Automated multi-category tunnel damage detection and report generation from ultra-high-resolution panoramic laser images

Tunnelling and Underground Space Technology 168 (2026) 107194

Contents lists available at ScienceDirect

Tunnelling and Underground Space Technology

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

Automated multi-category tunnel damage detection and report generation from ultra-high-resolution panoramic laser images

Zehao Ye [a], Mohammadhamed Mozafarian [b], Paola Alice Rosa Cavallaro [b], Kamil Altinay [c], Valentina Villa [b], Jelena Ninić [c,*]

[a] University of Birmingham, UK
[b] Politecnico di Torino, Italy
[c] Durham University, UK

ARTICLE INFO

Keywords:
Damage report
Tunnel lining
Defect detection
Evaluation method
Instance segmentation
Web-based platform

ABSTRACT

In the realm of ageing tunnel infrastructure, accurately assessing structural damage remains a pressing challenge due to the inherent subjectivity and time demands of manual inspections. Although reality capture technology allows for digital representation of as-is condition of assets, converting these rich data sources into actionable risk assessments demands still requires innovative solutions. In this paper, we introduce a comprehensive, web-based automated framework that uses ultra-high-resolution (UHR) panoramic tunnel images to automatically generate detailed damage records and risk assessment reports. A significant challenge in this domain is the observation that damage regions often lack sharply defined boundaries; instead, they exhibit gradual, blurred transitions, which is not well-suited to conventional segmentation evaluation. To address this, we formally define the challenge of inconsistency of damage annotation in complex real-world scenarios and propose a novel evaluation metric: Intersection over Union with buffer zone (IoUb). This metric relaxes the rigid boundary precision requirements of traditional evaluation methods, focusing more on capturing the overall damage. We evaluated several instance segmentation algorithms and recommend adopting a lower confidence threshold, as it reduces missed detections without significantly increasing false positives. We introduce post-processing methods that aggregate the predictions from multiple inferences to meet the demands of processing UHR panoramic images, resulting in a 3% improvement in Macro IoU and IoUb, along with a 90% damage recall. Experimental results on Italian road tunnels demonstrate that our framework enhances automated damage detection. We then categorize damage severity using a statistically grounded methodology, enable natural language queries of statistical damage results, and handle visualization and report export, all within a single end-to-end web-based platform. The proposed framework significantly enhances the efficiency of professionals in planning and monitoring ageing tunnel assets. Our code is available at https://github.com/zxy239/Auto-damage-report-generation.

1. Introduction

Tunnel infrastructure plays a vital role in ensuring connectivity for people and goods, and supporting economic activities worldwide, especially in regions with challenging topographies. Across Europe, the maintenance and safety of ageing tunnels within the Trans-European Transport Network (TEN-T) have become a pressing concern due to increasing traffic demands and evolving safety standards combined with accelerated infrastructure deterioration due to impacts of climate change (Schade et al., 2022). For example in Italy, the mountainous terrain makes tunnels indispensable for connecting urban centres with rural areas, particularly in regions such as the Alps and the Apennines. Approximately half of the total number and length of tunnel sections within the TEN-T are located in Italy, with about 50% of these having been in operation for over 30 years, and only 19% comply with the minimum safety requirements suggested by regulations (Ministro delle infrastrutture e dei trasporti, 2023). This underscores the urgent need for stakeholders to take action. Currently, Italy is conducting a large-scale survey and investigation to monitor the condition of tunnel infrastructure. Guidelines (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022) have been put forward, emphasizing a preventive approach aimed at reducing risks, prioritizing proactive maintenance over emergency interventions, and further highlighting the importance

* Corresponding author.
E-mail address: jelena.ninic@durham.ac.uk (J. Ninić).

https://doi.org/10.1016/j.tust.2025.107194
Received 20 May 2025; Received in revised form 18 September 2025; Accepted 17 October 2025
Available online 30 October 2025

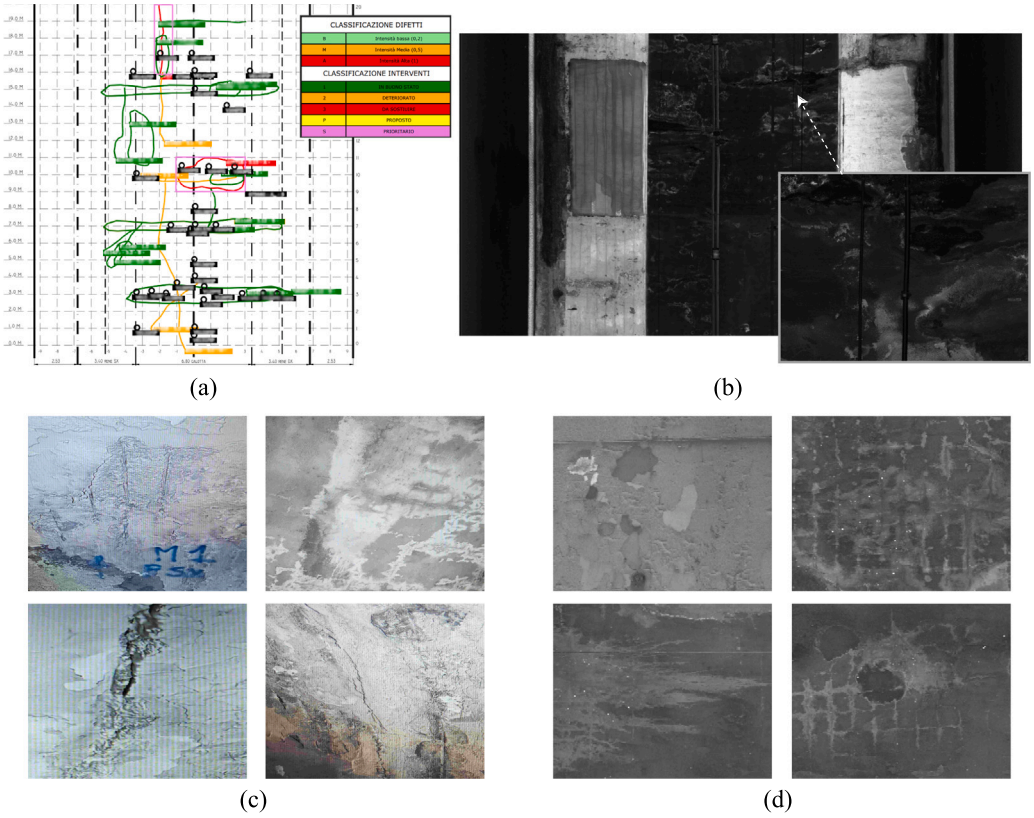

Fig. 1. Damage report and exemplar images of actual severe degradation damage: (a) an example of manually annotated damage report (20 m along the tunnel direction): different coloured curves represent various types of damage. The colours green, orange, and red represent the severity levels subjectively assessed by the engineer, with risk progressively increasing from green to red. Rectangular labels are both colour-coded and labelled to specify the type of damage. (b) A raw gray-scale local laser panoramic image of the tunnel section (20 m along the tunnel direction). (c) Examples of RGB images of tunnel damage (spalling, seepage, crack, and their combinations) from the database used in this paper. (d) Examples of locally enhanced gray-scale images after gamma correction exhibit improved visualization; damage boundaries are highly complex and blurred, making precise annotation difficult. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of continuous monitoring and advanced surveying technologies.

Advanced reality capture technologies are used to automatically collect panoramic tunnel data (e.g. point clouds, images), significantly improving the efficiency of data collection (Zhu et al., 2016; Ma and Liu, 2018; Foria et al., 2019; Ye et al., 2025). Among these, the image data are typically of Ultra-High Resolution (UHR), usually above 2–4 kilopixels (kpx) per section (i.e. image), which is much higher than the resolutions commonly used in general computer vision tasks, i.e. usually less than 1 kpx (Liu et al., 2024), and allows the tunnel to be captured and represented more comprehensively. However, damage labelling and recording are still primarily done manually. An example of the evaluated and labelled conditions is shown in Fig. 1(a). This is mainly generated based on field surveys, manual damage recording, and is prone to subjective grading, and scoring to identify high-risk areas for focused monitoring. The process is also time-consuming, labour-intensive, and prone to inconsistencies due to the subjectivity of evaluations and the experience of inspectors (Huang et al., 2021; Deng et al., 2022). Therefore, it is crucial to propose effective and highly automated methods for tunnel damage detection and risk assessment, as these evaluations are vital for maintaining the functionality and safety of these critical assets.

The use of computer vision algorithms for identifying tunnel surface damage is regarded as a key method for achieving automated detection, with many studies already exploring this approach (Attard et al., 2018; Jiang et al., 2023). We summarized in Table 1.

Among them, Deep Learning (DL) algorithms often demonstrate superior performance, particularly Transformer-based algorithms or models that incorporate attention mechanisms (Ye et al., 2024; Huang et al., 2024; Zhang et al., 2024). Various types of DL-based image recognition and segmentation algorithms have been applied, which can be categorized into object detection, semantic segmentation, and instance segmentation. Each type progressively increases the amount of information extracted from a single image, respectively ranging from describing damage locations with bounding boxes, to providing more precise boundary definitions, and ultimately to offering instance-level recognition and segmentation. Recent trends show a rising interest in instance segmentation, especially for non-crack defects, and an increasing emphasis on multi-defect detection.

Additionally, recent studies on deep learning for tunnel and underground infrastructure have moved beyond static defect detection to address predictive maintenance. Some studies have explored temporal modelling of defect evolution. For example, An and Kang (2024) used ConvLSTM to predict crack growth in concrete, linking image-based detection to long-term deterioration analysis. In tunnels, LSTM models have been applied to forecast surrounding rock deformation and vault settlement (He and Chen, 2023), showing deep learning's potential in capturing temporal patterns. Beyond sequence learning, statistical and machine learning approaches have been used to model deterioration, such as combining regression trees with optimization algorithms (Abdelkader et al., 2025) or comparing multiple models for tunnel degradation assessment (Ahmed et al., 2021), highlighting the

Table 1
Summary of tunnel defect recognition studies.

Author	Task	Defect type	Best model
Ren et al. (2020)	SS	Crack	CrackSegNet
Zhao et al. (2020)	IS	Moisture	Mask R-CNN
Li et al. (2021)	OD	Crack, Falling Block, Leakage	Improved Faster R-CNN
Foria et al. (2022)	SS	Seepage	U-Net
Liu et al. (2022)	OD	Crack	YOLOv5
Ouyang et al. (2023)	SS	Crack	DeepLab V3+
Xu et al. (2023)	SS	Leakage	Improved U-Net++
Geng et al. (2023)	IS	Leakage	Improved BleedMask
Zhou et al. (2023)	SS	Crack, Leakage, Peeling Fireproof Coating, Multiple Defects	MC-TLD
Feng et al. (2023)	SS	Crack, Leakage	Hybrid Model
Zhang et al. (2024)	IS	Spalling, Crack, Rebar Exposure, Water Seepage	YOLOv8-CM
Wang et al. (2024)	SS	Crack	Improved Cascade R-CNN
Feng et al. (2025)	SS	Crack	TCSegNet
Ouyang et al. (2025)	SS	Crack	DeepLab V3+
Lin et al. (2025)	IS	Leakage, Damage	Improved SOLOV2
Yang et al. (2025)	IS	Damp, Unfilled corner, Creep, Crack, Chip	YOLO-SH

Note: Object Detection (OD); Semantic Segmentation (SS); Instance Segmentation (IS).

value of predictive analytics in maintenance planning. MIRET-Tunnel AI employs convolutional neural networks for damage detection and integrates human supervision into the AI workflow to optimize maintenance scheduling (Foria et al., 2024; Glab et al., 2025). Overall, while there has been considerable research on tunnels, challenges remain in fully addressing their inherent complexities and translating predictive models into effective maintenance decision-making, due to issues such as limited long-term validation and scarce large-scale datasets.

To sum up, applying DL-based image algorithms to tunnel surface damage detection, along with explorations into predictive maintenance, has proven effective and promising. However, challenges remain when addressing real-world tunnel scenarios, particularly those with severe deterioration. Here, we summarize the three specific and pressing challenges identified in this study, which align well to the challenges faced by most researchers and engineers for practical applications: (i) the dilemma of data annotation: classification of defects, annotation efficiency and consistency in data annotation, (ii) the need for more focused training and evaluation methods, and (iii) the processing of ultra-high-resolution panoramic images and automation workflows. In the following we discuss each of these challenges in detail.

Firstly, annotation is especially difficult in real cases, making dataset creation challenging and potentially leading to inaccuracies when evaluating models on validation datasets. Many defect detection studies rely on well-labelled datasets with relatively simple damage classifications, such as cracks, thus overlooking the complexities of severely damaged real-life scenarios. In reality, structures with severe degradation typically contain multiple types of damage that are intertwined and overlapping. To achieve comprehensive detection, it is necessary to distinguish and label all types of damage, which complicates the annotation process. Additionally, unlike the relatively clear boundaries of objects in common computer vision datasets, damage in tunnels often lacks well-defined edges, shown in Fig. 1(b), (c) and (d). Some experiments (Kirillov et al., 2019) have shown that annotation inconsistencies can occur on common datasets, such as Cityscapes (Cordts et al., 2016), ADE20k (Zhou et al., 2017), and Vistas (Neuhold et al., 2017) datasets. This issue is even more pronounced when dealing with severely deteriorated tunnels. These inconsistencies occur even between different annotators or when the same annotator re-labels the images after some time. In particular, on large objects (with an area greater than 96^2 pixels), the similarity of the segmentation mask (Segmentation Quality from panoptic segmentation evaluation) reaches above 80%, while on small objects (less than 32^2 pixels), it is even below 70% (Kirillov et al., 2019).

Some research (Tsai and Chatterjee, 2017; He et al., 2024) also discussed this issue in damage detection tasks, for example, the context of crack recognition evaluation. Due to the narrow width of cracks (sometimes only a few pixels), inconsistencies between the annotation and prediction masks can, under extreme conditions, result in no overlap, meaning an Intersection Over Union (IoU) of zero. However, the crack may have actually been successfully detected near the annotation with the correct shape. To address this, Tsai and Chatterjee (2017) designed an enhanced Hausdorff distance-based evaluation method, which uses a piecewise function to evaluate performance, with the non-penalized and low-penalty buffer around the crack serving as the basis for the design. This method introduces some computational overhead and requires hyper-parameters, such as the actual image resolution, and is not fully compatible with current mainstream IoU-based evaluation methods. However, it offers valuable insights for addressing such issues.

The second challenge is adapting DL-based algorithms to better align with real-world engineering requirements, ensuring their practicality, reliability, and effectiveness in complex applications. Specifically, in the context of damage detection, engineers prioritize minimizing missed detections, even if it results in a small number of false positives. This approach is common in critical tasks such as medical diagnosis (Yankaskas et al., 2001), fraud detection (Chung and Lee, 2023), and spam filtering (Guzella and Caminhas, 2009), where the consequences of missed detections are severe. As a result, metrics like recall and related evaluation measures are given greater emphasis. Meanwhile, regarding the accuracy of damage boundaries, annotating these boundaries is, as mentioned earlier, a challenging and time-consuming task. Moreover, engineers are generally more concerned with whether the predicted masks cover the damage areas, rather than achieving exact matches of the edges. Therefore, a more suitable evaluation method should be introduced to meet these engineering requirements.

However, when we look at the related research in the field of computer vision, in contrast, more studies focus on how to conduct more detailed evaluations of mask quality. As a result, more refined evaluation methods have been proposed, including boundary-based evaluations such as Trimap IoU (Chen et al., 2017), Boundary IoU (Cheng et al., 2021), and F-measure (Perazzi et al., 2016). Trimap IoU calculates the IoU around the GT boundary only, considering the region near the boundary, and such region is obtained through morphological operations of erosion and dilation. Boundary IoU calculates IoU by applying dilation or erosion operations to both the predicted and GT boundaries, and then computing the IoU within this modified region. F-measure matches the predicted and GT contours if they are within a specified pixel distance threshold d. We summarize their characteristics and suitable use cases in Table 2, along with the general IoU.

Clearly, these boundary-based evaluation methods are very different from our application scenario and requirements, but they still provide many valuable insights for us. First, it is best to base the evaluation on IoU, as this would make the evaluation method compatible with most current segmentation tasks. Second, we can design more suitable evaluation method, IoU with buffer zones, by focusing on boundaries and

Table 2
Comparison of evaluation methods for mask segmentation.

Method	Characteristics	Use case
Trimap IoU	Focuses on boundary quality, ignores outer errors. The calculation is asymmetric; less sensitive to dilated predictions.	Evaluating the quality of the region around the boundary, especially in tasks with a focus on boundary accuracy.
Boundary IoU	Accurate for boundary overlap, considers geometric relationship. Sensitive to small objects or misalignments, high variance.	Boundary overlap evaluation, suitable for tasks requiring precise boundary shape assessment.
F-Measure	Tolerates small misalignments, robust to ambiguity. Sensitive to small errors, discontinuous for slight changes.	Boundary detection tasks, particularly in scenarios with tolerance for boundary misalignments.
General IoU	Evaluates the entire mask overlap. Insensitive to boundary misalignment, less accurate for fine details.	General segmentation tasks, overall mask evaluation.

their neighbourhoods. Additionally, in our task, we should prioritize recall. By integrating the impact of inconsistencies in data annotation from first challenge as well, the precision of the mask boundaries is not as critical. Accordingly, more robust and reliable evaluation methods should be used to assess the performance of DL-based image segmentation algorithms in our task.

The final challenge is that, while successful detection and effective algorithm evaluation are crucial, developing an end-to-end system is equally valuable. Panoramic tunnel images are typically the final output produced by the data acquisition devices used in tunnel monitoring, and they have a very high resolution (Zhu et al., 2016; Attard et al., 2018; Li et al., 2021; Foria et al., 2022). Foria et al. (2022) downscaled tunnel panoramas up to 20k to 12.5% of their original size and further cropped them into 320 × 320 tiles to train a U-Net for seepage detection. While this improves efficiency, it sacrifices a significant amount of detail. Therefore, to preserve data fidelity, the best approach is to perform training and prediction tasks with minimal downsampling whenever possible. A new system should fully automate the process of taking panoramic tunnel images as input, maintaining high fidelity throughout the processing pipeline, and generating comprehensive damage reports as output by following the related guideline, to effectively guide maintenance operations. Specially, it includes pre-processing and cropping of the UHR panoramic tunnel images, followed by post-processing to merge defect instances across images after inferences, and present results for each individual tunnel lining segment. Ideally, a long-term tracking system and a historical database should be established, enabling continuous monitoring of individual damage instances over time. Such data would serve as a critical source of information for deterioration modelling and predictive maintenance (Ninic et al., 2025). Eventually, the system provides all relevant outcomes necessary for the damage report, along with final risk assessment values, to guide subsequent focused monitoring efforts.

One of the aims of this paper is to develop a data processing platform for damage detection, quantification, and risk assessment that supports inspections. Therefore, we summarize some of the research gaps in current tunnel inspection platforms. Taking MIRET (Foria et al., 2022) as an example, it relies on semantic segmentation models, though instance segmentation is preferable, as it can both distinguish individual overlapping damages and provide instance-level damage identification. They also lack the capability for high-fidelity processing of UHR images without significant downsampling. Additionally, these platforms do not provide auxiliary annotation tools, despite the recent widespread adoption of methods such as SAM in various fields. Finally, standard IoU-based evaluation metrics excessively penalize ambiguous defect boundaries; hence, inspired by other fields, a new problem-specific metric is needed.

In this paper, we address the aforementioned challenges by proposing a highly integrated, web-based framework that takes UHR panoramic laser images of tunnels as input and executes a sequential workflow in automated way to perform instance-level, multi-category damage detection and generate statistical damage reports. To achieve practical defect detection in these complex real-world scenes, this paper presents the following contributions:

- We establish a dataset encompassing five categories of damage (seepage, corrosion, damaged joint, spalling and crack), covering all types of tunnel defects that are visually identifiable and distinguishable in UHR images. We employed Segment Anything Model (SAM)-based prompt annotation for instance labelling, enabling overlapping annotations, which greatly streamlined and accelerated the annotation process.
- We introduce a more reliable and practical metric, IoU with buffer zones. It is designed to emphasize holistic recognition of damage rather than precise boundary delineation. Several mainstream instance segmentation algorithms are comprehensively assessed by proposed metric on real-world case studies.
- We propose a multi-inference aggregation method inspired by ensemble learning. Single model performs multiple full-image inferences with different crop sizes, capturing various levels of contextual information. By overlapping and combining these results, we significantly improve segmentation accuracy on UHR images.
- We developed an end-to-end web-based platform that centralizes UHR image processing, integrating pre-processing, multi-inference aggregation, post-processing, and statistical analysis, with interactive visualization and exportable outputs in .pdf and .json formats for unified data management.

The remainder of this paper is structured as follows. Section 2 presents the methodology, detailing how damage detection tasks are performed on UHR tunnel panoramic images with a new evaluation method, and the processes for automated damage report and risk area generation on web-based platform. Section 3 describes the implementation details of the entire end-to-end system, including the establishment of the dataset, damage detection model training process and parameters, as well as the pre-processing, post-processing, and other settings for the model. Section 4 presents the model results along with a series of discussions based on ablation studies. A demonstration of the web-based damage report will also be provided. Section 5 concludes the paper, discussing its limitations and suggesting directions for future work.

2. Methodology

2.1. Overview

The overview of the methodology can be seen in Fig. 2. The first section focuses on data creation (see Section 2.2), including the process of collecting tunnel panoramic image and the use of SAM-based tools for annotating various types of tunnel damage. The second section discusses the introduction of IoUb (see Section 2.4), which aims to

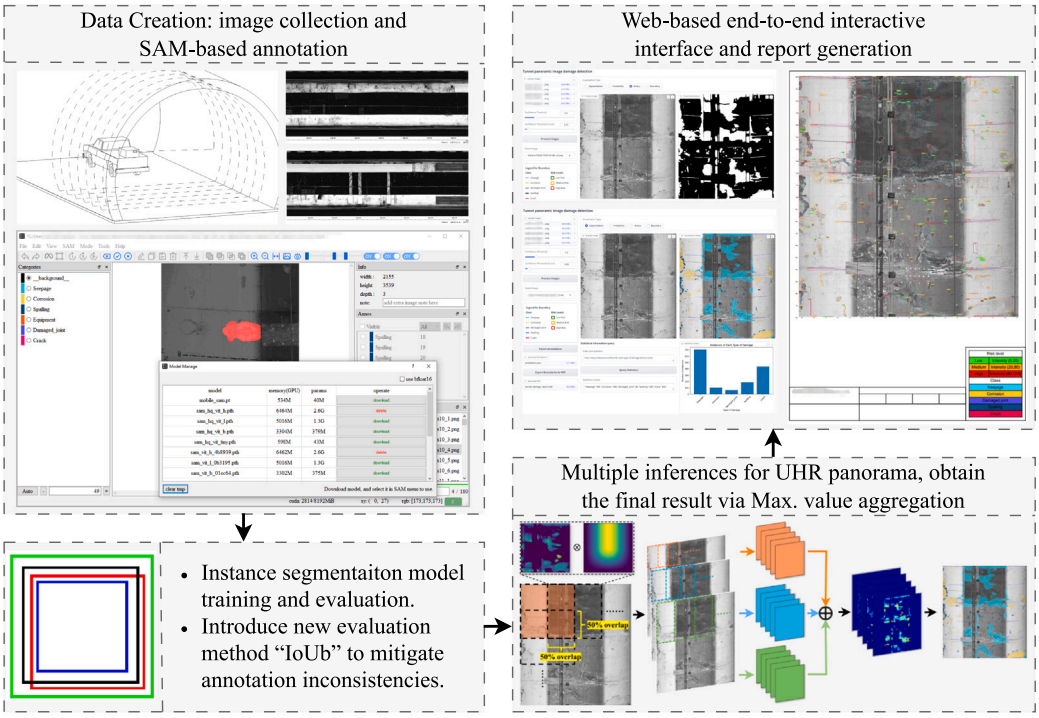

Fig. 2. Overview of proposed framework.

mitigate the impact of annotation inconsistencies, particularly in highly deteriorated tunnel environments. The third section explores the adaptation of common instance segmentation algorithms to UHR images and how multi-inference aggregation improves the final segmentation quality for UHR images (see Section 2.3). The final section presents a web-based interactive platform (see Section 2.5) that integrates the entire end-to-end algorithm pipeline, taking UHR images as input, visualizing the final segmentation results, and generating user-friendly damage assessment reports.

2.2. Dataset creation

The tunnel panoramic images in this study were collected using Mobile Laser Scanner, as laser point clouds are commonly used for data collection in the tunnel's limited lighting conditions. The scanner developed by Spacetec, referred to as TS4 (spacetec, 2019). This instrument generates a pair of sinusoidal laser pulses and measures distance by calculating the phase difference between the emitted and received waves. Due to potential limitation of the phase shift, the maximum effective scanning distance is limited to 15 m, which is sufficient for surveys in tunnels, even those with three lanes. With a field of view of 360-degree, shown in Fig. 3, the laser scanner can capture the entire tunnel vault and roadway. Data acquisition occurs while the vehicle moves at approximately 5 km/h. The scanner's head rotates at 200 revolutions per second, collecting 5,000 points per revolution, enabling the seamless collection of detailed point clouds representing the tunnel's surface, along with reflectance data for material characterization. The primary output of this technology is therefore a point cloud, and in addition, the device can directly produce files in other formats, including tunnel panoramic unwrapped images used for this study. These images are generated from a rigorous three-dimensional representation, ensuring precision and reliability for analysis.

The data evaluated in this study are 8-bit gray-scale images in .tif format using laser intensity as the colour value. Each image represents an entire tunnel with a fixed width of 10,000 pixels, while the height

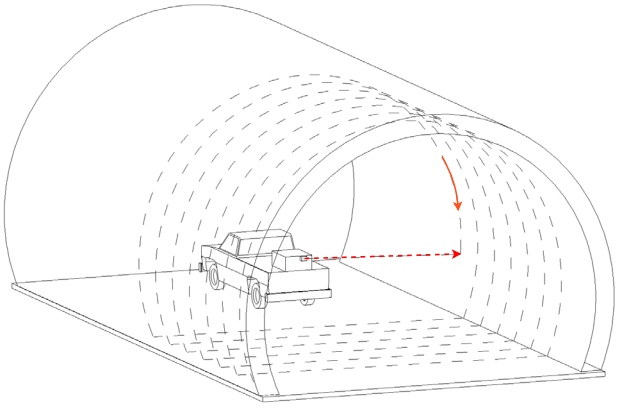

Fig. 3. An illustration of a collection vehicle equipped with a scanner performing 360-degree rapid scanning while moving through a tunnel.

varies depending on the actual length of the whole tunnel. More detailed information can be found in Mozafarian et al. (2025). The real-world size of each pixel can be calculated as L/p, where L is the total scanned tunnel length and p is from .tif files, recording the number of pixels along the tunnel. For example, if $L = 200$ m and $p = 100{,}000$, the real-world width/height of each pixel represents 2 mm. Therefore, based on the tunnel parameters and the .tif file, the real-world size of each pixel can be calculated. Pixel size may vary slightly between tunnels, as all .tif files have a fixed width of 10,000 pixels. According to the guidelines (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022), the tunnel is divided into sections with a 20-meter longitudinal spacing (along the direction of tunnel advancement), and the road sections on both sides of the panoramic image are cropped, leaving only the tunnel lining. The crown of road tunnels and the side walls exhibit a significant colour difference: the

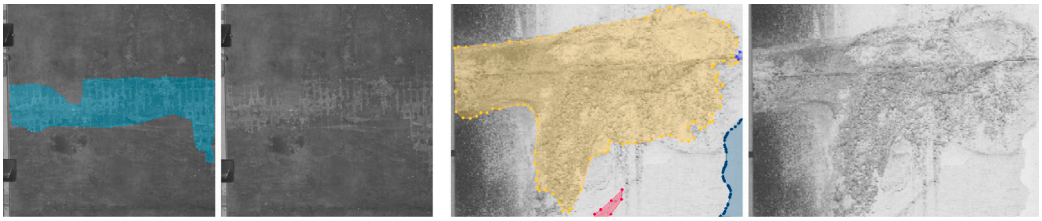

Fig. 4. Challenging damage annotations: (a) Overly complex damage boundaries, (b) Blurred transitions of damage boundaries. The annotation follows the principle of covering as much of the damage as possible. Blue represents seepage, yellow indicates corrosion.

top is dark, while the side walls are relatively lighter. To facilitate better annotation and training, we applied gamma correction (Bradski, 2000), as shown in Fig. 5(a), which significantly enhanced the visibility of the images, by making details in the dark areas more visible while preventing overexposure in the light areas.

It should be noted that, due to the inherent ambiguity and uncertainty in the boundaries of tunnel damage areas (see Fig. 4), defining and annotating these boundaries has always been one of the major challenges in this study. To address this issue and further enhance annotation efficiency, we introduced a SAM-based prompt annotation tool for labelling the damages (Kirillov et al., 2023; Ji and Zhang, 2023). This tool enables the generation of reasonably accurate masks with just a few manually provided prompt points. The annotation process was carried out using the original SAM with the ViT-H backbone, which has the largest parameter size (632 MB). Throughout the annotation process, we create masks automatically and then refrained them for intricate boundary details. This was an effective approach for two key considerations: first, the generated masks from prompt already provided sufficient coverage of the damaged areas, which partially alleviates inconsistencies among different annotators; second, defining highly complex boundaries is inherently challenging and is rather the opposite of actual inspection practices. However, this approach proved less effective for damage types with particularly irregular or elongated shapes, such as damage joints and cracks. For these categories, manual annotation remained essential. Overall, while SAM-based annotation cannot fundamentally resolve inconsistencies, it substantially mitigates this issue and greatly improves annotation efficiency. Importantly, our annotations are designed to capture the full extent of damage, even when boundaries are inherently ambiguous, as illustrated in Fig. 4.

The default input size for SAM is 1024 × 1024 pixels, so we divided each 20-meter tunnel section into multiple sub-images and controlled the side length of the cropped sub-images to ~2–3 k pixels, enabling SAM to generate masks with acceptable detail while also improving annotation efficiency. Two experts have been involved in the annotation process: the first was responsible for performing the initial labelling using the aforementioned SAM-based prompt annotation tool, mainly based on existing manually drawn damage reports, while the second expert was tasked with reviewing the annotations, and manually refining annotations that challenging for the SAM-based tool to generate accurately. After all annotations were completed, the sub-images were further cropped to a size with a side length of ~1 kpx. In the end, we obtained 1800 images, which included damage annotations for two full tunnels. Specifically, a 20 m section of the first tunnel (6465 × 7078) was divided into six non-overlapping sub-images (2155 × 3539 each) for annotation, and these were further split into approximately 1 kpx resolution images (1077 × 1179 each) for model training. Similarly, a 20 m section of the second tunnel (6432 × 7078) was divided into six sub-images (2144 × 3821 each) for annotation, which were then further split into 1 kpx resolution images (1072 × 1273 each) for training. We selected 5 tunnel sections (three from the first tunnel and two from the second one) for the validation set, which including 180 images. The distribution of the validation set is shown in Fig. 5(c). The remaining images, excluding those without annotations (no damages), amounted to 1430 and were used for training.

Computer vision methods can only identify damage types based on surface observation images. For example, damages like voids are difficult to observe from the surface and therefore were not included. Ultimately, we labelled five major types of tunnel damage that can be directly distinguished through images, including seepage, corrosion, damaged joint, spalling, and crack. Their typical examples are shown in Fig. 5(b). As mentioned earlier, we use an instance segmentation algorithm for damage detection, as instance-level segmentation is the most suitable for generating damage reports. According to the guidelines (Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022), the record of damage is at the instance level, and when two damages are close to each other and connected, they should be recognized as a single damage and their area calculated as a whole. Therefore, our annotations also follow this principle. Additionally, we simplified the instance definition by treating all connected areas of the same type of damage as a single instance, making annotation with SAM more efficient. SAM struggles with efficiency when instance regions are non-contiguous, and some damage areas in the panoramic image may be interrupted by auxiliary structures (tubes, pipes). While it is possible to annotate these separately and assign a consistent instance ID, the blurred edges of damage, the irregular shapes of defect instances, and the occlusion caused by auxiliary structures make this particularly challenging. Therefore, as a practical solution, we currently define each closed mask as a single instance, which helps reduce the complexity of annotation and training.

2.3. Instance segmentation on UHR tunnel panoramic images

We conduct instance segmentation algorithms task due to its unique advantages. Unlike object detection, instance segmentation provides damage-covering masks essential for area-based damage assessment. Compared to semantic segmentation, it offers instance-level information, better aligning with the need for damage evaluation based on individual instances. It also handles overlapping instances, reflecting real-world tunnel conditions where multiple types of damage often overlap. Additionally, unlike panoptic segmentation, it reduces unnecessary focus on background classes, prioritizing damage analysis.

The final report is generated section by section, with each tunnel section covering 20 m. Accordingly, we use the cropped panoramic images of tunnel sections at 20-meter intervals as input, removing the road portions from the images. In our dataset, these localized panoramic images have a resolution of ~6 k, qualifying as UHR images. Although some specialized algorithms (Cheng et al., 2020; Guo et al., 2022; Li et al., 2024) for UHR images exist, we opted for general instance segmentation algorithms combined with fixed pre-processing and post-processing steps to address this challenge through cropping and stitching. This choice ensures greater flexibility in integrating or replacing the latest algorithms.

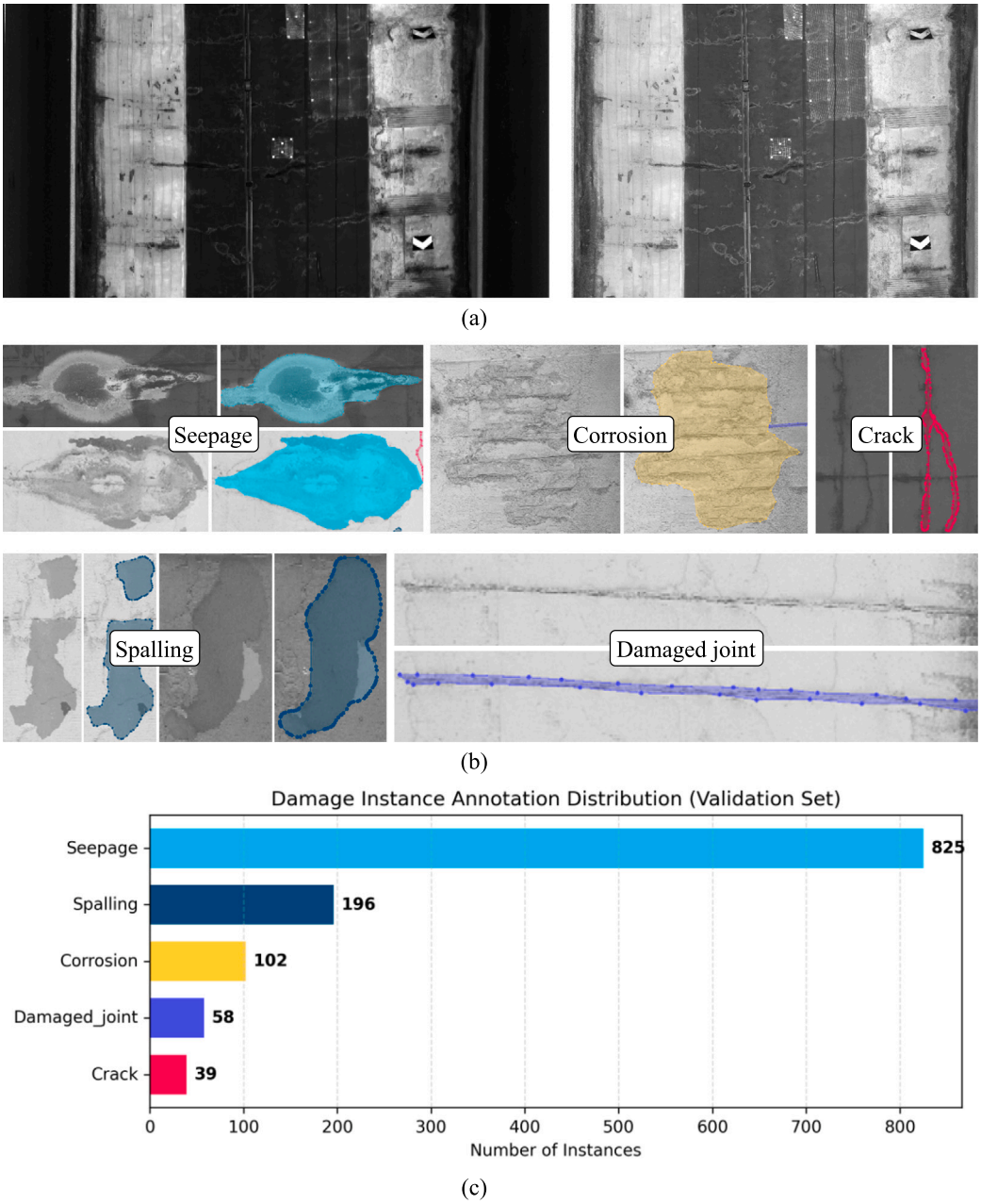

Fig. 5. Illustration of panoramic tunnel unfolded image and damage types (particularly clear case): (a) Original image (left) and gamma-corrected image (right); (b) Representative examples of five damage types. (c) The annotation distribution of damage instance for validation set.

In this paper, we focused on several mainstream and representative instance segmentation algorithms, including Mask R-CNN (He et al., 2018), Cascade Mask R-CNN (Cai and Vasconcelos, 2019), and Mask2Former (Cheng et al., 2022), to carry out our tasks, for testing our new model evaluation method (IoUb). Additionally, all models used the Swin-L backbone (Liu et al., 2021), and the actual input should be the image with a resolution of ~1 kpx.

During inference, we follow the process outlined in Fig. 6 to perform UHR tunnel panoramic image segmentation. Specifically, it includes pre-processing and post-processing based on conventional instance segmentation. As for pre-processing, in order to minimize the loss of contextual information during inference on individual images, we applied a cropping strategy with a 50% overlap along the both horizontal and vertical directions of the UHR image. All cropped images were then input into the instance segmentation algorithms for inference, producing logits (the raw prediction values of the model) for each image.

For post-processing, there are two main tasks: (1) Mask adjustment for overlapping regions: the predicted logits in overlapping areas are recalculated using a weighted approach, with the weights decreasing from the centre of the cropped image toward the edges (Chatterjee and Poullis, 2021). (2) Merging instances across images: masks in overlapping regions with some overlap and belonging to the same category are merged into a single instance. Finally, it generates the segmentation map of the UHR tunnel section.

Notably, for Region Proposal Network (RPN)-based networks like Mask R-CNN (He et al., 2018) and Cascade Mask R-CNN (Cai and Vasconcelos, 2019), the logits are multi-channel matrices. Each channel

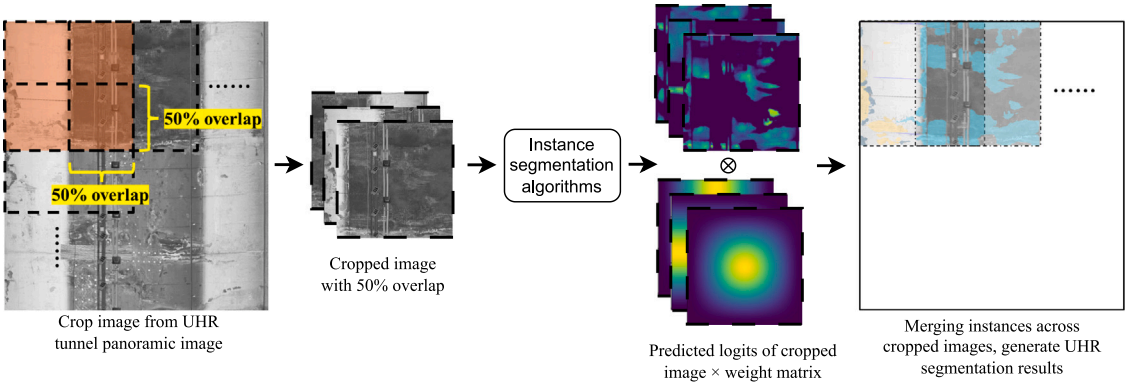

Fig. 6. Single inference process for UHR tunnel panoramic image.

corresponds to the confidence score of a specific class for each Region of Interest (RoI). The class logits ($\mathbf{L}_c$) form an $\mathbf{L}_c \in \mathbb{R}^{N\times(C+1)}$ matrix, where N is the number of instances, and $C+1$ includes C object classes and one background class. The mask logits ($\mathbf{L}_m$), on the other hand, form an $\mathbf{L}_m \in \mathbb{R}^{N\times C\times H\times W}$ matrix, where H and W are the height and width of the predicted mask for each instance and each class. The post-processing merging of such algorithms has been widely practised (Chatterjee and Poullis, 2021).

In query-based methods like Mask2Former (Cheng et al., 2022), there is a fundamental difference: each query corresponds to the complete mask logits and class logits for several potential instance. This approach differs from traditional RPN-based methods. Due to the limited research in this area, we provide a detailed explanation here: the mask logits are represented as $\mathbf{L}_m \in \mathbb{R}^{Q\times H\times W}$, where Q denotes the number of queries, and each query generates a mask for a specific instance. The class logits are given by $\mathbf{L}_c \in \mathbb{R}^{Q\times(C+1)}$, where each query corresponds to a class prediction for the respective instance. During training, Mask2Former does not apply activation functions when converting mask logits into binary images. Instead, it directly thresholds the discriminative values at 0, meaning that all values less than 0 are considered background.

By combining the class logits and mask logits, the model autonomously computes $Q \times C$ scores, as formulated in Eq. (1), and selects the top K results. Notably, a single query can be associated with multiple masks, eliminating the need for traditional Non-Maximum Suppression (NMS).

$$Score = S_{class} \times \frac{\sum \left(\sigma(Mask_{logits}) \times Mask_{binary}\right)}{\sum Mask_{binary} + 1e-6} \tag{1}$$

where, σ is the sigmoid activation function applied to the mask logits. The final detection score combines both the classification confidence and the quality of the predicted mask.

Then, based on the confidence threshold, the corresponding logits are extracted. For each cropped image, we construct an empty matrix $\mathbf{M} \in \mathbb{R}^{C\times H\times W}$ and sequentially populate it with mask logits for each class. For logits belonging to the same class, we apply a maximum value aggregation at the same pixel position, ensuring that only the maximum logits value is retained for each pixel:

$$\mathbf{M}_{c,h,w} = \max_i \mathbf{L}^{(i)}_{c,h,w}, \quad \forall c \in \{1,\ldots,C\}, \quad \forall (h,w) \in H \times W \tag{2}$$

where $\mathbf{L}^{(i)}$ represents the mask logits of the ith cropped image.

Once these matrices are obtained for all cropped images, we begin merging the logits class by class to generate the final C UHR mask logits for the local panoramic tunnel section. During this merging process, as mentioned before, a weighted matrix $\mathbf{W}$ is applied to adjust the logits from adjacent cropped images in the overlapping regions:

$$\mathbf{L}_{\text{merged},c,h,w} = \sum_i \mathbf{W}^{(i)}_{h,w} \cdot \mathbf{M}^{(i)}_{c,h,w} \tag{3}$$

An important observation was that using a fixed cropping size may result in over-segmentation or under-segmentation for certain instances located within the overlapping regions during the merging process. Therefore, inspired by the idea of ensemble learning (Géron, 2022) and Test Time Augmentation (TTA) (Sun et al., 2020), we perform three separate inferences for each tunnel panoramic image, using different cropping sizes, as shown in Fig. 7. Ensemble learning typically improves model robustness by aggregating predictions from multiple models or different inference settings. Similarly, in our approach, multiple inferences with varying cropping sizes serve as diverse "weak learners", helping to mitigate the over-segmentation and under-segmentation issues observed with a fixed cropping size. The final logits are obtained by merging the outputs from these different inferences using a maximum value aggregation across logits again at each pixel location, ensuring that the most confident prediction is retained. We also believe that this approach can more effectively capture all potential damage with higher recall, ultimately generating the most reliable mask logits.

Although it increases inference time, this approach allows the contextual information of each cropped image to vary during each inference, helping to more comprehensively avoid missing damage instances and ultimately improving segmentation quality. After completing the three inferences, we obtain three sets of different predicted logits. A detailed discussion can be found in Section 4.4. Next, we extract the instance masks by selecting the regions with values greater than 0.

Overall, this simple yet effective approach ensures accurate segmentation, even with the challenges posed by UHR images. This post-processing method makes it adaptable to most segmentation algorithms, addressing issues like overlapping regions and class-specific adjustments. By applying a weighted matrix to adjust logits in overlapping areas and integrating the results from multiple inferences with different cropping sizes, our method combines these outputs into final logits, helping to mitigate the loss of contextual information caused by image cropping and ensuring more accurate segmentation of the panoramic image. Its flexibility make it a valuable tool for handling large-scale, high-resolution data in many real-world applications.

2.4. IoU with buffer

As we discussed in Section 2.2, in the annotation of tunnel damage, boundaries are often not clearly defined but rather exhibit fuzzy or gradual transitions. These non-unique transition areas make it difficult to maintain consistency in manual annotations, leading to subtle yet significant discrepancies between annotators that can impact evaluation accuracy. Traditional IoU evaluation methods rely heavily on precise boundary matching, which can result in harsh penalties for model predictions that are otherwise reasonable when dealing with ambiguous boundaries. To address this issue, we propose an IoU with

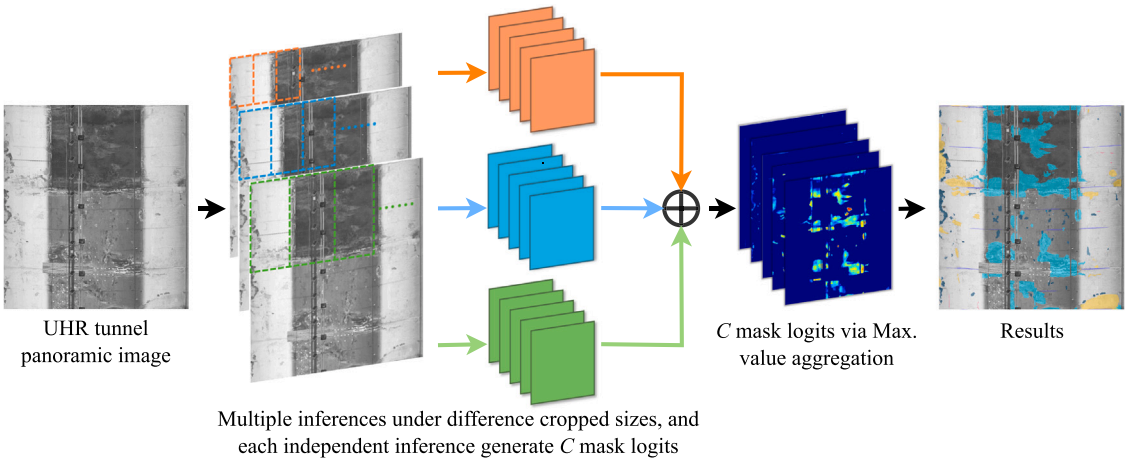

Fig. 7. Multiple inference process for UHR tunnel panoramic image and each inference can generate C mask logits $\mathbf{M} \in \mathbb{R}^{C\times H\times W}$. The final logits are generated through maximum value aggregation, producing the final segmentation mask.

buffer (IoUb) evaluation method, to optimize complexity for the given task. By introducing a buffer zone, we aim to strike a balance between strict boundary matching and the inherent ambiguity in damage annotations, ensuring a more robust and interpretable assessment of segmentation performance, mitigating evaluation bias caused by boundary uncertainty. This method better aligns with the real-world characteristics of damage distribution, ensuring a more stable and reliable evaluation. Meanwhile, it is also highly suitable for all current image segmentation tasks, as it aligns with the IoU-based evaluation used in semantic, instance, and panoptic segmentation. Inspired by enhanced Hausdorff distance–based quantitative performance evaluation method (Tsai and Chatterjee, 2017), our evaluation method is opposite to boundary-based IoU (Chen et al., 2017; Cheng et al., 2021). Specifically, Boundary IoU quantifies segmentation quality by computing the IoU only between predicted and GT boundaries within a defined range, emphasizing boundary accuracy rather than overall mask region overlap. In contrast, IoUb incorporates a buffer around the GT boundary, treating all pixels within this zone as True Positive (TP), thereby relaxing boundary precision requirements and prioritizing overall mask region consistency.

Similarly, IoUb constructs a boundary buffer zone around the GT boundary by applying dilation (δ) and erosion (ϵ) operations. A predicted pixel is considered correct if it falls within this buffer zone. Inconsistencies outside the buffer are evaluated using standard IoU. Let B_{GT}, M_{GT} and M_{pred} denote the GT boundaries, masks and predicted masks, and the boundary buffer zone is $\mathcal{B}$:

$$\mathrm{IoUb} = \begin{cases} M_{\mathrm{pred}} \subseteq \mathcal{B}, \quad \mathcal{B} = \delta(B_{\mathrm{GT}}) \cup \epsilon(B_{\mathrm{GT}}) & \text{predicted pixel inside buffer,} \\ \dfrac{\left|M_{\mathrm{GT}} \cap M_{\mathrm{pred}}\right|}{\left|M_{\mathrm{GT}} \cup M_{\mathrm{pred}}\right|} & \text{otherwise (standard IoU).} \end{cases} \tag{4}$$

Boundary IoU compares the boundaries of the predicted and GT masks by simultaneously dilating (or eroding) both boundaries to maintain symmetry, while our goal is to create a buffer zone to account for annotation errors, and constructing a buffer along the GT boundary is sufficient. Meanwhile, our comparison is still based on the entire mask, and unlike Trimap IoU, IoUb is equally sensitive to both overly large and overly small predictions.

$$IoUb = \frac{TP}{TP + FP + FN} \tag{5}$$

As shown in Fig. 8, a illustration is provided to demonstrate pixel-level evaluation. The evaluation formula follows Eq. (5). The black border represents the GT, the red border represents the prediction, and the green and blue borders correspond to the boundaries after dilation and erosion based on the GT, which are used to construct the buffer zone. Furthermore, TP refers to the predicted area that falls within the dilated or eroded GT area. False Positive (FP) refers to the part of the predicted area that lies outside the dilated GT area, and False Negative (FN) refers to the part of the GT area that is not covered by the predicted area, but excludes the part outside the eroded GT area. At this point, as shown in Fig. 8, the predictions lie entirely within the buffer zone. Consequently, according to Eq. (5), the IoUb achieves a value of 1, whereas the original IoU is approximately 0.6857. Similar to any boundary-based IoU (Cheng et al., 2021; Chen et al., 2017), the parameters for dilation and erosion could be implemented by extending a fixed number of pixels or in proportion to the image resolution or the size of the GT region. In this paper, we adopt the Boundary IoU setup (Cheng et al., 2021) and set the dilation and erosion distances for the buffer zone process to a fixed value of 15 pixels, which corresponds to $\sim$ 5cm in the real-world tunnel distance from our dataset. For cracks, which have a narrow and elongated instances, we adjust the buffer zone value to 5 pixels, equivalent to around 1.5 cm.

2.5. Web-based report generation

To facilitate the visualization and reporting of tunnel damage analysis, a web-based platform (see Fig. 9) was developed as part of this research. The platform is built based on Gradio (Abid et al., 2019). Our platform streamlines and integrates the process of batch image processing, performing model predictions, and generating comprehensive damage reports. It also provides a simple natural language interaction feature for viewing relevant statistics of the final detection results. Users can batch upload UHR panoramic images of tunnel sections or retrieve them directly from the internal server and set the detection parameters (Zone A in Fig. 9), and then, the platform automatically processes each image follow the step of Section 2.3. Therefore, the trained model will be applied in the platform to predict and analyse potential damage areas. The final output is the damage prediction results for tunnel sections based on a 20-meter length.

According to the encoding pattern of the file name, users can interactively explore the results by selecting any tunnel sections (Zone B in Fig. 9) for detailed visualization. The platform provides synchronized displays of the original image alongside various analysis outputs,

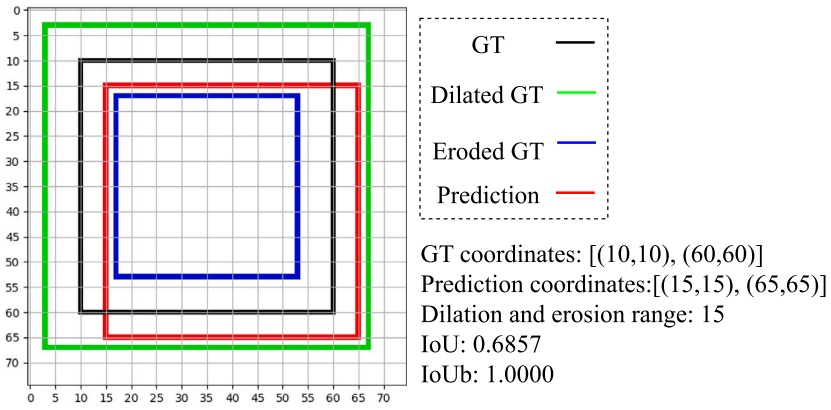

Fig. 8. Illustration of IoUb. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

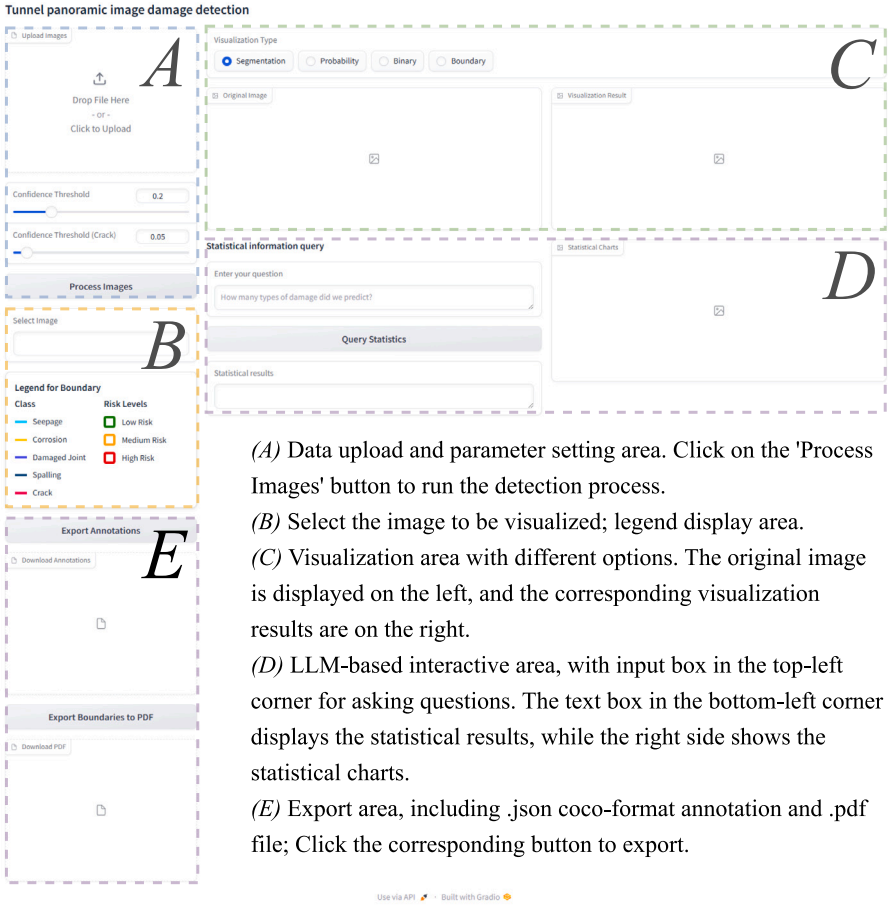

Fig. 9. A web-based interactive interface diagram for displaying tunnel panoramic image damage detection results.

including binary masks of damage, the damage probability map, and damage contour map (Zone C in Fig. 9). The damage probability map provides a standardized probability map for all damages corresponding to the UHR tunnel panorama in one go. It can be considered as a probability map distinguishing normal surfaces from damaged surfaces. This map is generated by aggregating the final logits from different classifications, which are derived from the merged UHR image, as described in Section 2.3.

The damage boundary map is the primary output and will be generated using a method similar to that shown in Fig. 1, with additional clearer visual improvements. One localized example can be seen in Fig. 10, where each contour (representing the predicted mask edge) will enclose a damage instance. These contour lines are colour-coded according to the damage type (see legend from Zone B in Fig. 9 or Fig. 10). Next, we draw bounding boxes for each damage instance using three colours: green, orange, and red. The damage intensity is determined through a statistical ranking method to assign the appropriate colour. For each type of damage, we rank individual instances based on their respective area measurements, using statistical distributions derived from the training set. The area of each damage instance is calculated by multiplying the actual area per pixel in the specific tunnel with the pixel area of the instance mask. The ranking follows a descending order, where larger damage areas receive higher ranks. According to the guidelines (Ministero delle Infrastrutture e della

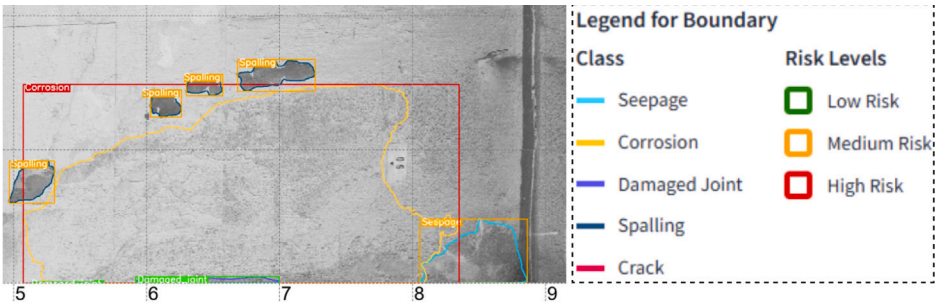

Fig. 10. A localized example of the damage boundary map and legend for boundary. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Mobilità Sostenibili, 2022), the top 20% of damages with the largest areas are classified as high-intensity (red), the bottom 20% as low-intensity (green), and the remaining as medium-intensity (orange). As a result, for each type of damage, we obtain two area thresholds, which determine the bounding box colour. Additionally, we label the damage type in the top-left corner of each bounding box.

This drawing information will also be saved back to the database for further dataset expansion. As the dataset expands, these area thresholds will become increasingly accurate. Ultimately, once a tunnel section has at least one type of damage classified as high risk, the segment will be marked as a key monitoring target. Overall, these digitization and visualization serve as key components of the tunnel damage assessment, offering multiple perspectives on the detected damage. In addition to interactive visualization, the platform supports the generation of exportable reports in .pdf format, which include all the analysis information mentioned above (Zone E in Fig. 9). Users can compile the visualized results into a Portable Document Format (PDF) report with additional coordinates, which consolidates the analysis outputs into a structured and professional format. Similarly, digital damage annotations can also be directly exported. Here, we record them in the COCO-format annotations, saving and exporting the predicted damage annotations as a .json file.

On the web platform interface, a simple natural language query function is provided for inquiring about detected damage, returning statistical results and visualized charts (Zone D in Fig. 9). The process begins with obtained COCO-format annotations, where the segmentation field (used to describe mask polygons) is removed to reduce unnecessary information (token) for subsequent statistical analysis. The processed .json file is then used as input for the natural language model. System prompts are designed to include at least the following points for supporting statistical queries:

- Generate results solely based on the provided processed .json file and refuse to answer unrelated questions.
- Convert user queries into Python code, which is then executed for statistical analysis. Store and return two variables: the statistical results and statistical charts.
- Optionally, describe the format of the input file.
- Optionally, provide examples of statistical charts.

When the user queries statistical questions, such as “how many instances were detected for each type of damage”, the large language model can automatically generates Python code to process the processed .json file, executing it in an isolated sandbox environment and returning statistical results and charts on the web platform. Converting user queries into Python code effectively prevents hallucinations from the language model and ensures statistical accuracy. Since COCO-format annotations are widely used, no additional explanation is required, whereas user-defined formats necessitate clarification. Meanwhile, providing sample statistical charts further ensures consistency in the final output.

Overall, these features provide valuable support to the owner (the infrastructure concession holder in our case), in making informed decisions about maintenance investment priorities. They facilitate the efficient sharing and documentation of tunnel damage assessments, making the platform a practical and effective tool for real-world applications.

3. Implementation

3.1. Conventional evaluation methods

We used instance segmentation algorithms to implement the damage detection task. Therefore, their mainstream evaluation will be introduced as well. The evaluation method for instance segmentation usually follows the widely recognized COCO (Microsoft Common Objects in Context) official evaluation criteria (Lin et al., 2014). A prediction is considered accurate if the IoU between the predicted instance’s result and the GT exceeds a certain threshold τ, and the predicted category matches the GT category. Related evaluation method is as follows:

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{6}$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{7}$$

$$AR(\tau) = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i(\tau)}{TP_i(\tau) + FN_i(\tau)} \tag{8}$$

$$AP = \frac{1}{101} \times \sum_{r\in 0,0.01,\ldots,1} p_{interpolation}(r) \tag{9}$$

where TP_i, FN_i and FP_i denote the true positive, false negative and false positive instances i, and $p_{\text{interpolation}}(r)$ is the precision obtained through interpolation at the given maximum recall level r. Here, $i = 1, 2, 3, \ldots, n$, where n represents the total number of instances. The Average Recall (AR) measures the average recall across a dataset and is computed as the mean recall over a set of predefined IoU thresholds τ or a single IoU level. It quantifies the model’s ability to detect or segment objects, considering both true positives and missed detections. The Average Precision (AP) quantifies the area under the precision–recall curve and is computed as the mean precision over 101 equally spaced recall levels: $[0.0, 0.01, 0.02, \ldots, 1.0]$. The evaluation of instance segmentation encompasses both bounding boxes and masks, with the results presented as AP_b for bounding boxes and AP_m for masks. Overall, we provide AR and AP at IoU or IoUb thresholds of 0.5, 0.75, and the average over 0.5 to 0.95 with a 0.05 interval.

Meanwhile, we also provide some pixel-level evaluation methods, which will be applied in Section 4.3 for comparison. These evaluation methods also include recall and precision, which are consistent with the aforementioned Eqs. (6) and (7). However, under current situation, the calculations are performed at the pixel level rather than the instance level. Accuracy, F1-score and F2-score, with the latter placing greater

Table 3
Results of Mask R-CNN (He et al., 2018), Cascade Mask R-CNN (Cai and Vasconcelos, 2019) and Mask2Former (Cheng et al., 2022) under IoU and IoUb. The table also shows the absolute and percentage increases from IoU to IoUb.

	AP_b	AP_b^{50}	AP_b^{75}	AP_m	AP_m^{50}	AP_m^{75}	AR_m	AR_m^{50}	AR_m^{75}
Mask R-CNN									
IoU	0.230	0.384	0.212	0.195	0.350	0.184	0.259	0.440	0.245
IoUb	0.352	0.429	0.361	0.384	0.444	0.395	0.467	0.523	0.474
Inc.	**+0.122**	+0.045	**+0.149**	+0.189	+0.094	+0.211	+0.208	+0.083	+0.229
Cascade Mask R-CNN									
IoU	0.221	0.314	0.233	0.188	0.310	0.179	0.228	0.354	0.217
IoUb	0.299	0.356	0.296	0.316	0.358	0.318	0.353	0.387	0.356
Inc.	+0.078	+0.042	+0.063	+0.128	+0.048	+0.139	+0.125	+0.033	+0.139
Mask2Former									
IoU	**0.258**	**0.426**	**0.249**	**0.224**	**0.413**	**0.211**	**0.366**	**0.632**	**0.348**
IoUb	**0.372**	**0.475**	**0.383**	**0.433**	**0.509**	**0.433**	**0.721**	**0.806**	**0.726**
Inc.	+0.114	**+0.049**	+0.134	**+0.209**	**+0.096**	**+0.222**	**+0.355**	**+0.174**	**+0.378**

emphasis on recall, will also be provided. The aforementioned pixel-level evaluation will also be conducted separately based on IoU and IoUb. The relevant formulas are as follows:

$$Accuracy = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{10}$$

$$F_\beta - Score = (1+\beta^2) \times \frac{\text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \beta = 1, 2 \tag{11}$$

3.2. Training parameter setting

All experiments were conducted using NVIDIA A100 Tensor Core GPUs. The software environment consists of MMCV 2.1 and PyTorch 2.0.1 with CUDA version 11.7. Three classic instance segmentation algorithms were employed to accomplish our task: Mask R-CNN (He et al., 2018), Cascade Mask R-CNN (Cai and Vasconcelos, 2019) and Mask2Former (Cheng et al., 2022). The official configuration files for all three algorithms from MMDetection (Chen et al., 2019) were utilized, applying the data augmentation strategies configured in the official setup. All models follow the strategies below during the training process. The model input is standardized to 1024 × 1024. The total number of epochs is fixed at 100. The AdamW optimizer (Loshchilov and Hutter, 2019) is employed during training, with the learning rate schedule following a cosine annealing strategy (Loshchilov and Hutter, 2017). Training begins with a linear warm-up phase lasting 1000 iterations, after which the learning rate is set to 1e−4 and gradually reduced to 1e−7.

4. Results and discussion

4.1. Result of different instance segmentation algorithms

The results of three classic instance segmentation algorithms: Mask R-CNN (He et al., 2018), Cascade Mask R-CNN (Cai and Vasconcelos, 2019) and Mask2Former (Cheng et al., 2022) are shown in Table 3. We selected several cropped images to display the results of the three algorithms, shown in Fig. 11. All models used Swin-L (Liu et al., 2021) as backbone, and were trained starting from weights pre-trained on the ImageNet-22k dataset (Deng et al., 2009). We selected the final epoch to report the results and present both IoU and IoUb. We also present the comparison between IoU and IoUb, showing the increase in values from IoU to IoUb. Additionally, the details for five categories are shown in Table 4. We also tested the impact of different backbone sizes and demonstrated that the model can learn a bit more information from our data as the number of parameters increases. For details, please refer to the Appendix A.

After a comprehensive comparison, we found that under the conditions of our current dataset, Mask2Former performs the best, followed by Mask R-CNN. The reason for this is because Mask2Former has a better Transformer-based architecture (Cheng et al., 2022). From IoU to IoUb, the differences between models are not only maintained but also amplified to some extent, particularly in AR_m-related metrics, where Mask2Former shows a more significant increase compared to Mask R-CNN. Obviously, there is some randomness in this process, as AP and AR-related metrics are calculated based on IoU/IoUb thresholds, and increases are only observed once a certain threshold is reached. However, the overall trend remains consistent. This pattern is also evident in Cascade Mask R-CNN, which performs the worst on the current dataset and exhibits the smallest increase from IoU to IoUb.

This seems counter-intuitive since our proposed evaluation method relaxes the boundary constraints by introducing a buffer zone around defect instances. Intuitively, tolerance increases, which should benefit weaker models more, as they often struggle with edge precision. In contrast, strong models, being already highly accurate, would likely gain less from this added tolerance. However, while IoUb relaxes boundary constraints and Boundary IoU (Cheng et al., 2021) focuses on edge precision, both enhance evaluation differences between models, but they do so in different ways. IoUb emphasizes overall shape coverage, while Boundary IoU prioritizes boundary refinement. As a result, models like Mask2Former outperform Mask R-CNN in both aspects, illustrating the multidimensional nature of segmentation performance. Ultimately, effectiveness of metrics depends on how well it aligns with the task's specific goals.

Moreover, we further examined the visualization results. Since instance segmentation algorithms treat individual instances as segmentation targets, overlaps between instances can occur. From the visualization, we observed that some larger instances were often fully covered by multiple smaller instances. This is primarily because damage detection under complex conditions is a highly challenging task, making it difficult for the algorithm to accurately distinguish individual instances. This is also why we introduced a simple post-processing step to merge instances in our methodology.

We can further observe the differences between categories in Table 4. The most significant increase are observed in damaged joints and crack. Damaged joint has become the highest-scoring category in both AP_m and AR_m, achieving a significant leap in detection performance. This is because the general shape and occurrence of "damaged joints" follow some pattern, making them relatively easier for model to learn and detect compared to other categories. Similarly, under the IoU-based evaluation, most cracks are nearly impossible to recall. However, with the addition of a small buffer zone in the IoUb evaluation, the AR_m for cracks grows from 0.062 to 0.636, a 925.8% increase in

Table 4
The model comparison under specific categories, where the left side of the slash represents IoU and the right side represents IoUb.

Algorithm	IoU/IoUb	Seepage	Corrosion	Damaged joint	Spalling	Crack
Mask R-CNN	AP_m	0.364/0.518	0.072/0.116	0.112/0.612	0.408/0.543	0.016/**0.132**
	AP_m^{50}	0.562/0.622	0.156/0.192	0.393/0.691	0.545/0.568	**0.094/0.147**
	AP_m^{75}	0.384/0.528	0.052/0.114	0.021/0.660	0.464/0.542	0.000/0.132
	AR_m	0.461/0.614	0.131/0.208	0.205/0.695	0.474/0.611	0.026/0.208
	AR_m^{50}	0.646/0.692	0.265/0.304	0.552/0.759	0.607/0.628	0.128/0.231
	AR_m^{75}	0.498/0.621	0.098/0.206	0.086/0.724	0.541/0.612	0.000/0.205
Cascade Mask R-CNN	AP_m	0.333/0.449	0.081/0.117	0.082/0.425	0.417/0.511	**0.025**/0.079
	AP_m^{50}	0.484/0.529	0.149/0.149	0.324/0.492	0.513/0.539	0.079/0.079
	AP_m^{75}	0.368/0.451	0.059/0.129	0.010/0.428	0.456/0.504	0.000/0.079
	AR_m	0.403/0.506	0.138/0.189	0.109/0.448	0.466/0.546	0.026/0.077
	AR_m^{50}	0.536/0.567	0.225/0.225	0.379/0.500	0.551/0.566	0.077/0.077
	AR_m^{75}	0.442/0.508	0.127/0.206	0.017/0.448	0.500/0.541	0.000/0.077
Mask2Former	AP_m	**0.396/0.580**	**0.088/0.161**	**0.158/0.701**	**0.476/0.631**	0.003/0.092
	AP_m^{50}	**0.624/0.697**	**0.210/0.273**	**0.561/0.783**	**0.656/0.676**	0.012/0.117
	AP_m^{75}	**0.420/0.579**	**0.063/0.132**	**0.054/0.733**	**0.518/0.628**	0.000/0.094
	AR_m	**0.573/0.792**	**0.299/0.468**	**0.279/0.903**	**0.619/0.805**	**0.062/0.636**
	AR_m^{50}	**0.816/0.886**	**0.520/0.657**	**0.759/0.931**	**0.811/0.837**	**0.256/0.718**
	AR_m^{75}	**0.623/0.801**	**0.275/0.471**	**0.155/0.914**	**0.689/0.801**	0.000/**0.641**

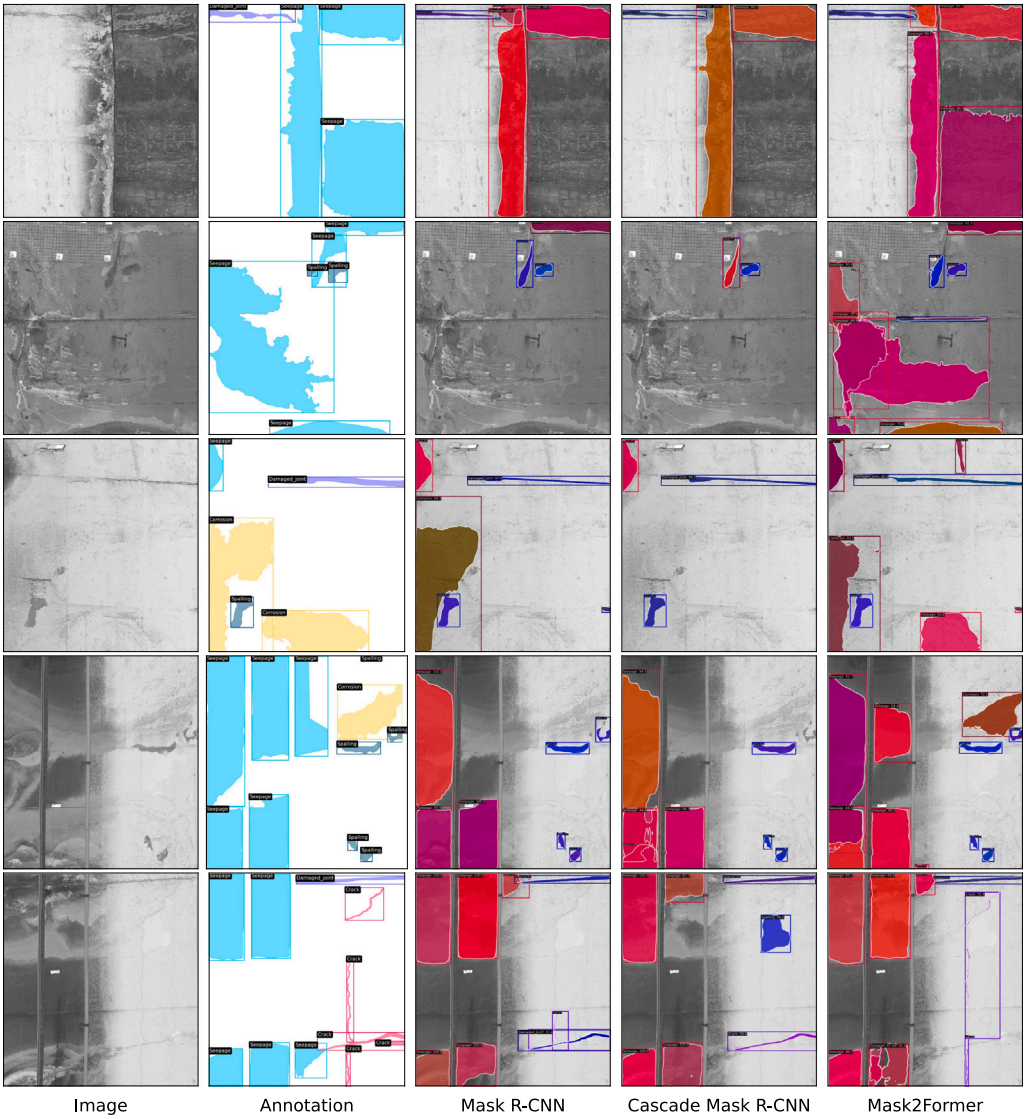

Fig. 11. Some visualization examples from different instance segmentation algorithms.

Mask2Former. This actually indicates that a large number of cracks can be detected, but the IoU penalty is too severe due to its shape,

does not accurately reflect real-world conditions. All of this can also be observed in the subsequent visualizations. Therefore, the introduction of the IoUb buffer zone mitigates the excessive impact of narrow and elongated shapes on detection accuracy and the inherent inaccuracies in damage annotations, leading to a more realistic representation of the results (Tsai and Chatterjee, 2017). The remaining categories, including seepage, corrosion, and spalling, show increases of 38%, 56.5%, and 30% in AR and 46.5%, 83% and 32.6% in AP, respectively. Among these, the boundary of corrosion is the most ambiguous during annotation due to its gradually transition with the background, while boundary of spalling is usually clearer.

4.2. Result of different confidence thresholds

Normalized confusion matrix under difference confidence score (0.5, 0.3, 0.2, 0.1) of three instance segmentation algorithms are shown in Figs. 12 and 13. The confusion matrix is calculated by comparing the predicted masks with the GT masks. An instance is recognized when the matched instance satisfies IoU or IoUb greater than 0.5. Taking the matrix in the last row and last column of Fig. 12 as an example (Mask2Former with a confidence threshold of 0.1), for the "seepage" category, 72% of instances were correctly identified, 2% were detected but misclassified as "corrosion", 1% were misclassified as "spalling", and 22% of instances were not detected, predicted as background.

Through these comparisons, we can observe the following findings. First, in most cases, whether using IoU or IoUb, lowering the confidence threshold alleviates missed detections without significantly increasing misclassification or false detections (identifying the background as damage). For example, in Mask R-CNN under IoU, when the confidence threshold is reduced from 0.5 to 0.1, the proportion of "seepage" instances misclassified as background decreases from 42% to 34%, a reduction of 8%, while false detections increase by only 2%. Second, we can analyse different categories by observing the matrix. It is evident that "corrosion" is the most frequently misclassified category, often mistaken with "seepage", whereas misclassification in other categories is not as pronounced. This observation aligns with real-world scenarios. Moreover, by transitioning from IoU to IoUb, the matrix provides a clearer representation of category-specific changes again, particularly for "damaged joint" and"crack", whose scores increase the most significantly. Third, confidence thresholds affect different categories with varying sensitivity. Specifically, for "crack", using IoUb as an example, a lower confidence threshold typically results in more detections. For instance, in Mask2Former, reducing the confidence threshold from 0.5 to 0.1 increases detected crack instances from 0% to 50%. On the other hand, "spalling" is the least sensitive, with its detection rate increasing by only 9%, from 65% to 74%. Similar trends are observed in the other two algorithms. One possible reason is that cracks are inherently difficult to detect due to their irregular and narrow and elongated shape. We present a more extreme example in Fig. 14, where the confidence threshold is lowered to 0.001. At this situation, almost all cracks are detected with limited misclassification. However, for other categories, it is clearly not recommended to use such a low confidence threshold for detection, as it would introduce too many misclassification. Therefore, it seems feasible to apply a lower confidence threshold specifically for crack category.

Overall, in damage detection tasks, IoUb adapts well to the task requirements, effectively reflecting the distinction between different models and providing a better representation of detection results for various damage categories. Lower confidence thresholds typically lead to more comprehensive detection, with only a very limited increase in misclassification. In other words, any defects detected by the model could have some basis, hence it is worth being questioned. For cracks, which are more difficult to detect, significantly higher recall is achieved at much lower confidence thresholds.

4.3. Result of post-processing

In this section, we compare the effectiveness of our proposed post-processing method for local tunnel panoramic images with the inference results obtained by directly stitching non-overlapping cropped images. Since instance information in the merged large images differs from that in the cropped images, making direct instance segmentation evaluation impractical. To ensure a meaningful and fair assessment, we adopt a pixel-level evaluation approach similar to semantic segmentation.

The first step involves reconstructing instance annotations for each local tunnel panorama sections. Specifically, we merge the annotations of all sub-images within each 20-meter section of the validation set, ensuring that adjacent cross-image instances are correctly combined. This process results in 5 local tunnel panoramas with instance segmentation annotation. However, such annotations have some instance overlap remains, hence cannot be directly converted to semantic segmentation. To address this, we transform instance segmentation into semantic segmentation by generating $C + 1$ separate binary masks (C object categories plus a background class) for both the GT and predictions. This approach ensures that pixels belonging to multiple instances of different categories are properly accounted for. The background mask is defined as:

$$M_{\text{bg}} = 1 - \max\left(M_1, M_2, \ldots, M_C\right), \tag{12}$$

where M_i represents the binary mask for category i. Next, we compute the Intersection over Union (IoU) and other pixel-based evaluation metrics separately for each of the $C + 1$ masks. Additionally, we report macro-averaged results over all categories:

$$\text{IoU}_{\text{macro}} = \frac{1}{C+1} \sum_{i=0}^{C} \text{IoU}_i. \tag{13}$$

The evaluation process for the cropped images follows the same methodology, with annotation conversion and calculations performed on all 180 validation images. The pixel-level evaluation results are presented in Tables 5 and 6. Among them, the confidence threshold is set to 0.2 in our post-processing setup, except for cracks (0.05). For directly stitched results, we use confidence threshold 0.3, which is the default value for most algorithms in MMCV (Chen et al., 2019).

We can observe that all macro-averaging metrics, except for the F2 score under IoUb, show improvement after post-processing, whether in terms of IoU or IoUb. Specifically for each category, all classes, except for the crack category, show an increase in performance after post-processing. This is because we significantly lowered the confidence threshold for the crack category, which leads to a decrease in the F2 score under IoU-based metrics and also affects the background class. This led to a substantial increase in recall under IoU (from 0.205 to 0.630), while this affects the precision, thereby impacting other metrics. As previously discussed in Section 4.2, we set this threshold for crack instances based on the premise of detecting as much as possible. If we maintain a consistent threshold with other categories, all metrics will show improvement. Under IoUb-based evaluation, we can achieve an IoUb of 0.304 for the crack category (an improvement of 0.073 compared to methods without post-processing), a Macro-F2 Score of 0.824 (a 0.037 improvement), and a Macro-IoUb of 0.642 (a 0.063 improvement). Although this would yield better scores, we prefer a more comprehensive detection approach.

Additionally, we observed that under IoUb, recall reached 1 regardless of whether post-processing was applied, meaning that all cracks we annotated were detected at the pixel level. As for why the instance segmentation result did not reach 100, this is because instance segmentation is evaluated based on individual instances, and pixel-level evaluations do not take instance-specific considerations into account. The pixels being fully captured could be the result of multiple instances being merged. Each instance may not be considered as TP compared with GT individually. Therefore, adjusting or merging instances could

Fig. 12. Normalized confusion matrices of three instance segmentation algorithms based on IoU at different confidence thresholds.

Table 5
Pixel-level evaluation of IoU and IoUb without post-processing.

	Background	Seepage	Corrosion	Damaged joint	Spalling	Crack	Macro
Under IoU							
Precision	0.938	0.840	0.502	0.389	0.613	0.133	0.569
Recall	0.966	0.805	0.557	0.772	0.704	0.205	0.668
Accuracy	0.924	0.937	0.962	0.997	0.994	0.999	0.969
F1 score	0.952	0.822	0.528	0.517	0.655	0.161	0.606
F2 score	0.960	0.812	0.545	0.645	0.683	0.185	0.638
IoU	0.908	0.698	0.359	0.349	0.487	0.088	0.481
Under IoUb							
Precision	0.958	0.879	0.538	0.530	0.646	0.231	0.630
Recall	0.978	0.868	0.641	0.988	0.873	1.000	0.891
Accuracy	0.949	0.956	0.968	0.998	0.996	0.999	0.978
F1 score	0.968	0.873	0.585	0.690	0.743	0.376	0.706
F2 score	0.974	0.870	0.617	0.843	0.816	0.600	0.787
IoUb	0.937	0.775	0.414	0.527	0.591	0.231	0.579

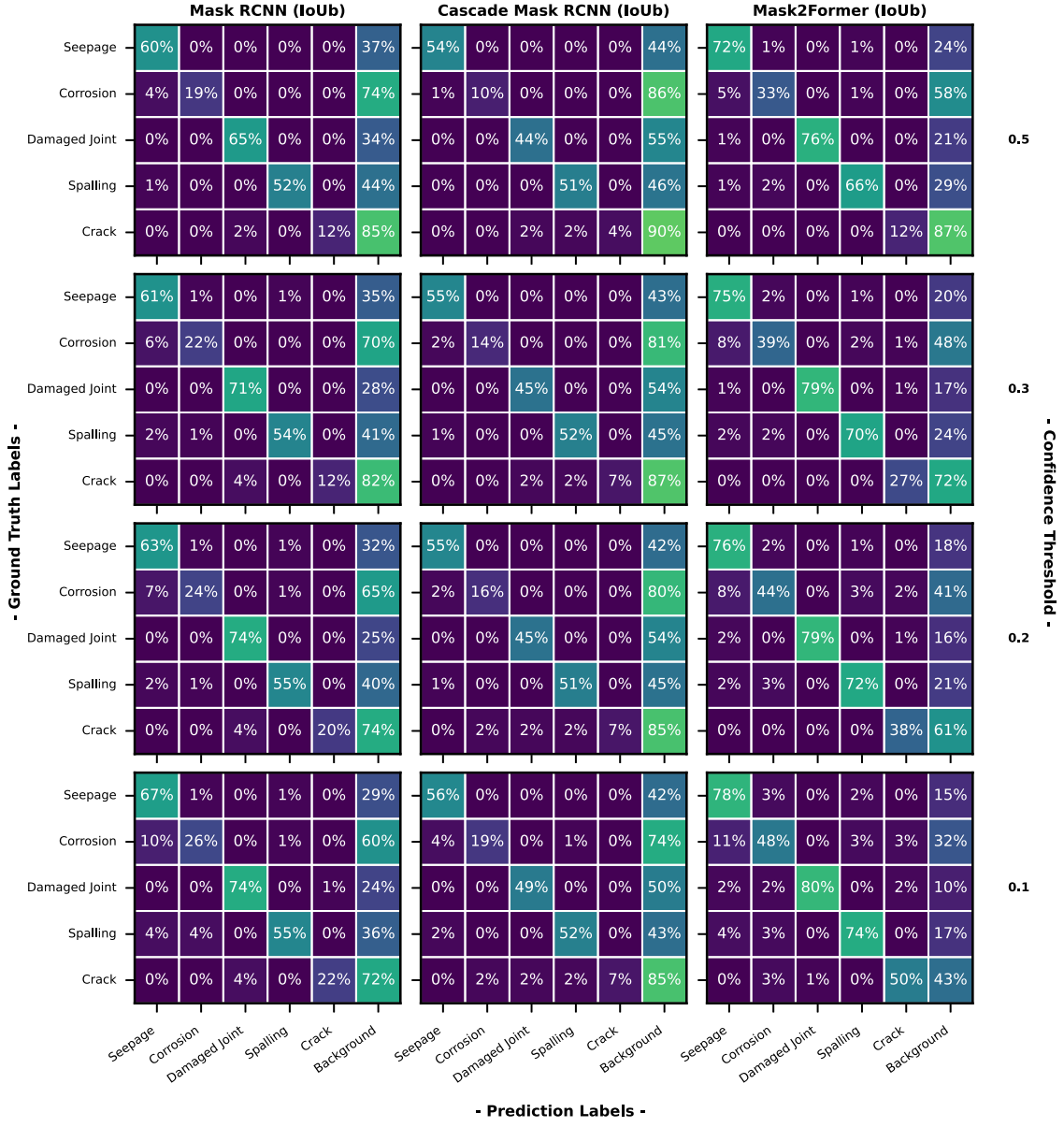

Fig. 13. Normalized confusion matrices of three instance segmentation algorithms based on IoUb at different confidence thresholds.

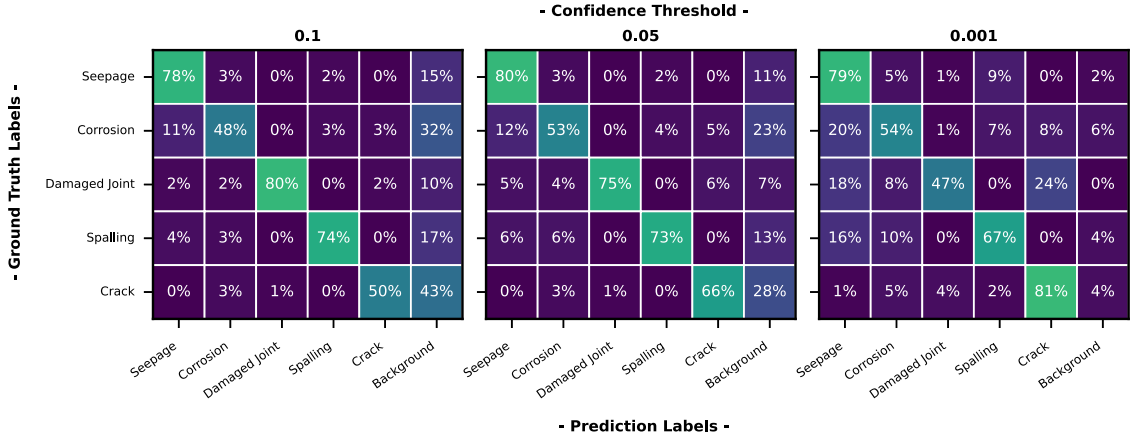

Fig. 14. Normalized confusion matrices of Mask2Former based on IoUb at lower confidence thresholds.

Table 6
Pixel-level evaluation of IoU and IoUb with post-processing (values with differences compared to results without post-processing).

	Background	Seepage	Corrosion	Damaged joint	Spalling	Crack	Macro
Under IoU							
Precision	0.942	0.867	0.678	0.473	0.622	0.069	0.608
	(+0.004)	(+0.027)	(+0.176)	(+0.084)	(+0.009)	(−0.064)	(+0.039)
Recall	0.958	0.816	0.570	0.781	0.709	0.630	0.744
	(−0.008)	(+0.011)	(+0.013)	(+0.009)	(+0.005)	(+0.425)	(+0.076)
Accuracy	0.922	0.944	0.973	0.998	0.995	0.996	0.971
	(−0.002)	(+0.007)	(+0.011)	(+0.001)	(+0.001)	(−0.003)	(+0.002)
F1 score	0.950	0.841	0.619	0.589	0.662	0.124	0.631
	(−0.002)	(−0.019)	(+0.091)	(+0.072)	(+0.007)	(−0.037)	(+0.025)
F2 score	0.955	0.826	0.589	0.691	0.689	0.239	0.665
	(−0.005)	(+0.014)	(+0.044)	(+0.046)	(+0.006)	(+0.054)	(+0.027)
IoU	0.904	0.726	0.448	0.418	0.495	0.066	0.510
	(−0.004)	(+0.028)	(+0.089)	(+0.069)	(+0.008)	(−0.022)	(+0.029)
Under IoUb							
Precision	0.961	0.910	0.721	0.674	0.665	0.126	0.676
	(+0.003)	(+0.031)	(+0.183)	(+0.144)	(+0.019)	(−0.105)	(+0.046)
Recall	0.971	0.878	0.657	0.990	0.879	1.000	0.896
	(−0.007)	(+0.010)	(+0.016)	(+0.002)	(+0.006)	(0.000)	(+0.005)
Accuracy	0.947	0.963	0.979	0.999	0.996	0.997	0.980
	(−0.002)	(+0.007)	(+0.011)	(+0.001)	(+0.000)	(−0.002)	(+0.002)
F1 score	0.966	0.894	0.687	0.802	0.757	0.224	0.722
	(−0.002)	(+0.021)	(+0.102)	(+0.112)	(+0.014)	(−0.152)	(+0.016)
F2 score	0.969	0.885	0.669	0.906	0.826	0.420	0.779
	(−0.005)	(+0.015)	(+0.052)	(+0.063)	(+0.010)	(−0.180)	(−0.008)
IoUb	0.935	0.809	0.524	0.670	0.609	0.126	0.612
	(−0.002)	(+0.034)	(+0.110)	(+0.143)	(+0.018)	(−0.105)	(+0.033)

be an area for further exploration in future research. Additionally, all the results based on IoUb maintain a similar level of distinction compared to those based on IoU.

Finally, we present a complete visualization example of a local panoramic tunnel in Fig. 15 to fully demonstrate the benefits of our introduced post-processing method. In Fig. 15(c), the directly stitched results of instance segmentation predictions on cropped images without overlap are presented. Different colours are used to distinguish instances, with the class label displayed at the top-left corner of each bounding box. Notably, for larger cross-image instances, direct merging is suboptimal. For example, the large "seepage" instance (indicated by the blue oval in Fig. 15(c)) near the lower central region of the image shows noticeable missing areas and demonstrates very poor continuity. Similarly, we observe that the loss of contextual information due to cropping leads to abrupt misalignments at the boundaries of many cross-image instances, and the random, unpredictable loss of various instances. Fig. 15(d) presents our post-processed merged result, which effectively resolves the cross-image instance issues. The instance edges are smooth, the overall integrity is well-defined, and no artifacts arise from contextual information loss. Although it is essentially the same model as in Fig. 15(c), enhancing the inference process has significantly improved the final results. Meanwhile, we significantly lowered the confidence threshold for cracks (0.05), and in the images, we even detected some cracks that were not annotated in the GT.

4.4. Local tunnel panoramas image merging

We conducted study on the merging strategy used in the process of merging 20-meter local panoramas image of tunnel. The following discussion is based on the Mask2Former (Cheng et al., 2022). To minimize the loss of contextual information between adjacent cropped images, we set a high overlap ratio of 50%. During the merging process, we tested three different strategies for handling logits in overlapping regions: averaging (mean), gradual adjustment (gradual), and maximum value aggregation (max). The gradual adjustment approach based on a cosine function. In this way, the final logits for each pixel in the overlapping region are contributed by all original logits covering that pixel. The contribution weights are assigned based on the pixel's distance from the centre of the corresponding cropped image, with a value of 1 at the centre and 0 at the edges. The visualization of weight matrices are shown in Fig. 16.

After tested the three merging strategies, we found out that with a fixed crop size, which implies fixed contextual information, it is hard to strike a balance between TN and FP when merging cropped images, regardless of the merging strategies employed. More detailed visual analyses and comparisons are provided in Appendix B. Therefore, to address this issue, our solution is to use multiple inferences combined with fusion. The multiple inferences employ different crop sizes, which introduces rich variations in the contextual information for each inference.

Specifically, during the merging of cropped images into a local tunnel panorama, we use a gradual adjustment overlay method. This method is less aggressive than the maximum value overlay and generally more effective than simple averaging, offering a robust and conservative stitching strategy. After multiple inferences (in our experiment, we conducted three inferences, each with square crop sizes of 1000, 1400, and 1800) on a local tunnel panorama, we apply the maximum value overlay to fuse the results. Since each single inference is already conservative, this ensures that no detected instances are missed during fusion. A local multiple inference example is presented in Fig. 17. In this example, we can see that using different crop sizes for single inference often misses some instances (marked in green and red). Merging multiple results gives more reliable outcomes.

4.5. Damage report

The front-end interactive interface is built using the widely adopted Gradio (Abid et al., 2019). As shown in Fig. 18, users can batch upload images, adjust unified confidence thresholds (with a dedicated threshold of 0.05 for "crack", as previously discussed, while a unified threshold of 0.2 is applied to all other categories), view legends for boundary maps, switch between different visualization modes, and

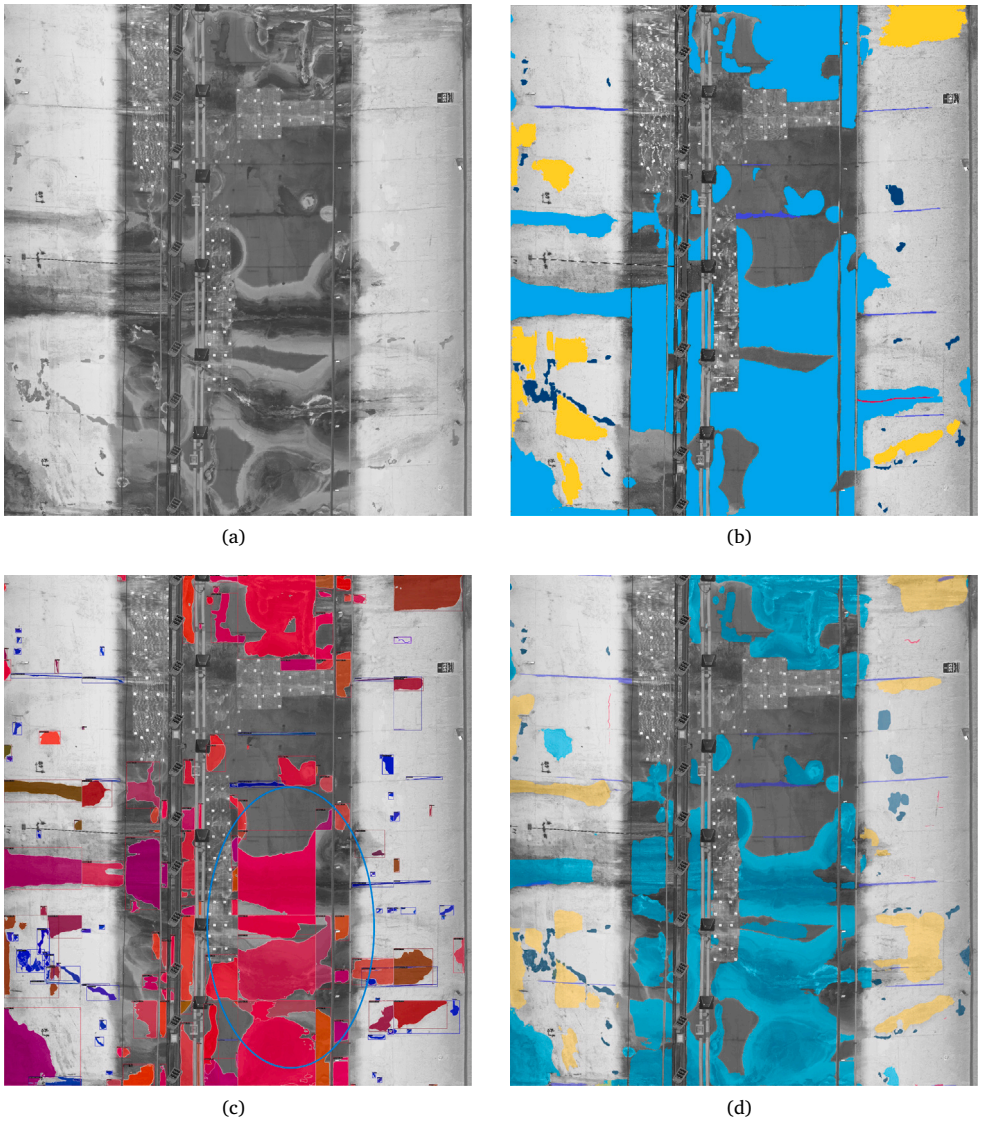

Fig. 15. A local panoramic tunnel example: (a) panoramic tunnel image; (b) manual annotation; (c) directly stitch from 1 kpx resolution images without post-processing; (d) proposed multi-inference stitching method.

export/download COCO format annotation file and PDF reports. The visualization area simultaneously displays the original image along with four selected visualization analysis maps, including segmentation maps, probability maps, binary maps, and boundary maps. The area below the visualization is the dialogue area for natural language query statistical results. The left side includes the question area and the returned statistical results, while the right side displays the statistical charts.

The process of generating PDF reports is based on PyPDF2 (Fenniak et al., 2022) and FPDF2 (py-pdf organization, 2021). We first created a template .pdf file that includes necessary legends and blank tables for information to be filled in. Then, we perform secondary processing on the boundary maps, where we draw a coordinate system with the bottom of the vertical axis set to 0 m and the top to 20 m. This serves as a reference for drawing the coordinate grid. For the horizontal axis, we set the centre of the image as 0, with the right side being positive, aligning with the format used in the manual report shown in Fig. 1. The processed image is then embedded into the template .pdf file. Finally, the necessary text information is printed in the corresponding table.

Taking one tunnel as an example, each 20 m section with a resolution of 6465 × 7078 requires about 270 cropped images for inference at three scales (50% overlap). Specifically, resolutions of 1000, 1400, and 1800 correspond to 156, 72, and 42 patches, respectively. The cropping principle ensures full edge coverage by adjusting the last patch to extend to the image boundary whenever the remaining area is smaller than 25% of the patch size (the minimum overlap). Using a single NVIDIA A100 Tensor Core GPU, Mask2Former achieves an inference speed of approximately 7 images per second (Ye et al., 2024), so processing 270 cropped images (one UHR image) takes roughly 40 s. With pre-processing and post-processing, the entire process is completed in under a minute. Unlike manual labelling methods, which are time-consuming and subjective, our approach automates damage detection for a 300 m-long tunnel (15 sections of 20 m) in just 15 min, providing consistent, objective, and repeatable results. An example of two pages from different tunnels are presented in Fig. 19. The overall

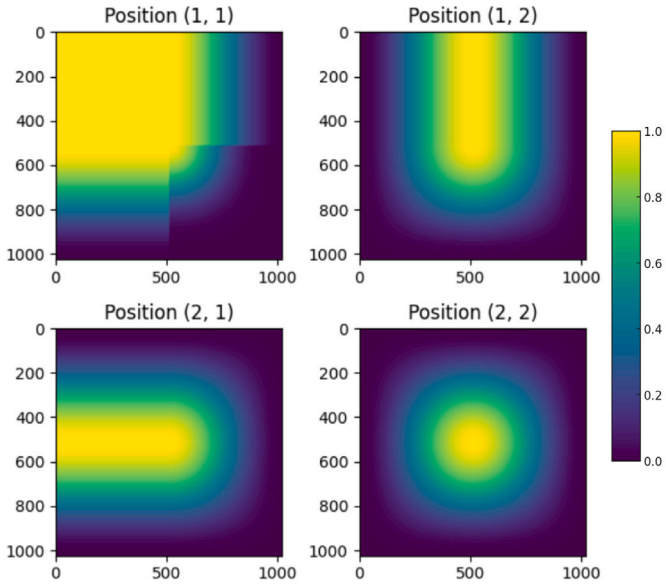

Fig. 16. Gradual adjustment weight matrices for overlapping area based on cosine function. These four weight maps represent the cropped images located at the top-left corner (1, 1), top edge (1, 2), left edge (2, 1), and non-edge and non-corner (2, 2).

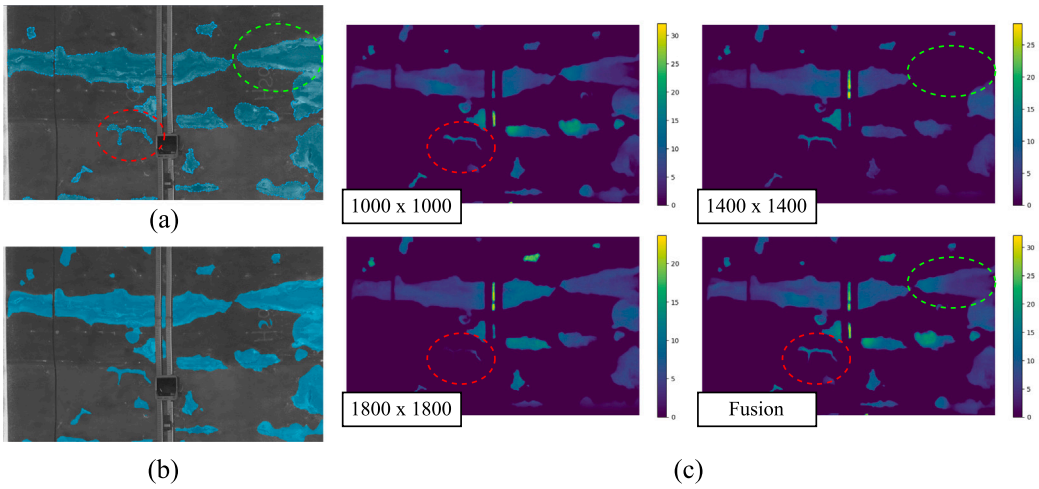

Fig. 17. A local multiple inference example: (a) local panoramic image with only "seepage" annotation; (b) prediction results; (c) predicted logits under different crop sizes and fusion results, and all values less than 0 are represented as 0. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

segmentation performance is notably strong in both tunnel, demonstrating clear delineation of damage locations and effectively distinguishing between different damage types. Furthermore, many potential cracks were captured, which is very beneficial.

Overall, in the current state, the platform provides an end-to-end solution for UHR tunnel image analysis, aiming to generate damage reports with quantitative and qualitative evaluations and digitized annotations to replace manual processing. It integrates hyper-parameter setting, pre-processing, multi-inference aggregation, post-processing, statistical analysis, and interactive features including visualization and natural language queries, with exportable outputs in .pdf and .json formats. Future development suggestions for the platform may include: (i) further optimizing the interactive front-end to allow online editing and correction of annotations; (ii) supporting historical data updates to automatically track changes for the same instances; (iii) continuously optimizing damage detection models; (iv) incorporating more quantitative and probabilistic risk metrics; (v) integrating with numerical simulation models to enable predictive analysis of damage evolution.

5. Conclusion

In this paper, we introduce a web-based framework for automated tunnel damage report generation based on UHR laser panoramic images, serving as a decision-support tool for the client. We also propose an new evaluation method, called IoU with buffer zone (IoUb), which reduces annotation inconsistencies in defects and more effectively as-

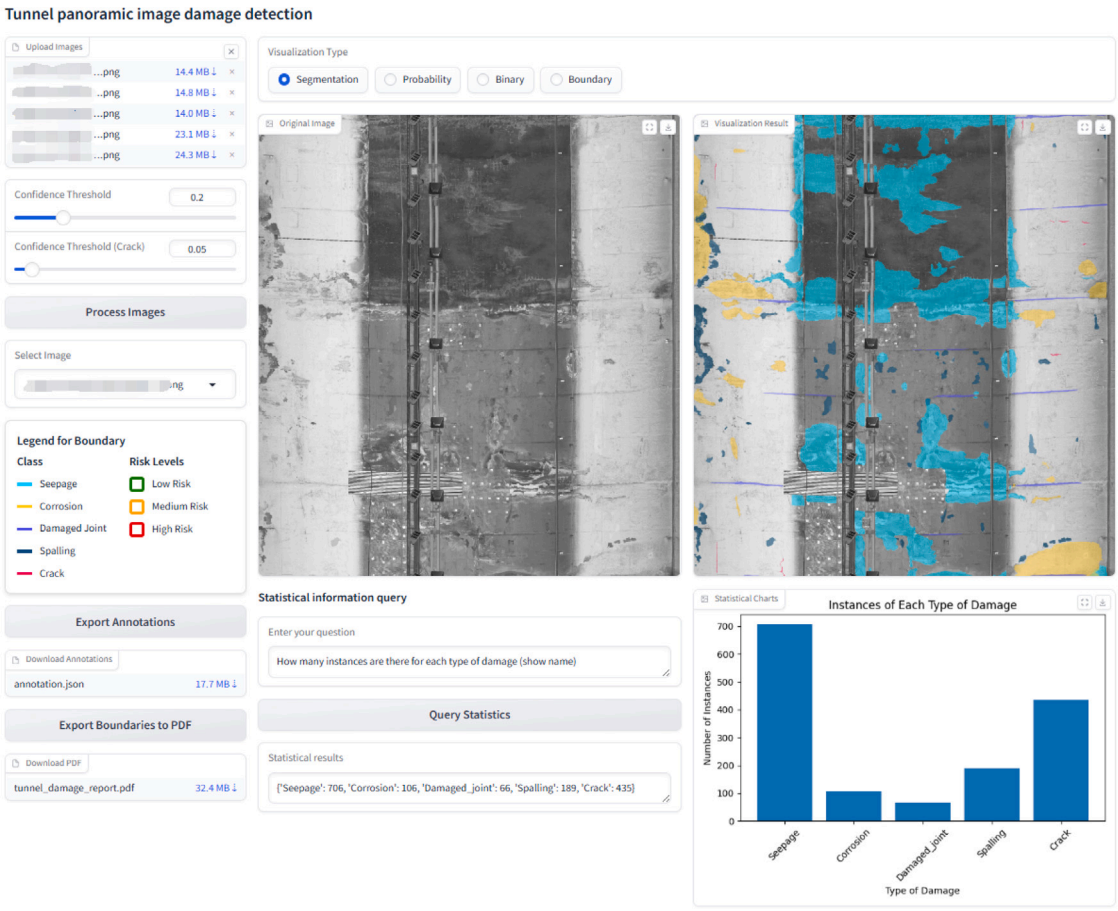

Fig. 18. Interactive interface for tunnel panoramic image damage detection.

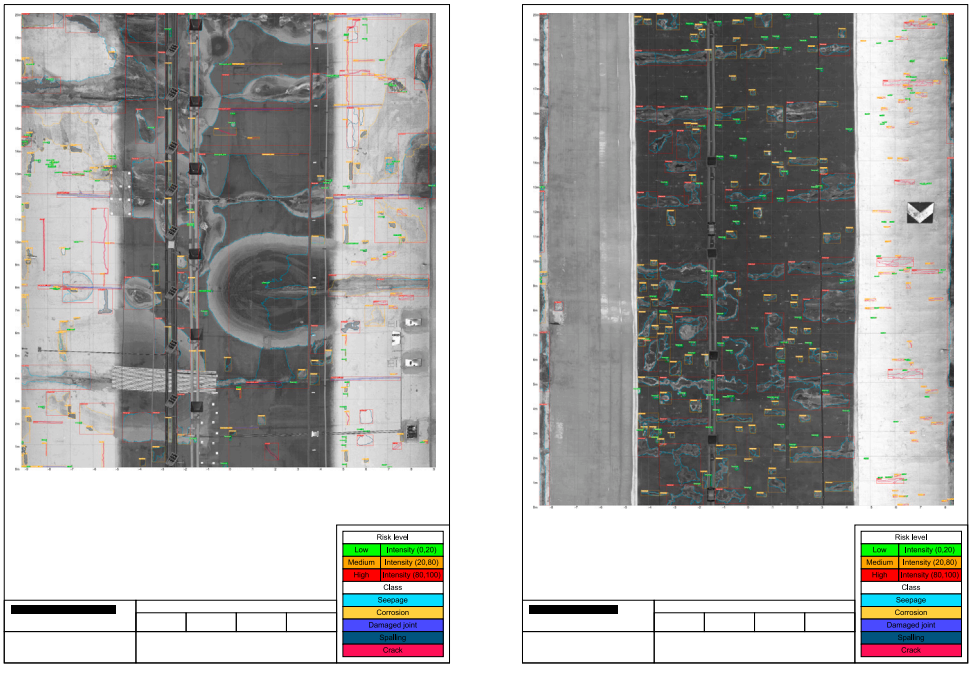

Fig. 19. Automatically generated damage report examples from two different tunnels based on 20 m tunnel local panoramic images.

sesses the defect prediction models, while allowing flexibility in boundary precision. We recommend a lower confidence threshold for better defect recall while it does not add excessive false positives. The main conclusions of this paper are:

- We present an end-to-end platform that automates the processing of UHR tunnel panoramic images, from pre-processing and post-processing to statistical-based multi-class damage severity estimation and report generation. Our approach enables the evaluation of UHR tunnel panoramas to segment highly complex scenes with multiple damage categories, an extremely challenging computer vision task. The system seamlessly outputs a detailed PDF report with digital damage annotations, eliminating the need for manual intervention, and supports centralized data management to facilitate unified storage and access. Hence, one of our contributions lies in integrating these steps into a fully automated pipeline for UHR images in the field of tunnel monitoring.
- In highly deteriorated tunnels, damage boundaries are often unclear, with gradual transitions, leading to inconsistent manual annotations and evaluation issues. We propose a new evaluation method that adds a buffer zone to the conventional IoU, focusing on the overall shape of the damage rather than potentially unreliable boundaries. This approach enables more objective evaluations in complex tunnels and better distinguishes between models. We found that in our task, better instance segmentation models lead to a greater absolute increase in IoUb-based evaluation scores.
- We introduced a post-processing method for UHR image segmentation, which can adapt to both query-based and NMS-based decoder, where we perform multiple inferences on different crop sizes and then aggregate the results. This approach yields better evaluation and visualization results. Compared to direct inference and stitching, the average IoU and IoUb improved from 0.481 and 0.579 to 0.510 and 0.612, respectively. Ultimately, it can recall approximately 90% of the damage based on Mask2Former.
- We generate damage reports with our framework that clearly map the locations of five types of damage, ranking each instance by area to classify its severity. Unlike the current process, which is difficult, time-consuming, and highly subjective, our approach automates this task in one minutes per 20 m section, delivering consistent, objective, and repeatable results. This transformation significantly enhances the efficiency of maintenance and monitoring efforts. Moreover, this system revolutionizes the management of inspections and lays the groundwork for creating a database for predictive maintenance. As a key component of the decision-support platform (Villa et al., 2025), the tool supports the concessionaire in making strategic decisions about budget distribution for both routine and extraordinary maintenance across the tunnel infrastructure spread throughout Italy.

There are some limitations in our current method. The first aspect is algorithmic. During annotation and inference, connected damaged areas of the same type are treated as a single instance, which may misrepresent risk in cases where damage is interrupted by pipes or maintenance structures. Additionally, multi-inference on UHR images requires large overlaps and multiple passes, increasing computational cost and leaving room for optimization. Second, the current workflow has not been designed to track historical information, which is important for observing changes in the same instance over time and provides valuable data for time-series deterioration models. Third, the current risk classification is statistically derived from damage characteristics and serves only as a reference rather than a definitive basis for maintenance prioritization.

To further develop the framework of fully automated damage detection and evaluation,future work will revolve around : (a) A more suitable method for identifying damage instances involves considering the occlusion relationships with tunnel accessories, while also taking into account tunnels with different structural characteristics and potentially extending the proposed UHR image processing approach to other types of infrastructure. (b) Using multiple sensors, such as thermal cameras and RGB-D cameras, to generate different types of data and fuse them together, combining the advantages of each modality to provide better data for inspection and analysis; (c) Developing specialized UHR algorithms for tunnels to improve the accuracy and inference efficiency of tunnel panoramic image recognition; (d) Establishing numerical simulation models based on identified surface damage to more accurately estimate risks; (e) Integrate this system with the Tunnel Asset Management platform to enable the seamless transfer of real-time data captured from the images to the control dashboard, allowing for efficient monitoring, analysis, and decision-making.

CRediT authorship contribution statement

Zehao Ye: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mohammadhamed Mozafarian:** Writing – review & editing, Data curation. **Paola Alice Rosa Cavallaro:** Writing – review & editing, Writing – original draft, Data curation. **Kamil Altinay:** Writing – review & editing, Data curation. **Valentina Villa:** Writing – review & editing, Supervision, Resources. **Jelena Ninić:** Writing – review & editing, Supervision, Resources, Methodology.

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.

Acknowledgements

The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC, United Kingdom and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham. The computations described in this paper were also performed using the University of Birmingham's BlueBEAR (http://www.birmingham.ac.uk/bear) HPC service, which provides a High Performance Computing service to the University's research community. The engineering company TECNE underpinned the research and contributed all necessary data. Their ongoing collaboration has made it possible to engage with practical applications relevant to the field. This publication is part of the project PNRR-NGEU, which has received funding from MUR-DM 118/2023.

Appendix A. Impact of different backbone size

We evaluated the model's performance on IoU and IoUb across different backbone sizes to analyse the impact of model performance on IoUb. Using varying backbone sizes provides a controlled way to adjust model capacity, as larger backbones are expected to yield progressively better results. The results are shown in Tables A.1 and A.2.

In most metrics, Swin-L still demonstrates the best performance. However, the overall gap is not substantial, which can be attributed to the limitations of the dataset (with a total of 1,430 images). A larger backbone cannot capture significantly more information due to the size of the dataset. Nevertheless, the most comprehensive metric, AP_m, reveals that results under IoUb show a clear and well-defined increase in differences across the various backbone sizes. A larger dataset with more precise annotations could potentially lead to greater differentiation.

Table A.1
Results of Mask2Former (Cheng et al., 2022) across different backbone sizes under IoU; Swin-T and Swin-S backbones are pre-trained on ImageNet-1k dataset (Deng et al., 2009).

Backbone	AP_b	AP_b^{50}	AP_b^{75}	AP_m	AP_m^{50}	AP_m^{75}	AR_m	AR_m^{50}	AR_m^{75}
Swin-T	0.247	0.404	0.242	0.221	0.411	0.210	**0.371**	**0.659**	**0.353**
Swin-S	0.240	0.382	0.235	0.219	0.405	0.208	0.362	0.621	0.348
Swin-B	0.255	0.400	**0.260**	0.217	0.391	**0.212**	0.363	0.635	0.351
Swin-L	**0.258**	**0.426**	0.249	**0.224**	**0.413**	0.211	0.366	0.632	0.348

Table A.2
Results of Mask2Former (Cheng et al., 2022) across different backbone sizes under IoUb.

Backbone	AP_b	AP_b^{50}	AP_b^{75}	AP_m	AP_m^{50}	AP_m^{75}	AR_m	AR_m^{50}	AR_m^{75}
Swin-T	0.360	0.459	0.359	0.422	0.501	0.427	0.717	**0.813**	**0.729**
Swin-S	0.352	0.450	0.348	0.424	0.503	0.432	0.701	0.799	0.708
Swin-B	0.367	0.462	0.365	0.429	**0.510**	0.426	0.699	0.778	0.703
Swin-L	**0.372**	**0.475**	**0.383**	**0.433**	0.509	**0.433**	**0.721**	0.806	0.726

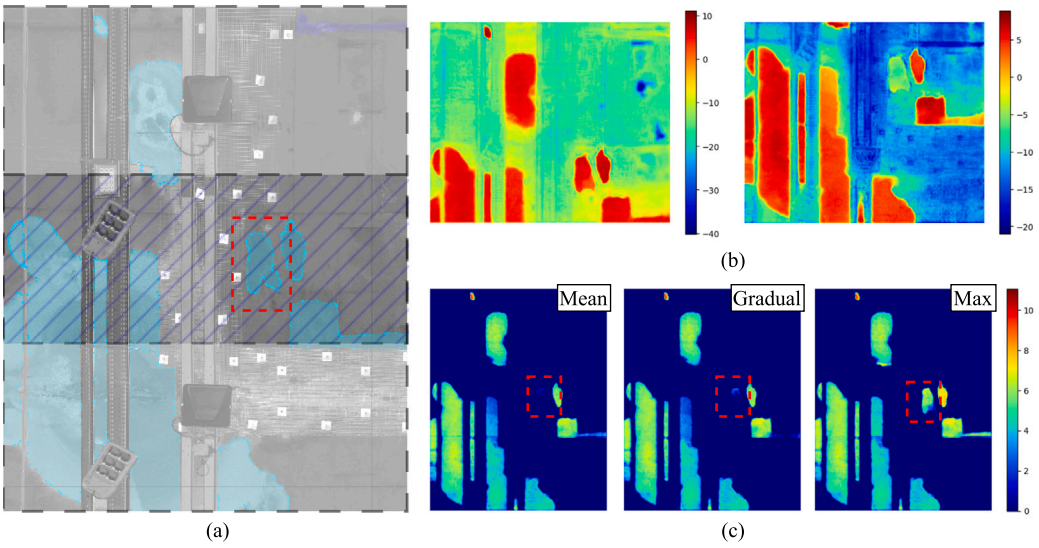

Fig. B.1. Under-segmentation example: (a) the two adjacent cropped images with annotations overlap with each other, and the instances marked with red dashed box is the one being discussed. (b) the original logits of the two cropped images, with the left side representing the upper one and the right side representing the lower one. (c) the merging of the two cropped images using three strategies, with only the portions greater than 0 are visualized. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Appendix B. Under-segmentation and over-segmentation

Here, we provide examples of under-segmentation and over-segmentation for the seepage category with a fixed crop, as shown in Figs. B.1 and B.2, to illustrate why we designed this post-processing workflow. In Fig. B.1, only when using the maximum value overlay can we capture the seepage instance (red dashed box) near the centre of the overlapping region. This is because in the predictions of the two adjacent cropped images, only the top image correctly recognized this instance. Although the bottom image also roughly outlines the instance and shows significant differences from the surrounding background, the overall logits are below 0, meaning it is still considered as background. Therefore, both the averaging and progressive overlay methods lead to the loss of this seepage instance. Only with the maximum value overlay can it be successfully captured.

However, for the example in Fig. B.2, the bottom image is cropped right at the edge of a large seepage instance due to the cropping. This loss of contextual information, a highly uncommon case, causes the bottom cropped image to incorrectly segment the boundary of the seepage, leading the top-right corner of the cropped image to be mistakenly considered within the range of the seepage instance. In contrast, the top image, with its more holistic semantic information for this instance, better outlines the edges of the seepage. When both images are overlaid, the reduction effect from the region considered as background in the top image allows the final logits from both the averaging and progressive methods to better distinguish the edges of this instance. However, using the aggressive maximum value overlay results in over-segmentation.

In summary, using multiple inferences with different crop sizes helps address the challenge of balancing TN and FP during image merging by introducing varied contextual information. We can also see in Fig. B.1, the most aggressive method, max, is the best, but the gradual method also detects some instances and outperforms the average method. While in Fig. B.2, it is clear that the gradual method yields the best results. Therefore, we selected the gradual method for its stability and robustness, as it provides a more consistent stitching approach in most cases during a single inference process. We present the local prediction results in Fig. B.3. Through the aforementioned method, we successfully captured both instances from the examples in Figs. B.1 and B.2.

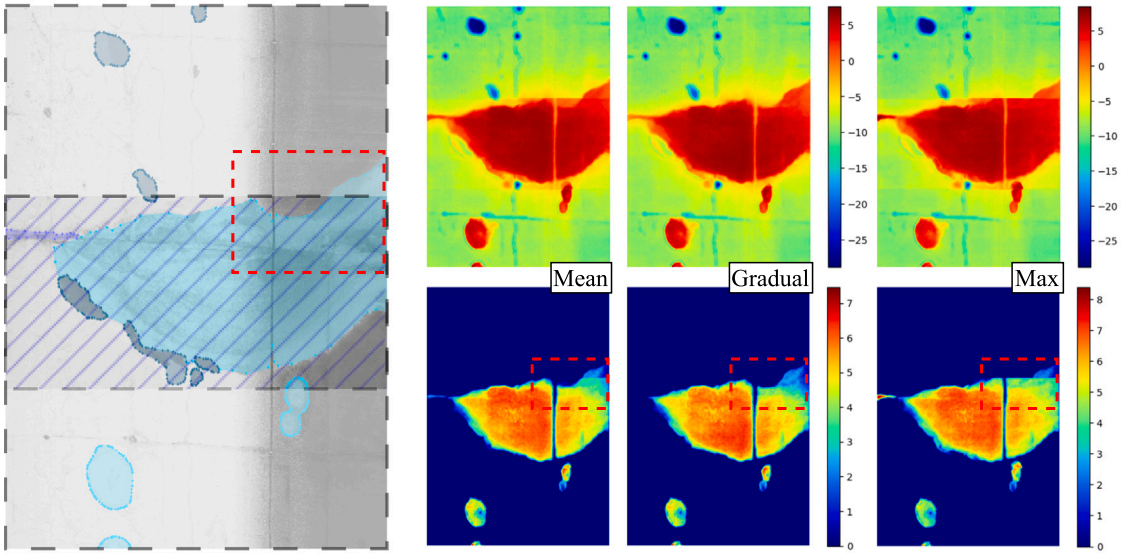

Fig. B.2. Over-segmentation example: (a) the two adjacent cropped images with annotations overlap with each other, and the area marked with red dashed box is the one being discussed. (b) the upper part visualizes the merged logits directly, while the lower part visualizes only the portions greater than 0. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

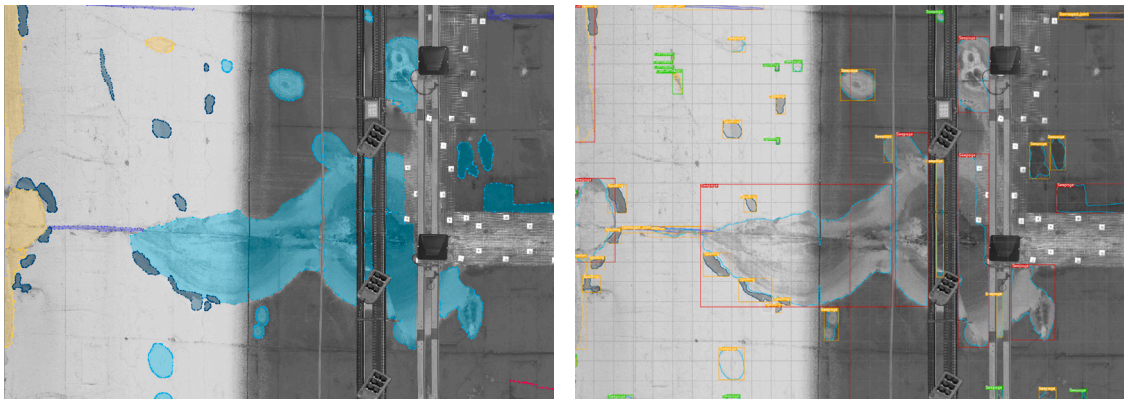

Fig. B.3. The local prediction results demonstrate that our method effectively balances TN and FP: (a) image with GT, (b) boundary map of prediction results.

Data availability

Data will be made available on request.

References

Abdelkader, E.M., Al-Sakkaf, A., Elkabalawy, M., Omar, A., Alfalah, G., 2025. Simulating the deterioration behavior of tunnel elements using amalgamation of regression trees and state-of-the-art metaheuristics. Mathematics 13 (7), 1021.

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., Zou, J., 2019. Gradio: Hassle-free sharing and testing of ML models in the wild. arXiv preprint arXiv:1906.02569.

Ahmed, M.O., Khalef, R., Ali, G.G., El-Adaway, I.H., 2021. Evaluating deterioration of tunnels using computational machine learning algorithms. J. Constr. Eng. Manag. 147 (10), 04021125.

An, Y.-K., Kang, M.-S., 2024. Crack growth prediction on a concrete structure using deep convlstm. Smart Struct. Syst. an Int. J. 33 (4), 301–311.

Attard, L., Debono, C.J., Valentino, G., Di Castro, M., 2018. Tunnel inspection using photogrammetric techniques and image processing: A review. ISPRS J. Photogramm. Remote Sens. 144, 180–188.

Bradski, G., 2000. The opencv library. Dr. Dobb's J. Softw. Tools.

Cai, Z., Vasconcelos, N., 2019. Cascade R-CNN: High quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 43 (5), 1483–1498.

Chatterjee, B., Poullis, C., 2021. Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks. Comput. Vis. Image Underst. 210, 103251.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans. Pattern Anal. Mach. Intell. 40 (4), 834–848.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al., 2019. MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155.

Cheng, H.K., Chung, J., Tai, Y.-W., Tang, C.-K., 2020. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8890–8899.

Cheng, B., Girshick, R., Dollar, P., Berg, A.C., Kirillov, A., 2021. Boundary iou: Improving object-centric image segmentation evaluation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR, pp. 15334–15342.

Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R., 2022. Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299.

Chung, J., Lee, K., 2023. Credit card fraud detection: an improved strategy for high recall using KNN, LDA, and linear regression. Sensors 23 (18), 7788.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Ieee, pp. 248–255.

Deng, J., Singh, A., Zhou, Y., Lu, Y., Lee, V.C.-S., 2022. Review on computer vision-based crack detection and quantification methodologies for civil structures. Constr. Build. Mater. 356, 129238.

Feng, S.J., Feng, Y., Zhang, X.L., Chen, Y.H., 2023. Deep learning with visual explanations for leakage defect segmentation of metro shield tunnel. Tunn. Undergr. Space Technol. 136, 105107.

Feng, Y., Zhang, X.-L., Feng, S.-J., Zhang, W., Hu, K., Da, Y.-W., 2025. Intelligent segmentation and quantification of tunnel lining cracks via computer vision. Struct. Heal. Monit. 24 (3), 1896–1926.

Fenniak, M., Stamy, M., pubpub-zz, Thoma, M., Peveler, M., exiledkingcc, PyPDF2 Contributors, 2022. The PyPDF2 library. See https://pypdf2.readthedocs.io/en/latest/meta/CONTRIBUTORS.html for all contributors.

Foria, F., Avancini, G., Ferraro, R., Miceli, G., Peticchia, E., 2019. ARCHITA: an innovative multidimensional mobile mapping system for tunnels and infrastructures. In: MATEC Web of Conferences, vol. 295, EDP Sciences, p. 01005.

Foria, F., Miceli, G., Calicchio, M., Brichese, M., 2024. Decarbonization and climate change analysis of tunnels in an asset management framework through MIRET. Procedia Struct. Integr. 62, 1069–1076.

Foria, F., Miceli, G., Nascetti, A., Loprencipe, G., Crespi, M., Belloni, V., Ravanelli, R., Cordaro, S., et al., 2022. Digitalization and defects analysis for the maintenance of mechanized tunnels. In: Proceedings World Tunnel Congress, WTC2022. ITA-AITES.

Geng, P., Jia, M., Ren, X., 2023. Tunnel lining water leakage image segmentation based on improved BlendMask. Struct. Heal. Monit. 22 (2), 865–878.

Géron, A., 2022. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc..

Glab, K., Westerlund, K., Shivasami, A., Foria, F., Dewangan, A., Robert, F., Menozzi, A., Karlovsek, J., 2025. ITA development in ML and AI in tunnelling. In: Tunnelling Into a Sustainable Future–Methods and Technologies. CRC Press, pp. 158–165.

Guo, S., Liu, L., Gan, Z., Wang, Y., Zhang, W., Wang, C., Jiang, G., Zhang, W., Yi, R., Ma, L., et al., 2022. Isdnet: Integrating shallow and deep networks for efficient ultra-high resolution segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4361–4370.

Guzella, T.S., Caminhas, W.M., 2009. A review of machine learning approaches to spam filtering. Expert Syst. Appl. 36 (7), 10206–10222.

He, Y., Chen, Q., 2023. Construction and application of LSTM-based prediction model for tunnel surrounding rock deformation. Sustainability 15 (8), 6877.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2018. Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. 42 (2), 386–397.

He, Y., Tan, Y., Yang, M., Wang, Y., Xu, Y., Yuan, J., Li, X., Chen, W., Kang, G., 2024. Accurate prediction of discontinuous crack paths in random porous media via a generative deep learning model. Proc. Natl. Acad. Sci. 121 (40), e2413462121.

Huang, H., Cai, Y., Zhang, C., Lu, Y., Hammad, A., Fan, L., 2024. Crack detection of masonry structure based on thermal and visible image fusion and semantic segmentation. Autom. Constr. 158, 105213.

Huang, M., Ninić, J., Zhang, Q., 2021. BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives. Tunn. Undergr. Space Technol. 108, 103677.

Ji, S., Zhang, H., 2023. ISAT with segment anything: An interactive semi-automatic annotation tool. URL https://github.com/yatengLG/ISAT_with_segment_anything, Updated on 2023-06-03.

Jiang, Y., Wang, L., Zhang, B., Dai, X., Ye, J., Sun, B., Liu, N., Wang, Z., Zhao, Y., 2023. Tunnel lining detection and retrofitting. Autom. Constr. 152, 104881.

Kirillov, A., He, K., Girshick, R., Rother, C., Dollar, P., 2019. Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., et al., 2023. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026.

Li, Q., Cai, J., Luo, J., Yu, Y., Gu, J., Pan, J., Liu, W., 2024. Memory-constrained semantic segmentation for ultra-high resolution uav imagery. IEEE Robot. Autom. Lett..

Li, D., Xie, Q., Gong, X., Yu, Z., Xu, J., Sun, Y., Wang, J., 2021. Automatic defect detection of metro tunnel surfaces using a vision-based inspection system. Adv. Eng. Informat. 47, 101206.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, pp. 740–755.

Lin, L., Zhu, H., Ma, Y., Peng, Y., Xia, Y., 2025. Surface feature and defect detection method for shield tunnel based on deep learning. J. Comput. Civ. Eng. 39 (3), 04025019.

Liu, W., Li, Q., Lin, X., Yang, W., He, S., Yu, Y., 2024. Ultra-high resolution image segmentation via locality-aware context fusion and alternating local enhancement. Int. J. Comput. Vis. 132 (11), 5030–5047.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022.

Liu, J., Zhao, Z., Lv, C., Ding, Y., Chang, H., Xie, Q., 2022. An image enhancement algorithm to improve road tunnel crack transfer detection. Constr. Build. Mater. 348, 128583.

Loshchilov, I., Hutter, F., 2017. SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. In: International Conference on Learning Representations.

Ma, Z., Liu, S., 2018. A review of 3D reconstruction techniques in civil engineering and their applications. Adv. Eng. Informat. 37, 163–174.

Ministero delle Infrastrutture e della Mobilità Sostenibili, 2022. Linee guida per la classificazione e gestione del rischio, la valutazione della sicurezza ed il monitoraggio delle gallerie esistenti (Guidelines for risk classification and management, safety assessment, and monitoring of existing tunnels). Technical Report Parere n. 29/2022, Consiglio Superiore dei Lavori Pubblici, Rome, Italy, Espresso dall'Assemblea Generale in data 08.04.2022.

Ministro delle infrastrutture e dei trasporti, 2023. Relazione concernente lo stato di attuazione degli interventi relativi all'adeguamento delle gallerie stradali della rete transeuropea. https://www.senato.it/service/PDF/PDFServer/DF/426541.pdf. (Accessed: 15 March 2024).

Mozafarian, M.H., Desiderio, G., Ye, Z., Ninic, J., Villa, V., 2025. Image-based multi-damage detection in tunnels: a deep learning dataset for structural health monitoring. In: European Conference on Computing in Construction (EC3) 2025. Porto, Portugal.

Neuhold, R., Ollmann, P., Rusu, R.B., Smeulders, A.W.M., 2017. The vistas dataset for semantic scene understanding. In: Proceedings of the IEEE International Conference on Computer Vision. ICCV, pp. 753–761.

Ninic, J., Ye, Z., Bui, H.-G., Altinay, K., Cavallaro, P.A.R., Villa, V., 2025. Digital twin for damage modelling of tunnel linings. In: Digital Twins in Engineering & Artificial Intelligence and Computational Methods in Applied Science (DTE - AICOMAS 2025). Paris, France.

Ouyang, A., Di Murro, V., Cull, M., Cunningham, R., Osborne, J.A., Li, Z., 2023. Automated pixel-level crack monitoring system for large-scale underground infrastructure–a case study at CERN. Tunn. Undergr. Space Technol. 140, 105310.

Ouyang, A., Di Murro, V., Daakir, M., Osborne, J.A., Li, Z., 2025. From pixel to infrastructure: Photogrammetry-based tunnel crack digitalization and documentation method using deep learning. Tunn. Undergr. Space Technol. 155, 106179.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A., 2016. A benchmark dataset and evaluation methodology for video object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 724–732.

py-pdf organization, 2021. fpdf2: A python library for generating PDF documents. URL https://github.com/PyPDF2/fpdf2.

Ren, Y., Huang, J., Hong, Z., Lu, W., Yin, J., Zou, L., Shen, X., 2020. Image-based concrete crack detection in tunnels using deep fully convolutional networks. Constr. Build. Mater. 234, 117367.

Schade, W., Rothengatter, W., Stich, M., Streif, M., Himmelsbach, M., Lindberg, N., Stasio, C., Fermi, F., Maffii, S., Zani, L., Bielanska, D., Skinner, I., 2022. Analysis accompanying the impact assessment for the revision of regulation (EU) n° 1315/2013 on union guidelines for the development of the trans-European transport network – final report. Publications Office of the European Union.

spacetec, 2019. Spacetec. URL https://www.spacetec.de/en/. (Accessed on 23 January 2025).

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M., 2020. Test-time training with self-supervision for generalization under distribution shifts. In: International Conference on Machine Learning. PMLR, pp. 9229–9248.

Tsai, Y.-C., Chatterjee, A., 2017. Comprehensive, quantitative crack detection algorithm performance evaluation system. J. Comput. Civ. Eng. 31 (5), 04017047.

Villa, V., Chiaia, B., Cavallaro, P., Rahmanpour, N., 2025. A digital approach for effective asset management of tunnel infrastructure. In: New Frontiers of Construction Management. Proceedings of CMW 24. Springer Cham.

Wang, Y., Liao, W., Dong, A., Xu, L., Zhu, L., Shi, H., Yu, Z., 2024. High-speed acquisition and intelligent tunnel surface defects recognition. Tunn. Undergr. Space Technol. 144, 105572.

Xu, L., Wang, Y., Dong, A., Zhu, L., Shi, H., Yu, Z., 2023. Image-based intelligent detection of typical defects of complex subway tunnel surface. Tunn. Undergr. Space Technol. 140, 105266.

Yang, H., Wang, L., Pan, Y., Chen, J.-J., 2025. A teacher-student framework leveraging large vision model for data pre-annotation and YOLO for tunnel lining multiple defects instance segmentation. J. Ind. Inf. Integr. 44, 100790.

Yankaskas, B.C., Cleveland, R.J., Schell, M.J., Kozar, R., 2001. Association of recall rates with sensitivity and positive predictive values of screening mammography. Am. J. Roentgenol. 177 (3), 543–549.

Ye, Z., Lin, W., Faramarzi, A., Ninić, J., 2025. Automated digital twin reconstruction for tunnel inspection and maintenance. In: Proceedings of the World Tunnel Congress 2025. CRC Press.

Ye, Z., Lovell, L., Faramarzi, A., Ninić, J., 2024. Sam-based instance segmentation models for the automation of structural damage detection. Adv. Eng. Informat. 62, 102826.

Zhang, C., Chen, X., Liu, P., He, B., Li, W., Song, T., 2024. Automated detection and segmentation of tunnel defects and objects using YOLOv8-CM. Tunn. Undergr. Space Technol. 150, 105857.

Zhao, S., Zhang, D.M., Huang, H.W., 2020. Deep learning–based image instance segmentation for moisture marks of shield tunnel lining. Tunn. Undergr. Space Technol. 95, 103156.

Zhou, B., Elhoseiny, A.M., Yang, M.F., Yuille, A.L., 2017. Scene parsing through ADE20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 633–641.

Zhou, Z., Yan, L., Zhang, J., Zheng, Y., Gong, C., Yang, H., Deng, E., 2023. Automatic segmentation of tunnel lining defects based on multiscale attention and context information enhancement. Constr. Build. Mater. 387, 131621.

Zhu, Z.-H., Fu, J.-Y., Yang, J.-S., Zhang, X.-M., 2016. Panoramic image stitching for arbitrarily shaped tunnel lining inspection. Computer-Aided Civ. Infrastruct. Eng. 31 (12), 936–953.