POLITECNICO DI TORINO
Repository ISTITUZIONALE

Maintenance-oriented tunnel digital model generation via panoptic segmentation of ultra-high-resolution images

Original
Maintenance-oriented tunnel digital model generation via panoptic segmentation of ultra-high-resolution images / Ye, Z., Li, Q.i., Desiderio, G., Huange, M., Liuc, W., Villa, V., Nini, J.. - In: COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING. - ISSN 1467-8667. - 43:(2026), pp. 1-14. [10.1016/j.cacaie.2026.100032]

Availability:
This version is available at: 11583/3012164 since: 2026-06-17T15:08:16Z

Publisher:
Elsevier

Published
DOI:10.1016/j.cacaie.2026.100032

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

06 September 2026

Computer-Aided Civil and Infrastructure Engineering 43 (2026) 100032

Contents lists available at ScienceDirect

Computer-Aided Civil and Infrastructure Engineering

journal homepage: www.sciencedirect.com/journal/computer-aided-civil-and-infrastructure-engineering

Research article

Maintenance-oriented tunnel digital model generation via panoptic segmentation of ultra-high-resolution images

Zehao Ye [a,b,1], Qi Li [c,1], Giuseppe Desiderio [d], Mengqi Huang [e], Wenxi Liu [c], Valentina Villa [d], Jelena Ninić [a,*]

[a] Department of Engineering, Durham University, Durham, UK
[b] Department of Civil Engineering, University of Birmingham, Birmingham, UK
[c] Department of Computer Science, Fuzhou University, Fuzhou, China
[d] Department of Civil Engineering, Politecnico di Torino, Torino, Italy
[e] Department of Civil and Construction Engineering, Swinburne University of Technology, Melbourne, Australia

ARTICLE INFO

Keywords:
Ultra-high-resolution segmentation
As-damaged BIM
Tunnel monitoring
Defect detection
Panoptic segmentation
Segment anything model 2

ABSTRACT

Ultra-high-resolution (UHR) panoramic imaging enables detailed capture of tunnel surface conditions. Nevertheless, damage detection and reporting still rely on manual annotations after the inspection, as existing automated algorithms typically require downsampling (losing fine details) or patch-based processing (losing global context) to handle the massive computational load of UHR data. This trade-off restricts the effective use of such high-fidelity data, leaving comprehensive reporting reliant on manual annotation. To address this gap, a novel framework has been proposed that directly operates on UHR panoramic images for automated damage detection and 3D reconstruction employing Building Information Modelling to create digital model with annotated defects. At its core is a flexible segmentation architecture with a side network that enables context extraction from larger image patches, supporting panoptic segmentation of images over 6K resolution and accurate detection of tunnel components and five main damage types. With the Segment Anything Model 2 backbone, performance further improves, raising panoptic quality from 53.14 to 56.99. Industry Foundation Classes open data format has been expanded to support the standardized and interoperable representation of damaged tunnels, based on which an as-damaged BIM model is constructed to effectively record and visualize inspection outcomes and quantitative significance levels, thereby enhancing the efficiency of tunnel monitoring and management. The source code is publicly accessible at https://github.com/zxy239/UHR-segmentation-for-tunnel.

1. Introduction

Tunnels are a critical component of transport infrastructure, serving both road and railway systems and enabling connectivity across challenging terrain and within dense urban environments (Ye et al., 2026). Tunnels contribute significantly to transport development by reducing travel distances and construction costs while minimizing negative environmental impacts (Chen et al., 2025). As a result, the tunnelling market is expanding at an annual rate of approximately 9%, 2.5 times faster than the overall growth of the global construction industry (ITA-AITES, 2019). Europe has an extensive and long-serving tunnel network now facing ageing-related rehabilitation, further exacerbated by climate-induced extreme weather such as heavy rainfall and flooding (Schade et al., 2020). Across the Trans-European Transport Network, Italy hosts nearly half of the system's tunnel mileage, a significant portion of which has been operating for more than three decades. Yet only about 19% of these tunnels fulfil the minimum regulatory safety requirements, prompting the report (MIT, 2023) to call for preventive maintenance, early risk mitigation, and the use of continuous monitoring together with modern inspection technologies.

Panoramic imaging and point cloud acquisition have become the typical advanced surveying techniques for documenting tunnel surfaces in recent years (Lin et al., 2024, 2025; Ye et al., 2025; Zhu et al., 2016). These methods enable efficient acquisition of information about tunnel surface conditions with high fidelity and completeness, capturing fine-grained details while preserving the overall structural context, and providing consistent, continuous records for reliable analysis and mapping.

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

[1] Equal contribution.

https://doi.org/10.1016/j.cacaie.2026.100032
Received 9 December 2025; Accepted 18 February 2026
Available online 21 April 2026
1093-9687/© 2026 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Such comprehensive documentation lays the foundation for subsequent tasks such as Structural Health Monitoring (SHM), deformation analysis, and digital twin construction (Babanagar et al., 2025; Ye et al., 2025; Zhu et al., 2025). However, processing this massive volume of UHR data presents a significant computational challenge. While deep learning-based detection has advanced rapidly, standard algorithms struggle to handle 4K or higher resolutions directly due to memory constraints. Common workarounds, such as downsampling (Foria et al., 2022), inevitably erode the visibility of micro-defects, while simple patch-cropping disrupts the semantic continuity of large-scale damage (Xu et al., 2025). Due to these technical bottlenecks, high-fidelity damage detection and risk evaluation remain manual and subjective in practice, making the process time-consuming, labour-intensive, and prone to inconsistencies. Studies have reported a high level of subjectivity with 16–42% variation between inspectors when assessing the same visual features (Kirillov et al., 2019), which might significantly influence tunnel condition evaluation. Consequently, such reliance can cause delays and extra costs in maintenance planning, compromising timely safety interventions (Huang et al., 2021).

Deep learning-based image analysis provides an effective means for detecting structural surface damage and addressing this challenge (Cha et al., 2017; Chai et al., 2025; Jiang et al., 2024; Liang, 2019; Ouyang et al., 2023, 2025; Spencer Jr et al., 2025; Ye et al., 2024; Zhu et al., 2024). Despite these advances, deploying such techniques in real-world tunnel environments, especially in ageing and deteriorated infrastructure, remains highly challenging. Occlusions caused by maintenance staff, inspection vehicles and Mechanical, Electrical and Plumbing (MEP) installations, such as cable trays, lighting fixtures and ventilation ducts, together with complex surface textures, variable lighting conditions, and the coexistence of multiple damage types significantly complicate automated detection (Ye et al., 2026). Therefore, a comprehensive framework is needed to interpret panoramic tunnel images and efficiently manage detected components and damage.

Hence, the first practical challenge is that tunnel surfaces are typically documented as UHR panoramic images, exceeding 2K or even 4K in resolution (Liu et al., 2024). While these images provide rich and detailed visual information, their large size introduces additional computational and methodological challenges (Ouyang et al., 2025), especially considering that conventional segmentation algorithms typically operate on images smaller than 1K. Effectively processing these images requires not only robust feature extraction and segmentation strategies but also the ability to maintain accuracy across the entire spatial extent of the tunnel, ensuring that both subtle defects and extensive damage are reliably detected. Previous approaches have included direct downsampling (Foria et al., 2022), inference after simple cropping (Ouyang et al., 2023), or cropping with high overlap, followed by combining the predictions after multiplying by a fixed weighted matrix (Chatterjee & Poullis, 2021). These methods either excessively lose fine details and contextual information or increase computational costs. Consequently, several segmentation methods (Cheng et al., 2020; Li et al., 2024) have been developed for UHR images, but most remain highly specialized and are not easily adaptable to stronger Vision Foundation Models (VFMs), limiting their generalizability. In particular, with the recent emergence of VFMs such as SAM (Kirillov et al., 2023), SAM2 (Ravi et al., 2024), and DINOv3 (Siméoni et al., 2025) pretrained on massive datasets, it has been demonstrated that transferring their backbone knowledge and applying further task-specific training can substantially improve models (Ge et al., 2024; Ye et al., 2024).

Another important challenge concerns the effective management and documentation of detected damage information (Ouyang et al., 2025). Building Information Modeling (BIM) is a well-established approach for the digital management of a building's structural and operational features, providing a unified information model that can be used throughout the project lifecycle (Bui et al., 2024; Fang et al., 2025; Ninić et al., 2024). Within this context, the Industry Foundation Class (IFC) standard serves as the backbone of BIM data exchange, offering an open and interoperable schema for representing physical and functional attributes of infrastructure assets (buildingSMART International, 2024). Prior studies (Alsharif et al., 2025; Tan et al., 2022; Zeibak-Shini et al., 2016) integrated BIM into various aspects of infrastructure management to enhance information efficiency, with applications extending to risk management. Maharjan et al. (2025) developed a domain-adaptive self-supervised learning framework for corrosion detection in steel tunnels, mapping 2D inspection results into 3D BIM via Structure-from-Motion and Multi-View Stereo (SfM-MVS) to facilitate digital-twin-based maintenance. Accordingly, integrating rich semantics, in terms of both damage elements and categorical information, from UHR tunnel image into BIM is a promising way for management.

Additionally, although tunnel linings are relatively uniform, multiple damage types and interference from auxiliary structures (e.g., MEP, signs, steel reinforcement mesh) complicate detection. Because assessment is required to be performed at the instance level (MIT, 2023), panoptic segmentation was adopted to obtain both semantic and instance information. Unlike semantic segmentation, panoptic segmentation is instance-aware and can recover complete instances under partial occlusion. Secondly, compared with instance segmentation, it also recognises stuff categories (amorphous regions, such as auxiliary structures inside tunnel) by distinguishing them from damage instances (things) (Kirillov et al., 2019). Non-damage elements such as continuous tubes or reinforcing mesh, though not individual instances, provide important context and are also reconstructed in BIM model to improve scene understanding.

In summary, three main research gaps remain in tunnel damage detection from UHR images: i) Resolution Gap: conventional segmentation models (standard input in ~1K) cannot accommodate the high resolution of modern tunnel imagery. ii) Limited Adaptability: existing UHR-oriented segmentation methods often adopt highly specialized architectures that limit scalability and complicate integration with VFMs. iii) Lack of BIM Integration: detected defects are not systematically documented and integrated into BIM for tunnel asset management.

To address these gaps, a generalized algorithmic framework was proposed, which is designed to directly process UHR tunnel panoramic images for panoptic segmentation. This approach enables comprehensive damage detection, automated condition evaluation, and subsequent data integration for asset management within BIM. The fully automated process classifies all pixels into five damage categories and auxiliary structures, providing instance-level outputs aligned with risk assessment requirements. The proposed generalized framework is a dual-branch network, which includes a main network consistent with conventional segmentation algorithms, a lightweight side network dedicated to extracting contextual information, and a context-guided module. Therefore, SAM2 (Ravi et al., 2024) can be easily integrated as the backbone to enhance segmentation performance in our framework. The final result is an IFC-based 3D as-damaged tunnel BIM model, incorporating auxiliary structures, embedded damage information, and corresponding risk evaluations.

2. Related work

2.1. UHR segmentation algorithms

Currently, mainstream approaches to UHR segmentation can be categorized into two paradigms. The first paradigm involves directly downsampling the UHR image and applying conventional segmentation models (Cheng et al., 2022; Xie et al., 2021) for global inference, followed by upsampling the segmentation result back to the original image resolution. While this approach reduces computational overhead, it often results in a considerable reduction of detailed information. To alleviate this problem, we take cues from efficient semantic segmentation (Fan et al., 2021; Yu et al., 2021). Several works (Guo et al., 2022; Ji et al., 2023; Li et al., 2024, 2025; Qin et al., 2025) have extended this method by introducing dual-stream networks, comprising deep and shallow branches. Specifically, the downsampled image is processed

through the deep branch to capture global semantics while the original UHR image is simultaneously fed into the lightweight shallow branch to preserve spatial details. These two complementary feature streams are then fused to produce the final segmentation result. However, this paradigm still involves a trade-off between performance and computational efficiency, rendering it ineffective for real-world UHR image processing.

The second paradigm divides the original image into smaller local patches, performs segmentation on each patch independently using conventional models, and then aggregates the results. This approach effectively preserves fine details and reduces GPU memory consumption. However, due to the lack of global context, it struggles with recognizing large-scale objects. To address this limitation, recent efforts (Chen et al., 2019; Huynh et al., 2021; Li et al., 2021; Liu et al., 2024) have introduced additional contextual inputs, such as surrounding regions, into each local patch, thereby enriching local predictions with contextual cues and alleviating the limited receptive field inherent in local inference. This strategy significantly enhances object recognition and overall segmentation accuracy. Nevertheless, the inclusion of extensive contextual information and the intricate interaction mechanisms between local patches and contextual regions often results in overly complex architectures, leading to a substantial increase in model parameters and computational cost.

Additionally, some algorithms (Cheng et al., 2020; Shen et al., 2022) refine segmentation outputs through post-processing models, but they require careful parameter tuning and complex training pipelines, limiting their practical deployment. Meanwhile, most existing UHR segmentation models (e.g., ISDNet (Guo et al., 2022), WSDNet (Ji et al., 2023), RUE (Qin et al., 2025)) are highly specialized, lacking generalizability and scalability, hindering their effective integration with emerging VFMs to further improve performance.

2.2. As-damaged BIM application in tunnels

The integration of BIM is now well-established across the planning, design, and construction phases of projects, offering proven benefits in collaboration, coordination, and cost management (Sacks et al., 2018). In tunnel engineering, BIM applications have progressed from early-stage functions such as 3D geological modelling and segment collision detection to comprehensive lifecycle management (Gall et al., 2025, 2026; Huang et al., 2023; Sanfilippo et al., 2025). Moreover, its role has evolved from traditional *as-built* documentation to refined and dynamic asset management, where static models are transformed into *as-damaged* or *as-is* representations that integrate and visualize structural health information (Sun et al., 2025; Tan et al., 2022; Teng et al., 2022; Zeibak-Shini et al., 2016), supporting more intelligent and data-driven maintenance decision-making (Huang et al., 2021). These as-damaged BIM applications are summarized into three aspects: sensor information management, detected damage information recording, and the digital documentation of the structure itself.

Sensor information management integrates live data from SHM systems directly into tunnel BIM. Tunnels are equipped with numerous sensors that measure stress, movement, and leaks (Gómez et al., 2020; Yin et al., 2020; Zhang et al., 2022; Zhu et al., 2024), often complemented by advanced non-destructive evaluation techniques such as infrared thermography for concrete defect detection (Sirca Jr & Adeli, 2018). To enhance signal interpretation accuracy, these inputs are frequently coupled with deep learning algorithms (Rosso et al., 2024). By mapping these onto BIM, stakeholders can visually pinpoint where stresses or deformations are occurring, transforming the static 3D model into a dynamic “Digital Shadow” or “Digital Twin” that mirrors the tunnel’s real-world condition. This data-rich environment supports advanced diagnostics, ranging from global and local health assessments (Rafiei & Adeli, 2018) to precise structural response predictions using Bayesian-trained recurrent neural networks (Perez-Ramirez et al., 2019). Ultimately, such integrated systems provide a powerful tool for predictive maintenance and risk alerts (Zhang et al., 2024; Zhu et al., 2024), aligning with the broader vision of smart cities by consolidating structural control, monitoring, and energy harvesting (Javadinasab Hormozabad et al., 2021).

As for detected damage information recording, although wide research has been conducted on damage detection, studies that integrate detected results into BIM models remain limited in the tunnel context. Relevant approaches in other domains, such as buildings and bridges, provide useful references. For instance, based on the IFC standard, a Damage Information Model (DIM) has been developed to represent bridge damage and support data exchange between BIM and finite element (FE) environments, enabling automated assessment and visualization of residual capacity in deteriorated reinforced concrete bridges (Alsharif et al., 2025). Tan et al. (2022) used custom parametric families to generate polylines mapping façade cracks onto BIM models, while Sun et al. (2025) employed Mask R-CNN with camera pose information to project detected damages onto bridge components within simplified BIM models. Both methods, however, are limited to planar surfaces. Similarly, Artus et al. (2022) detected spalling on bridge point clouds, triangulated the surrounding points, aligned them with the BIM model using the GoICP algorithm, and subtracted the spalling mesh from the main girder to construct an as-damaged BIM model. However, in severely deteriorated tunnels, it can be difficult to differentiate spalling from deformation in point clouds, as structural displacement may also occur. Moreover, curved tunnel surfaces make simple 2D projection methods largely ineffective, and the lack of a standardized open data format further limits practical application.

The last aspect concerns digital documentation. As-damaged models with projected defects provide clear visual maps that support repair planning, cost estimation, and monitoring of damage progression. Beyond damage mapping, several studies (Kerle et al., 2019; Zeibak-Shini et al., 2016) have also explored the application of Scan-to-BIM techniques to severely damaged or deteriorated structures to reconstruct as-damaged BIM models at the component level. Additionally, digital documentation (Gao et al., 2015; Tang et al., 2010) also serves as an important management approach for ageing structures, especially for HBIM (Heritage BIM) (Yang et al., 2020). The integration of both approaches can be regarded as a more comprehensive form of digital documentation. BIM plays an increasingly important role in tunnel management, particularly for aged tunnels where accurate drawings may be incomplete or unavailable. Mobile laser scanning enables precise as-is 3D reconstruction, and inspection-identified defects can be systematically recorded to form a digital health record (Guo et al., 2024; Rimella et al., 2025). However, a fully integrated workflow that consolidates digital documentation and semantic information without redundancy is still lacking.

In summary, integrating real-world damage information into tunnel BIM models represents a crucial step toward data-informed, systematic, and automation-driven optimization of infrastructure management. Although significant progress has been made in data visualization and defect detection across various fields, considerable challenges remain in tunnel applications. These challenges include developing effective methods for processing UHR scanned images and automatically linking detected defects and components to their corresponding BIM elements. This study aims to address these challenges.

3. Methodology

3.1. Overview

In this section, a detailed description of proposed approach is presented to address the identified challenges as shown in Fig. 1, including: (a) the development of a general algorithmic framework for processing UHR tunnel images (Section 3.2), and (b) the development and generation of IFC-based as-damaged tunnel BIM via projection of detected results (Section 3.3). This approach systematically bridges the gap between 2D images and 3D BIM environment.

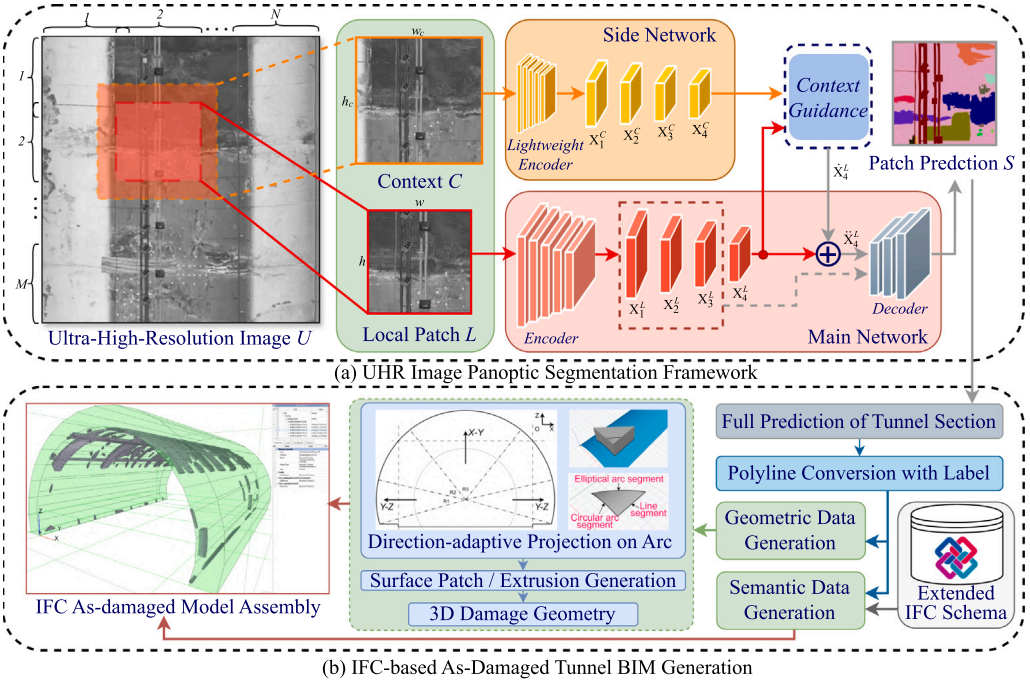

Fig. 1. Proposed UHR segmentation framework and IFC-based as-damaged tunnel BIM generation.

Panoptic segmentation enables comprehensive segmentation of tunnel damages (things) and auxiliary components (stuff) in a single step, which makes it well suited for tunnel recognition. Hence, panoptic segmentation has been selected as task. To enable direct processing of UHR images, a dual-branch network was proposed, where a simple branch is dedicated to extracting contextual information that is subsequently incorporated into the main network via a context guidance module for panoptic segmentation. Next, the detected 2D masks are converted into labelled closed polylines, whose vertices are direction-adaptively projected onto the 3D tunnel surface and connected along the curved tunnel geometry to generate the damage representations. These representations are then integrated into a dynamic, data-rich as-damaged BIM through our extended IFC schema tailored to represent tunnels' as-damaged condition, enabling intuitive visualization of the tunnel's structure.

3.2. Proposed UHR image segmentation framework

The proposed UHR image segmentation framework comprises three stages, shown in Fig. 1. Initially, the UHR panoramic image U of dimensions $W \times H$ is uniformly partitioned into $N \times M$ overlapping local patches, denoted as L_{ij} of size $w \times h$, arranged along the horizontal and vertical axes. Here, the indices i and j correspond to the row and column indices of the patches, respectively, with $i \in \{1, 2, \dots, N\}$ and $j \in \{1, 2, \dots, M\}$. Second, the proposed model performs panoptic segmentation on each local patch. Third, all predicted local results are placed back into their corresponding positions to form the final ultra-high-resolution panoptic segmentation map.

Compared to the full panoramic image, individual local patches have a limited field of view and often lack essential contextual cues. This restriction impairs the model's ability to achieve a holistic semantic understanding, potentially leading to inaccurate segmentation or incomplete representation of structural details. Moreover, it may cause inconsistent predictions for adjacent patches, assigning different damage types to the same region. To address this issue, in the first stage, extended regions centred on each local patch are simultaneously extracted and incorporated into the segmentation model to provide additional contextual information. As shown in Fig. 1, given a cropped local patch L, its context C is not smaller than L and covers L, which has the width w_c and height h_c ($w_c \geq w$, $h_c \geq h$). Specifically, when C is beyond the UHR panoramic image, the area beyond is filled with zeros. To improve the segmentation of L, local patch information is associated with corresponding contextual information. C is first downsampled to align spatially with L for computational efficiency, then both components are simultaneously input into the proposed model. This dual-stream input preserves detailed local features while maintaining context awareness.

The proposed panoptic segmentation framework consists of two networks: a **main network** and a **side network**. In the main network, each L undergoes initial processing via a general-purpose visual encoder to generate multi-level feature representations. These features, denoted as $X_i^L \in \mathbb{R}^{h_i \times w_i \times c_i}$, $i \in \{1, 2, 3, 4\}$, capture hierarchical visual information necessary for fine-grained understanding. The extracted features, after being guided and enhanced by contextual information, are then passed to a standard panoptic segmentation decoder, such as Mask2Former (Cheng et al., 2022), which produces the final panoptic prediction. In the side network, the corresponding C is processed by a lightweight encoder to extract multi-scale contextual features. These features, denoted as $X_i^C \in \mathbb{R}^{h_i \times w_i \times c_i'}$, $i \in \{1, 2, 3, 4\}$, introduce rich contextual information while incurring only minimal computational overhead. No additional supervision is introduced for the side network. This auxiliary context enhances the edge features of each L and contributes to improved segmentation accuracy.

Empirically, the deepest feature map among the extracted multi-scale features contains the most semantically meaningful information and is thus exclusively utilized to enhance the local patch representation (Liu et al., 2024). Specifically, the local features X_4^L and contextual features X_4^C are fed into the context guidance module. To facilitate effective feature interaction, the contextual features are first projected to match the channel dimensionality of the local features via a linear projection layer.

Both feature maps are then reshaped and linearly projected into a unified token sequence format of size $N_{tokens} \times C$. A multi-head deformable attention mechanism is subsequently applied to establish fine-grained, non-local correspondences, where the local features serve as the queries (Q) and the contextual features act as both keys (K) and values (V):

$$\begin{aligned}\dot{X}_4^L &= \text{DeformAttn}(Q, K, V) \\ &= \text{DeformAttn}(Q_{X_4^L}, K_{X_4^C}, V_{X_4^C})\end{aligned} \tag{1}$$

Through this sparse attention mechanism, each local region adaptively aggregates the most semantically relevant contextual information while suppressing redundant cues. Following the context guidance, the context-enhanced local features are reshaped back to dimensions $h_4 \times w_4 \times c_4$ and fused with the original local features using a residual connection.

$$\ddot{X}_4^L = X_4^L + \gamma \dot{X}_4^L \tag{2}$$

where γ is a learnable weighting parameter. Finally, the enhanced local features $\ddot{X}_4^L$, together with the original multi-scale features $X_i^L, i \in \{1, 2, 3\}$ are input to the decoder in the main network to produce local panoptic segmentation map S.

In the final stage, all local predictions S are assembled into a full panoptic map. Each patch is placed in its spatial position from left to right and top to bottom, and in overlapping regions existing labels are preserved to avoid overwriting. Overlaps are used only for propagating instance IDs: when overlapping pixels share the same class, their instance IDs are merged. This prevents duplicate or fragmented instances and ensures consistent, unique instance segmentation across the panoramic image, resulting in a complete UHR panoptic map.

3.3. As-damaged tunnel digital model generation

BIM models incorporate both **geometric** and **semantic data** to describe components and their relationships (Borrmann et al., 2018). This study considers five visually identifiable and distinguishable damage types in UHR tunnel images: seepage, cracks, spalling, damaged joints, and corrosion. The detailed modelling process is described below.

3.3.1. Geometric data generation

After obtaining the complete UHR panoptic segmentation map, the first step is to convert the masks from the map into closed 2D polylines represented by continuous coordinates. These polylines also include class and instance labels decoded from the panoptic segmentation map. For interrupted masks with the same instance ID, the segments are merged using their outer envelope, forming a single, unified instance. Building on this representation, the vertices of the transformed 2D closed polylines are subsequently projected into 3D coordinates. We consider a tunnel aligned such that the Z-axis corresponds to the vertical direction, the Y-axis follows the tunnel's longitudinal direction, and the X-axis points to the right, as shown in "Direction-adaptive Projection on Arc" in Fig. 1(b). The tunnel cross-section is represented as a sequence of N connected curve segments, where each arc R_i has its own centre $(X_{c,i}, Z_{c,i})$ and radius R_i, for $i = 1, 2, \ldots, N$. While some segments may be straight lines, these can be equivalently treated as circular arcs with $R_i \to \infty$.

A 2D closed polyline representing a damage instance is denoted as

$$\mathcal{P} = \{(x_j, y_j)\}_{j=1}^{M}, \tag{3}$$

where j serves as the vertex index of damage polyline and M is the total number of vertices. The coordinates x_j and y_j correspond to the tunnel's transverse and longitudinal (Y) directions, respectively. The objective is to project each vertex (x_j, y_j) onto the corresponding 3D tunnel surface. For a vertex (x_j, y_j) located on the i-th circular arc R_i with centre $(X_{c,i}, Z_{c,i})$, the 3D projection is given by:

$$\begin{aligned}X_j^{3D} &= X_{c,i} + R_i \frac{x_j - X_{c,i}}{\sqrt{(x_j - X_{c,i})^2 + (Z_j - Z_{c,i})^2}}, \\ Y_j^{3D} &= y_j, \\ Z_j^{3D} &= Z_{c,i} + R_i \frac{Z_j - Z_{c,i}}{\sqrt{(x_j - X_{c,i})^2 + (Z_j - Z_{c,i})^2}},\end{aligned} \tag{4}$$

where $(X_j^{3D}, Y_j^{3D}, Z_j^{3D})$ is the projected 3D coordinate. If two consecutive polyline vertices (x_j, y_j) and (x_{j+1}, y_{j+1}) lie on different arcs R_i and R_{i+1} with distinct centres, two intermediate vertices $(x_j^{*,i}, y_j^{*,i})$ and $(x_j^{*,i+1}, y_j^{*,i+1})$ are inserted at slightly extended positions on either side of the boundary between the arcs. This ensures that the resulting local solids for R_i and R_{i+1} overlap, preserving continuity and facilitating their merging. The 3D coordinates of each intermediate vertex are projected according to the arc it belongs to. Formally, the set of projected 3D vertices is:

$$\mathcal{P}^{3D} = \{(X_j^{3D}, Y_j^{3D}, Z_j^{3D})\}_{j=1}^{M'}, \tag{5}$$

where $M' \geq M$ includes both the original vertices and any inserted intersection points. This procedure guarantees a continuous and geometrically consistent mapping of 2D damage instances onto the 3D tunnel surface.

The next step is to generate 3D damage instances conforming to the tunnel geometry. Each instance is first processed within each circular R_i of the tunnel cross-section and then merged into a single 3D instance. For each pair of consecutive 3D vertices $(\mathbf{V}_j, \mathbf{V}_{j+1})$, a local plane (cutting plane) Π_j is established. The intersection of Π_j with the tunnel surface S defines a spatial curve C_j:

$$C_j = \Pi_j \cap S. \tag{6}$$

Since a plane through two points is not unique, and tunnel cross-sections may exceed 180°, an arbitrary plane may intersect the surface in two curves. To simplify computation and maintain consistency, we restrict planes to two orientations, as shown in "Direction-adaptive Projection on Arc" in Fig. 1(b):

1. Perpendicular to the X-Y plane, generating curves along the tunnel cross-section.
2. Perpendicular to the Y-Z plane, with its cutting region restricted to avoid intersecting the opposite tunnel wall.

The boundaries of the two projection cutting regions can be aligned with a radial division line where R_i changes, or with any other user-defined boundary, as long as it prevents redundant intersection. Although the most conservative approach would be to enforce that the cutting plane always passes through the centre of the corresponding circular arc R_i, this would require repeated angle calculations and increase computational cost. The direction-adaptive projection balances accuracy and efficiency. Then, the spatial intersection curves C_j fall into three fundamental types:

1. **Line segment:** plane aligned with the tunnel longitudinal direction (Y).
2. **Circular arc segment:** plane parallel to the cross-section (X-Z).
3. **Elliptical arc segment:** all other orientations.

and no other curve types occur due to the tunnel being composed of circular arcs and straight segments.

Each damage instance is processed separately within its corresponding circular segments R_i. For each R_i, a closed polyline is formed by the

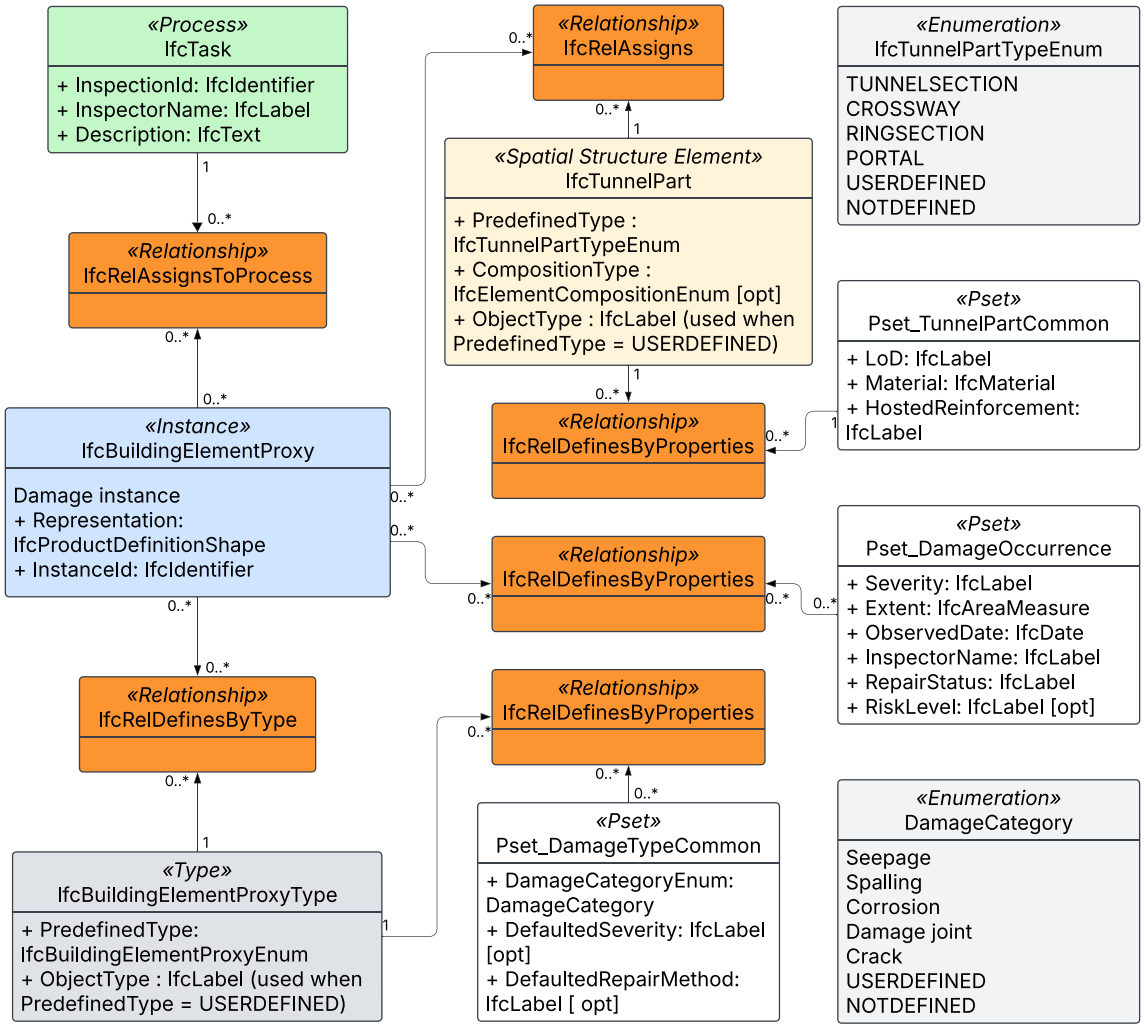

Fig. 2. UML class diagram for information model of tunnel inspection and damage representation.

union of consecutive spatial curves $\bigcup C_j$, which connects the damage vertices to define a local surface patch $S_{\text{local},i}$. This patch is sampled using Poisson sampling, refined with Lloyd relaxation for a more uniform distribution, and then extruded outward by an offset h. The final 3D damage instance S_{damage} is obtained by:

$$S_{\text{damage}} = \bigcup_i \mathcal{V}_{\text{local},i} = \bigcup_i \text{Extrude}\Big(\bigcup_j C_j, h\Big), \tag{7}$$

where $i \in \{1, \ldots, N\}$ is the index of the circular segments R_i, and $j \in \{1, \ldots, M_i\}$ is the index of the vertices belonging to the i-th segment. The inner union $\bigcup_j$ topologically connects the M_i vertices to form the closed boundary for each $S_{\text{local},i}$, while the outer union $\bigcup_i$ merges all local patches across the N segments into the final continuous 3D geometric damage instance. The parameter h denotes the radial offset and extrusion thickness. Seepage is represented as a surface element, while other damage types are modelled as volumetric voids. Components inside the tunnel are reconstructed similarly using their panoptic segmentation masks and predefined BIM categories.

3.3.2. Semantic data generation

The IFC standard, developed by buildingSMART, has been increasingly adopted for exchanging and visualising both geometric and semantic information related to the visual-degradation of infrastructure (Alsharif et al., 2025; Bellon et al., 2025; Isailović et al., 2020). This vendor-neutral standard has been adopted to reduce data loss and support future inspections. The digitisation of damage segmentation results is implemented using the IFC format, establishing semantic relationships for tunnel inspection and damage (Fig. 2). This Unified Modelling Language (UML) class diagram presents a dynamic extension of the *IFC 4x3 Add2* schema (buildingSMART International, 2024) for representing inspection processes, tunnel components, and associated damage information in tunnel asset management. The dynamic extension is achieved through the Property Set (*Pset*) mechanism (Borrmann et al., 2018), which enables the independent definition of domain-specific properties, such as those for the damage domain, without schema changes.

The UML diagram begins with the inspection workflow. The *IfcTask* entity, representing each inspection activity and recording metadata such as the inspector, inspection ID, and description, is associated with the observed damage elements through the relationship *IfcRelAssignsToProcess*, establishing a semantic link between the inspection process and the damaged components it identifies. At the instance level, individual damage occurrences are represented using *IfcBuildingElementProxy*, which provides a flexible placeholder for elements (in this case, damages) not explicitly defined in the IFC schema. Each proxy instance captures geometric and semantic information for a specific damage observation, including its 3D representation (via *IfcProductDefinitionShape*) and related quantitative attributes such as severity, extent, observed date, inspector name, and repair

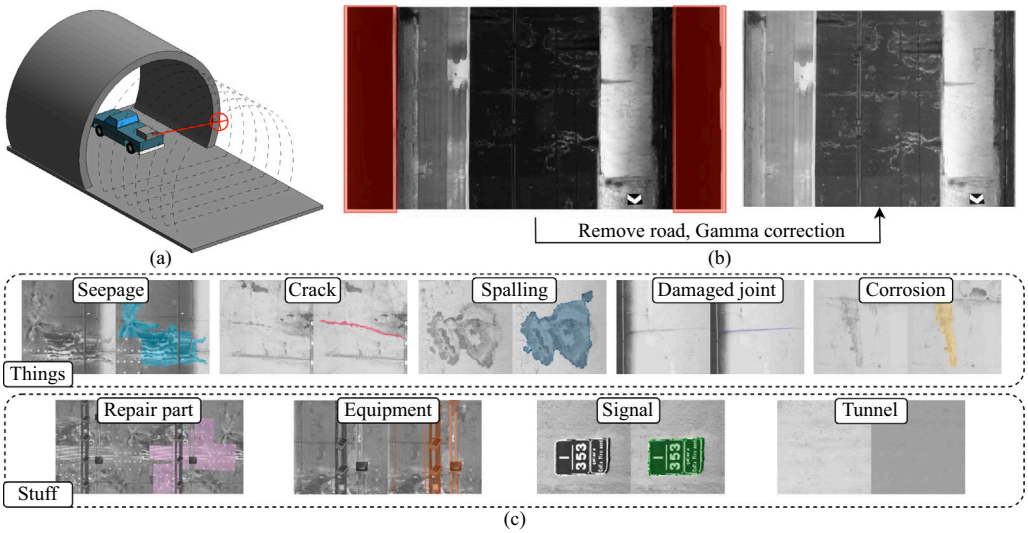

Fig. 3. Data collection: (a) A vehicle-mounted scanner captures tunnel point cloud data. (b) The point cloud is converted into a tunnel panorama, with the road removed and gamma correction applied. (c) Typical annotated "Things" and "Stuff".

status through the relationship *IfcRelDefinesByProperties*, referencing the *Pset_DamageOccurrence* property set.

Reusable definitions of damage categories are represented using *IfcBuildingElementProxyType*, following the IFC schema by linking type objects to shared properties through *IfcRelDefinesByProperties* and the *Pset_DamageTypeCommon* property set. This property set includes attributes such as Damage Category Enumeration, Defaulted Severity (optional), and Defaulted Repair Method (optional), with *DamageCategoryEnum* enumerating the primary five damage categories observed in the study.

The *IfcBuildingElementProxy* instances reference their corresponding *IfcBuildingElementProxyType* definitions so that similar damage occurrences share the same classification and property set. They can also be spatially related to *IfcTunnelPart*, a concrete subtype of *IfcFacilityPart*, which represents discrete physical and functional components of the tunnel (e.g., Tunnel Section, Portal, Crossway, Ring Section). This linkage provides a complete semantic and spatial context for damage representation within tunnel asset models. The related attributes such as the level of detail (LoD), material, and reinforcement status are included through the relationship *IfcRelDefinesByProperties*, referencing the *Pset_TunnelPartCommon* property set.

Two types of geometric forms have been implemented for damage feature modelling, with seepage realised with poly-surfaces and the others as created by volumetric additions. The entity *IfcProductDefinitionShape* is used to represent the geometry, employing entities such as *IfcTessellatedFaceSet* for complex polygonal shapes. Collectively, these extensions provide a standardized and interoperable data structure for digitally representing and managing tunnel damage in BIM.

4. Experiments and results

4.1. Data collection and dataset creation

In this study, a vehicle-mounted laser scanner traveling at approximately 5 km/h was used to acquire 3D point clouds of the tunnel (Fig. 3(a)). These point clouds were geometrically corrected and registered using fixed markers, then projected to generate UHR panoramic tunnel unfold images. The final output consisted of 8-bit grayscale *.tif* files containing laser intensity values, which were subsequently processed with gamma correction (Fig. 3(b)). For efficient annotation, each 20-meter tunnel segment was split into 6 smaller images, annotated using a SAM (Segment Anything Model)-based tool (Ji & Zhang, 2024; Kirillov et al., 2023), and then merged to form the complete annotation. The annotations were based on on-site inspection reports from engineers, with detailed labelling performed by a trained expert (cracks were manually annotated) and a final review conducted by another trained expert. Further details are available in (Mozafarian et al., 2025; Ye et al., 2026).

Table 1
Data distribution of training and validation set based on UHR image.

Class	Type	Instance count (train/val)	Pixel count (Million) (train/val)	Pixel ratio (%) (train/val)
Seepage	Things	2245/403	124/43	5.99/18.51
Corrosion	Things	364/51	39/9	1.90/3.81
Spalling	Things	1288/162	16/2	0.75/0.78
Damaged joint	Things	328/29	7/0.5	0.33/0.19
Crack	Things	191/33	0.9/0.1	0.04/0.06
Tunnel	Stuff	45/5	1,723/161	82.83/68.60
Equipment	Stuff	45/5	91/10	4.40/4.57
Repair part	Stuff	35/4	75/7	3.63/3.32
Signal	Stuff	43/5	2/0.3	0.14/0.16

As illustrated in Fig. 3(c), five damage types are annotated as "things", with contiguous regions of the same damage treated as single instances even under partial occlusion (e.g., seepage behind pipes). Auxiliaries such as steel mesh, lighting, pipes, signals, and intact lining are categorized as "stuff", since instance identification is not required or appropriate for these objects. Although some are countable, grouping all non-damage elements as "stuff" allows clear and intuitive instance-level damage evaluation. After sub-image merging, instances spanning multiple regions were unified. Two tunnels were annotated, producing 50 images at $\sim 6K \times 6K$ resolution, with 5 for validation and the rest for training. The data distribution is shown in Table 1. For stuff, only one ID is used for each image.

4.2. Evaluation method

The evaluation of the proposed UHR panoptic segmentation approach follows the standard metrics defined by Kirillov et al. (2019): Segmentation Quality (SQ), Recognition Quality (RQ), and Panoptic Quality (PQ).

Segmentation Quality (SQ) assesses the geometric precision of the predicted segments, independent of detection errors. It is defined as the

Table 2
Summary of network configurations, experimental setups, and abbreviation definitions in ablation study.

Component	Configuration	Notes
Main Encoder	Swin-L backbone (Liu et al., 2021); Hiera-L variant (Ryali et al., 2023) from SAM2 (Ravi et al., 2024)	"swin" denotes Swin-L backbone (197M parameters), which is widely used one; while "sam2" indicates the SAM2-L backbone (224M parameters), representing the latest VFM backbone.
Side Encoder	Transformer-based EfficientViT backbone (Liu et al., 2023); CNN-based STDCNet backbone (Fan et al., 2021)	"b0"–"b3" denote EfficientViT variants, "stdc1net" and "stdc2net" denote STDCNet variants. "2l" means guidance from last two layers of side network; others use only the final one.
Decoder	Mask2Former panoptic segmentation decoder (Cheng et al., 2022)	Widely adopted panoptic segmentation head.
Patch Extraction	1024×1024 resolution for random cropping; fixed cropping divides the full image into a 6×6 grid.	"fix" and "rd" refer to fixed and random cropping strategies for generating training inputs, respectively.
Context Size	Expanded by 128, 256, 512, 768, and 1024 pixels (12.5-100% of patch size)	"128", "256", etc., denote context expansion sizes used to evaluate contextual influence.
Baseline	Mask2Former (Cheng et al., 2022) without contextual guidance	Used for direct comparison without contextual expansion or side network input, labelled as "no context".

average Intersection over Union (IoU) calculated only for the matched segments (True Positives):

$$SQ = \frac{\sum_{(p,g)\in TP} \mathrm{IoU}(p,g)}{|TP|} \tag{8}$$

where p and g represent the predicted and ground-truth segments respectively, and TP denotes the set of true positive matches.

Recognition Quality (RQ) evaluates the instance detection performance. It penalizes both false positives (FP) and false negatives (FN):

$$RQ = \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{9}$$

Panoptic Quality (PQ) serves as the unified metric, computed as the product of SQ and RQ for each category. To obtain the final evaluation metric for the entire dataset, we compute the macro-average over all C semantic categories:

$$PQ = \frac{1}{|C|} \sum_{c\in C} (SQ_c \times RQ_c) \tag{10}$$

where PQ represents the final reported value used in our experiments.

4.3. Implementation details

All experiments were performed on NVIDIA A100. The main software setup consisted of PyTorch 2.0.1 and CUDA 11.7. The implementation is built on MMDetection, using data augmentation techniques from the official Mask2Former setup. The models were trained on a dual-GPU setup with a batch size of 2 per GPU, resulting in a total batch size of 4. The AdamW optimizer was employed with a cosine annealing learning rate that increased to 1e-4 in the first 250 iterations and then decayed to 1e-7. Mixed-precision training was applied throughout, proving effective without compromising accuracy. No gradient accumulation was employed.

Table 2 summarizes the network configurations and baseline. Repeatedly reading UHR images and extracting local patches with context during training is inefficient. Therefore, all UHR images were preprocessed to generate local patches and their corresponding contexts of various sizes in advance, reducing training time to about one-sixth on our dataset, and minimizing randomness in ablation studies. A fixed extraction strategy divides each 6K image into a 6×6 grid of non-overlapping patches with contexts, trained for 100 epochs determined empirically through testing (Ye et al., 2026). In the randomly sampled strategy, 3600 annotated patch–context pairs were randomly extracted from each UHR image and used only once during training, consistent with the fixed extraction strategy. In context guidance experiments, using the final layer alone and the last two layers were evaluated. During inference, patches overlapped by 25% to aid instance merging.

4.4. Results and ablation study

This section presents the experimental results, ablation studies, and comparative evaluations against existing benchmark methods, with the optimal iteration chosen based on PQ on the validation set. The experimental results show that incorporating contextual regions generally enhances performance, with EfficientViT-b2 achieving the best overall results among side-network configurations, improving PQ by 2.85%. The flexible framework design also enables seamless integration of the SAM2 backbone, leading to a further 1% improvement in PQ.

Table 3
Comparison of different sizes of context.

Model	PQ	SQ	RQ
No context-swin-rd	53.14	73.52	62.77
128-swin-b2-rd	53.68	78.30	64.42
256-swin-b2-rd	**55.99**	79.58	**67.14**
512-swin-b2-rd	54.10	79.38	64.51
768-swin-b2-rd	53.58	**79.93**	63.82
1024-swin-b2-rd	52.49	73.06	62.67

Table 4
Comparison of different side networks.

Model	Params	PQ	SQ	RQ
No context-swin-rd	–	53.14	73.52	62.77
256-swin-b0-rd	0.7M	52.17	78.46	62.81
256-swin-b1-rd	4M	53.32	**80.42**	63.02
256-swin-stdc1net-rd	5M	53.40	79.80	62.66
256-swin-stdc2net-rd	9M	53.63	72.28	64.17
256-swin-b2-rd	15M	**55.99**	79.58	**67.14**
256-swin-b2-2l-rd	15M	54.30	78.38	64.82
256-swin-b3-rd	39M	54.23	78.19	65.31

Specifically, in Table 3, the effects of using different context sizes were compared. It can be observed that expanding the patch by 256 pixels yielded the most effective balance in our dataset, resulting in improvements of 2.85 in PQ, 6.06 in SQ, and 4.37 in RQ compared to the no-context random cropping baseline. However, performance degraded as the context size continued to increase, with a clear decline at 1024 pixels, indicating that excessive context failed to provide useful guidance in this scenario. This performance drop stems from two factors: first, using a 1024-pixel context requires aggressive downsampling of the original patch, which blurs fine-grained details crucial for detection; second, distant context contributes little to the segmentation of the central patch, leading to diminishing returns. The optimal context size may vary across tasks and datasets, and thus should be tailored through empirical evaluation. For different side-networks shown in Table 4, the choice between Transformer-based and CNN-based architectures appears to

Table 5
Comparison of with/without context, including things(Th) and stuff(St).

Model	PQ	PQ^{St}	PQ^{Th}	SQ	SQ^{St}	SQ^{Th}	RQ	RQ^{St}	RQ^{Th}
No context-swin-rd	53.14	**84.13**	28.34	73.52	**88.19**	61.78	62.78	95.00	37.00
256-swin-b2-rd	**55.99**	83.95	**33.63**	**79.58**	87.84	72.98	**67.14**	95.00	**44.87**
No context-swin-fix	52.40	83.16	27.78	71.07	86.90	58.41	62.75	95.00	36.96
256-swin-b2-fix	53.62	83.37	29.82	79.51	87.15	**73.41**	63.73	95.00	38.71

Table 6
Comprehensive comparison of Swin and SAM2 backbones in terms of accuracy and efficiency.

Model	PQ	SQ	RQ	FPS	GFLOPs	Params(M)	Memory(MB)
No context-swin-rd	53.14	73.52	62.77	4.21	971	215	3225
256-swin-b2-rd	55.99	**79.58**	67.14	3.86	1010	237	3277
No context-sam2-rd	55.49	79.03	66.93	3.91	977	232	3365
256-sam2-b2-rd	**56.99**	78.94	**68.68**	3.60	1013	251	3365

have little impact on performance (e.g., EfficientViT-b1 vs. STDC1Net), while overall results remain correlated with model size, with larger models generally performing better, possibly because the difference between these two architectures is minor at smaller model sizes (Mehta & Rastegari, 2022). However, this trend does not always hold. Among the variants, EfficientViT-b2 achieves the best results, while further increasing the size to b3 (20% of the main network backbone's parameters), leads to performance degradation. Furthermore, applying context guidance only at the final layer proved to be the most effective configuration, as high-level features contain richer semantic information and are better suited to integrate contextual cues without disturbing low-level spatial details (Lin et al., 2017). We report the PQ, SQ, and RQ results for both things and stuff in Table 5. We can see that random cropping consistently outperforms fixed cropping, regardless of whether contextual information is included. For the no-context model, applying random cropping improves PQ from 52.40 to 53.14, yielding a 0.74% improvement. The improvement is more pronounced for models with context (e.g., from 53.62 to 55.99 for 256-swin-b2-fix vs. 256-swin-b2-rd, a gain of 2.37). The improvement of RQ is significant, while the marginal gain in SQ is primarily due to a "dilution effect". As RQ improves, the model detects more challenging instances with complex boundaries. These new samples typically yield lower IoU scores, which pull down the overall average SQ.

Results for the SAM2 backbone integration can be found in Table 6, which enhances performance both with and without contextual guidance, ultimately achieving the highest PQ of 56.99 among all tested models, representing an improvement of 3.85% over the baseline model without contextual guidance. The results regarding detection efficiency are also provided in Table 6. To ensure a consistent comparison, inference is measured on a per-local patch basis, with each patch sized 1024 $\times$ 1024 $\times$ 3. The testing setup uses Python 3.11.7, PyTorch 2.7.0, CUDA 12.6, and an NVIDIA A40 GPU. For Frames Per Second (FPS) and memory measurement, 100 inference runs are first performed as a warm-up, and the average is then calculated over 200 subsequent runs. It is observable that the incorporation of the lightweight side network incurs minimal computational cost.

We conducted an examination of the overlapping regions and their merging strategies, and the results are presented in Table 7. "First/Last" retains the result from the first/last inference in the overlapping region. "Max" selects the class with the highest confidence for each pixel, whereas "Mean" averages the class scores across all inferences. The results indicate that applying fusion strategies (Max or Mean) leads to more stable results than relying on a single inference, although the improvement over "First" is marginal. This is consistent with the principle of ensemble learning; therefore, we recommend using the Max strategy.

Table 7
Comparison of PQ across different merging strategies in overlapping regions.

Model	First	Last	Max	Mean
256-swin-b2-rd	56.99	56.44	57.03	56.99
256-sam2-b2-rd	55.99	55.37	56.26	56.16

Fig. 4 shows the class-wise PQ comparison of the best models (256-swin-b2-rd and 256-sam2-b2-rd) against the no-context baseline (rd). Overall, contextual guidance improves performance across most categories except "Signal", possibly due to randomness caused by its very

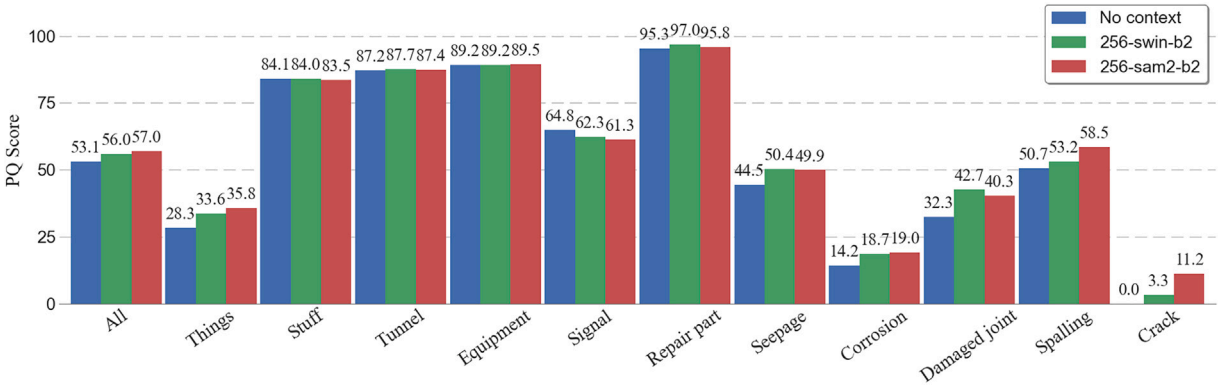

Fig. 4. Model performance comparison by PQ for each category.

Table 8
Comparison with leading UHR segmentation and panoptic segmentation models.

Model	PQ	SQ	RQ
No context-sam2-rd	55.49	79.03	66.93
256-sam2-b2-rd	**56.99**	78.94	**68.68**
FCtL++	55.41	78.50	66.58
RUE	53.10	73.20	62.65
No context-sam2 + PFPN-rd	53.84	**80.75**	63.62
256-sam2 + PFPN-b2-rd	54.69	80.24	64.84

limited pixel representation in the validation set. The primary advantage of SAM2 is its enhanced recognition of damage-related classes, particularly showing substantial gains in cracks.

We provide comparisons with leading methods specifically designed for UHR segmentation, including FCtL++ (Liu et al., 2024) and RUE (Qin et al., 2025). FCtL++ comprises a local network and two contextual networks, all employing identical yet independent backbone networks as encoders. It further incorporates two carefully designed feature fusion modules to enable deep interaction between local and contextual features. The core architecture of RUE utilizes a deep network to process images at a downsampled resolution for semantic information extraction, while simultaneously employing a lightweight shallow network to handle images at the original resolution to capture fine-grained details, thereby achieving efficient inference. For a fairer comparison, we made the following adaptations during the experimental process. All backbone networks in FCtL++ adopt SAM2, whereas in RUE, the deep network employs SAM2 while the shallow network remains unchanged. Additionally, we unified the decoder by using the Mask2Former panoptic segmentation head. Furthermore, we evaluated another panoptic segmentation model, PanopticFPN (Kirillov et al., 2019), using the SAM2 backbone to further assess the generalization capability of our side network. All results can be seen in Table 8.

In FCtL++, the contextual network struggles to effectively extract contextual information due to an excessive number of parameters (three sam2 backbone) and consequently fails to utilize the advantages of the VFM, yielding no improvement in segmentation accuracy. In particular, this highlights the advantage of our proposed lightweight side network. Conversely, RUE suffers from significant semantic information loss due to downsampling, and its lightweight shallow network is unable to effectively capture complex detailed information, leading to suboptimal segmentation performance. Additionally, incorporating our lightweight side network into PanopticFPN demonstrates the generalization capability of the side network architecture, with PQ improving from 53.84 to 54.69.

Table 9, presents a comparison with a multi-inference aggregation post-processing method with varying context sizes based on instance segmentation (Ye et al., 2026). This post-processing strategy applies sliding windows of three sizes with 50% overlap to perform local inference on UHR images and then merges all results through weighted aggregation. While it effectively exploits the full potential of the prediction model, it introduces considerable redundancy and computational cost, requiring more than three times the inference runs compared to the proposed method. Due to differences in task definitions and instance partitioning rules, the results are compared using mean Intersection over Union (mIoU), following (Cheng et al., 2022), focusing solely on

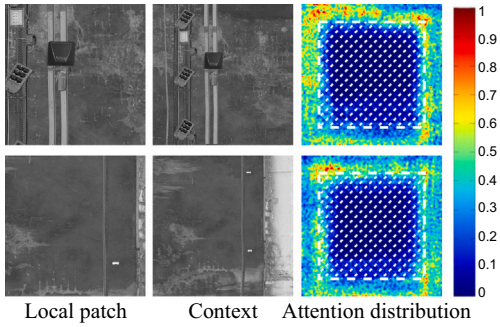

Fig. 5. The visualized normalized attention distribution pattern while processing the outermost three pixels of a local patch (outlined with white dashed lines).

semantic segmentation capability. For our panoptic segmentation results, all stuff classes are treated as background, and thing classes are converted to a single semantic region by removing instance IDs. Our method achieves consistent performance with the post-processing approach. The latter has been evaluated under "IoU with buffer" metric (Ye et al., 2026), which was specifically designed for damage recognition, and achieves a 90% damage recall, demonstrating practical value for asset management. In contrast, our approach requires only a single inference pass and no aggregation procedure, while still maintaining consistent performance.

This demonstrates that the incorporation of the lightweight side network effectively addresses the loss of contextual information in local inference. The SAM2-based variant shows no mIoU advantage over the Swin-based version. This gap primarily stems from differences in pretraining. SAM2 is trained with mask-only supervision, which emphasizes instance regions, boundaries, and fine-grained spatial details, whereas Swin is pretrained on ImageNet-21K (Deng et al., 2009) with semantic labels and thus focuses more on high-level semantic representations. Although Swin achieves competitive performance on semantic-oriented metrics, SAM2 demonstrates clear advantages in PQ, producing more coherent instance predictions and preserving finer structural details, such as thin and discontinuous cracks. This indicates that SAM2's superiority is not merely reflected in higher PQ values, but also in qualitatively better instance completeness and boundary fidelity, which are critical for fine-crack recognition.

4.5. Visualization of analysis results

The model's attention distribution was examined while processing the outer three pixels of a local patch (the edge region) in Fig. 5. The model focuses mainly on the surrounding context beyond the local patch. This suggests that edge feature extraction depends on external contextual cues to compensate for the limited information available within the patch itself, supporting the design rationale of our framework.

As shown in Fig. 6, incorporating contextual information improves patch-level predictions and reduces errors caused by missing context. Without it, several damaged regions remain missing or incomplete, and even entire patches may remain unclassified as illustrated in Fig. 6(d). Both Swin- and SAM2-based models benefit from contextual guidance, with SAM2 producing more complete instances.

Table 9
Comparison of proposed method with post-processing method (Ye et al., 2026) in mIoU.

Method	Background	Seepage	Corrosion	Damaged joint	Spalling	Crack	mIoU
without post-processing	0.908	0.698	0.359	0.349	0.487	0.088	0.481
with post-processing	0.904	0.726	0.448	0.418	0.495	0.066	0.510
256-swin-b2-rd	0.905	0.719	0.358	0.520	0.493	0.082	0.513
256-sam2-b2-rd	0.902	0.723	0.339	0.477	0.523	0.096	0.510

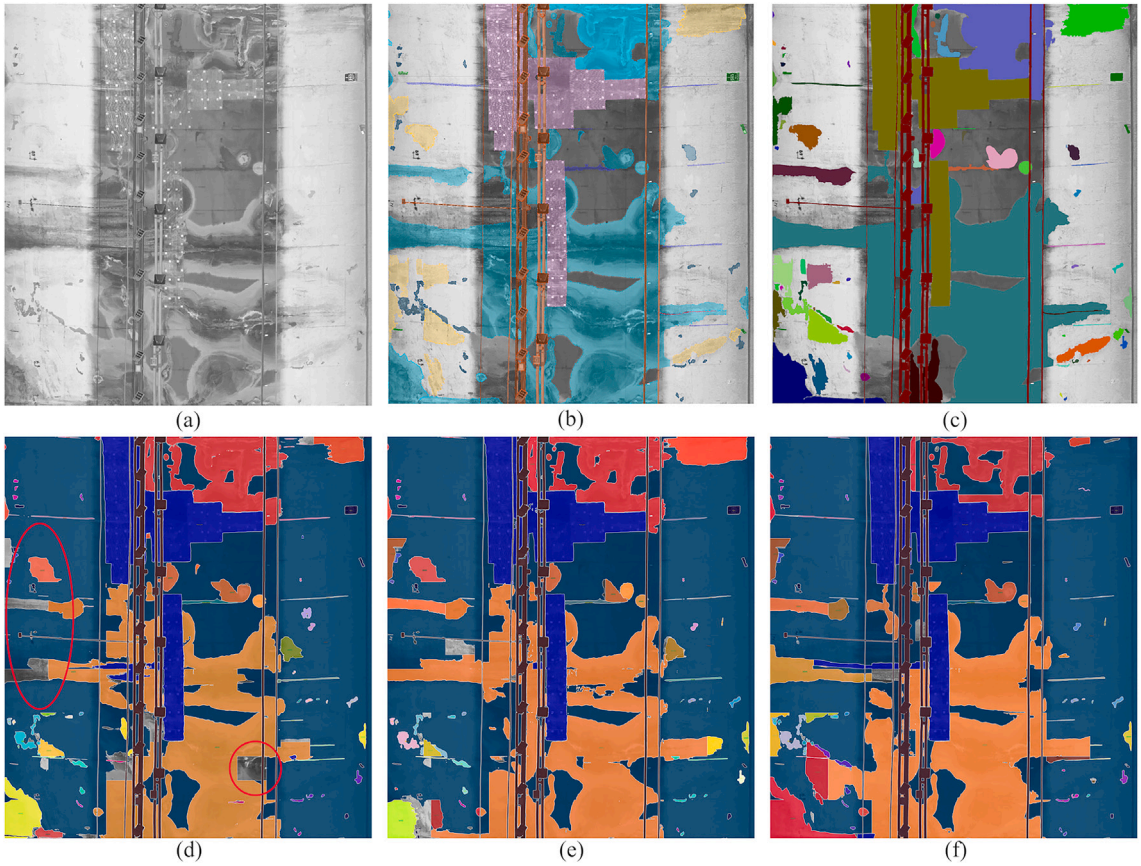

Fig. 6. The visualized results: (a) original UHR image; (b) classification annotation; (c) instance annotation; (d) panoptic segmentation results without context; (e) panoptic segmentation results of 256-swin-b2-rd, and (f) panoptic segmentation results of 256-sam2-b2-rd.

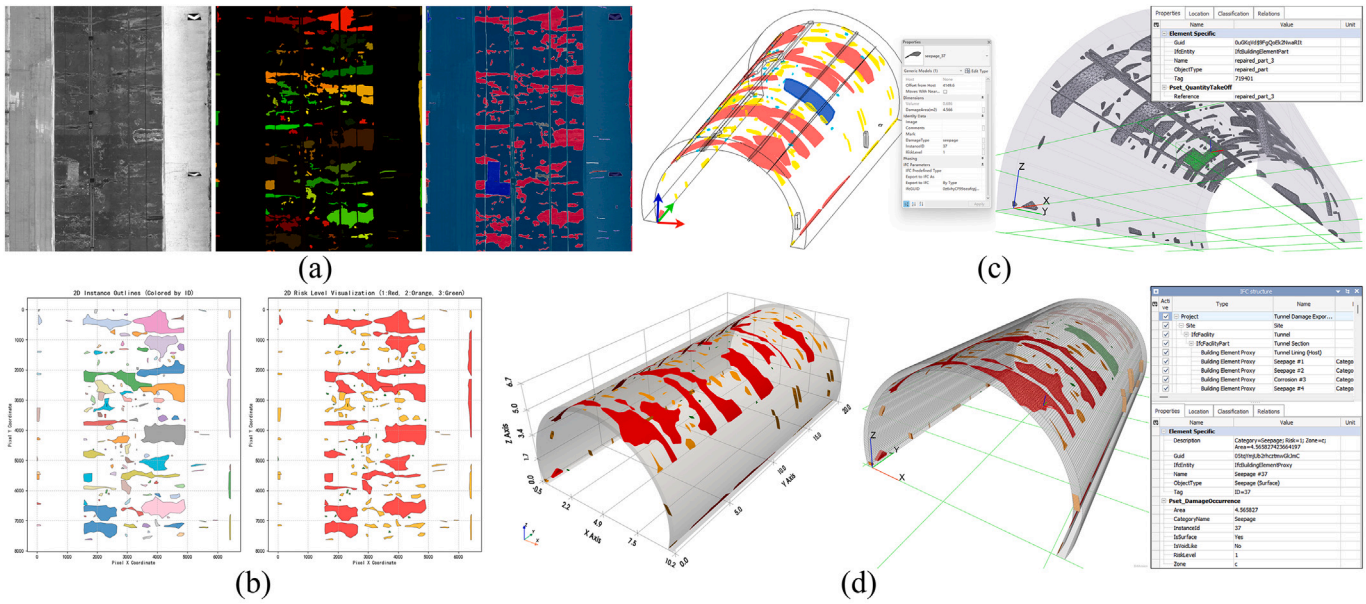

Fig. 7. As-damaged tunnel BIM model generation: (a) original UHR image and UHR panoptic segmentation results; (b) simplified prediction results in 2D; (c) model generated via Dynamo with damage (in Revit) and repaired pairs (in BIMVision) highlighted, separately; and (d) PyVista-based geometric solids and IFC format-based results generated from IfcOpenshell.

4.6. As-damaged tunnel digital model

Following our methodology, an automated workflow was implemented to process the UHR panoptic segmentation results, as shown in Fig. 7(a). After projected 3D vertices were obtained $\mathcal{P}^{3D}$, damage instance generation was provided using both of the two most commonly adopted approaches. One approach uses Dynamo to create the geometric solid, which is then imported into Revit to embed information (instance ID, category name, quantitative significance level and detected area)

and exported in IFC format. The other approach generates the geometric solids directly in Python environment using PyVista (Sullivan & Kaszynski, 2019), which are then converted and saved as IFC files using IfcOpenShell (IfcOpenShell contributors, 2023), following Fig. 2. Quantitative significance levels are assigned based on the area distribution of each category in the training set: for every category, the largest 20% of instances are marked as high-risk (red), the smallest 20% as low-risk (green), and the remaining instances as medium-risk (orange) (MIT, 2023). These rankings define two area thresholds per category for visualization. This method is presented as a visualization heuristic and an initial screening tool designed to facilitate rapid data prioritization in large-scale inspections. The generation of damage instances is conducted on the virtual inner surface of the tunnel. To balance model size and computational efficiency, the enclosing polylines of the damage regions are simplified by constraining their geometric complexity. A convex hull algorithm is applied to merge and simplify the defects, while adjacent points on each polyline are spaced at least 5 cm apart. This approach ensures a consistent reduction in point density while preserving the overall geometric characteristics of the damage instance, as shown in Fig. 7(b). Overall, after performing the 3D projection based on our direction-adaptive method, the results can be seen in Fig. 7(c) and (d). Since the input data come only from images without 3D information, damage regions other than seepage are modelled with a fixed thickness. Other tunnel elements, such as tubes and signage, are also positioned through projection due to the lack of detailed 3D geometry for modelling.

5. Conclusion

In this paper, we propose, for the first time, an architecture for panoptic segmentation of ultra-high-resolution (UHR) images, enabling automated, high-fidelity damage detection and as-damaged BIM generation for tunnel infrastructure. The key findings and contributions are summarized as follows:

- **Significant Performance Gains via Architecture Design:** A dual-branch architecture was proposed to resolve the conflict between ultra-high resolution and computational constraints. By introducing a lightweight side network for context guidance, the framework facilitates the integration of VFMs like SAM2. This approach significantly outperforms baselines without contextual guidance, achieving substantial improvements of 3.85, 5.42, and 5.91 in PQ, SQ, and RQ, respectively, particularly in detecting fine-grained cracks.
- **Efficiency Verification:** The evaluation demonstrates that our context-guided patch inference enables the accurate detection of both large-scale and cross-patch damage. Critically, compared to algorithms specifically designed for UHR segmentation and traditional multi-inference aggregation methods, the proposed approach maintains comparable segmentation accuracy while significantly reducing inference computational costs, making it viable for large-scale industrial deployment.
- **Automated As-Damaged BIM Generation:** An automated workflow was developed to bridge pixel-level detection with asset management. By projecting detected instances onto the tunnel geometry and populating semantic properties in standard IFC format, the framework successfully generates 3D as-damaged BIM models, providing a data-driven foundation for intelligent tunnel maintenance.

The limitations of this study are as follows: (a) The selection of context size is dataset-dependent and globally fixed, which may not represent the optimal context for every patch. (b) In the absence of 3D tunnel data and with only the unfolded drawings as input, the base tunnel BIM model is built from design parameters and thus does not reflect actual displacements or deformations. (c) The absence of 3D information prevents depth characterization of damage and limits accurate modelling of tunnel components that would otherwise be defined as precise BIM families. (d) The proposed priority-level classification, based on statistical measures, is intended as a heuristic for prioritization, rather than an exact risk assessment method, as it overlooks physics and structural interpretability such as location and orientation. Future studies will incorporate spatial sensitivity and numerical analysis to refine the evaluation. (e) Our method currently does not support effective contextual guidance on non-hierarchical ViT-based backbones due to the lack of inherent multi-scale features.

The future work could be: (a) For tunnel-specific applications, UHR algorithms can be further adapted by incorporating post-processing techniques or multi-model ensemble learning strategies to refine the segmentation results. (b) By integrating point clouds and images through a combined 2D–3D data fusion approach, a more detailed and automated workflow for generating the as-damaged model can be achieved. (c) Establishing numerical simulation models based on the obtained as-damaged tunnel BIM to more accurately assess risks, incorporating spatial sensitivity and mechanical analyses to refine the evaluation.

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 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 authors would like to thank a leading infrastructure management company for providing the data used in this research. The authors would also like to thank Tiancheng Wang for his valuable support with the IFC modelling. This publication is part of the project PNRR-NGEU, which has received funding from MUR-DM 118/2023.

References

Alsharif, M. A., Thermou, G., Ninic, J., & Tizani, W. (2025). Data integration framework for multi-level information modelling and numerical analysis of deteriorated RC bridges. *Advances in Engineering Software*, *202*, 103860.

Artus, M., Alabassy, M. S. H., & Koch, C. (2022). A BIM based framework for damage segmentation, modeling, and visualization using ifc. *Applied Sciences*, *12*(6), 2772.

Babanagar, N., Sheil, B., Ninić, J., Zhang, Q., & Hardy, S. (2025). Digital twins for urban underground space. *Tunnelling and Underground Space Technology*, *155*, 106140.

Bellon, F. G., Martins, A. C. P., de Carvalho, J. M. F., de Souza, C. A. F., Ribeiro, J. C. L., Júnior, K. M. L. C., & de Oliveira, D. S. (2025). Ifc framework for inspection and maintenance representation in facility management. *Automation in Construction*, *174*, 106157.

Borrmann, A., König, M., Koch, C., & Beetz, J. (2018). Building information modeling: Why? What? How? In *Building information modeling: Technology foundations and industry practice* (pp. 1–24). Springer.

Bui, H.-G., Ninić, J., Koch, C., Hackl, K., & Meschke, G. (2024). Integrated bim-based modeling and simulation of segmental tunnel lining by means of isogeometric analysis. *Finite Elements in Analysis and Design*, *229*, 104070.

buildingSMART International. (2024. Available at: *Industry foundation classes (ifc) 4.3 addendum 2 specification*. Online, https://ifc43-docs.standards.buildingsmart.org/.

Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Computer-aided Civil and Infrastructure Engineering*, *32*(5), 361–378.

Chai, C., Gao, Y., Xiong, G., Liu, J., & Li, H. (2025). Domain knowledge-driven image captioning for bridge damage description generation. *Automation in Construction*, *174*, 106116.

Chatterjee, B., & Poullis, C. (2021). Semantic segmentation from remote sensor data and the exploitation of latent learning for classification of auxiliary tasks. *Computer Vision and Image Understanding*, *210*, 103251.

Chen, W., Jiang, Z., Wang, Z., Cui, K., & Qian, X. (2019). Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8924–8933).

Chen, X., Huang, M., Xiao, F., Bai, Y., & Zhang, Q.-B. (2025). Sustainability of underground infrastructure–part 2: Digitalisation-based integration and optimisation for low carbon design. *Tunnelling and Underground Space Technology*, *159*, 106479.

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).

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).

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* (pp. 248–255). Ieee.

Fan, M., Lai, S., Huang, J., Wei, X., Chai, Z., Luo, J., & Wei, X. (2021). Rethinking Bisenet for real-time semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9716–9725).

Fang, Y., Mitoulis, S.-A., Boddice, D., Yu, J., & Ninic, J. (2025). Scan-to-bim-to-sim: Automated reconstruction of digital and simulation models from point clouds with applications on bridges. *Results in Engineering*, *25*, 104289.

Foria, F., Miceli, G., Nascetti, A., Loprencipe, G., Crespi, M., Belloni, V., Ravanelli, R., & Cordaro, S. (2022). Digitalization and defects analysis for the maintenance of mechanized tunnels. In *Proceedings World Tunnel Congress, wtc2022*. ITA-AITES.

Gall, V. E., Mleczko, W., Salmi, J., Ye, Z., Angerer, W., Robert, F., & Karlovšek, J. (2025). Continued development of the ITA guideline for BIM in tunnelling. In *Tunnelling into a sustainable future–methods and technologies* (pp. 1397–1404). CRC Press.

Gall, V. E., Mleczko, W., Robert, F., Angerer, W., Salmi, J., Ye, Z., Grasmick, J., Rives, M., Aldrian, W., Dohmen, P., Jacobs, A., Ninic, J., & Karlovšek, J. (2026). Ita-aites-BIM in tunnelling–guideline for mechanised and conventional tunnels. *Tunnelling and Underground Space Technology*, *168*, 106714.

Gao, T., Akinci, B., Ergan, S., & Garrett, J. (2015). An approach to combine progressively captured point clouds for BIM update. *Advanced Engineering Informatics*, *29*(4), 1001–1012.

Ge, K., Wang, C., Guo, Y. T., Tang, Y. S., Hu, Z. Z., & Chen, H. B. (2024). Fine-tuning vision Foundation model for crack segmentation in civil infrastructures. *Construction and Building Materials*, *431*, 136573.

Gómez, J., Casas, J. R., & Villalba, S. (2020). Structural health monitoring with distributed optical fiber sensors of tunnel lining affected by nearby construction activity. *Automation in Construction*, *117*, 103261.

Guo, S., Liu, L., Gan, Z., Wang, Y., Zhang, W., Wang, C., Jiang, G., Zhang, W., Yi, R., Ma, L., & Xu, K. (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).

Guo, Z., Wang, G., Liu, Z., Liu, L., Zou, Y., Li, S., Yang, R., Hu, X., Li, S., & Wang, D. (2024). The automated inspection of precast utility tunnel segments for geometric quality based on the BIM and lidar. *Buildings*, *14*(9), 2717.

Huang, M. Q., Ninić, J., & Zhang, Q. (2021). BIM, machine learning and computer vision techniques in underground construction: Current status and future perspectives. *Tunnelling and Underground Space Technology*, *108*, 103677.

Huang, M. Q., Chen, X. L., Ninić, J., Bai, Y., & Zhang, Q. B. (2023). A framework for integrating embodied carbon assessment and construction feasibility in prefabricated stations. *Tunnelling and Underground Space Technology*, *132*, 104920.

Huynh, C., Tran, A. T., Luu, K., & Hoai, M. (2021). Progressive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16755–16764).

IfcOpenShell contributors. (2023). *Bonsai: Add-on to blender providing a graphical native ifc authoring platform*. https://bonsaibim.org/ (accessed 17 October 2025). GPL-3.0-or-later.

Isailović, D., Stojanovic, V., Trapp, M., Richter, R., Hajdin, R., & Döllner, J. (2020). Bridge damage: Detection, ifc-based semantic enrichment and visualization. *Automation in Construction*, *112*, 103088.

ITA-AITES. (2019). *Tunnel market survey 2019*. https://nff.no/wp-content/uploads/sites/2/2021/01/Market-Survey-2019.pdf (Accessed: 30 September 2025).

Javadinasab Hormozabad, S., Gutierrez Soto, M., & Adeli, H. (2021). Integrating structural control, health monitoring, and energy harvesting for smart cities. *Expert Systems*, *38*(8), e12845.

Ji, D., Zhao, F., Lu, H., Tao, M., & Ye, J. (2023). Ultra-high resolution segmentation with ultra-rich context: A novel benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 23621–23630).

Ji, S., & Zhang, H. (2024). *ISAT with segment anything: An interactive Semi-Automatic annotation tool*. https://github.com/yatengLG/ISAT_with_segment_anything (Updated on 7 February 2025).

Jiang, Y., Pang, D., Li, C., & Wang, J. (2024). A method of concrete damage detection and localization based on weakly supervised learning. *Computer-aided Civil and Infrastructure Engineering*, *39*(7), 1042–1060.

Kerle, N., Nex, F., Gerke, M., Duarte, D., & Vetrivel, A. (2019). Uav-based structural damage mapping: A review. *ISPRS International Journal of Geo-information*, *9*(1), 14.

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6399–6408).

Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2019). Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9404–9413).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollar, P., & Girshick, R. (2023). Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer vision* (pp. 4015–4026).

Li, Q., Yang, W., Liu, W., Yu, Y., & He, S. (2021). From contexts to locality: Ultra-high resolution image segmentation via locality-aware contextual correlation. In *Proceedings of the IEEE/CVF International Conference on Computer vision* (pp. 7252–7261).

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 Robotics and Automation Letters*, *9*(2), 1708–1715.

Li, Q., Luo, J., Chen, C., Cai, J., Yang, W., Yu, Y., He, S., & Liu, W. (2025). Playing to the strengths of high-and low-resolution cues for ultra-high resolution image segmentation. *IEEE Robotics and Automation Letters*, *10*(8), 7787–7794.

Liang, X. (2019). Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. *Computer-aided Civil and Infrastructure Engineering*, *34*(5), 415–430.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on Computer vision and pattern recognition* (pp. 2117–2125).

Lin, W., Sheil, B., Zhang, P., Zhou, B., Wang, C., & Xie, X. (2024). Seg2tunnel: A hierarchical point cloud dataset and benchmarks for segmentation of segmental tunnel linings. *Tunnelling and Underground Space Technology*, *147*, 105735.

Lin, W., Sheil, B., Zhang, P., Chang, J., & Xie, X. (2025). Automated digital reconstruction of high-fidelity present-day geometries for segmental tunnel linings based on segmented point clouds. *Tunnelling and Underground Space Technology*, *164*, 106859.

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. *International Journal of Computer Vision*, *132*(11), 5030–5047.

Liu, X., Peng, H., Zheng, N., Yang, Y., Hu, H., & Yuan, Y. (2023). Efficientvit: Memory Efficient vision transformer with cascaded group attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14420–14430).

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).

Maharjan, S., Inadomi, S., Itakura, K., & Chun, P.-J. (2025). Domain-adaptive self-supervised learning for corrosion detection and 3d building information model mapping in steel tunnels. *Computer-aided Civil and Infrastructure Engineering*, *40*(26), 4425–4447.

Mehta, S., & Rastegari, M. (2022). Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *International Conference on Learning representations*.

MIT. (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.

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 *Ec3 conference 2025* (Vol. 6, pp. 0). European Council on Computing in Construction.

Ninić, J., Gamra, A., & Ghiassi, B. (2024). Real-time assessment of tunnelling-induced damage to structures within the building information modelling framework. *Underground Space*, *14*, 99–117.

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. *Tunnelling and Underground Space Technology*, *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. *Tunnelling and Underground Space Technology*, *155*, 106179.

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., & Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, *178*, 603–615.

Qin, R., Liu, X., Shi, J., Lin, L., & Yang, J. (2025). Boosting the dual-stream architecture in ultra-high resolution segmentation with resolution-biased uncertainty estimation. In *Proceedings of the Computer vision and pattern recognition Conference* (pp. 25960–25970).

Rafiei, M. H., & Adeli, H. (2018). A novel unsupervised deep learning model for global and local health condition assessment of structures. *Engineering Structures*, *156*, 598–607.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., & Feichtenhofer, C. (2024). Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714.

Rimella, N., Rimella, L., & Osello, A. (2025). Machine learning method for as-is tunnel information model reconstruction. *Automation in Construction*, *172*, 106039.

Rosso, M. M., Aloisio, A., Randazzo, V., Tanzi, L., Cirrincione, G., & Marano, G. C. (2024). Comparative deep learning studies for indirect tunnel monitoring with and without Fourier pre-processing. *Integrated Computer-Aided Engineering*, *31*(2), 213–232.

Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y. L., & Feichtenhofer, C. (2023). Hiera: A hierarchical vision transformer without the bells-and-whistles. In *International Conference on machine Learning* (pp. 29441–29454). PMLR.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*. John Wiley & Sons.

Sanfilippo, R., Robert, F., Menozzi, A., Paskaleva, G., Garbutt, D., Foria, F., Glab, K., Folly, M., Svensson, M., Spaeh, J., & Karlovsek, J. (2025). ITA guideline for sustainable BIM approaches in lifecycle management of tunnelling projects. In *Tunnelling into a sustainable future–methods and technologies* (pp. 4604–4611). CRC Press.

Schade, W., Khanna, A. A., Mader, S., Streif, M., Abkai, T., de Stasio, C., Fermi, F., Bielanska, D., Deidda, C., Thiery, W., & Maatsch, S. (2020). *Support study on the climate adaptation and cross-border investment needs to realise the ten-t network*. Technical report, EUR policy report on TEN-T infrastructure and resilience, Luxembourg: European Commission, Directorate-General for Mobility and Transport (DG MOVE).

Shen, T., Zhang, Y., Qi, L., Kuen, J., Xie, X., Wu, J., Lin, Z., & Jia, J. (2022). High quality segmentation for ultra high-resolution images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1310–1319).

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A. …Bojanowski, P. (2025). Dinov3. arXiv preprint arXiv:2508.10104.

Sirca Jr, G. F., & Adeli, H. (2018). Infrared thermography for detecting defects in concrete structures. *Journal of Civil Engineering and Management*, *24*(7), 508–515.

Spencer Jr, B. F., Sim, S.-H., Kim, R. E., & Yoon, H. (2025). Advances in artificial intelligence for structural health monitoring: A comprehensive review. *KSCE Journal of Civil Engineering*, *29*(3), 100203.

Sullivan, B., & Kaszynski, A. (May 2019). PyVista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (VTK). *Journal of Open Source Software*, *4*(37), 1450. https://doi.org/10.21105/joss.01450

Sun, Y., Hou, M., Chen, H., Gao, C., & Nallbani, A. (2025). Damage information mapping from images to BIM models for the detection of ancient stone bridges. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *48*, 1459–1466.

Tan, Y., Li, G., Cai, R., Ma, J., & Wang, M. (2022). Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection. *Automation in Construction*, *139*, 104284.

Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, *19*(7), 829–843.

Teng, Y., Xu, J., Pan, W., & Zhang, Y. (2022). A systematic review of the integration of building information modeling into life cycle assessment. *Building and Environment*, *221*, 109260.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, *34*, 12077–12090.

Xu, S., Zhao, K., Loney, J., Li, Z., & Visentin, A. (2025). Image-based large language model approach to road pavement monitoring. *Computer-aided Civil and Infrastructure Engineering*, *40*(26), 4448–4464.

Yang, X., Grussenmeyer, P., Koehl, M., Macher, H., Murtiyoso, A., & Landes, T. (2020). Review of built heritage modelling: Integration of hbim and other information techniques. *Journal of Cultural Heritage*, *46*, 350–360.

Ye, Z., Faramarzi, A., Ninić, J., & Lin, W. (2025). Automated digital twin reconstruction for tunnel inspection and maintenance. In *Tunnelling into a sustainable future–methods and technologies* (pp. 517–524). CRC Press.

Ye, Z., Lovell, L., Faramarzi, A., & Ninić, J. (2024). Sam-based instance segmentation models for the Automation of structural damage detection. *Advanced Engineering Informatics*, *62*, 102826.

Ye, Z., Lin, W., Faramarzi, A., Xie, X., & Ninić, J. (2025). Sam4tun: No-training model for tunnel lining point cloud component segmentation. *Tunnelling and Underground Space Technology*, *158*, 106401.

Ye, Z., Mozafarian, M., Cavallaro, P. A. R., Altinay, K., Villa, V., & Ninić, J. (2026). Automated multi-category tunnel damage detection and report generation from ultra-high-resolution panoramic laser images. *Tunnelling and Underground Space Technology*, *168*, 107194.

Yin, X., Liu, H., Chen, Y., Wang, Y., & Al-Hussein, M. (2020). A bim-based framework for operation and maintenance of utility tunnels. *Tunnelling and Underground Space Technology*, *97*, 103252.

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., & Sang, N. (2021). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, *129*(11), 3051–3068.

Zeibak-Shini, R., Sacks, R., Ma, L., & Filin, S. (2016). Towards generation of as-damaged BIM models using laser-scanning and as-built BIM: First estimate of as-damaged locations of reinforced concrete frame members in masonry infill structures. *Advanced Engineering Informatics*, *30*(3), 312–326.

Zhang, L., Guo, J., Fu, X., Tiong, R. L. K., & Zhang, P. (2024). Digital Twin enabled real-time advanced control of tbm operation using deep learning methods. *Automation in Construction*, *158*, 105240.

Zhang, Y., Karlovšek, J., & Liu, X. (2022). Identification method for internal forces of segmental tunnel linings via the combination of laser scanning and hybrid structural analysis. *Sensors*, *22*(6), 2421.

Zhu, H., Huang, M., & Zhang, Q.-B. (2024). Tungpr: Enhancing data-driven maintenance for tunnel linings through synthetic datasets, deep learning and BIM. *Tunnelling and Underground Space Technology*, *145*, 105568.

Zhu, H., Huang, M., Ji, P., Xiao, F., & Zhang, Q.-B. (2025). Transforming the maintenance of underground infrastructure through digital twins: State of the art and outlook. *Tunnelling and Underground Space Technology*, *161*, 106508.

Zhu, Z.-H., Fu, J.-Y., Yang, J.-S., & Zhang, X.-M. (2016). Panoramic image stitching for arbitrarily shaped tunnel lining inspection. *Computer-aided Civil and Infrastructure Engineering*, *31*(12), 936–953.