecise boundaries, human experts may have different judgments in drawing bounding boxes that label them. To compensate for differences of this kind and for the purposes of the experiments, we introduced a level of tolerance by applying to each bounding box a 3-pixel expansion in all directions.

The IDEA dataset contains seven main classes of damage, subdivided in various sub-classes. In agreement with experts, we focused on the four relevant classes of damage specified in Section 3, namely crack, spalling, corrosion/exposed rebars and leaching. The occurrences per class in the dataset are summarized in Table 3.

### 4.2. Semi-synthetic dataset

The semi-synthetic dataset contains images created through the generation method presented in Section 3.1. 3D models of four building compounds and four bridges were used as input. Such models were

**Table 3**

Number of images and occurrences for each class for the two datasets.

Dataset	Images	Crack	Spalling	Cor./E.R.	Leaching
IDEA	4946	2540	2606	1292	696
Semi-Synthetic	33 836	135 786	149 210	25 491	15 805

obtained by applying the Structure for Motion (SfM) photogrammetric technique (Pepe and Costantino, 2020) on images acquired during routine surveys in the period between 2018 and 2022. Various UAS devices (Dji Mavic 2 Pro, Air 2S, Mini 2) were employed for this purpose. Agisoft Metashape was the photogrammetry software used for generating the 3D mesh, while Meshlab (Cignoni et al., 2008) was used for refinements and for cleaning minor defects and post-processing inaccuracies.

This image dataset was obtained by applying the tunable data augmentation strategy described in Section 3.2. At the end of the iterative process, the dataset contained 33 836 semi-synthetic images (30 059 damaged building/bridges and 3777 non-damaged building/bridges). The occurrences per class are summarized in Table 3.

## 5. Experimental results

### 5.1. Damage detection

To assess the potential advantages of the proposed tunable data augmentation strategy, we carried out a series of comparative experiments, training various DCNN-based damage detectors.

The IDEA dataset was split with a proportion 80:20 into two subsets. The training set included 3952 images, while the testing set comprised 994 images. This subdivision was performed by experts through manual selection, to ensure balancing of damage classes and structure types, to the maximum extent possible, and a similar proportion of images with and without damage. No split was performed on the semi-synthetic dataset since it was used only during training as data augmentation.

As DCNN, we tested the off-the-shelf YOLOv8 (Jocher et al., 2023), YOLOv9 (Wang and Liao, 2024) and YOLOv10 (Wang et al., 2024) architectures. For all of them, we employed the small model, with input

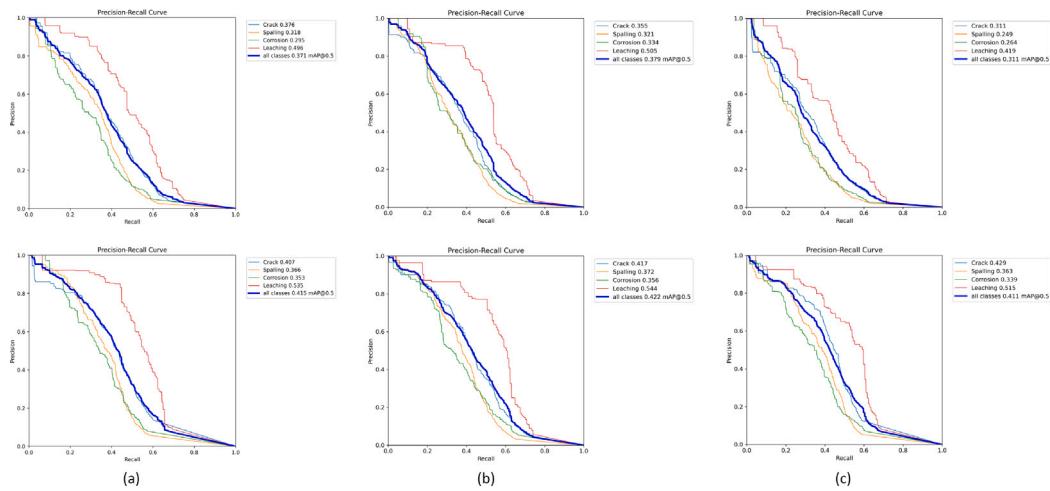


Fig. 12. Precision-Recall (PR) curves for (a) YOLOv8-small, (b) YOLOv9-small and (c) YOLOv10-small. Top row, baseline model; Bottom row, augmented model.

Table 4

Performances of the baseline model with various YOLO architectures. Best results are highlighted in bold, second-best are underlined.

Class	Network	Precision	Recall	F1-score	AP@0.5
Cracks	YOLOv8s	<b>0.615</b>	<u>0.398</u>	<b>0.484</b>	<b>0.376</b>
	YOLOv9s	<u>0.591</u>	<b>0.413</b>	<b>0.486</b>	<u>0.355</u>
	YOLOv10s	<b>0.703</b>	<u>0.293</u>	<u>0.413</u>	<u>0.311</u>
Spalling	YOLOv8s	<u>0.415</u>	<b>0.274</b>	<b>0.330</b>	<u>0.318</u>
	YOLOv9s	<u>0.367</u>	<u>0.243</u>	<u>0.293</u>	<b>0.321</b>
	YOLOv10s	<b>0.424</b>	<u>0.118</u>	<u>0.185</u>	<u>0.249</u>
Corr./Exp. Reb.	YOLOv8s	<u>0.218</u>	<u>0.091</u>	<u>0.128</u>	<u>0.295</u>
	YOLOv9s	<u>0.211</u>	<b>0.110</b>	<b>0.145</b>	<u>0.334</u>
	YOLOv10s	<b>0.264</b>	<u>0.065</u>	<u>0.105</u>	<u>0.264</u>
Leaching	YOLOv8s	<u>0.483</u>	<u>0.474</u>	<u>0.479</u>	<u>0.496</u>
	YOLOv9s	<b>0.524</b>	<b>0.500</b>	<b>0.512</b>	<b>0.505</b>
	YOLOv10s	<u>0.482</u>	<u>0.377</u>	<u>0.423</u>	<u>0.419</u>
All Classes	YOLOv8s	<u>0.471</u>	<u>0.298</u>	<b>0.366</b>	<u>0.371<sup>a</sup></u>
	YOLOv9s	<u>0.444</u>	<b>0.299</b>	<u>0.358</u>	<b>0.379<sup>a</sup></b>
	YOLOv10s	<b>0.527</b>	<u>0.191</u>	<u>0.280</u>	<u>0.311<sup>a</sup></u>

<sup>a</sup> mAP@0.5 instead of AP@0.5.

image size 1088 px, batch size 24, and 50 epochs for training (with best model saving). For optimal performance, all other hyperparameters were tuned separately for each detector. The choice of the small model was dictated by the requirement of having the smallest possible computation footprint for the detector to be used in the field and with potentially limited computational capabilities.

Each DCNN was trained in two ways: (i) as baseline model, using only real images from the training split of the IDEA dataset; (ii) as semi-synthetic augmented model, using the same training split with augmentation of semi-synthetic images. Testing was always performed using the test split of the IDEA dataset.

We started by training the baseline model. Table 4 shows a comparison among the three YOLO architectures considered in terms of Precision, Recall, F1-score and Average Precision (AP), with the Intersection over Union (IoU) threshold set at 0.5. YOLOv8 and YOLOv9 was almost on par, with an mAP@0.5 of 0.371 and 0.379, respectively. Instead, YOLOv10 obtained lower scores, with a mAP@0.5 of 0.311. Comparing the results for each individual class, we noticed that YOLOv9 achieved better scores than YOLOv8 for corrosion/exposed rebars, but worse ones for cracks. Fig. 12 (top row) shows the best Precision-Recall (PR) curves obtained by each YOLO network.

For the semi-synthetic augmented models, we followed the tunable training strategy described in Section 3.2. We analyzed the detector's outcomes during the iterative training to identify problematic cases and updated the semi-synthetic dataset accordingly. For instance, during

Table 5

Performances of the semi-synthetic augmented model with various YOLO architectures. Best results are highlighted in bold, second-best are underlined.

Class	Network	Precision	Recall	F1-score	AP@0.5
Cracks	YOLOv8s	<b>0.726</b>	<u>0.363</u>	<b>0.484</b>	<u>0.407</u>
	YOLOv9s	<u>0.682</u>	<b>0.427</b>	<b>0.525</b>	<u>0.417</u>
	YOLOv10s	<b>0.737</b>	<u>0.406</u>	<u>0.523</u>	<b>0.429</b>
Spalling	YOLOv8s	<u>0.441</u>	<u>0.229</u>	<u>0.302</u>	<u>0.366</u>
	YOLOv9s	<u>0.434</u>	<b>0.264</b>	<u>0.329</u>	<b>0.372</b>
	YOLOv10s	<b>0.457</b>	<u>0.259</u>	<b>0.330</b>	<u>0.363</u>
Corr./Exp. Reb.	YOLOv8s	<u>0.265</u>	<b>0.125</b>	<b>0.170</b>	<u>0.355</u>
	YOLOv9s	<u>0.179</u>	<u>0.108</u>	<u>0.135</u>	<b>0.356</b>
	YOLOv10s	<u>0.237</u>	<u>0.119</u>	<u>0.158</u>	<u>0.339</u>
Leaching	YOLOv8s	<u>0.526</u>	<u>0.461</u>	<u>0.491</u>	<u>0.535</u>
	YOLOv9s	<u>0.562</u>	<b>0.526</b>	<b>0.544</b>	<u>0.544</u>
	YOLOv10s	<b>0.571</b>	<u>0.468</u>	<u>0.514</u>	<u>0.515</u>
All Classes	YOLOv8s	<u>0.517</u>	<u>0.276</u>	<u>0.360</u>	<u>0.415<sup>a</sup></u>
	YOLOv9s	<u>0.490</u>	<b>0.314</b>	<u>0.382</u>	<u>0.422<sup>a</sup></u>
	YOLOv10s	<b>0.529</b>	<u>0.301</u>	<b>0.384</b>	<u>0.411<sup>a</sup></u>

<sup>a</sup> mAP@0.5 instead of AP@0.5.

Table 6

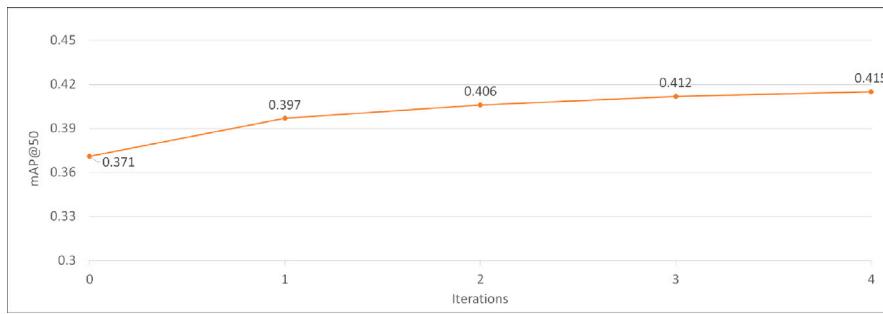
Comparison among the various YOLO architectures, for both baseline and semi-synthetic augmented model, in terms of AP@0.5. Best results are highlighted in bold, second-best are underlined.

Network	Model	Crack	Spalling	Cor./E.R.	Leach.	All
YOLOv8s	Baseline	<b>0.376</b>	<u>0.318</u>	<u>0.295</u>	<u>0.496</u>	<u>0.371<sup>a</sup></u>
YOLOv9s	Baseline	<u>0.355</u>	<b>0.321</b>	<u>0.334</u>	<u>0.505</u>	<u>0.379<sup>a</sup></u>
YOLOv10s	Baseline	<u>0.311</u>	<u>0.249</u>	<u>0.264</u>	<u>0.419</u>	<u>0.311<sup>a</sup></u>
YOLOv8s	Augmented	<u>0.407</u>	<b>0.366</b>	<u>0.355</u>	<u>0.535</u>	<u>0.415<sup>a</sup></u>
YOLOv9s	Augmented	<u>0.417</u>	<b>0.372</b>	<u>0.356</u>	<u>0.544</u>	<u>0.422<sup>a</sup></u>
YOLOv10s	Augmented	<b>0.429</b>	<u>0.363</u>	<u>0.339</u>	<u>0.515</u>	<u>0.411<sup>a</sup></u>

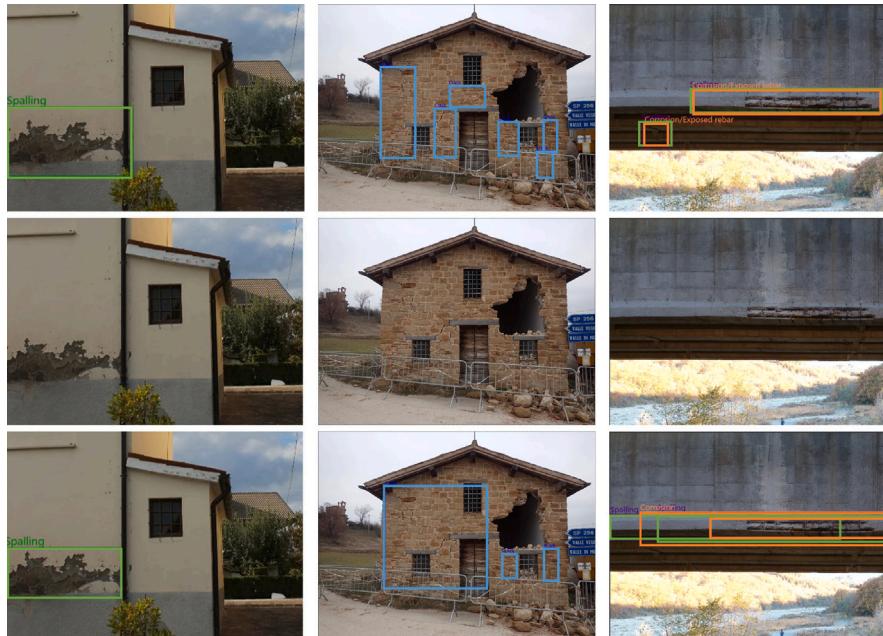
<sup>a</sup> mAP@0.5 instead of AP@0.5.

the initial stages, the DCNN detector frequently missed cracks on brick masonry walls. Therefore, we tuned the parameters of the meta-annotations to create more semi-synthetic images of that particular kind. Similar adjustments were made also for the other classes of damage. Variations in ambient conditions and flight paths also helped to handle the complexities of the scenario considered. Since, in the end, semi-synthetic images largely outnumbered the real ones by a ratio of 1:6, we oversampled the real images to match the number of the semi-synthetic ones and avoid biasing the detector. Fig. 13 shows an example of the improvement in the mAP@0.5 score as iterations progress.

Table 5 shows the results obtained by the three YOLO architectures for the semi-synthetic augmented model, while Fig. 12 (bottom



**Fig. 13.** Example of iterative tuning of the semi-synthetic image generation process. Iteration 0 shows the baseline result before augmentation.



**Fig. 14.** Comparative examples of the predictions of the two detectors: ground truth (top row); prediction of the baseline model (middle row); prediction of the semi-synthetic augmented model (bottom row).

row) shows the corresponding best PR curves. Comparing these scores with the baseline results (Fig. 12 (top row) and Table 4), we can notice an improvement in all metrics for all the networks, proving the effectiveness of the proposed augmentation strategy. More precisely, the summary comparison in Table 6 highlights a significant growth in AP@0.5 for all classes, with an average increase of 12% for YOLOv8 and YOLOv9, and 30% for YOLOv10 that shows the greater improvement overall, reaching results comparable (or better) to the other two networks. Fig. 14 shows some examples of the enhanced detection capabilities of the augmented model with respect to the baseline on both buildings and bridges.

## 5.2. Many-to-Many metrics

By comparing the top and bottom rows of Fig. 14, it can be noticed that ground-truth and predicted bounding boxes differ significantly in both number and size, even when the damage is detected, nonetheless. This outcome is common in the scenario considered. In fact, unlike standard target objects like cars or pedestrians, damage instances may have ambiguous shapes and boundaries, which could be annotated and detected in several yet equally valid ways. For instance, a long crack could be labeled with a single box by one expert and by two or more smaller boxes by another.

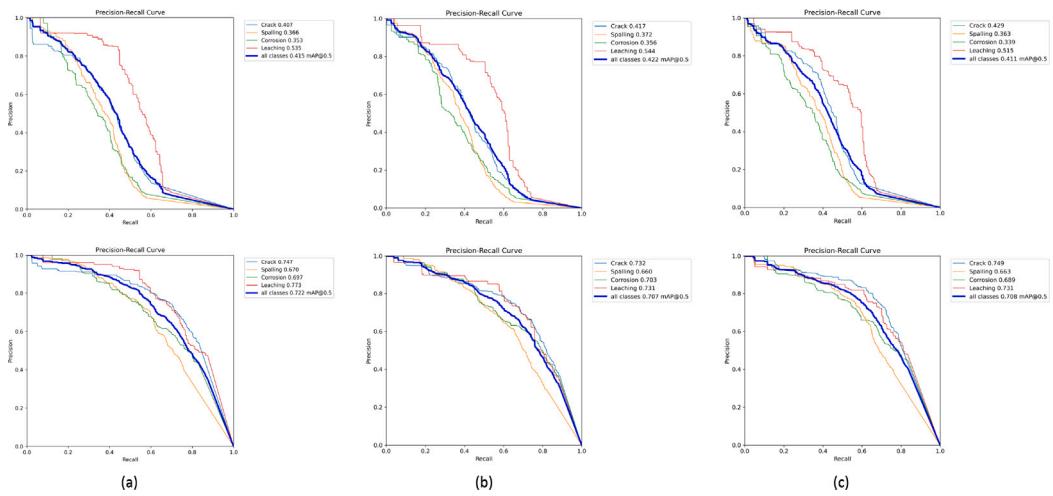
If this aspect is not considered, the performance of a detector may be significantly underestimated. In fact, standard metrics based on

IoU assume a one-to-one correspondence between ground-truth and predicted bounding boxes. In damage detection, on the other hand, multiple ground-truth boxes (GTBs) can be matched by multiple prediction boxes (PBs), and vice versa. To deal with this, Dondi et al. (2023) proposed the alternative Many-to-Many metrics. In this approach, Precision and Recall are computed using two different variants of the IoU metric: precision is evaluated using *Intersection over Prediction* (IoP, see Eq. (1)), while recall is evaluated using *Intersection over Ground Truth* (IoG, see Eq. (2)). Note that, in both IoP and IoG the *area of overlap* may relate to either multiple PBs or multiple GTBs.

$$\text{IoP} = \frac{\text{area of overlap}}{\text{area of PB}} \quad (1)$$

$$\text{IoG} = \frac{\text{area of overlap}}{\text{area of GTB}} \quad (2)$$

Table 7 compares the performance of the semi-synthetic augmented models using the standard metrics and the Many-to-Many metrics, fixing at 0.5 the threshold for IoU, IoP and IoG. We can notice a significant increase in all values for all the networks. Overall, YOLOv8 is the network with the highest growth in mAP@0.5, from 0.415 to 0.722. Even considering the single classes it always reaches the best or the second best score. Accordingly, the PR curves computed with the Many-to-Many metrics are smoother and more compact than the PR curves computed with the standard one (see Fig. 15).



**Fig. 15.** Precision-Recall (PR) curve of the semi-synthetic augmented model for (a) YOLOv8-small, (b) YOLOv9-small and (c) YOLOv10-small. Top row, computation with the standard metrics; Bottom row, computation with the Many-to-Many metrics.

**Table 7**

Comparison among various YOLO architectures for the semi-synthetic augmented model in terms of AP@0.5, computed using both standard and Many-to-Many (MtoM) metrics. Best results for each category are highlighted in bold, second-best are underlined.

Network	Metrics	Crack	Spalling	Cor./E.R.	Leach.	All
YOLOv8s	Standard	0.407	<u>0.366</u>	<u>0.355</u>	<u>0.535</u>	<u>0.415<sup>a</sup></u>
YOLOv9s	Standard	0.417	<u>0.372</u>	<u>0.356</u>	<u>0.544</u>	<u>0.422<sup>a</sup></u>
YOLOv10s	Standard	<b>0.429</b>	0.363	0.339	0.515	<u>0.411<sup>a</sup></u>
YOLOv8s	MtoM	<b>0.747</b>	<b>0.670</b>	<u>0.697</u>	<b>0.773</b>	<b>0.722<sup>a</sup></b>
YOLOv9s	MtoM	0.732	0.660	<u>0.703</u>	<u>0.731</u>	<u>0.707<sup>a</sup></u>
YOLOv10s	MtoM	<b>0.749</b>	<u>0.663</u>	0.689	<u>0.731</u>	<u>0.708<sup>a</sup></u>

<sup>a</sup> mAP@0.5 instead of AP@0.5.

**Table 8**

Performances of the semi-synthetic augmented model with various YOLO architectures, computed using the Many-to-Many metrics. Best results are highlighted in bold, second-best are underlined.

Class	Network	Precision	Recall	F1-score	AP@0.5
Cracks	YOLOv8s	<u>0.873</u>	0.522	0.653	<u>0.747</u>
	YOLOv9s	0.812	<b>0.614</b>	<b>0.699</b>	0.732
	YOLOv10s	<b>0.892</b>	<u>0.557</u>	0.686	<b>0.749</b>
Spalling	YOLOv8s	<u>0.819</u>	0.454	0.584	<b>0.670</b>
	YOLOv9s	0.768	<b>0.501</b>	<u>0.606</u>	0.660
	YOLOv10s	<b>0.821</b>	<u>0.491</u>	<b>0.614</b>	<u>0.663</u>
Corr./Exp. Reb.	YOLOv8s	<u>0.836</u>	0.414	0.554	<u>0.697</u>
	YOLOv9s	<b>0.758</b>	<b>0.521</b>	<b>0.618</b>	<b>0.703</b>
	YOLOv10s	0.823	<u>0.456</u>	<u>0.587</u>	0.689
Leaching	YOLOv8s	<b>0.809</b>	<u>0.623</u>	<b>0.704</b>	<b>0.773</b>
	YOLOv9s	0.747	<u>0.662</u>	<u>0.702</u>	<u>0.731</u>
	YOLOv10s	<u>0.800</u>	0.610	0.692	<u>0.731</u>
All	YOLOv8s	<u>0.839</u>	0.484	0.614	<u>0.722<sup>a</sup></u>
	YOLOv9s	0.780	<b>0.558</b>	<u>0.651</u>	<u>0.707<sup>a</sup></u>
	YOLOv10s	<b>0.844</b>	<u>0.517</u>	<u>0.641</u>	<u>0.708<sup>a</sup></u>

<sup>a</sup> mAP@0.5 instead of AP@0.5.

**Table 8** shows the detailed performance per class and per networks using the Many-to-Many metrics. In reference to the examples **Fig. 14**, the Many-to-Many metrics appear to yield a more effective estimation of the capabilities of the damage detector.

### 5.3. Damage tracking and reporting system

The objective of developing a DCNN-based damage detector is to implement a screening tool that could be used by human experts on field, to rapidly analyze a series of post-earthquake videos acquired during UAS surveys.

To validate the tracking and reporting system described in Section 3.3, we used ten sample videos provided by EUCENTRE. Four of these videos relate to bridges (one masonry and three RC) and six to buildings. In passing, some of the latter were shot in locations hit by the 2016–2017 Central Italy seismic sequences. All sample videos were recorded by cameras mounted on an UAS at full-HD resolution (1920 × 1080). Since, as in the real case, videos were not annotated, the validation of the results was performed through a review made by human experts.

By comparing the produced reports (**Fig. 16**) with the complete video footages, according to experts, it resulted that the tracking system could detect the most significant damage sequences on moderate-to-severely damaged structures. In their opinion, only an acceptable number of misdetections and false positives were present. Furthermore, they also acknowledged that the tracking and reporting system could effectively process a large amount of video data, thereby accelerating the inspection process.

In addition, we tested the system on a PC with limited computational capabilities. Specifically, a machine with an Intel Xeon E5-1630 v3 CPU, 32 GB of RAM and a GTX 1080Ti GPU. As DCNN, we used the semi-synthetic augmented model with YOLOv8 small. The measured inference time was 16 ms per frame, on average, which was deemed acceptable for practical purposes.

### 6. Limitations and applicability conditions

In the approach presented, we aimed to automate the image generation and annotation process as much as possible. Nonetheless, setting up the semi-synthetic image generation system still requires substantial human expert intervention that needs to be translated into Blender instructions. The meta-annotation phase and inner-structure specification, in particular, require structural engineering expertise for proper placement and appropriate layering according to the type of structure and damage considered. Additionally, the generation of well-tuned semi-synthetic images may need several iterations of the process described in **Fig. 7**.

The requirement for high resolution 3D models of real-world buildings and bridges is another potential limitation of the semi-synthetic image generation method. However, the digitization of civil structures is becoming increasingly common in the field of structural engineering (Rakha and Gorodetsky, 2018; Pepe and Costantino, 2020; Jayasinghe et al., 2024). Consequently, the availability of high-resolution 3D models is projected to grow in the coming years.

The tracking and reporting system is intended to work with UAS-acquired videos taken at low altitude in a range of 2–7 m from the

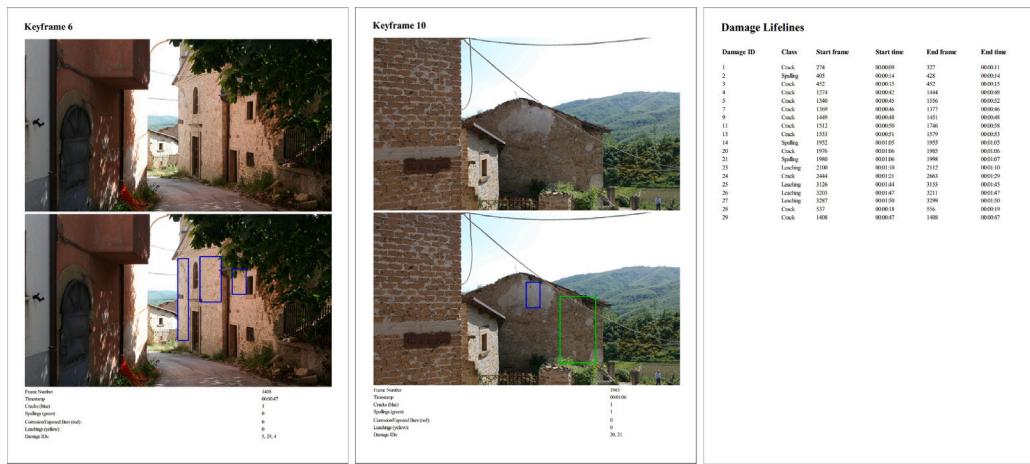


Fig. 16. Some sample pages from a report obtained by one of the videos analyzed.

damaged structures. Experiments performed with the analysis of such videos revealed that the camera angle can influence the quality of the detection. More precisely, it became evident that the detector may miss damage instances when seen from a too small angle. Consequently, it follows the recommendation to UAS pilots to frame target structures as frontally as possible. Experiments also showed that low flight speeds could improve the performance of temporal filtering.

## 7. Conclusions

In this paper, we introduced a novel system capable of detecting four indicative classes of structural damage. Our approach was specifically designed to assist first responders during post-earthquake assessments, with the goal of expediting the inspection of impacted areas.

To train our DCNN-based damage detector, we developed an iterative tunable data augmentation strategy that employs semi-synthetic images. Such images were generated starting from real-world 3D models of buildings and bridges with the application of parametric meta-annotations. The results of the experiments performed on various DCNN architectures show that the proposed augmentation method leads to improved performance of automatic damage detectors, and thus could be effective in addressing the scarcity of real-world datasets.

In our opinion, a key advantage of the proposed semi-synthetic image generation method is the use of meta-annotations. This approach allows for the creation of a vast number of diverse images with varying damage instances from a limited amount of human input, all while respecting structural engineering constraints.

As future works, we plan to extend the method described to further relevant classes of damage, such as debris or more complex collapse effects, and to other structural typologies. Additionally, we aim to enhance the DCNN-system's detection capabilities by exploring the integration of the Many-to-Many metrics into the loss function. Improvements to the tracking and reporting system, following experts' suggestions, will be considered as well.

## CRediT authorship contribution statement

**Piercarlo Dondi:** Writing – original draft, Visualization, Software, Methodology, Formal analysis. **Alessio Gullotti:** Visualization, Software, Methodology. **Michele Inchegolo:** Software, Methodology, Investigation. **Ilaria Senaldi:** Validation, Methodology, Investigation, Funding acquisition. **Chiara Casarotti:** Project administration, Funding acquisition. **Luca Lombardi:** Writing – review & editing, Supervision. **Marco Piastra:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization.

## Declaration of competing interest

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

## Acknowledgments

This work has been carried out under the financial support of the project TeamAware (European Union's Horizon 2020 research and innovation program, grant agreement No 101019808) and of the Italian Civil Protection, within the framework of the Executive Project 2022–2023 and 2024–2026. We would like to thank André Raucher and Martin Raynaud for their help in the initial development of the tracking system.

## Data availability

The IDEA dataset will be made available from the EUCENTRE Foundation web site (<https://www.eucentre.it/>). Access will be granted upon reasonable request. Access to the semi-synthetic dataset generated for the experimentation will be made available in conjunction to the access to the IDEA dataset, upon explicit request.

## References

- Agyemang, I.O., Zeng, L., Chen, J., Adjei-Mensah, I., Acheampong, D., 2024. Multi-visual modality micro drone-based structural damage detection. Eng. Appl. Artif. Intell. 133, 108460. <http://dx.doi.org/10.1016/j.engappai.2024.108460>.
- Aharon, N., Orfaig, R., Bobrovsky, B.-Z., 2022. BoT-SORT: Robust associations multi-pedestrian tracking. <http://dx.doi.org/10.48550/arXiv.2206.14651>, arXiv: 2206.14651.
- Arya, D., Maeda, H., Ghosh, S.K., Toshniwal, D., Sekimoto, Y., 2021. RDD2020: An annotated image dataset for automatic road damage detection using deep learning. Data Brief 36, 107133. <http://dx.doi.org/10.1016/j.dib.2021.107133>.
- Avci, O., Abdeljaber, O., Kiranyaz, S., Hussein, M., Gabouj, M., Inman, D.J., 2021. A review of vibration-based damage detection in civil structures: From traditional methods to machine learning and deep learning applications. Mech. Syst. Signal Process. 147, 107077. <http://dx.doi.org/10.1016/j.ymssp.2020.107077>.
- Bai, Y., Sezen, H., Yilmaz, A., 2021. Detecting cracks and spalling automatically in extreme events by end-to-end deep learning frameworks. ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. V-2-2021, 161–168. <http://dx.doi.org/10.5194/isprs-annals-V-2-2021-161-2021>.
- Bai, Y., Zha, B., Sezen, H., Yilmaz, A., 2023. Engineering deep learning methods on automatic detection of damage in infrastructure due to extreme events. Struct. Heal. Monit. 22 (1), 338–352. <http://dx.doi.org/10.1177/14759217221083649>.
- Bayomi, N., Fernandez, J.E., 2023. Eyes in the Sky: Drones applications in the built environment under climate change challenges. Drones 7 (10), <http://dx.doi.org/10.3390/drones7100637>.

- Bhatta, S., Dang, J., 2024. Multiclass seismic damage detection of buildings using quantum convolutional neural network. *Comput.-Aided Civ. Infrastruct. Eng.* 39 (3), 406–423. <http://dx.doi.org/10.1111/mice.13084>.
- Bianchi, E., Hebdon, M., 2021. COCO-Bridge 2021+ Dataset. University Libraries, Virginia Tech, <http://dx.doi.org/10.7294/16624495.v1>.
- Bianchi, E., Hebdon, M., 2022. Visual structural inspection datasets. *Autom. Constr.* 139, 104299. <http://dx.doi.org/10.1016/j.autcon.2022.104299>.
- Blender Team, 2023. Blender, Version 3.6 LTS. Blender Foundation, URL <http://www.blender.org>.
- Boikov, A., Payor, V., Savelev, R., Kolesnikov, A., 2021. Synthetic data generation for steel defect detection and classification using deep learning. *Symmetry* 13 (7), <http://dx.doi.org/10.3390/sym13071176>.
- Branikas, E., Murray, P., West, G., 2023. A novel data augmentation method for improved visual crack detection using generative adversarial networks. *IEEE Access* 11, 22051–22059. <http://dx.doi.org/10.1109/ACCESS.2023.3251988>.
- Canchila, C., Zhou, S., Song, W., 2024. Hyperparameter optimization and importance ranking in deep learning-based crack segmentation. *J. Comput. Civ. Eng.* 38 (2), 04023042. <http://dx.doi.org/10.1061/JCCEE5.CPENG-5512>.
- Cha, Y.-J., Choi, W., Büyüköztürk, O., 2017. Deep learning-based crack damage detection using convolutional neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* 32 (5), 361–378. <http://dx.doi.org/10.1111/mice.12263>.
- Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O., 2018. Autonomous structural visual Inspection Using Region-based deep learning for detecting multiple damage types. *Comput.-Aided Civ. Infrastruct. Eng.* 33 (9), 731–747. <http://dx.doi.org/10.1111/mice.12334>.
- Choi, W., Cha, Y.-J., 2020. SDDNet: Real-time crack segmentation. *IEEE Trans. Ind. Electron.* 67 (9), 8016–8025. <http://dx.doi.org/10.1109/TIE.2019.2945265>.
- Chow, J., Su, Z., Wu, J., Tan, F., Mao, X., Wang, Y., 2020. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Adv. Eng. Inform.* 45, 101105. <http://dx.doi.org/10.1016/j.aei.2020.101105>.
- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G., 2008. MeshLab: an open-source mesh processing tool. In: Scarano, V., Chiara, R.D., Erra, U. (Eds.), Eurographics Italian Chapter Conference. The Eurographics Association, pp. 129–136. <http://dx.doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136>.
- Dais, D., Bal, İ.E., Smyrou, E., Sarhosis, V., 2021. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Autom. Constr.* 125, 103606. <http://dx.doi.org/10.1016/j.autcon.2021.103606>.
- Davoudi, R., Miller, G.R., Kutz, J.N., 2018a. Data-driven vision-based inspection for reinforced concrete beams and slabs: Quantitative damage and load estimation. *Autom. Constr.* 96, 292–309. <http://dx.doi.org/10.1016/j.autcon.2018.09.024>.
- Davoudi, R., Miller, G.R., Kutz, J.N., 2018b. Structural load estimation using machine vision and surface crack patterns for shear-critical RC beams and slabs. *J. Comput. Civ. Eng.* 32 (4), 04018024. [http://dx.doi.org/10.1061/\(ASCE\)CP.1943-5487.0000766](http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000766).
- Dewan, T., Anirudh, G., Tembhurne, J.V., 2023. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools Appl.* 82 (6), 9243–9275. <http://dx.doi.org/10.1007/s11042-022-13644-y>.
- Dondi, P., Gullotti, A., Inchigolo, M., Senaldi, I., Casarotti, C., Lombardi, L., Piastri, M., 2024. Improving post-earthquake crack detection using semi-synthetic generated images. arXiv preprint [arXiv:2412.05042](https://arxiv.org/abs/2412.05042).
- Dondi, P., Senaldi, I., Lombardi, L., Piastri, M., 2023. Many-to-Many metrics: A new approach to evaluate the performance of structural damage detection networks. In: Foresti, G.L., Fusillo, A., Hancock, E. (Eds.), Image Analysis and Processing – ICIAP 2023, Lecture Notes in Computer Science. Vol. 14234, Springer Nature Switzerland, Cham, pp. 144–155. [http://dx.doi.org/10.1007/978-3-031-43153-1\\_13](http://dx.doi.org/10.1007/978-3-031-43153-1_13).
- Dung, C.V., Sekiya, H., Hirano, S., Okatani, T., Miki, C., 2019. A vision-based method for crack detection in gusset plate welded joints of steel bridges using deep convolutional neural networks. *Autom. Constr.* 102, 217–229. <http://dx.doi.org/10.1016/j.autcon.2019.02.013>.
- Dunphy, K., Fekri, M.N., Grolinger, K., Sadhu, A., 2022. Data augmentation for deep-learning-based multiclass structural damage detection using limited information. *Sensors* 22 (16), <http://dx.doi.org/10.3390/s22166193>.
- Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* 111, 98–136. <http://dx.doi.org/10.1007/s11263-014-0733-5>.
- Flah, M., Nunez, I., Ben Chaabene, W., Nehdi, M.L., 2021. Machine learning algorithms in civil structural health monitoring: a systematic review. *Arch. Comput. Methods Eng.* 28, 2621–2643. <http://dx.doi.org/10.1007/s11831-020-09471-9>.
- Gao, Y., Li, H., Fu, W., 2023. Few-shot learning for image-based bridge damage detection. *Eng. Appl. Artif. Intell.* 126, 107078. <http://dx.doi.org/10.1016/j.engappai.2023.107078>.
- Gao, Y., Mosalam, K.M., 2020. PEER Hub ImageNet: A large-scale multiattribute benchmark data set of structural images. *J. Struct. Eng.* 146 (10), 04020198. [http://dx.doi.org/10.1061/\(ASCE\)ST.1943-541X.0002745](http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0002745).
- Halder, S., Afsari, K., 2023. Robots in inspection and monitoring of buildings and infrastructure: A systematic review. *Appl. Sci.* 13 (4), <http://dx.doi.org/10.3390/app13042304>.
- Hoskere, V., Narazaki, Y., Hoang, T., Spencer, B., 2020. Madnet: multi-task semantic segmentation of multiple types of structural materials and damage in images of civil infrastructure. *J. Civ. Struct. Heal. Monit.* 10 (5), 757–773. <http://dx.doi.org/10.1007/s13349-020-00409-0>.
- Hoskere, V., Narazaki, Y., Spencer, B.F., Smith, M.D., 2019. Deep learning-based damage detection of miter gates using synthetic imagery from computer graphics. In: 12th International Workshop on Structural Health Monitoring: Enabling Intelligent Life-Cycle Health Management for Industry Internet of Things (IIOT), IWSHM 2019. DEStech Publications Inc., pp. 3073–3080.
- Hu, X., Liu, W., Wen, H., Yuen, K.-V., Jin, T., Junior, A.C.N., Zhong, P., 2024. AF-Net: An active fire detection model using improved object-contextual representations on unbalanced UAV datasets. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* 17, 13558–13569. <http://dx.doi.org/10.1109/JSTARS.2024.3406767>.
- Jain, S., Seth, G., Paruthi, A., Soni, U., Kumar, G., 2022. Synthetic data augmentation for surface defect detection and classification using deep learning. *J. Intell. Manuf.* 1–14. <http://dx.doi.org/10.1007/s10845-020-01710-x>.
- Jayasinghe, S., Mahmoodian, M., Sidiq, A., Nanayakkara, T., Alavi, A., Mazaheri, S., Shahrivar, F., Sun, Z., Setunge, S., 2024. Innovative digital twin with artificial neural networks for real-time monitoring of structural response: A port structure case study. *Ocean Eng.* 312, 119187. <http://dx.doi.org/10.1016/j.oceaneng.2024.119187>.
- Jocher, G., Chaurasia, A., Qiu, J., 2023. YOLO by ultralytics. URL <https://github.com/ultralytics/ultralytics>.
- Kang, D., Benipal, S.S., Gopal, D.L., Cha, Y.-J., 2020. Hybrid pixel-level concrete crack segmentation and quantification across complex backgrounds using deep learning. *Autom. Constr.* 118, 103291. <http://dx.doi.org/10.1016/j.autcon.2020.103291>.
- Kim, J.-H., Lee, J., 2023. Efficient dataset collection for concrete crack detection with spatial-adaptive data augmentation. *IEEE Access* 11, 121902–121913. <http://dx.doi.org/10.1109/ACCESS.2023.3328243>.
- Li, B., Guo, H., Wang, Z., 2024. Data augmentation using CycleGAN-based methods for automatic bridge crack detection. *Structures* 62, 106321. <http://dx.doi.org/10.1016/j.istruc.2024.106321>.
- Li, S., Zhao, X., Zhou, G., 2019. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. *Comput.-Aided Civ. Infrastruct. Eng.* 34 (7), 616–634. <http://dx.doi.org/10.1111/mice.12433>.
- Lv, X., Duan, F., Jiang, J.-j., Fu, X., Gan, L., 2020. Deep metallic surface defect detection: The new benchmark and detection network. *Sensors* 20 (6), <http://dx.doi.org/10.3390/s20061562>.
- Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., Omata, H., 2021. Generative adversarial network for road damage detection. *Comput.-Aided Civ. Infrastruct. Eng.* 36 (1), 47–60. <http://dx.doi.org/10.1111/mice.12561>.
- Mandriola, M., Casarotti, C., Peloso, S., Lanese, I., Brunesi, E., Senaldi, I., 2022. Use of UAS for damage inspection and assessment of bridge infrastructures. *Int. J. Disaster Risk Reduct.* 72, 102824. <http://dx.doi.org/10.1016/j.ijdrr.2022.102824>.
- Mandal, T.G., Jahanshahi, M.R., Wu, Z.Y., 2023. Deep learning-based RGB-D fusion for multimodal condition assessment of civil infrastructure. *J. Comput. Civ. Eng.* 37 (4), 04023017. <http://dx.doi.org/10.1061/JCCEE5.CPENG-5197>.
- Mundt, M., Majumder, S., Murali, S., Panetsos, P., Ramesh, V., 2019. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, IEEE Computer Society, Los Alamitos, CA, USA, pp. 11188–11197. <http://dx.doi.org/10.1109/CVPR.2019.01145>.
- Nguyen, N.H.T., Perry, S., Bone, D., Le, H.T., Nguyen, T.T., 2021. Two-stage convolutional neural network for road crack detection and segmentation. *Expert Syst. Appl.* 186, 115718. <http://dx.doi.org/10.1016/j.eswa.2021.115718>.
- Pepe, M., Costantino, D., 2020. Techniques, tools, platforms and algorithms in close range photogrammetry in building 3D model and 2D representation of objects and complex architectures. *Comput. Aided Des. Appl.* 18, 42–65. <http://dx.doi.org/10.1473/cadaps.2021.42-65>.
- Polovnikov, V., Alekseev, D., Vinogradov, I., Lashkia, G.V., 2021. DAUNet: Deep augmented neural network for pavement crack segmentation. *IEEE Access* 9, 125714–125723. <http://dx.doi.org/10.1109/ACCESS.2021.3111223>.
- Rakha, T., Gorodetsky, A., 2018. Review of unmanned aerial system (UAS) applications in the built environment: Towards automated building inspection procedures using drones. *Autom. Constr.* 93, 252–264. <http://dx.doi.org/10.1016/j.autcon.2018.05.002>.
- Seo, J., Duque, L., Wacker, J., 2018. Drone-enabled bridge inspection methodology and application. *Autom. Constr.* 94, 112–126. <http://dx.doi.org/10.1016/j.autcon.2018.06.006>.
- Sony, S., Dunphy, K., Sadhu, A., Capretz, M., 2021. A systematic review of convolutional neural network-based structural condition assessment techniques. *Eng. Struct.* 226, 111347. <http://dx.doi.org/10.1016/j.engstruct.2020.111347>.
- Su, G., Qin, Y., Xu, H., Liang, J., 2024. Automatic real-time crack detection using lightweight deep learning models. *Eng. Appl. Artif. Intell.* 138, 109340. <http://dx.doi.org/10.1016/j.engappai.2024.109340>.
- Terven, J., Córdoba-Esparza, D.-M., Romero-González, J.-A., 2023. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 5 (4), 1680–1716. <http://dx.doi.org/10.3390/make5040083>.

- Thai, H.-T., 2022. Machine learning for structural engineering: A state-of-the-art review. *Structures* 38, 448–491. <http://dx.doi.org/10.1016/j.istruc.2022.02.003>.
- Wan, H., Gao, L., Yuan, Z., Qu, H., Sun, Q., Cheng, H., Wang, R., 2023. A novel transformer model for surface damage detection and cognition of concrete bridges. *Expert Syst. Appl.* 213, 119019. <http://dx.doi.org/10.1016/j.eswa.2022.119019>.
- Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G., 2024. YOLOv10: Real-time end-to-end object detection. arXiv preprint [arXiv:2405.14458](https://arxiv.org/abs/2405.14458).
- Wang, C.-Y., Liao, H.-Y.M., 2024. YOLOv9: Learning what you want to learn using programmable gradient information. arXiv preprint [arXiv:2402.13616](https://arxiv.org/abs/2402.13616).
- Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., Chen, X., 2019. Automatic bridge crack detection using a convolutional neural network. *Appl. Sci.* 9 (14), <http://dx.doi.org/10.3390/app9142867>.
- Xu, Y., Turkan, Y., 2020. BRIM and UAS for bridge inspections and management. *Eng. Constr. Archit. Manag.* 27 (3), 785–807. <http://dx.doi.org/10.1108/ECAM-12-2018-0556>.
- Xu, J., Yuan, C., Gu, J., Liu, J., An, J., Kong, Q., 2023. Innovative synthetic data augmentation for dam crack detection, segmentation, and quantification. *Struct. Heal. Monit.* 22 (4), 2402–2426. <http://dx.doi.org/10.1177/14759217221122318>.
- Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., Yang, X., 2018. Automatic pixel-level crack detection and measurement using fully convolutional network. *Comput.-Aided Civ. Infrastruct. Eng.* 33 (12), 1090–1109. <http://dx.doi.org/10.1111/mice.12412>.
- Yang, G., Liu, K., Zhang, J., Zhao, B., Zhao, Z., Chen, X., Chen, B.M., 2022. Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for crack classification, segmentation, and detection. *Constr. Build. Mater.* 356, 129226. <http://dx.doi.org/10.1016/j.conbuildmat.2022.129226>.
- Yang, Q., Shi, W., Chen, J., Lin, W., 2020. Deep convolution neural network-based transfer learning method for civil infrastructure crack detection. *Autom. Constr.* 116, 103199. <http://dx.doi.org/10.1016/j.autcon.2020.103199>.
- Ye, X.-W., Jin, T., Chen, P.-Y., 2019. Structural crack detection using deep learning-based fully convolutional networks. *Adv. Struct. Eng.* 22 (16), 3412–3419. <http://dx.doi.org/10.1177/1369433219836292>.
- Ye, X.W., Jin, T., Li, Z.X., Ma, S.Y., Ding, Y., Ou, Y.H., 2021. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *J. Struct. Eng.* 147 (11), 04721008. [http://dx.doi.org/10.1061/\(ASCE\)ST.1943-541X.0003140](http://dx.doi.org/10.1061/(ASCE)ST.1943-541X.0003140).
- Yeum, C.M., Dyke, S.J., Ramirez, J., 2018. Visual data classification in post-event building reconnaissance. *Eng. Struct.* 155, 16–24. <http://dx.doi.org/10.1016/j.engstruct.2017.10.057>.
- Zhao, M., Xu, X., Bao, X., Chen, X., Yang, H., 2024. An automated instance segmentation method for crack detection integrated with CrackMover data augmentation. *Sensors* 24 (2), <http://dx.doi.org/10.3390/s24020446>.
- Zheng, M., Lei, Z., Zhang, K., 2020. Intelligent detection of building cracks based on deep learning. *Image Vis. Comput.* 103, 103987. <http://dx.doi.org/10.1016/j.imavis.2020.103987>.