Tunnelling and Underground Space Technology 153 (2024) 106043

Contents lists available at ScienceDirect

Tunnelling and Underground Space Technology incorporating Trenchless Technology Research

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

Automated masonry spalling severity segmentation in historic railway tunnels using deep learning and a block face plane fitting approach

Jack Smith [a,*], Chrysothemis Paraskevopoulou [a], Anthony G. Cohn [b,c], Ryan Kromer [a], Anmol Bedi [d], Marco Invernici [d]

[a] School of Earth and Environment, University of Leeds, Leeds, UK
[b] School of Computing, University of Leeds, Leeds, UK
[c] Alan Turing Institute, London, UK
[d] Bedi Consulting Ltd., London, UK

ARTICLE INFO

Keywords:
Masonry
Condition assessment
Deep learning
Semantic segmentation
Historic tunnels

ABSTRACT

Masonry lined tunnel condition assessment is a predominantly manual process. It consists primarily of a visual inspection followed by a lengthy and subjective manual defect labelling process. There is therefore much potential for automation. Masonry spalling is a key indicator of a masonry tunnel's condition. To obtain actionable detail about a tunnel's condition, it is also necessary to determine the spalling severity, defined by the depth of spalling. This study presents an automated workflow to identify the depth of spalling from masonry tunnel 3D point cloud data obtained by lidar. Firstly, a tunnel point cloud is unrolled using a cylindrical projection and the points are rasterised into a 2D image taking pixel values of the offset of each point from the cylinder. Then, a 2D U-Net pretrained on both real and synthetic masonry lining data, is used to segment masonry joint locations to isolate individual blocks. A separate U-Net is used to segment areas of masonry damage and data obstructions, which are then masked out before a surface plane representing the theoretical undamaged surface location is fitted to each masonry block from the remaining points. This allows the depth of spalling to be measured directly. As a result, this method can automatically determine the depth of spalling despite the curved and often deformed nature of a masonry tunnel profile. Experiments show results competitive with those obtained by human assessors.

1. Introduction

A substantial proportion of the world's railway tunnels are masonry lined and were constructed before the creation of modern design standards during the rapid expansion of railway networks in the mid-19th century. The variety of their design and often lack of construction documentation makes them difficult and costly to analyse. In addition, due to their age, the tunnel linings have deteriorated and often contain large areas of defects including deformation, cracking, spalling and efflorescence (Atkinson et al., 2021; Chiu et al., 2015). It is therefore vital to conduct regular condition assessments to identify the location and severity of damages to monitor the overall health of the structure (NR, 2016). Spalling is one of the most time-consuming damages to identify and analyse, relying on manual visual inspection of the structure on site and from lidar surveys. In addition, the presence of extensive spalling often makes it difficult to visually determine the severity of other damages. For example, when a deviation in the lining surface location is observed, it can be difficult to differentiate between brick loss caused by spalling and movements of the lining which may be an indicator of more serious structural deficiencies.

Automation of condition assessment tasks has the potential to reduce costs and improve the repeatability of the assessment results (Koch et al., 2015). While past research has been conducted investigating automated concrete spalling detection (Dong and Catbas, 2021; Koch et al., 2015; Sjölander et al., 2023; Spencer et al., 2019), the authors could not find published research that investigates automatic determination and localisation of masonry spalling depth. This paper presents an automated workflow for masonry lined tunnel spalling severity assessment. Expanding upon work conducted by Smith et al. (2023), this paper investigates how deep learning methods can be effectively integrated into a robust industry standard condition assessment workflow. The developed workflow combines deep learning with a geometric block

* Corresponding author.
E-mail address: eejmws@leeds.ac.uk (J. Smith).

https://doi.org/10.1016/j.tust.2024.106043
Received 10 November 2023; Received in revised form 9 July 2024; Accepted 17 August 2024
Available online 2 September 2024

orientation analysis to automatically extract the spalling locations and label the depth of spalling. The scientific contribution of the work performed is threefold:

- A new automated workflow is proposed for masonry spalling severity segmentation that has the potential to reduce the labour required for masonry lined tunnel condition assessment, while improving the repeatability and explainability of the process.
- Deep learning methods for automated masonry joint segmentation and automated masonry defect segmentation from lidar data are investigated. Optimum trained networks are proposed for use in the workflow.
- A method of isolating individual masonry blocks within a point cloud is proposed. This will aid with tunnel documentation.

2. Background

The world's railway networks provide connectivity vital for economic prosperity (Oxera, 2014; Williams Rail Review, 2019). As one of the most environmentally friendly modes of transport, encouraging passenger growth and freight usage by increasing railway network reliability and reducing costs is important to encourage the modal shift away from road travel required to meet upcoming net-zero targets. In addition, with increasing urbanisation, more efficient use must be made of the underground space to ensure that our cities are sustainable into the future (Paraskevopoulou et al., 2019; 2022a). This includes upgrading underground railway infrastructure to cope with projected increased passenger numbers. Since the construction of new tunnels can be both costly (Paraskevopoulou and Benardos, 2013; Paraskevopoulou and Boutsis, 2020; Paraskevopoulou et al., 2022b) and disruptive, where possible, it is more sustainable to better utilise existing and disused tunnels by extending their lifespan.

Within the UK, there is approximately 31,251 km of operational railway track (ORR, 2021), the majority of which was constructed in the second half of the 19th century when most railway tunnels were masonry lined. The condition of these tunnels has often deteriorated after around 150 years of usage. Ideally, the tunnels would be upgraded to modern standards (Atkinson et al., 2021). However, due to funding constraints, in most cases they are monitored and subject to ad-hoc repairs to prevent further deterioration. It is therefore important to accurately characterise the serviceability state of each tunnel to facilitate timely remediation work and prevent disruptive and costly line closures.

While there are a range of non-destructive and in-situ testing methods available to analyse the condition of masonry (Hussain and Akhtar, 2017; Schuller, 2003); the majority of these require specialist equipment with expert operators and substantial post-analysis. Due to the extent of masonry condition assessment required within the UK's railway tunnels, these techniques are not routine and are commonly only applied to areas of particular concern. As a result, the industry methodology for masonry lined tunnel condition assessment consists primarily of an onsite visual and limited tactile inspection, with tunnel pathologies indicated by the presence of observable lining defects. Within the UK, this is followed by completion of a TCMI (Tunnel Condition Marking Index) in a standardised process outlined within NR/L3/CIV/006 (NR, 2016) and shown in Fig. 1.

Increasingly, lidar surveys are conducted during the inspection to enable further post-inspection analysis from a 3D point cloud and confirm damage severity classifications. The severity scores obtained can be used to prioritise maintenance and inform future monitoring approaches. The TCMI also uses these to calculate an overall condition score for the structure, which can be used for planning larger scale improvement works and inform asset management. Despite this standardisation, identification of defect locations and determination of severity values is still highly labour intensive and subjective, with conclusions based on the perception and engineering judgement of individual assessors. As a result, the outcome of assessments is highly variable and can result in deterioration going unnoticed. This leads to more challenging repairs being required in the future and is a known problem across condition assessment disciplines as noted by Laefer et al. (2010) and Phares et al. (2004). There is therefore a need for an explainable and consistent workflow that can reduce the manual labour required for masonry tunnel condition assessment.

Masonry spalling is typically the most widespread defect observed on a masonry tunnel's lining (Chiu et al., 2015) and as a result, is one of the most time-consuming to document. Spalling is generally caused by chemical and moisture content changes within the masonry causing expansion and shrinking of the blocks. This results in a deterioration in the masonry's structural properties leading to overloading (McKibbins et al., 2010). This may also be more directly caused by changes in loading conditions. Deep spalling reduces the effective thickness of the masonry arch and therefore reduces the overall loading capacity of the tunnel. Deformations in the lining or changes in the lining thickness will adjust the location of the arch thrust line. A significant change in the thrust line can also result in arch instability, which in extreme cases can lead to the formation of a mechanism and ultimately collapse (Heyman, 2014). Depth of spalling is therefore used to quantify spalling severity.

Due to the time-limited nature of tunnel inspections, instead of manually measuring spalling depth onsite, it is increasingly calculated post-inspection directly from a point cloud obtained by lidar. This requires an engineer to manually draw a surface representing the undamaged profile of the tunnel to measure off the depth of spalling. However, it is complicated by two factors. Firstly, the as built tunnel profile is typically unknown and has also often substantially deformed since construction. This is in addition to local deformations to the lining caused by changes in the loading conditions, voids behind the lining or mortar/brick loss. This makes determining the unspalled block face profile challenging. Secondly, mortar and crack locations need to be manually masked out in order to only consider block face areas and isolate brick spalling from mortar loss or masonry cracking.

Due to the need to reduce costs yet increase the efficiency and effectiveness of condition assessment tasks, there has been substantial research across engineering disciplines into developing improved inspection and assessment methodologies for all asset types. There is a particularly significant potential to reduce costs through automation of the damage detection and classification step of the condition assessment process. Research has been aided by the recent increased accessibility and effectiveness of computer vision methods using deep learning (Schmidhuber, 2022). Multiple reviews of the field have been conducted (Deng et al., 2022; Dong and Catbas, 2021; Koch et al., 2015; Spencer et al., 2019; Ye et al., 2019).

2.1. Machine learning for tunnel condition assessment

Neural networks are a machine learning method consisting of layers of interconnected neurons, each containing a simple equation (Lecun

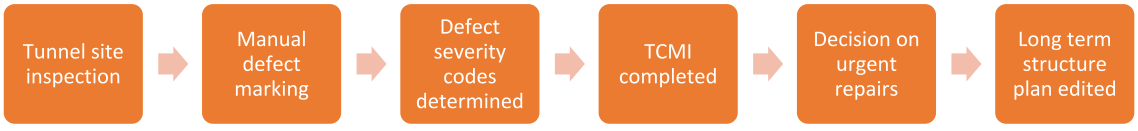


Fig. 1. Typical UK industry workflow for railway tunnel condition assessment.

et al., 2015). Backpropagation allows the equation parameters to be optimised for transforming data from an input to a desired output. Deep learning refers to neural networks with many layers. Since these networks typically have millions of trainable parameters, the reasoning behind their outputs for each specific task is challenging to interrogate and so they are largely a black box method. Nevertheless, they have revolutionised machine learning research, by enabling more complex relationships to be understood within larger volumes of data. Convolutional Neural Networks (CNNs) are an effective neural network design for computer vision tasks. They involve layers of image convolutions, effectively image filters, being applied at varying scales to the image to bring out salient features and build a description of the image (Simonyan and Zisserman, 2014). The nature of the filters is learnt during the training process. Much research has been conducted using deep learning and CNNs for defect segmentation in concrete tunnel linings (Sjölander et al., 2023). The majority of studies consider concrete cracking, spalling, or water ingress and take advantage of the homogeneous appearance of a concrete surface and relative lack of tunnel damage or deformation found in newer concrete lined tunnels.

Initially, CNN architectures were only developed for classification tasks, so studies such as Protopapadakis et al. (2019) and Huang et al. (2018) split tunnel lining images up into overlapping patches. Damaged areas were identified by sliding a window of focus between these patches and using the difference in predicted damage likelihoods to determine the exact location of damage on the tunnel lining. Later research using R-CNNs enabled boundary boxes to be placed around identified damaged areas (Gao et al., 2019). Subsequently, Mask R-CNNs were developed that could semantically segment damage within each boundary box (Xu et al., 2021). Finally, encoder-decoder based networks, such as the U-Net (Feng et al., 2023) and domain specific adaptions including CrackSegNet (Ren et al., 2020) or U-CliqueNet (Li et al., 2020) have enabled pixelwise semantic segmentation of damage in lining images and form the most popular and well documented damage identification and localisation vision technique. Encoder-decoder designs follow the standard CNN encoder architecture with a decoder that rebuilds a segmented image from the description generated by the encoder.

Many of these methods, however, do not perform well when faced with less homogenous tunnel lining surfaces and the poor, uneven lighting conditions found in many tunnels. Some researchers have modified standard deep learning semantic segmentation networks to overcome this. Dong et al. (2019), for example, developed Fl-SegNet for concrete tunnel lining multi damage detection from photographic data. By combining the existing SegNet with a focal loss function, they increased the mean intersection over union score for crack and spalling segmentation in challenging low-light conditions.

An alternative approach is to investigate the tunnel geometry directly using 3D point clouds obtained by lidar or photogrammetry. Lidar is extensively used for documentation and analysis during the construction of new tunnels, as it produces a comprehensive 3D dataset within a short time and for a reasonable cost. It is therefore considered the preferred data collection method for automated tunnel analysis workflows (Allen et al., 2023). Lidar technology is also increasingly being integrated into autonomous robotic and drone-based inspection systems (Sjölander et al., 2023), so working with lidar data sets the groundwork for end to end automation of the inspection and condition assessment process.

2.2. Deep learning with 3D point clouds

There are four main approaches for applying defect detection and segmentation algorithms to 3D data: point based, voxel, multiview and single surface projection. A comparison of the key methods is shown in Table 1, while more detailed reviews of the application of deep learning to point cloud classification and segmentation can be found in Che et al. (2019), Guo et al. (2021) and Zhang et al. (2023).

Single surface projection has been used for segmenting damage on concrete-lined tunnel point clouds (Feng et al., 2023; Huang et al., 2020; Zhou et al., 2021) by using a cylindrical tunnel projection strategy. The studies rasterised radial offset, colour and intensity values to segment

Table 1
Methods of preparing 3D point clouds for semantic segmentation using deep learning.

Method	Description	Major use cases	Relevant architectures	Advantages and Disadvantages
Pointwise	Operates directly on unstructured 3D point cloud data.	Segmentation of 3D shapes, e.g., Identification of tunnel structural components (Grandio et al., 2022; Soilán et al., 2020). Limited use for concrete cracking and spalling segmentation (Bolourian et al., 2023).	PointNet (Charles R Qi et al., 2017), Pointnet++ (Charles R. Qi et al., 2017)	Takes full advantage of spatial location data. No data loss during preprocessing. Low efficiency at operating on features of different scales (Guo et al., 2021).
Voxel	Points are rasterised onto a 3D grid.	Segmentation of objects within 3D internal scans. Used for underground pipe, subsidence and crack detection from GPR data (Li et al., 2021).	VoxNet (Maturana and Scherer, 2015), GPR-RCNN (Li et al., 2021)	Works well when data points inside an object are involved. Grid structure retains spatial relations of points. Large memory requirements. Algorithms struggle with data sparsity. Down sampling often required, leading to information loss (Guo et al., 2021).
Multiview projection	The point cloud is rasterised onto multiple 2D surfaces. 3D information is captured by training neural networks to understand the different image perspectives.	Object and shape classification. Point Cloud to BIM workflows	MVCNN, MHBN, View-GCN	Integration of standard 2D CNNs enables good segmentation of local features in addition to global 3D shapes. The field is relatively undeveloped. Success is dependent on view selection; some points may be lost through occlusion.
Single surface projection	The point cloud is projected onto a surface and unwrapped, then rasterised into pixels on a 2D image. 3D information is captured by setting the pixel values as the projection distance. Standard 2D computer vision techniques are then applied.	Segmentation of surface features and textures on an isolated object. Segmentation of cracking, spalling and leakage on concrete tunnel linings using a cylindrical projection (Feng et al., 2023; Huang et al., 2020; Zhou et al., 2021).	CNNs (Simonyan and Zisserman, 2014), U-Net (Ronneberger et al., 2015), U-Net++ (Zhou et al., 2018)	Performs well at segmentation of local surface features and on objects with a simple global shape. It is possible to use and finetune pretrained networks from the well-developed field of 2D image segmentation. Loses some spatial relativity during rasterisation. Not effective at object segmentation within a larger 3D scene. Success is dependent on the unwrapping strategy (Gao et al., 2019).

tunnel lining damage using 2D image based neural networks. Since the majority of concrete-lined tunnels are built with a cylindrical profile and have little lining deformation, cylindrical projection can accurately flatten the point cloud. However, the heavily deformed nature of a typical historic masonry-lined tunnel's profile prohibits direct application of their research to masonry spalling segmentation. Despite this, compared to pointwise, voxel and multiview methods, single surface projection methods are better developed for surface texture characterisation (Zhou et al., 2021). In addition, they have the lowest compute requirements.

2.3. Machine learning for masonry tunnel assessment

While much research has been conducted on automating concrete lined tunnel condition assessment, Smith et al. (2023) is the only publicly available study that could be found to have used deep learning for analysing masonry tunnel surfaces or to have applied deep learning to masonry point cloud data. The uneven and varied nature of masonry surfaces coupled with the variety of masonry materials and geometries has rendered deep learning based masonry condition assessment challenging. Masonry joints complicate defect segmentation, as they can be misidentified as cracks. Mortar loss can also easily be confused with masonry spalling. Ibrahim et al. (2020), Loverdos and Sarhosis (2022) and Smith et al. (2023) have trained encoder-decoder CNNs to detect joint locations within masonry walls as a necessary prerequisite to further analysis. This enables individual block locations to be isolated, allowing analysis to focus either on the blocks themselves or at the boundary between them. Nevertheless, masonry wall crack detection using deep learning has mostly been conducted without prior block isolation (Brackenbury and Dejong, 2018; Dais et al., 2021; Hallee et al., 2021; Karimi et al., 2023; Loverdos and Sarhosis, 2022). These studies used encoder-decoder networks trained on colour photographic data of a variety of masonry walls. Valero et al. (2019) and Wang et al. (2019) used patch based RCNN methods to segment masonry containing spalling.

2.4. Automated severity assessment

While identification of the presence and location of structural defects is useful, for condition assessments to provide actionable information about the structural condition it is important to classify defects into meaningful severity categories. However, there has been little research considering automation of structural defect severity classification. For concrete structures, Flah et al. (2020) used deep learning to identify concrete crack sizes and orientations. These parameters were then used within existing codified severity assessment procedures to assess structural damage severity. In assessing concrete spalling severity, Nguyen and Hoang (2022) considered the existence of exposed rebars as indicative of severe concrete spalling and trained deep learning networks to detect these.

However, despite the existence of codified masonry tunnel spalling severity classifications (NR, 2016), no research could be found that investigated masonry spalling severity location segmentation or looked at automated generation of spalling depth maps. Determination of spalling depth is a vital part of the routine masonry lined tunnel condition assessment process, as it provides a strong indication of the health of the tunnel without requiring destructive testing that is unsustainable over the long lifespan of the structure. Given the current laborious process for spalling depth determination and severity classification outlined in section 2.1, an automated method would reduce the labour cost and improve the reliability of the assessment procedure. For an automated method to be successful, it must be applied in a standard procedure that is easily repeatable, understandable for operators and can generalise to work with most masonry tunnels on the railway network.

The often deformed and non-homogenous surface of masonry lined tunnels precludes the direct application of existing deep learning based concrete lined tunnel spalling detection methods to the task of masonry tunnel spalling segmentation. To obtain the benefits of condition assessment automation within masonry lined tunnels, this paper proposes a fully automated workflow that is robust to tunnel deformations and varying masonry joint geometries. By combining a geometric analysis of block locations and orientations with the use of deep learning for joint detection and damage detection, the workflow enables robust masonry spalling segmentation for each block of masonry on the tunnel lining. In addition, the workflow is the first to automate the analysis of masonry spalling severity by segmenting and classifying masonry spalling depth.

3. Methodology

The proposed workflow automatically locates the areas of spalling that fall into each severity category on the lining of a masonry tunnel from a lidar survey of the structure. It then visualises these locations on a 3D point cloud. The workflow replaces the manual spalling labelling and severity categorisation steps of the condition assessment process shown in Fig. 1 with those shown in green within Fig. 2. To deal with the wide variety of masonry tunnels and reduce the amount of data labelling required when a new tunnel needs to be analysed, it is proposed that the method can be trained on a short 10 m section of tunnel, before being applied to other areas of tunnel with similar masonry styles and surface characteristics.

The method operates on 3D point cloud data using the following theory. Spalling severity is defined by the depth of spalling which is calculated as the offset from where the masonry face would be if it was unspalled. Therefore, the workflow must determine what the surface location and orientation of each block would be if it was unspalled. This differs from the as-built tunnel lining surface location, as the tunnel shape has likely since deformed. The unspalled lining surface is obtained by identifying undamaged surface points within the tunnel using a 2D convolutional neural network. The network uses a single-surface projection method to operate on the 3D point cloud data. An unspalled surface is fitted to the undamaged points. It is necessary, however, to account for differing masonry block orientations and geometries. As a result, the workflow uses a separate convolutional neural network to isolate individual masonry block locations by segmenting out the masonry joints. This enables a flat plane to be fitted to each individual block.

The complete workflow is shown in Fig. 3 and consists of 5 primary stages:

1. Unfold the tunnel 3D point cloud and rasterise point cloud parameters onto a 2D image
2. Use deep learning to identify masonry joint locations and isolate individual blocks
3. Use deep learning to identify areas of block face damage and remove damaged locations
4. Fit planes to the remaining undamaged points on each block representing the unspalled masonry surface
5. Measure the depth of spalling from the unspalled masonry face planes to calculate damage severity

There are multiple advantages of this method. Firstly, by utilising point cloud data that is increasingly routinely collected by lidar during structural inspections, it is possible to slot this workflow into the current assessment methodology without requiring a change in overall procedure or expensive new equipment. This paves the way for autonomous structural inspections that would reduce the disruption to railway operations from line closure. As historic tunnels can be a hazardous environment with dangers from falling masonry and dust, it also has positive impacts on the health and safety of tunnel inspections. The modularity of the workflow enables future developments in masonry damage

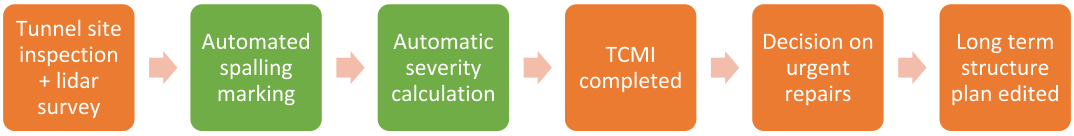


Fig. 2. Automated workflow within overall condition assessment procedure.

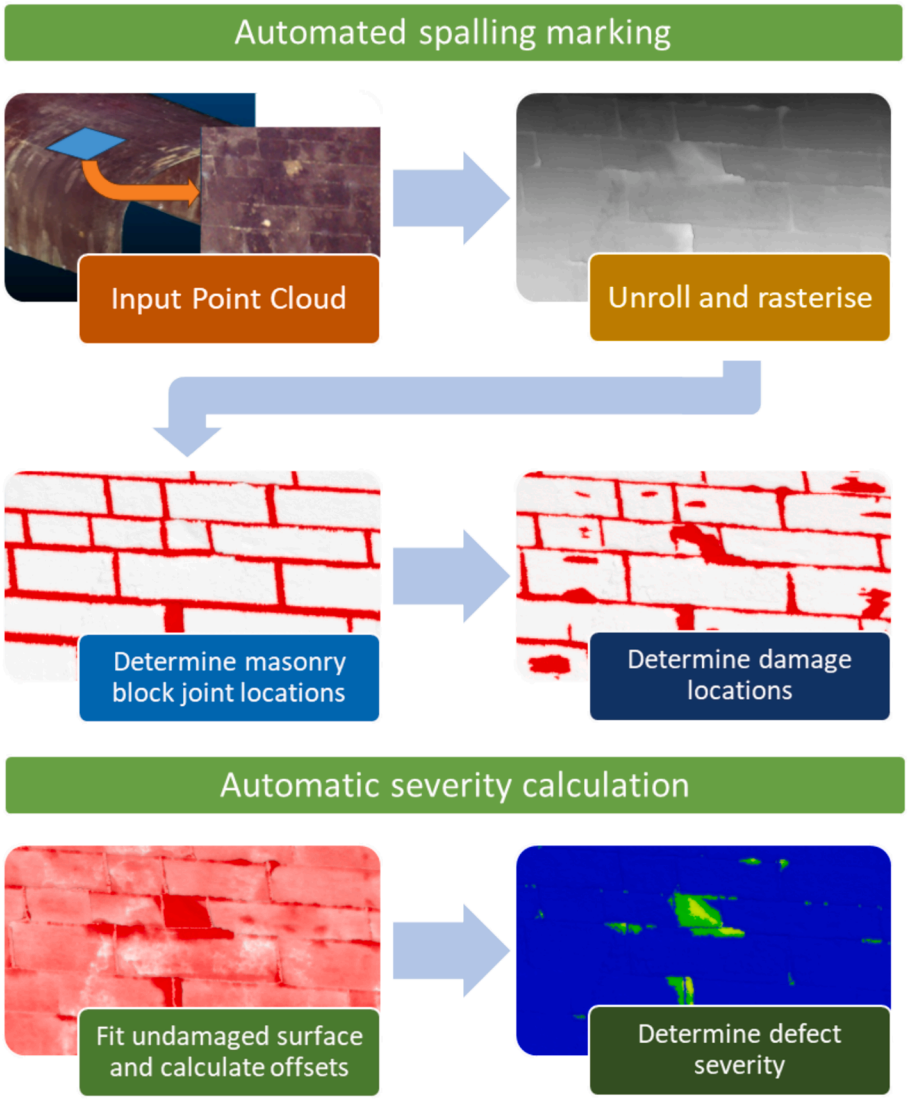


Fig. 3. Key steps of the automated workflow.

segmentation and masonry joint segmentation to be easily integrated into the method, by switching out the relevant deep learning network. The method also creates multiple additional outputs such as the location of each masonry block and the locations of masonry deformations. These provide additional documentation on the state of the structure and are useful analysis aids for an engineer. While it may also be possible to use deep learning to directly determine the depth of spalling, due to the black box nature of deep learning, it would be challenging to verify the results without a complete manual reassessment. It is easier to visually check the joint and damage detection steps separately for erroneous results. In addition, since the output defect depth value has a geometric basis, it produces a useful visualisation for the engineer and is easy to adjust to differing severity definitions.

3.1. Dataset

3D point cloud data of the linings of two masonry lined tunnels were provided by Bedi Consulting Ltd with the support of Network Rail for this study. They both contain operational railway lines and are named here only as Tunnel 1 (T1) and Tunnel 2 (T2). Point clouds of 20 m long sections of each were obtained by lidar and used as case studies to investigate the effectiveness of the workflow. T2 is predominantly lined with brick masonry, while T1 has a stone masonry lining. The age and condition of these tunnels is typical of those in operation on the British railway network and they represent the two most common masonry lining designs. Due to the geometric and surface texture differences between the two tunnels, the machine learning aspects of the workflow were trained on sections from each of them to enable the workflow to operate robustly on both types.

3.1.1. Tunnel 1 (T1)

T1 is located at a shallow depth in an urban area and was built in the 1870s. The tunnel is predominantly lined with blocks of oolitic limestone, although some areas have been infilled with brick masonry. The circumference of the tunnel lining surface is 18.96 m. There is a substantial quantity of spalling and efflorescence on the tunnel lining, although in most areas it is of low severity. It is apparent from visualising the point cloud data that the tunnel was constructed as a three centred arch. The presence of mortar loss and joint opening near to the crown suggests that inwards deformation is also present.

3.1.2. Tunnel 2 (T2)

Built during the 1850s, T2 is a deeper tunnel located in a suburban area. The tunnel lining cross-section forms a circular arc on top of curved sidewalls. It appears to have been originally lined with stone, but has been largely relined at some point with brick masonry. This masonry is in a relatively good condition with small areas of spalling.

3.1.3. Point cloud preparation

The obtained point clouds contain complete scans of the tunnel interior. These point clouds were manually cropped to remove the track bed, service cables and any vegetation growth to leave only points representing the tunnel lining. Isolated outliers were also removed manually. The resulting cloud had substantially larger point densities closer to the scanner locations. Therefore, to reduce the computational cost and reduce the impact of varying point densities on the results, the cloud was subsampled to minimum 4 mm point spacings. Further local outlier removal using statistical outlier removal was not undertaken, as it was found to reduce the number of points representing mortar and damaged areas, hindering the performance of the algorithm. Finally, the clouds were split into separate 10 m lengths for training and testing. Sections of these were used for assessment both of the overall workflow, and the neural network aspects.

3.2. Manual spalling segmentation

Both tunnels were assessed to NR/L3/CIV/006 (NR, 2016) by an engineer at Bedi Consulting Ltd using a manual workflow. The output of this analysis was used as the ground truth for assessment of the proposed automated workflow. The manual analysis was conducted by fitting a hypothetical undamaged surface to the tunnel lining through a combination of surface fitting and smoothing algorithms. Spalling depth was then measured as any outwards offset of points from this surface. Initially, areas of spalling, masonry joints and lining deformations were highlighted. The surface location was then refined using trial and error by adjusting the smoothing parameters until local spalling and masonry joints were visually distinguished from large scale tunnel deformations. Spalling locations and severities were determined using thresholding. A final visual analysis was conducted to remove masonry joints from the detected spalling areas.

4. Workflow development

The development and testing of each stage of the workflow is described here using the provided data as a case study. The workflow was implemented in Python and takes a point cloud cleaned of abnormalities and non-lining points as input. The cleaned T2 and T1 point clouds are visualised in Fig. 4.

4.1. Point cloud flattening

The first step of the workflow is to transform the 3D point cloud data such that it can be rasterised onto a 2D image compatible with the 2D semantic segmentation neural networks used for joint and damage segmentation. This must be conducted in a way that preserves the lining's local morphology. While the tunnel lining data is 3D, it forms a single surface, so a single surface projection method can be used. Due to its computational simplicity and broad similarity to a tunnel profile, a cylindrical projection was chosen. For most masonry tunnels, this should capture every part of the lining without occlusion, ensuring no data loss during the rasterisation step. While the deep learning stages of the workflow are designed to operate with distorted images, the morphology of the unrolled surface must be constrained to limit distortions in the rasterised image. As a result, the maximum deviation of the unrolled tunnel surface from a plane is set as 30 degrees to ensure that the deep learning models operating on the rasterised point cloud can be trained in a reasonable time with the provided volume of data. To ensure that this limit is not exceeded, the tunnel point cloud must first be split into sections where the alignment deviates a maximum of 30 degrees from a straight line. As the alignment curvature of T1 and T2 is less than this, the full sections are used.

For the case of tunnels on adhesion railways, the maximum incline is typically 4%, so the cylinder can be aligned horizontally, with the centre set as the centroid of the point cloud. Rotation of the cylinder in the horizontal plane is then set using principal component analysis of the cloud's convex hull. The direction of the 1st eigenvector is set as the tunnel centreline orientation.

Each tunnel point is projected onto the cylinder then unrolled using cylindrical co-ordinates. The offset of each point from the cylinder forms the offset depth parameter used for analysis in the subsequent stages of the workflow. Due to varying as-built profiles and long-term lining deformations, the unrolling produces an uneven surface for most tunnels as shown in Fig. 5. As a result, for the workflow to operate effectively, the neural network stages must be trained for a variety of uneven data. Steps taken to improve neural network generalisation performance can be applied to enable the networks to accurately segment local features from

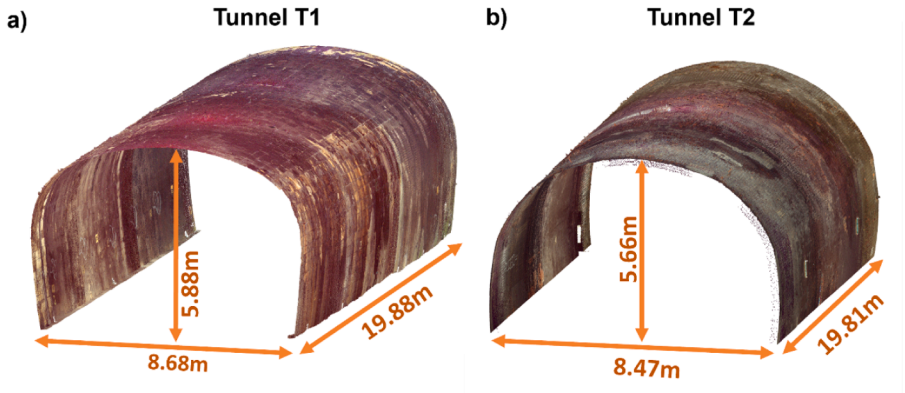


Fig. 4. Cleaned point clouds of T1 and T2 used as input to workflow (colour data shown to aid visualisation).

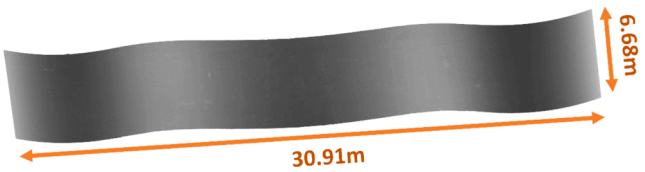


Fig. 5. A section of T1 unrolled using the cylindrical projection approach.

the wavy global profile of the offset depth. Provided that the relevant areas are still visibly prominent, this enables the workflow to operate on a range of tunnel lining profiles.

4.2. Masonry joint and damage segmentation

This part of the workflow uses one neural network to semantically segment masonry joint locations and another to segment locations of lining damage. The process used for designing each neural network is visualised within Fig. 6, while each stage is explained in detail in the following subsections.

4.2.1. Rasterisation

The offset of each point from the fitted cylinder was chosen as the parameter to rasterise into a 2D image compatible with the 2D neural network approach chosen. The work conducted within Smith et al. (2023) identified a combination of defect depth, normal variation, surface roughness and intensity parameters forming a 4-channel image to be the most effective, with a SegAN (Xue et al., 2017) generative adversarial segmentation network achieving the best segmentation performance. Despite this, the single channel defect depth parameter (depth offset) was adopted, as it was used in the best performing single channel network. A single channel network is substantially faster to train, with lower memory requirements. In addition, it enables the method to operate when colour or intensity data is not available or of poor quality.

For training and testing both neural networks, input data was taken from the unwrapped T1 and T2 point clouds. Each point offset value was projected vertically downwards onto a 2D plane and rasterised onto a 2D, 32bit single-precision floating-point format image. The resolution of the rasterised image can be manually set, although a minimum best case resolution with no data loss is automatically calculated. Points are binned into their nearest neighbour pixel. In the case where multiple points are assigned to the same pixel, then an average value is taken. Empty pixels receive a linearly interpolated value. The transformation from point cloud to image was recorded so that output images can be accurately projected back onto the initial point cloud to aid visualisation. A 11.7x13.9 m section of T1 and a 3.7x3.2 m section of T2 were taken for training the neural networks. These included 10,353,454 and 633,222 3D points respectively, before rasterisation. Because T2′s bricks are smaller than the predominantly stone blocks found within T1, more T1 than T2 data was used, with the generated images containing approximately 500 individual masonry blocks per wall. A 5x1.5 m section of T1 and a 0.9x2.3 m section of T2 were used for testing.

4.2.2. Synthetic data

There are three well documented methods of preparing the dataset and neural network to improve performance and generalisability once trained (Seib et al., 2020):

- Transfer learning by pretraining networks on large general datasets and tasks before fine-tuning for the relevant task and domain.
- Applying image augmentations to transform the dataset into a variety of realistic alternatives.
- Creating synthetic data with similar features to the real dataset and combining it with the real training data.

All three of these methods were used in this study. Synthetic data representing damaged masonry was generated and added to the training and validation sets. This data was created in 7 stages:

1. 1024x1024 images were created and random horizontal and vertical joint spacings were applied.
2. Masonry joints were modelled as normal distributions. Horizontal joints were placed first, followed by vertical joints between them. The vertically repeating vertical joint offset pattern was given a 50% likelihood of being randomised. Otherwise, a random standard masonry bonding pattern was selected.
3. Areas of damage were added to the images. Randomly placed control points were used to generate Bezier curves delineating the area of damage. These were defined as either representing spalling or efflorescence and can overlap each other. Damage areas representing spalling were curtailed at the joint locations.
4. Within an area of spalling, the depth of spalling was modelled as having a randomised uniform gradient that is generated by a distance transform within the boundary of the spalling area. Further deviations were added by introducing small areas with random offsets. These were smoothed into the image, by applying a Gaussian blur.
5. Efflorescence was modelled by taking the magnitude of small scale Perlin noise and subtracting it from the image. Perlin noise (Perlin, 1985) was developed for generating pseudorandom computer

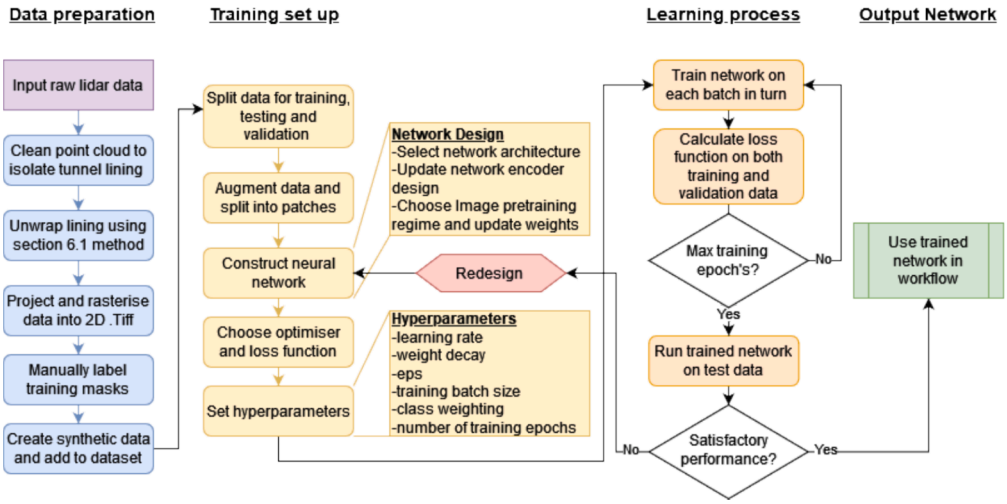


Fig. 6. Neural network training procedure.

graphics textures. It has a more organic appearance than standard noise functions and tunable levels of smoothness.
6. Large scale Perlin noise was applied to model both deformations of the lining and the overall gradient in the real training images caused by the tunnel unwrapping method.
7. Gaussian noise was applied to model the masonry roughness.

Realistic upper and lower bounds to the randomised parameters were obtained through trial and error with comparison to the T1 and T2 data. These are outlined in Table 2. A visualisation of the underside of a synthetically generated image is shown in Fig. 7 and a comparison between a real and synthetic data patch is shown in Fig. 8.

4.2.3. Masonry joint segmentation labelling

In order to train the masonry joint segmentation network, binary masks were drawn representing the ground truth joint locations. While typically for semantic segmentation tasks pixels are labelled individually, in this case often the areas of mortar are very thin, and the exact pixel location of a joint can be challenging to determine. Since the aim of this stage is to isolate each masonry block by determining the joint locations, ensuring that the full length of each joint is identified is more important than generating exact pixel wise segmentation of the width of the mortar area in each joint. To assist the network in predicting joint locations that are difficult to identify independently without knowledge of adjacent joint locations, the joint masks were created containing lines of constant thickness at the joint locations.

Table 2
Synthetic data parameter limits.

Stage	Description	Parameter	Lower Bound	Upper Bound
1	Block length	Number of pixels	50	350
1	Block height	Number of pixels	Block_length/2.6	Block_length/1.3
2	Horizontal offset between vertical joints	Number of pixels	0	Block_length
3	Bezier curve control points	Scale = 0.8	–	–
3	Bezier curve control points	Number of points	3	15
3	Generate Bezier curve	Rad	0	1
3	Generate Bezier curve	Edgy	0	10
4,5	Probability of damage	p = 0.7	–	–
4,5	Number of damaged locations	N	0	17
4,5	Depth of damage	Scale factor	0	5
4,5	Height of damage	Number of pixels	Block_length/6	Block_length*4
4,5	Width of damage	Number of pixels	Block_length/6	Block_length*4
6	Add small scale Perlin noise	Scale = 1 Octaves = 8 Persistence = 0.85 Lacunarity = 2.0 Variation = 0.3	–	–
6	Add large scale Perlin noise	Scale = 1 Octaves = 2 Persistence = 0.2 Lacunarity = 2.0 Variation = 0.3	–	–
7	Add Gaussian noise	Limit	0	0.15

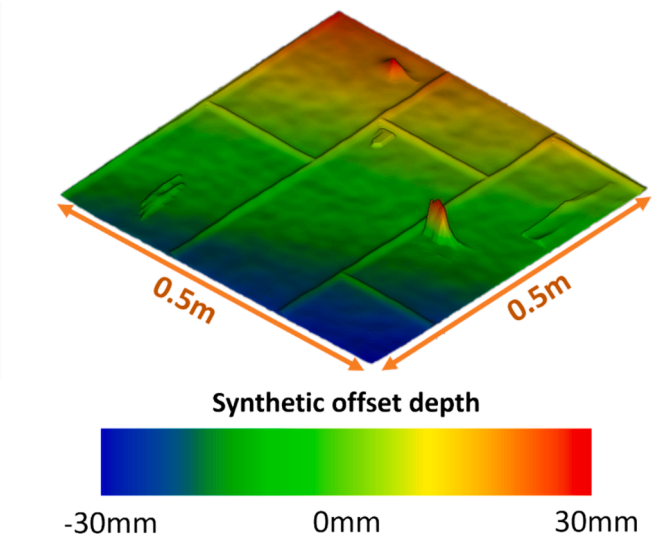


Fig. 7. 3D visualisation of area of synthetically generated masonry (offset depth axis stretched to aid visualisation).

4.2.4. Damage segmentation labelling

The aim of the damage segmentation network is to identify areas of damaged masonry in order to remove their pixels from each block. The training masks were created by defining damage as any area with an offset of 5 mm from a manually determined undamaged block surface. Areas of masonry joints were included. Since masonry joint locations are also identified and removed by the joint detection network, including joint detection in the damage detection network has no impact on the operation of the overall workflow.

4.2.5. Data augmentation

The training images were augmented to increase the effective amount of training data and improve the generalisability of the method. The parameters and probabilities of the transforms being applied were optimised to create a wide variety of training images, while ensuring that the images were representative of possible real world masonry configurations and conditions. The following augmentations were applied using the Albumentations Python library:

- Vertical and horizontal flips. Since masonry typically follows horizontally and vertically aligned patterns, these transformations produce realistic geometries.
- Elastic transform and grid distortion. These transformations were applied with a low intensity to model deformation of the masonry in the plane of the tunnel lining and non-flat masonry courses. Grid distortion was applied before the image cropping to better represent deformations over a larger scale.
- Random brightness shifts to represent varying depth magnitudes dependant on the diameter of the tunnel and the cross-sectional profile relative to a cylinder.
- Random contrast shifts to represent varying magnitudes of masonry homogeneity and varying mortar depths.
- Perlin noise to represent deformations perpendicular to the plane of the lining and those resulting from the cylindrical unwrapping process in different tunnel geometries.
- Small levels of gaussian noise representing possible variations of true point locations within the accuracy limit of the lidar equipment.

An example of the augmentation workflow on a section of T2 is shown in Fig. 9 and optimum augmentation parameters are shown in Table 3.

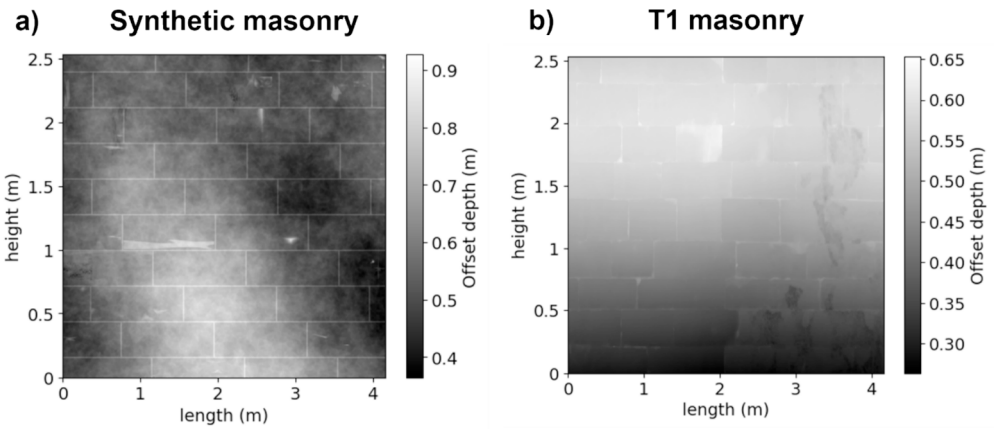


Fig. 8. 2D depth offset images used for neural network training.

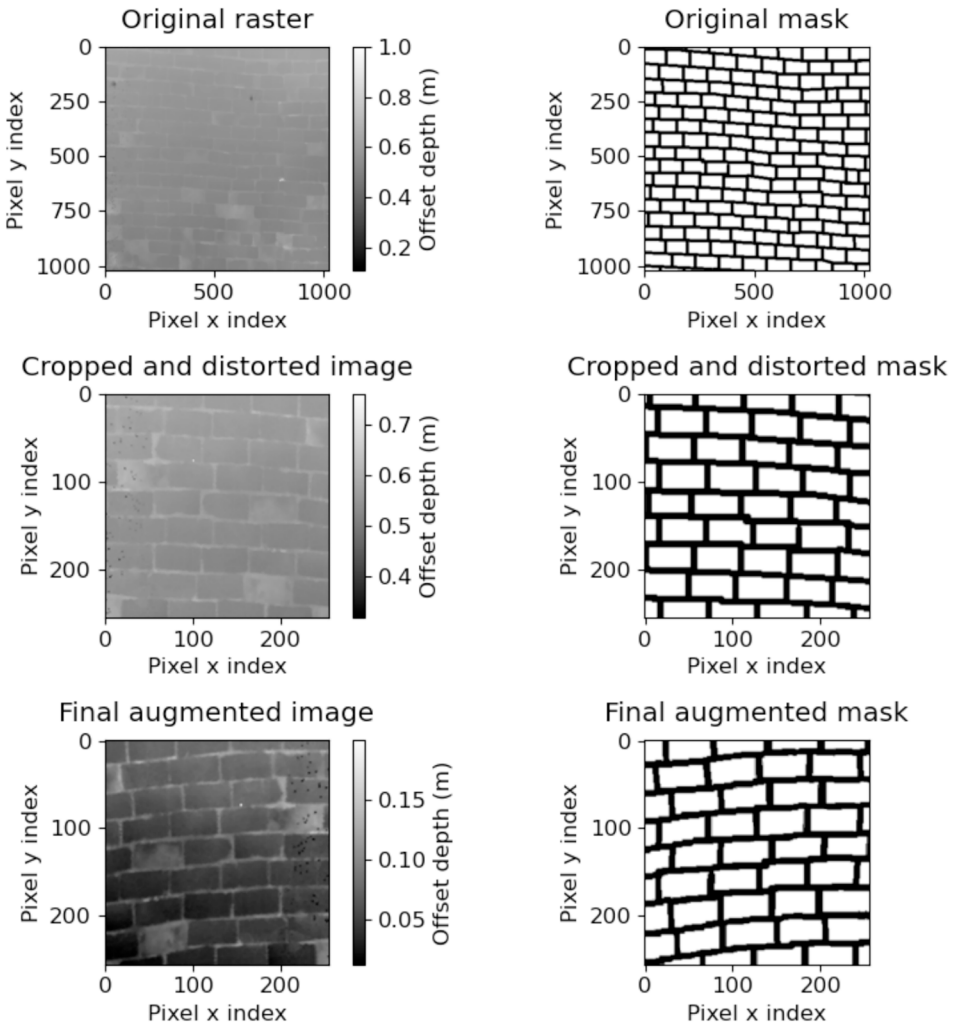


Fig. 9. Crop and augment workflow with all possible augmentations applied to a patch of T2.

4.2.6. Neural network design

In order to ensure that the optimum neural network design was chosen for each task, a variety of neural network architectures were assessed. The U-Net developed by Ronneberger et al. (2015) and the U-Net++ adaption by Zhou et al. (2018) were tested with different encoders as their backbone. A comparison of the different encoders is shown in Table 4. While there are many architectures that achieve state of the art semantic segmentation performance in a range of domains, these networks were chosen to examine the workflow potential due to their popularity and past performance on masonry images (Ibrahim et al., 2020; Loverdos and Sarhosis, 2022; Smith et al., 2023). The workflow is designed such that other neural networks could replace these following developments in semantic segmentation technology.

The U-Net is one of the most popular architectures for semantic

Table 3
Data augmentation parameters.

Augmentation	Probability	Parameter
Horizontal Flip	0.5	–
Vertical Flip	0.5	–
Elastic Transform	0.5	Alpha = 500, Sigma = 40, Alpha_affine = 8
Grid Distortion	0.5	Num_steps = 5
Random Brightness	0.7	Brightness_limit = 0.5
Random Contrast	0.7	Contrast_limit = 0.2
Gaussian Noise	0.7	Var_limit = [0,0.15]
Perlin Noise	0.4	Variation = 0.3, Octaves = 2, Persistence = 0.2, Lacunarity = 2.0

Table 4
Encoders assessed.

Encoder	No. of parameters	No. of layers	Date	Description
VGG19	20M	19	2014	This was the first very deep CNN and created the baseline CNN encoder architecture. Greater depth without an unreasonable number of parameters is achieved by using small 3x3 convolution filters
ResNet34	21M	34	2015	ResNet introduced residual blocks into the encoder design. These add skip connections between layers that bypass image convolutions allowing higher level convolutions to remain relevant deeper into the network. This allows deeper networks as it helps to overcome the vanishing gradient problem.
ResNet101	45M	101	2015	This is a deeper version of resnet34 and was tested to evaluate the effect of using a network with more trainable parameters.
ResNet18	11M	18	2015	This is a shallower version of resnet34 and was tested to evaluate the effect of using a network with fewer trainable parameters.
ResNext50_32x4d	22M	50	2017	Introducing the variable of cardinality, ResNeXt adds a split-transform-merge strategy to ResNet
MobileNet_v2	2M	15	2018	With only 2 million parameters, Mobilenet was developed to operate in low memory environments by introducing inverted bottleneck residual modules.

segmentation, as it was the first to demonstrate excellent results with transfer learning and only limited domain specific training. It follows a standard encoder-decoder design, with added skip connections between the encoder and decoder. These enable the decoder to incorporate spatial details from the input at the relevant encoder level with feature information from the deeper levels of the network. This produces more accurate pixelwise segmentation at class boundaries than designs without skip connections.

The U-Net++ is a more complex development of the original U-Net design which was developed to give the network more flexibility. The design fills in the centre of the U-Net with further convolutional and upsampling modules that enables it to perform in effect as a series of nested U-Nets of varying depths. This helps the overall network better characterise features at different scales.

While parameter gradients are calculated through backpropagation, optimisers provide the method for iterating towards a set of parameters that minimises the loss function. Three different optimisers were trialled, SGD, ADAM (Kingma and Ba, 2014) and ADAMW (Loshchilov and Hutter, 2017). A soft dice loss function was chosen, as it has been shown to be effective for semantic segmentation tasks with large class imbalances (Sudre et al., 2017).

4.2.7. Training configuration

The neural networks were trained using the PyTorch library on a Nvidia GTX 970 GPU, with 4 GB of VRAM. The system used an Intel i7 2600k CPU with 16 GB of RAM. The created training images were split between training and validation sets in a 4:1 ratio. There is a compromise between the speed of training, the training image dimensions used and the convergence of the training loss. If a larger image dimension is used for training, then the network will be better able to capture larger scale image features, such as whole masonry block geometries. However, the number of training images will be reduced, potentially leading to overfitting and poor generalisation performance. As a result, it was determined that randomly sized and located crops of 1024x1024 patches would be taken and then rescaled to 256x256 for training. This enables a batch size of 4, while training the network on both larger and smaller scale features.

For each task, the U-Net and U-Net++ were trained with the selected encoders for a maximum of 500 epochs. A grid search was performed on the network, encoder and optimiser combinations, by selecting the top performing trained model at the epoch with the lowest validation loss for testing. For the task of masonry joint segmentation, the network hyperparameters of the best performing networks were optimised using trial and error. The selected hyperparameters are shown in Table 5 and were reused for the damage segmentation task.

4.2.8. Joint segmentation results

Joint segmentation results are shown in Table 6. The Intersection over Union score (IoU) was used to quantify semantic segmentation performance. This is calculated by dividing the union of predicted and ground truth classified pixels on an image, by their intersection. A visualisation of intersection and union is shown in Fig. 10. This rewards true positives and punishes both false positives and false negatives. A score of 0.5 or above is considered a standard performance target.

All of the fully optimised networks in Table 6 achieve acceptable performance. These results are in line with or superior to those produced by (Smith et al., 2023) using a manually flattened point cloud for the same task. As their method had no distortions or offsets in the training and testing images, these results show that appropriate augmentations and synthetic data enable the networks to learn to work effectively with minorly deformed raster images. Although the larger ResNet34 encoder performs better on the more complex T1 task, MobileNet_v2 performs best on T2 and on average over both tunnels. The smaller MobileNet_v2

Table 5
Selected hyperparameters.

Parameter	Value
Learning rate	0.001
Image dimension	256 x 256
Weight decay	0
Pretrain	ImageNet
Eps	0.00000001
Dropout	0
Learning rate	0.001
Batch size	4
Loss function	Soft Dice

Table 6
Top performing U-Net and U-Net++ models for masonry joint segmentation.

Tunnel	T1		T2		Combined average	
Network	U-Net	U-Net++	U-Net	U-Net++	U-Net	U-Net++
IoU	0.513	0.5554	0.600	0.635	0.569	0.570
Encoder	ResNet34	ResNet34	MobileNetv2	MobileNetv2	MobileNetv2	MobileNetv2
Optimiser	Adam	Adam	AdamW	Adam	AdamW	Adam
Number of trainable parameters	24430097	26072337	6628369	6824145	6628369	6824145

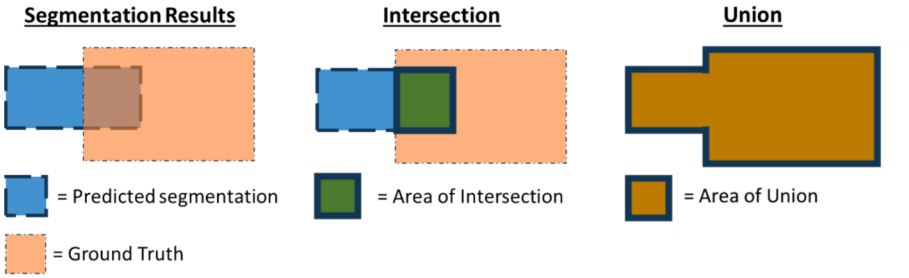


Fig. 10. Intersection over Union (IoU) visualisation.

generalises better due to not overfitting to the training data. The better flexibility of U-Net++ allows it to outperform the U-Net model on average.

To help select the best network for both datasets, it is important to understand the typical training pathways. Fig. 11 shows the output of the ResNet34 U-Net when trained with the ADAM optimiser on the test images at different epochs. The network initially generates a large proportion of false positive pixels for both T1 and T2 when trained for only 40 epochs. T1 requires more training than T2 for the IoU to plateau and final performance is better on T2 than T1. This demonstrates the increased training requirements caused by the complexity of the stone masonry in the T1 dataset, compared to the more regular and less damaged masonry in the T2 dataset. The best performer overall, the U-Net++ MobileNet_v2 network, is chosen as the optimum network.

4.2.9. *Damage segmentation results*

The top performing networks for damage segmentation are shown in Table 7. As the joint locations were determined in the previous stage of the workflow, the joints were masked out of the images prior to IoU calculation. The impact of this is shown in Fig. 12. A U-Net++ with a MobileNet_v2 encoder was determined to be the best overall and is chosen for the workflow, as it achieves an IoU of 0.3890 on average between T1 and T2. The network is shown in Fig. 12 to identify the locations of the largest areas of spalling. Damage segmentation performance is lower than joint segmentation performance, although is qualitatively considered adequate due to the robustness of the overall workflow.

4.3. *Block isolation*

Following joint detection, it is necessary to identify the location of each masonry block. However, since small areas of masonry joint are often undetected by the masonry joint detection network, many blocks do not have the complete closure required for block isolation. (Ibrahim

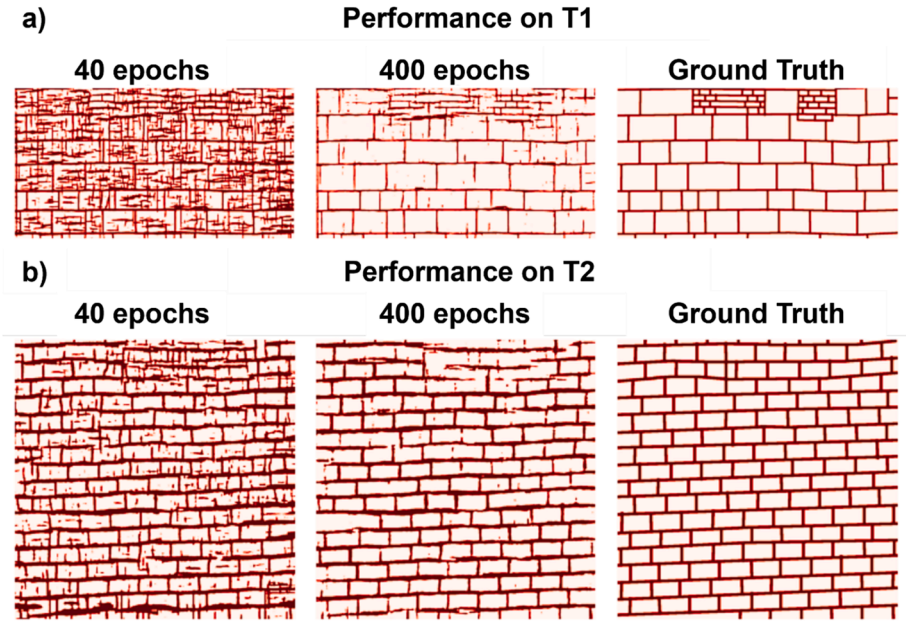


Fig. 11. ResNet34 U-Net masonry joint semantic segmentation prediction on test data after varying numbers of epochs.

Table 7
Top performing U-Net and U-Net ++ models for masonry damage segmentation.

Tunnel	T1		T2		Combined average	
Network	U-Net	U-Net++	U-Net	U-Net++	U-Net	U-Net++
IoU	0.413	0.459	0.321	0.319	0.367	0.389
Encoder	ResNext50	MobileNet_v2	ResNext50	MobileNet_v2	ResNext50	MobileNet_v2
Optimiser	Adam	Adam	Adam	Adam	Adam	Adam
Number of trainable parameters	31986705	6824145	31986705	6824145	31986705	6824145

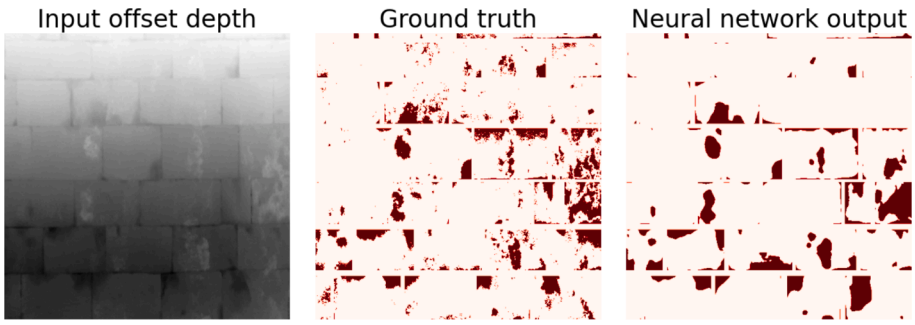


Fig. 12. Section of masonry damage semantic segmentation on T1.

et al., 2020) tackled this by applying watershed segmentation to the output of the joint detection network. Their method was trialled; However, it was necessary to manually determine different watershed algorithm parameters when applied to each of T1 or T2. As a result, a simpler geometric joint fitting algorithm was developed. This method relies on masonry being oriented with the horizontal and vertical joints being close to perpendicular. Lines are fitted in the positions of the masonry course joints if the output of the joint detection network produces a mask that covers at least 80% of a horizontal line across an image. Then between these lines, vertical lines are drawn at each location there is a greater than 90 % connection between each masonry course. Although the method fails at locations where the masonry type changes from brick to stone or if large deformations are present, overall the joint detection IoU increases for the top performing U-Net++ MobileNet_v2 network trained with ADAM. The performance on T1 increases from 0.5255 to 0.5361 and the output is shown in Fig. 13. A key restriction on this method is that it can only operate when masonry vertical and horizontal joints are perpendicular and there is limited in-plane deformation over each image patch. It fails at boundaries between blocks with different heights.

A connected components analysis is used to isolate each individual block instance after joint fitting has ensured that each block is completely enclosed. This uses the 8-way connectivity Spaghetti labelling algorithm developed by Bolelli et al. (2020) and was implemented using the OpenCV library.

4.4. Plane fitting and spalling severity calculation

The final step of the workflow is to identify the spalling severity of each point on the original 3D point cloud. There are 6 stages, each demonstrated in Fig. 14:

1. All pixels representing areas of masonry joints or lining damage are removed from the raster image generated in section 4.2.1, using the masks generated by the neural networks in sections 4.2.7 and 4.2.8.
2. Using the pixel coordinates of each block calculated in section 4.3, the remaining pixels within each block are isolated. Stage 3 is then applied to each block individually. If the proportion of a block's pixels that contain damage is greater than the damage proportion threshold (DPT) parameter, then 3(a) is conducted, otherwise 3(b) is applied.
3. (a) If a block is not too badly damaged, a best fit plane is applied to the depth values of the remaining pixels of the block. Each block is iterated over individually using a 2D linear least squares fit. This is evaluated using the LAPACK dgelsd routine ("LAPACK: dgelsd," n. d.). The resulting best fit planes create an image representing the location of the hypothetical unspalled surface.
3. (b) It is determined that the block is too heavily damaged to reliably calculate the block's undamaged surface using its points alone. The block's undamaged surface is therefore interpolated from adjacent blocks that are not heavily damaged. The nearest horizontally adjacent blocks that have less than the defined DPT% damage are identified each side of this block. The right edge pixels of the left side block and the left edge pixels of the right-side block are taken from those blocks' fitted undamaged planes and an average gradient is calculated from the top to the bottom of the block. With the vertical gradient set, the horizontal gradient of the block is set as a best fit between the two adjacent undamaged blocks.
4. The pixel offset values from step 1 of the overall workflow are subtracted from the fitted plane location values to produce a map of offsets from the undamaged surface. Pixels with positive values show spalling depth.
5. Depending on whether the wall consists of stone or brick masonry, the spalling severity categories are obtained from the table shown in

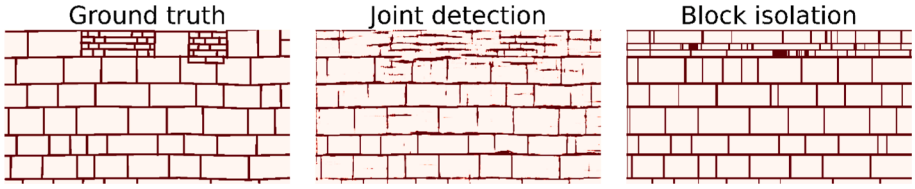
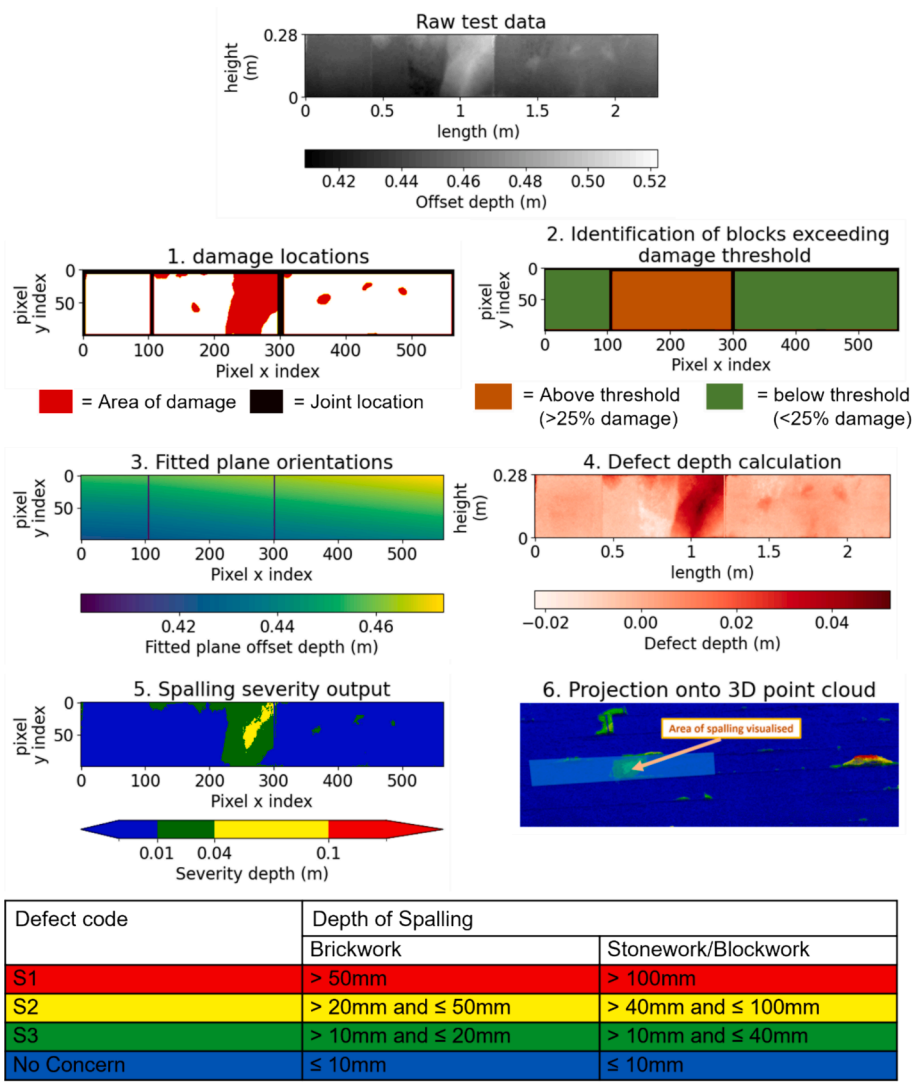


Fig. 13. Masonry joint segmentation before and after application of joint fitting algorithm.



Defect code	Depth of Spalling	
	Brickwork	Stonework/Blockwork
S1	> 50mm	> 100mm
S2	> 20mm and ≤ 50mm	> 40mm and ≤ 100mm
S3	> 10mm and ≤ 20mm	> 10mm and ≤ 40mm
No Concern	≤ 10mm	≤ 10mm

Fig. 14. Top: Output of each stage of the plane fitting and severity thresholding algorithm for a three block section of T1. Bottom: Network Rail masonry spalling severity thresholds (No Concern category non-standard, but generally accepted).

Fig. 14 (NR, 2016) and applied as thresholds to the offset image. This creates maps showing the locations of masonry that have each level of spalling severity.

6. Using the 3D point index to pixel location mapping saved during the rasterisation step of the complete workflow, the spalling severity maps are projected back onto the relevant points on the original 3D point cloud and set as a scalar field for easy visualisation. The images generated during the intermediate steps, such as joint positions, areas of damage and undamaged surface plane locations can also be projected onto the point cloud in this way.

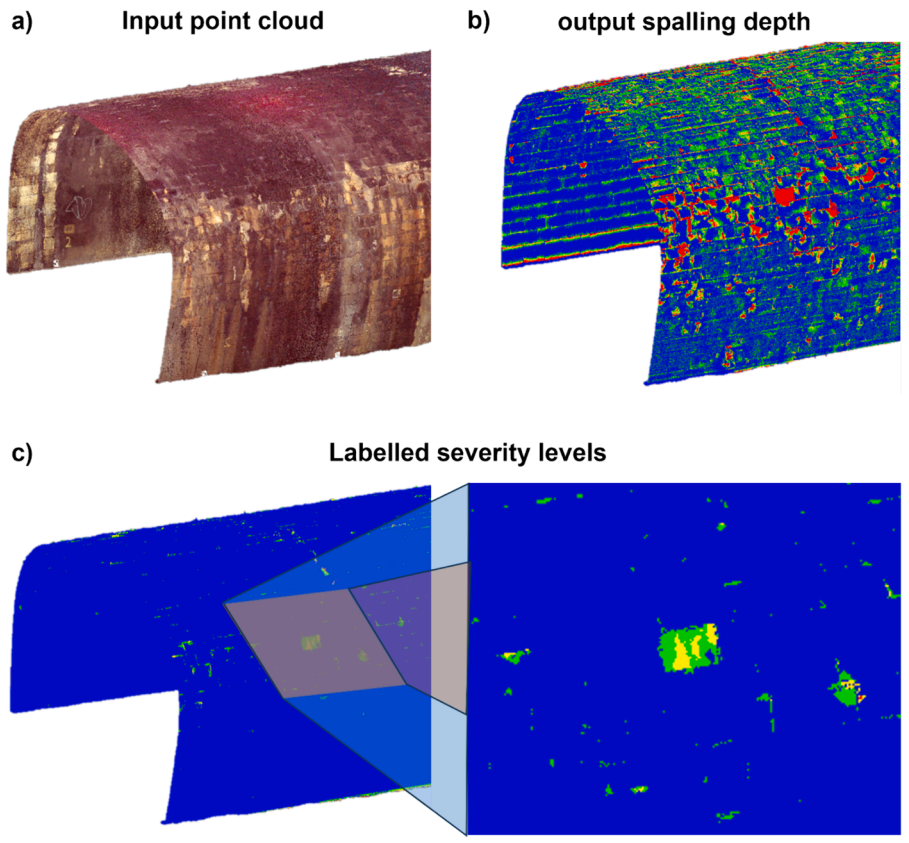
5. Workflow results and discussion

The full workflow takes approximately 30 min to evaluate on a 10 m length of T1 with no manual input, using default parameters. It produces a final 3D visualisation of spalling severity locations as shown in Fig. 15. With the manual workflow output set as the ground truth, Table 8 shows the IoU of the automated workflow for each level of spalling severity segmented. As performance is evaluated on the damage locations reprojected onto the point cloud, the IoU is evaluated per point. For T1, S1 performance is greater than S2 and S3, although all IoU scores are acceptable.

Performance on T2 is worse than T1. For S3 spalling, this is most likely because of poorer damage segmentation performance on T2 due to the smaller size and resulting lower resolution of each masonry block. The accuracy of the lidar scanner used appears to be insufficient in some areas, with substantial levels of noise limiting the neural network masonry joint segmentation performance. The Leica ScanStation P30/P40 series is a typical lidar scanner used for tunnel condition assessments. They have a 3D point location accuracy of 3 mm at 50 m range. Since each scan would usually be taken of a section of tunnel shorter than 40 m, and 3 mm is less than the minimum S3 spalling threshold of 10 mm, the range does not cause a major issue. However, the point location accuracy is one third of the range of values used for thresholding S3 level spalling (20 mm > S3 > 10 mm) in brickwork, so the accuracy will have a notable effect on the level of S3 spalling segmented. Furthermore, some tunnels may have the mortar joints offset only a couple of millimetres from the masonry face, so the lidar accuracy may be insufficient for effective joint segmentation.

The small amount of S2 and S1 spalling within the T2 test data yields the S1 and S2 results for T2 inconclusive.

For both tunnels, S3 spalling is substantially more extensive than S1 and S2. It is also more difficult to manually segment, as the small 10 mm threshold makes it challenging to identify against noise. For the automated workflow, to achieve a similar level of S3 segmentation performance, the damage segmentation neural network performance needs to



Defect code	Depth of Spalling	
	Brickwork	Stonework/Blockwork
S1	> 50mm	> 100mm
S2	> 20mm and ≤ 50mm	> 40mm and ≤ 100mm
S3	> 10mm and ≤ 20mm	> 10mm and ≤ 40mm
No Concern	≤ 10mm	≤ 10mm

Fig. 15. Progression of the overall workflow (a) Input point cloud (colour data shown). (b) Output of workflow showing defect depth values. (c) Spalling severity output (key in table).

Table 8
Pointwise IoU score for different severity categories for each tunnel.

Plane fitting damage proportion threshold	IoU for spalling greater than S1		IoU for spalling greater than S2		IoU for spalling greater than S3	
	T1 (>100 mm)	T2 (>50 mm)	T1 (>40 mm)	T2 (>20 mm)	T1 (>10 mm)	T2 (>10 mm)
20% DPT	0.482	None	0.494	0.122	0.500	0.307
40% DPT	0.648	None	0.504	0.126	0.504	0.317
60% DPT	0.648	None	0.507	0.237	0.498	0.432
80% DPT	0.675	None	0.540	0.045	0.406	0.055

be greater than that for S2 and S1. This is because small changes in the fitted plane location will have a greater impact on the predicted S3 spalling locations.

Conversely, poor S1 and S2 performance is typically caused by the spatial extent of spalling being greater than around S3. This leads to adjacent blocks being used for plane fitting, an approximation which impacts the accuracy of the output defect depth map. S1 and S2 severities have a substantially greater significance to the health of the structure than S3 and typically require imminent repairs after an inspection. For S3 spalling, during manual analysis small quantities are typically ignored and larger areas do not need accurate localisation. Only a notification of the presence of S3 in a broad area is required, to mark it for future inspections. Table 9 shows a comparison of the extent of spalling within each severity category against the ground truth. As expected, the performance of total S3 spalling extent is greater for both tunnels than the pixelwise segmentation performance and the workflow achieves adequate S3 localisation accuracy against that typically required for both T1 and T2. For all severity levels, T1 produces a more

Table 9
Measure of the difference in predicted spatial quantity of spalling, calculated within each severity category by taking the total number of predicted spalling points minus the number of ground truth points and dividing by the total number of ground truth spalling points within each test segment.

Deviation in predicted spatial quantity of spalling					
S1		S2 and S1		S3, S2 and S1	
T1 (>100 mm)	T2 (>50 mm)	T1 (>40 mm)	T2 (>20 mm)	T1 (>10 mm)	T2 (>10 mm)
−18.0%	N/A	−37.3%	+91.5%	−31.4%	−7.98%

conservative segmentation, with more false negatives than false positives, while the reverse is true for T2.

The damage proportion threshold (DPT) is an important parameter that must be optimised to maximise workflow performance. Higher DPT values would be expected to perform better, as this reduces the number of blocks that must approximate their fitted unspalled surface from neighbouring blocks. However, a higher DPT also places more emphasis on points close to the edge of damaged areas for unspalled face plane fitting. The damage segmentation is more likely to be inaccurate close to the edges of damaged areas, so in these areas a higher DPT will use more points that are likely to be incorrect for unspalled plane fitting. As a result, a lower DPT should have a moderating influence, reducing segmentation accuracy, but preventing large error magnitudes on the fitted planes and defect depth maps. Ideally, the performance of different tunnels would be robust to variations in the selected DPT values. A DPT value sensitivity analysis was therefore conducted, and the key results are shown in Table 7. While the IoU drops significantly for some very low or high values of DPT, overall the IoU is not considerably sensitive to the DPT chosen. For T1, a higher DPT yields better performance on severe S1 spalling, while for S3 a DPT of around 40% is optimal. This is because for deeper spalling, the damage detection network is likely to achieve a more accurate segmentation. As a result, the remaining

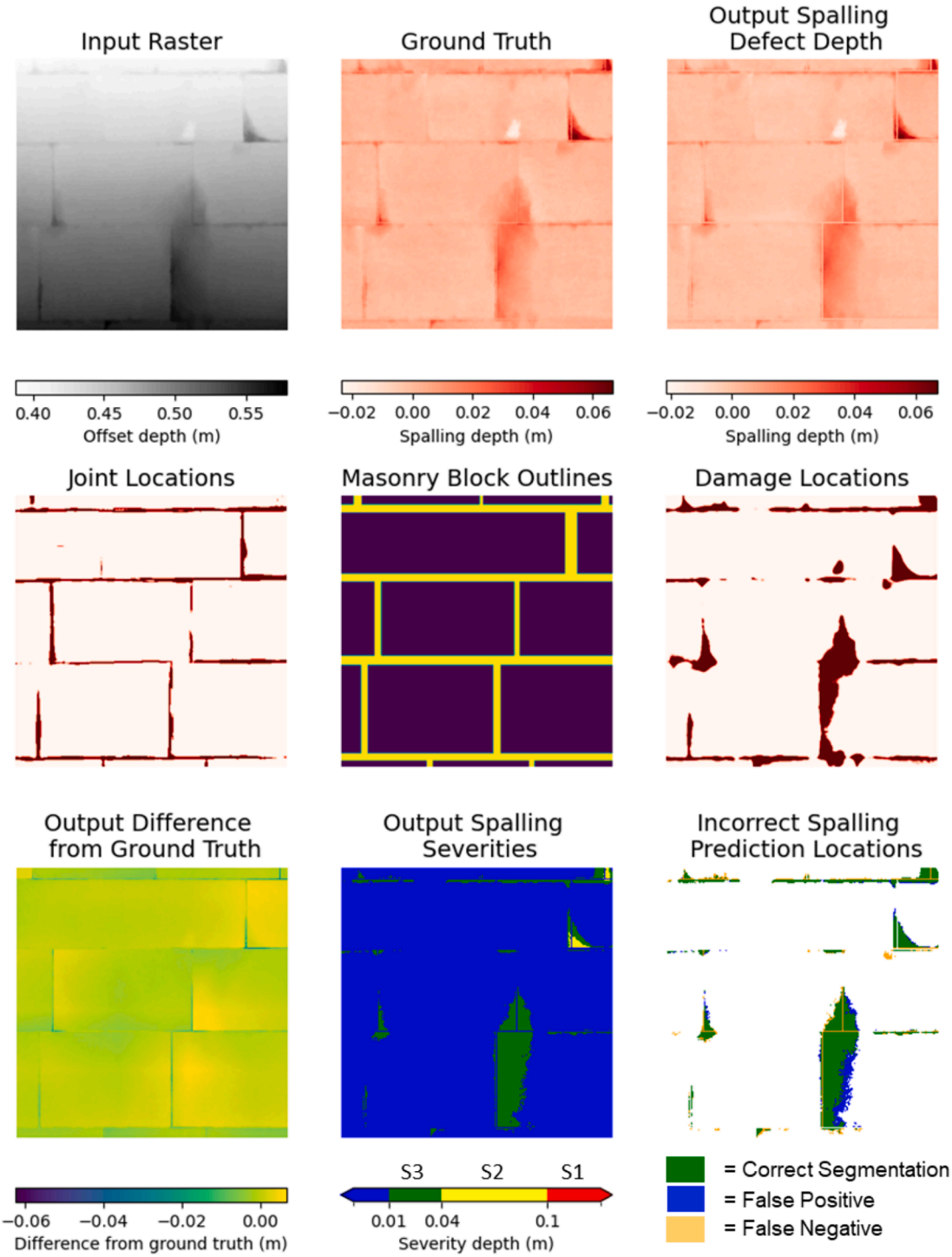


Fig. 16. Section of T1 with good algorithm performance.

undamaged points have a higher level of certainty, so the plane fitting method can still achieve accurate results with fewer points. A DPT of 60% achieves acceptable results for each severity category of T1.

An in-depth comparison of workflow performance on two areas of tunnel is visualised in Figs. 16 and 17. A comparison of the algorithm output on the entire testing dataset is included in Appendix A. The area of tunnel in Fig. 16 shows a qualitatively good segmentation with only small errors in the extent of each area of damage and no falsely classified regions of damage. There are also no regions where the difference between the output and ground truth has a significant magnitude.

Fig. 17 shows a typical cause of algorithm failure in T2. A noisy area in the top left of the image is too challenging to manually segment any defects or masonry joints. The block and damage segmentation neural networks both fail here. However, despite good performance by both neural networks outside of this area, some of the unspalled face planes are poorly fitted to the right of this area. This is due to the joint detection network not placing a joint at the edge of the noisy area, so the unspalled block face plane is fitted using some of the noisy points. In addition, there is a false negative area of spalling near to the middle of the image. This is caused by too many false negatives within the damage segmentation neural network causing a underprediction of the spatial extent of damage. As it is challenging to determine the accuracy of the manual severity assessment output, it is possible that in some areas the automated method is producing more accurate results. Additionally, unlike

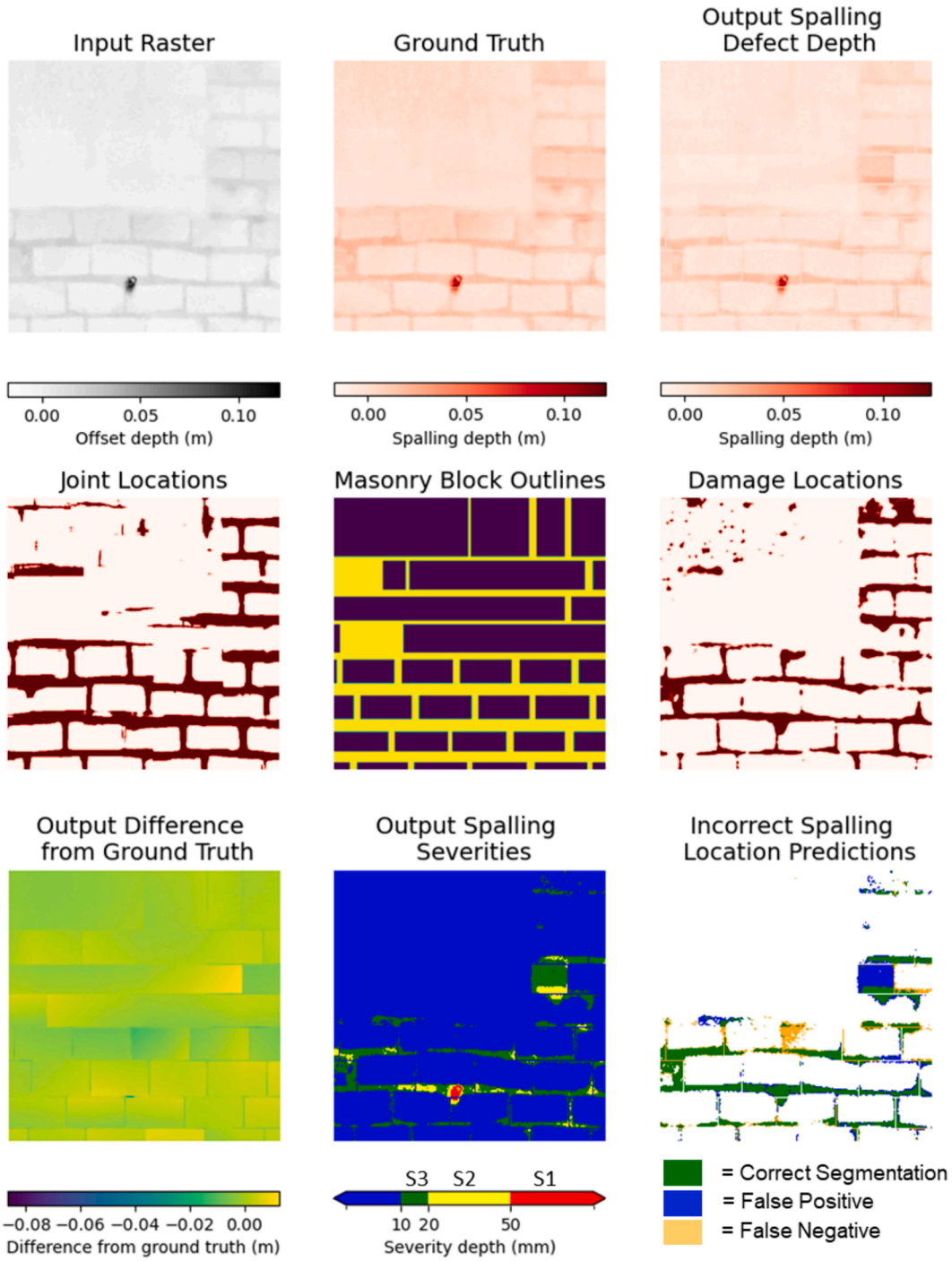


Fig. 17. Section of T2 showing poor algorithm performance.

the manual method, the algorithmic nature of the automated method ensures that the results are repeatable and understandable.

In order to better understand how the number of false positive and false negative predictions can be balanced, it is necessary to determine whether adjusting the defect depth threshold for each severity category could achieve a higher IoU. Fig. 18 shows how the optimal threshold varies for determining the amount of spalling that is greater than a corresponding defect depth target representing a hypothetical severity level within T1. The peak IoU appears at a threshold value lower than the semantically expected one for every severity level. This suggests that the workflow has too many false negative predictions, as a lower threshold would display a larger area of spalling. This could be counteracted by empirically lowering the threshold for each spalling severity level.

Overall, accuracy could be improved by encouraging the damage and joint segmentation CNNs to generate fewer false negatives, but more false positives during training. This would prevent unspalled block face planes from being fitted to potentially damaged areas of masonry, or to incorrect blocks. However, too many false positives would result in not enough undamaged points remaining within each block for accurate undamaged face plane fitting.

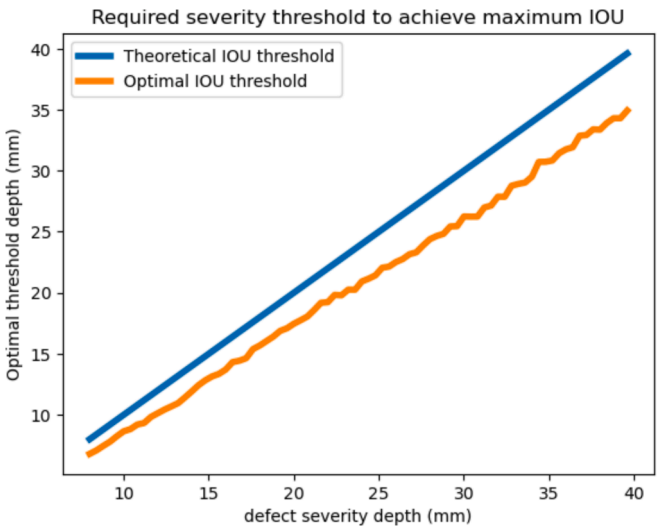


Fig. 18. Defect depth threshold that achieves optimal performance for semantically segmenting each target level of defect depth within T1.

6. Concluding remarks

This paper provides a proof of principle workflow for a fully automated method of masonry spalling detection, localisation and severity classification in masonry lined tunnels. The method uses a novel block isolation and undamaged surface fitting scheme to identify the depth of masonry spalling. Operating on 3D point cloud data circumvents the limitations of current state of the art photograph-based deep learning masonry damage segmentation methods that are unable to reliably determine spalling boundaries on noisy data. As a result, our method expands beyond masonry damage location segmentation to consider automation of masonry spalling depth map generation and masonry spalling severity classification. The benefits of this are twofold:

1. Accurately generating spalling severity maps enables asset managers to easily prioritise maintenance work and track the progression of damages between inspections.
2. Creating a spalling depth map enables the impact of spalling to be easily removed from a tunnel lining point cloud. Without the clutter of surface damages, an Engineer can then more easily analyse the location of structurally critical larger scale lining deformations.

As one of the most time-consuming desk-based tasks for the assessing engineer, this workflow has the potential to provide substantial labour savings to the overall condition assessment process. By combining geometrical and deep learning-based methods, the workflow is robust to localised areas of unclear data and provides explanation to the output. The method produces multiple intermediate deliverables including joint locations, damage locations, a record of the co-ordinates of each masonry block, and the location of a predicted unspalled masonry surface that may be utilised for lining deformation analysis. These can all be visualised on the original 3D point cloud, enabling each stage of the workflow to be fully scrutinised.

The workflow has potential application to masonry tunnel condition assessment worldwide and may be further modified for use with other masonry structures, such as masonry arch bridges. It was trained and tested on data from two different tunnels in the UK and was shown to perform particularly well overall on the stone lined tunnel test data. High severity spalling is captured well, however lower severity spalling has more uncertainty due to the higher accuracy required from the block isolation and damage detection steps in addition to greater uncertainties in the benchmark manual results.

Before a similar workflow can be applied within industry, there are three areas where further research is required. Firstly, an acceptable level of accuracy for an automated method needs to be determined. It is known that human assessors can obtain differing condition assessment results, however more research needs to be conducted to determine whether an automated method would fit acceptably within the existing levels of uncertainty. Secondly, as this study trained and tested the machine learning aspects of the workflow on only 2 different tunnels, it is unknown how well the trained models would generalise to other masonry tunnels. Although the data contained both stone and brick masonry and was augmented to mimic a variety of other masonry conditions, historic masonry tunnels can contain a wide variety of different materials, levels of damage and geometries. In addition, while the method is trained for deviations from a cylindrical profile of up to 30 degrees, it is unknown how well the method will perform when larger deformations are present. The geometrical limitations of the vertical and horizontal joint fitting method also needs to be addressed. Finally, research needs to be conducted into determining and fulfilling practical considerations to further reduce the required operator input. Required point cloud densities, processing power, development of a graphical user interface, and better adaption of the method to differing tunnel geometries should be investigated.

CRediT authorship contribution statement

Jack Smith: Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Chrysothemis Paraskevopoulou:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Anthony G. Cohn:** Writing – review & editing, Supervision. **Ryan Kromer:** Writing – review & editing, Supervision. **Anmol Bedi:** Supervision, Resources. **Marco Invernici:** Investigation.

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.

Data availability

The authors do not have permission to share data.

Acknowledgements

This project was funded by an EPSRC Environment Doctoral Training Partnership grant EP/T517860/1, project number 2601289: Stability assessment for sustainable and resilient tunnelling using AI. The authors would like to thank Network Rail and Bedi Consulting Ltd. for collecting and sharing point cloud data of operational railway tunnels.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.tust.2024.106043.

References

Allen, A., Paraskevopoulou, C., Smith, J., Bedi, A., Invernici, M., 2023. Utilising remote sensing to digitally map discontinuities in tunnelling. Expanding Underground - Knowledge and Passion to Make a Positive Impact on the World- Proceedings of the ITA-aites World Tunnel Congress, WTC 2023 2381–2389. DOI: 10.1201/9781003348030-286/Utilising-Remote-Sensing-Digitally-Map-Discontinuities-Tunnelling-ALLEN-Paraskevopoulou-Smith-Bedi-Invernici.

Atkinson, C., Paraskevopoulou, C., Miller, R., 2021. Investigating the rehabilitation methods of Victorian masonry tunnels in the UK. Tunn. Undergr. Space Technol. 108, 103696 https://doi.org/10.1016/J.TUST.2020.103696.

Bolelli, F., Allegretti, S., Baraldi, L., Grana, C., 2020. Spaghetti Labeling: Directed Acyclic Graphs for Block-Based Connected Components Labeling. IEEE Trans. Image Process. 29, 1999–2012. https://doi.org/10.1109/TIP.2019.2946979.

Bolourian, N., Nasrollahi, M., Bahreini, F., Hammad, A., 2023. Point Cloud-Based Concrete Surface Defect Semantic Segmentation. J. Comput. Civ. Eng. 37 https://doi.org/10.1061/JCCEE5.CPENG-5009.

Brackenbury, D., Dejong, M., 2018. Mapping Mortar Joints in Image Textured 3D Models to Enable Automatic Damage Detection of Masonry Arch Bridges. 17th International Conference on Computing in Civil and Building Engineering.

Che, E., Jung, J., Olsen, M.J., 2019. Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art. Review. https://doi.org/10.3390/s19040810.

Chiu, Y.C., Wang, T.T., Huang, T.H., 2015. Investigating continual damage of a nineteenth century masonry tunnel. https://doi.org/10.1680/feng.13.00030 167, 109–118. DOI: 10.1680/FENG.13.00030.

Dais, D., Bal, İ.E., Smyrou, E., Sarhosis, V., 2021. Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. Autom. Constr. 125 https://doi.org/10.1016/J.AUTCON.2021.103606.

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 https://doi.org/10.1016/J.CONBUILDMAT.2022.129238.

Dong, C.Z., Catbas, F.N., 2021. A review of computer vision–based structural health monitoring at local and global levels. Struct Health Monit 20, 692–743. DOI: 10.1177/1475921720935585/ASSET/IMAGES/10.1177_1475921720935585-IMG6.PNG.

Dong, Y., Wang, J., Wang, Z., Zhang, X., Gao, Y., Sui, Q., Jiang, P., 2019. A Deep-Learning-Based Multiple Defect Detection Method for Tunnel Lining Damages. IEEE Access 7, 182643–182657. https://doi.org/10.1109/ACCESS.2019.2931074.

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 https://doi.org/10.1016/J.TUST.2023.105107.

Flah, M., Suleiman, A.R., Nehdi, M.L., 2020. Classification and quantification of cracks in concrete structures using deep learning image-based techniques. Cem. Concr. Compos. 114, 103781 https://doi.org/10.1016/J.CEMCONCOMP.2020.103781.

Gao, X., Jian, M., Hu, M., Tanniru, M., Li, S., 2019. Faster multi-defect detection system in shield tunnel using combination of FCN and faster RCNN. DOI: 10.1177/1369433219849829 22, 2907–2921. DOI: 10.1177/1369433219849829.

Grandio, J., Riveiro, B., Soilán, M., Arias, P., 2022. Point cloud semantic segmentation of complex railway environments using deep learning. Autom. Constr. 141, 104425 https://doi.org/10.1016/J.AUTCON.2022.104425.

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Trans. Pattern Anal. Mach. Intell. 43, 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434.

Hallee, M.J., Napolitano, R.K., Reinhart, W.F., Glisic, B., 2021. Crack Detection in Images of Masonry Using CNNs. Sensors (Basel) 21. https://doi.org/10.3390/S21144929.

Heyman, J., 2014. The stone skeleton: Structural engineering of masonry architecture. Stone Skeleton: Struct. Eng. Masonry Architecture 1–160. https://doi.org/10.1017/CBO9781107050310.

Huang, H., Cheng, W., Zhou, M., Chen, J., Zhao, S., 2020. Towards Automated 3D Inspection of Water Leakages in Shield Tunnel Linings Using Mobile Laser Scanning Data. Sensors 2020, Vol. 20, Page 6669 20, 6669. DOI: 10.3390/S20226669.

Huang, H. wei, Li, Q. tong, Zhang, D. ming, 2018. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunnelling and Underground Space Technology 77, 166–176. DOI: 10.1016/J.TUST.2018.04.002.

Hussain, A., Akhtar, S., 2017. Review of Non-Destructive Tests for Evaluation of Historic Masonry and Concrete Structures. Arab. J. Sci. Eng. 42, 925–940. https://doi.org/10.1007/S13369-017-2437-Y.

Ibrahim, Y., Nagy, B., Benedek, C., 2020. Deep Learning-Based Masonry Wall Image Analysis. Remote Sensing 2020, Vol. 12, Page 3918 12, 3918. DOI: 10.3390/RS12233918.

Karimi, N., Valibeig, N., Rabiee, H.R., 2023. Deterioration Detection in Historical Buildings with Different Materials Based on Novel Deep Learning Methods with Focusing on Isfahan Historical Bridges. DOI: 10.1080/15583058.2023.2201576. DOI: 10.1080/15583058.2023.2201576.

Kingma, D.P., Ba, J.L., 2014. Adam: A Method for Stochastic Optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. https://doi.org/10.48550/arxiv.1412.6980.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., Fieguth, P., 2015. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Adv. Eng. Inf. 29, 196–210. https://doi.org/10.1016/J.AEI.2015.01.008.

Laefer, D.F., Gannon, J., Deely, E., 2010. Reliability of Crack Detection Methods for Baseline Condition Assessments. J. Infrastruct. Syst. 16, 129–137. https://doi.org/10.1061/(ASCE)1076-0342(2010)16:2(129).

LAPACK: dgelsd [WWW Document], n.d. URL https://netlib.org/lapack/explore-html/d7/d3b/group__double_g_esolve_ga94bd4a63a6dacf523e25ff617719f752.html (accessed 6.21.23).

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 2015 521:7553 521, 436–444. DOI: 10.1038/nature14539.

Li, G., Ma, B., He, S., Ren, X., Liu, Q., 2020. Automatic Tunnel Crack Detection Based on U-Net and a Convolutional Neural Network with Alternately Updated Clique. Sensors 2020, Vol. 20, Page 717 20, 717. DOI: 10.3390/S20030717.

Li, H., Li, N., Wu, R., Wang, H., Gui, Z., Song, D., 2021. GPR-RCNN: An Algorithm of Subsurface Defect Detection for Airport Runway Based on GPR. IEEE Robot Autom Lett 6, 3001–3008. https://doi.org/10.1109/LRA.2021.3062599.

Loshchilov, I., Hutter, F., 2017. Decoupled Weight Decay Regularization. 7th International Conference on Learning Representations, ICLR 2019.

Loverdos, D., Sarhosis, V., 2022. Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. Autom. Constr. 140, 104389 https://doi.org/10.1016/J.AUTCON.2022.104389.

Maturana, D., Scherer, S., 2015. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. IEEE International Conference on Intelligent Robots and Systems 2015-December, 922–928. DOI: 10.1109/IROS.2015.7353481.

McKibbins, L., Elmer, R., Roberts, K., 2010. Tunnels : inspection, assessment and maintenance (C671)). CIRIA.

Nguyen, H., Hoang, N.D., 2022. Computer vision-based classification of concrete spall severity using metaheuristic-optimized Extreme Gradient Boosting Machine and Deep Convolutional Neural Network. Autom. Constr. 140, 104371 https://doi.org/10.1016/J.AUTCON.2022.104371.

NR, 2016. NR_L3_CIV_006_4C - Structures, Tunnels and Operational Property Examinations: Recording of Tunnel Condition Marking Index (TCMI).

ORR, 2021. Rail Infrastructure and Assets 2020-21.

Oxera, 2014. What is the contribution of rail to the UK economy?.

Paraskevopoulou, C., Benardos, A., 2013. Assessing the construction cost of Greek transportation tunnel projects. Tunn. Undergr. Space Technol. 38, 497–505. https://doi.org/10.1016/J.TUST.2013.08.005.

Paraskevopoulou, C., Boutsis, G., 2020. Cost Overruns in Tunnelling Projects: Investigating the Impact of Geological and Geotechnical Uncertainty Using Case Studies. Infrastructures (Basel) 5. https://doi.org/10.3390/INFRASTRUCTURES5090073.

Paraskevopoulou, C., Cornaro, A., Admiraal, H., Paraskevopoulou, A., 2019. Underground space and urban sustainability: an integrated approach to the city of the future. In: International Conference on Changing Cities IV. University of Thessaly, pp. 198–208.

Paraskevopoulou, C., Dallavalle, M., Konstantis, S., Spyridis, P., Benardos, A., 2022. Assessing the failure potential of tunnels and the impacts on cost overruns and project delays. Tunn. Undergr. Space Technol. 123, 104443 https://doi.org/10.1016/J.TUST.2022.104443.

Paraskevopoulou, A., Cornaro, A., Paraskevopoulou, C., 2022a. Underground Space and Street Art towards resilient urban environments, in: Proceedings of the International Conference on Changing Cities IV: "Making Our Cities Resilient in Times of Pandemics." Corfu, Greece.

Perlin, K., 1985. An image synthesizer. ACM SIGGRAPH Computer Graphics 19, 287–296. https://doi.org/10.1145/325165.325247.

Phares, B.M., Washer, G.A., Rolander, D.D., Graybeal, B.A., Moore, M., 2004. Routine Highway Bridge Inspection Condition Documentation Accuracy and Reliability. J. Bridg. Eng. 9, 403–413. https://doi.org/10.1061/(ASCE)1084-0702(2004)9:4(403).

Protopapadakis, E., Voulodimos, A., Doulamis, A., Doulamis, N., Stathaki, T., 2019. Automatic crack detection for tunnel inspection using deep learning and heuristic image post-processing. Appl. Intell. 49, 2793–2806. https://doi.org/10.1007/S10489-018-01396-Y/TABLES/8.

Qi, Charles R, Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). DOI: 10.1109/CVPR.2017.16.

Qi, Charles R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Adv Neural Inf Process Syst 2017-December, 5100–5109.

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 https://doi.org/10.1016/J.CONBUILDMAT.2019.117367.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 . Munich, pp. 234–241.
Schmidhuber, J., 2022. Annotated History of Modern AI and Deep Learning.
Schuller, M.P., 2003. Nondestructive testing and damage assessment of masonry structures. Prog. Struct. Eng. Mater. 5, 239–251. https://doi.org/10.1002/PSE.160.
Seib, V., Lange, B., Wirtz, S., 2020. Mixing Real and Synthetic Data to Enhance Neural Network Training - A Review of Current Approaches. ArXiv.
Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.
Sjölander, A., Belloni, V., Ansell, A., Nordström, E., 2023. Towards Automated Inspections of Tunnels: A Review of Optical Inspections and Autonomous Assessment of Concrete Tunnel Linings. Sensors 2023, Vol. 23, Page 3189 23, 3189. DOI: 10.3390/S23063189.
Smith, J., Paraskevopoulou, C., Bedi, A., Invernici, M., 2023. Deep learning for masonry lined tunnel condition assessment. Expanding Underground - Knowledge and Passion to Make a Positive Impact on the World- Proceedings of the ITA-AITES World Tunnel Congress, WTC 2023 2910–2917. DOI: 10.1201/9781003348030-351.
Soilán, M., Nóvoa, A., Sánchez-Rodríguez, A., Riveiro, B., Arias, P., 2020. Semantic Segmentation of Point Clouds with Pointnet and Kpconv Architectures Applied to Railway Tunnels. ISPRS Annals Photogramm. Remote Sens. Spatial Inf. Sci. 5, 281–288. https://doi.org/10.5194/ISPRS-ANNALS-V-2-2020-281-2020.
Spencer, B.F., Hoskere, V., Narazaki, Y., 2019. Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. Engineering 5, 199–222. https://doi.org/10.1016/J.ENG.2018.11.030.
Sudre, C.H., Li, W., Vercauteren, T., Ourselin, S., Cardoso, M.J., 2017. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10553 LNCS, 240–248. DOI: 10.1007/978-3-319-67558-9_28.
Valero, E., Forster, A., Bosché, F., Hyslop, E., Wilson, L., Turmel, A., 2019. Automated defect detection and classification in ashlar masonry walls using machine learning. Autom. Constr. 106, 102846 https://doi.org/10.1016/J.AUTCON.2019.102846.
Wang, N., Zhao, X., Zhao, P., Zhang, Y., Zou, Z., Ou, J., 2019. Automatic damage detection of historic masonry buildings based on mobile deep learning. Autom. Constr. 103, 53–66. https://doi.org/10.1016/J.AUTCON.2019.03.003.
Williams Rail Review, 2019. The role of the railway in Great Britain.
Xu, Y., Li, D., Xie, Q., Wu, Q., Wang, J., 2021. Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN. Measurement 178, 109316. https://doi.org/10.1016/J.MEASUREMENT.2021.109316.
Xue, Y., Xu, T., Zhang, H., Long, R., Huang, X., 2017. SegAN: Adversarial Network with Multi-scale L_1 Loss for Medical Image Segmentation. Neuroinformatics 16, 383–392. https://doi.org/10.1007/s12021-018-9377-x.
Ye, X.W., Jin, T., Yun, C.B., 2019. A review on deep learning-based structural health monitoring of civil infrastructures. Smart Struct. Syst. 24, 567–585. https://doi.org/10.12989/SSS.2019.24.5.567.
Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., Bai, X., 2023. Deep learning-based 3D point cloud classification: A systematic survey and outlook. Displays 79, 102456. https://doi.org/10.1016/J.DISPLA.2023.102456.
Zhou, M., Cheng, W., Huang, H., Chen, J., 2021. A Novel Approach to Automated 3D Spalling Defects Inspection in Railway Tunnel Linings Using Laser Intensity and Depth Information. Sensors (basel) 21, 5725. https://doi.org/10.3390/S21175725.
Zhou, Z., Siddiquee, R., Tajbakhsh, N., Liang, J., 2018. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. DOI: 10.1007/978-3-030-00889-5_1.

Further reading

Admiraal, H., Cornaro, A., 2020. Future cities, resilient cities – The role of underground space in achieving urban resilience. Underground Space 5, 223–228. https://doi.org/10.1016/J.UNDSP.2019.02.001.